Last Name_____ Student Number _____

University of Toronto
Faculty of Applied Science and Engineering
Edward S. Rogers Sr. Department of Electrical and Computer Engineering

Final Examination
ECE 241F - Digital Systems

Examiners: S. Brown, J. Rose, K. Truong and B. Wang
December 13, 2006

Duration: 2.5 Hours

ANSWER ALL QUESTIONS. USE ONLY THESE SHEETS, USING THE BACKS IF NECESSARY.

- Exam Type D, these specific aids allowed:
  i. Original Versions (no photocopies) of the course text, **Fundamentals of Digital Logic with Verilog Design,** by Brown & Vranesic, ISBN 0-07-282315-1.
  ii. One 8.5 x 11" two-sided aid sheet.

- The amount of marks available for each question is given in square brackets [].

LAST NAME: _____Solutions_____

FIRST NAME: _____

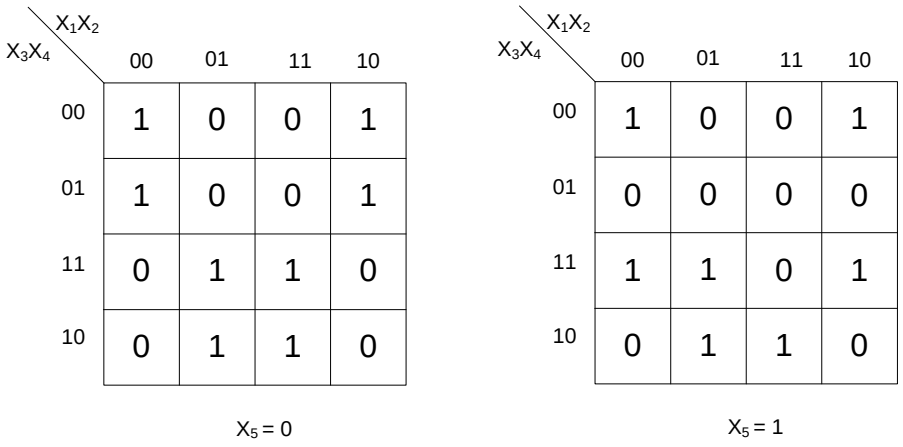
STUDENT NUMBER: _____

Lecture Section: Section 01 (Rose) [ ]
Section 02 (Wang) [ ]
Section 03 (Brown) [ ]
Section 04 (Truong) [ ]

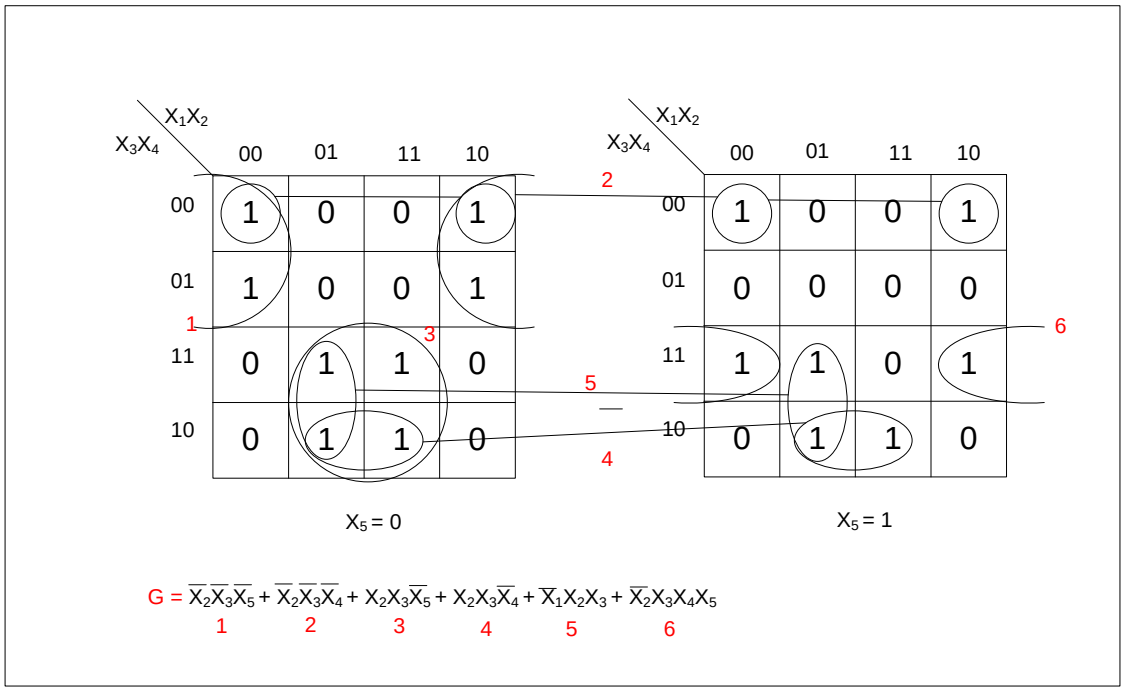
Question	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**Total**
Maximum Mark	**7**	**3**	**6**	**10**	**8**	**7**	**8**	**5**	**8**	**9**	**7**	**78**
Mark Obtained												

Last Name_____ Student Number _____

[3] **Q1 (i)** Consider the five variable function g = f(x_1, x_2, x_3, x_4, x_5) as specified in the K-map below. Give the minimum sum-of-products (SOP) expression for the function g.

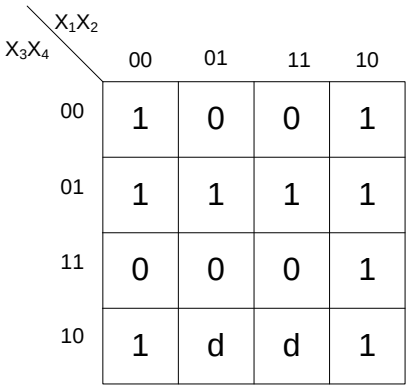


SOLUTION: 0.5 mark for each correct minimum product expression, total of 3 marks.

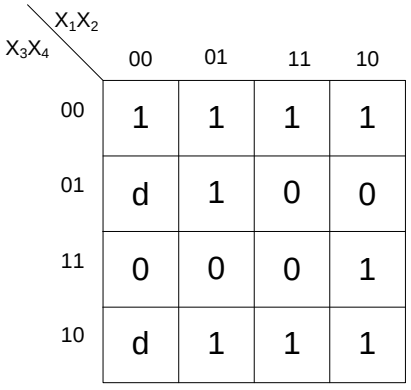


Last Name_____ Student Number _____

[4] **(ii)** Consider the K-maps for the **f** and **g** outputs below. You are to derive logic functions for **f** and **g**, such that the total cost (# of gates + # of inputs) of implementing the two functions is minimal. Both **f** and **g** are to be implemented in either SOP or POS form, whichever is better. Assume that the complement form of the input is free. Note: d is a don't care output value.



f



g

ANSWER:

Group Zeros. Xs in f are 0. Xs in g are 1.

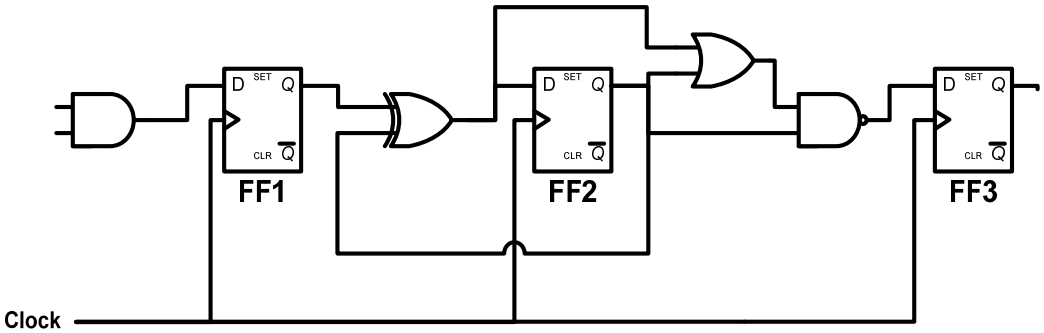
(Underlined is the shared term)
F= (x1+~x3+~x4)(~x2+~x3+~x4)(~x2+x4)
G=(x1+~x3+~x4)(~x2+~x3+~x4)(~x1+x3+~x4)

4 marks for the answer above
3 marks for the POS form correct but not optimal
2 marks for any correct answer POS or SOP
1 mark for partially correct

Last Name_____ Student Number _____

[3] **Q2**. Determine the minimum clock period for the correct operation of the circuit below given the following parameters. T_{SU} is setup time; T_H is hold time.

Parameter	Max (ns)	Min (ns)
T_{and}	4	1
T_{xor}	5	1.2
T_{nand}	3	0.8
T_{or}	4	1
$T_{Clock\ to\ Q}$	5	3.5
T_{SU}	2	
T_H	3	



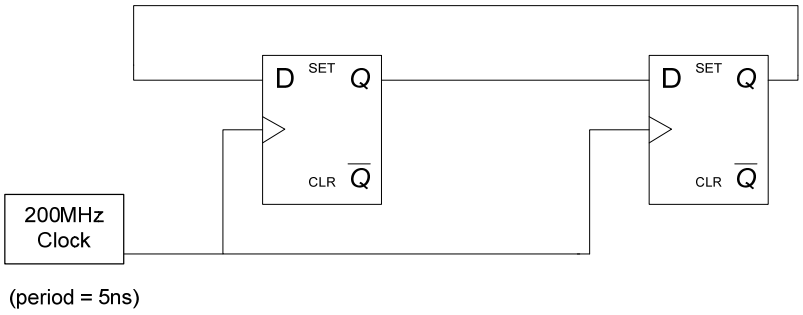
ANSWER:

Critical path: $T_{min} = t_{clocktoq} + t_{xor} + t_{or} + t_{nand} + t_{setup} = 5 + 5 + 4 + 3 + 2 = 19$ ns

3 marks for the answer above
-1 for each term that was not consider to a max of -3.

Last Name_____ Student Number _____

[6] **Q3.** Consider the circuit below, in which the Clock-To-Q time ($T_{Clock\_to\_Q}$) of both flip-flops is 3ns.



This circuit functions correctly, as a shift register connected in a ring, when the clock is running at the frequency (and period) given.

[2] **(i)** If this circuit is still to work correctly, what is the largest value for the set-up time (T_{SU})?

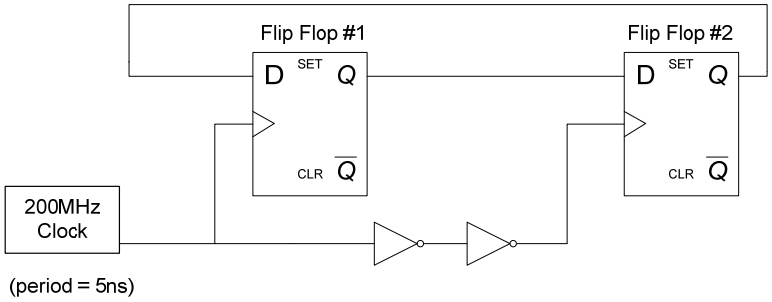
SOLUTION: The largest value of T_{SU} can be Period – Clock-to-Q = 5 – 3 = 2ns
2 marks for correct solution

[2] **(ii)** If this circuit is still to work correctly, what is the largest value for the hold time (T_H)?

SOLUTION: The largest value of T_H can be is Clock-to-Q = 3ns
2 marks for correct solution

Last Name_____ Student Number _____

Q3 continued. Now consider the following circuit which has inverters on the clock path to the second flip-flop.

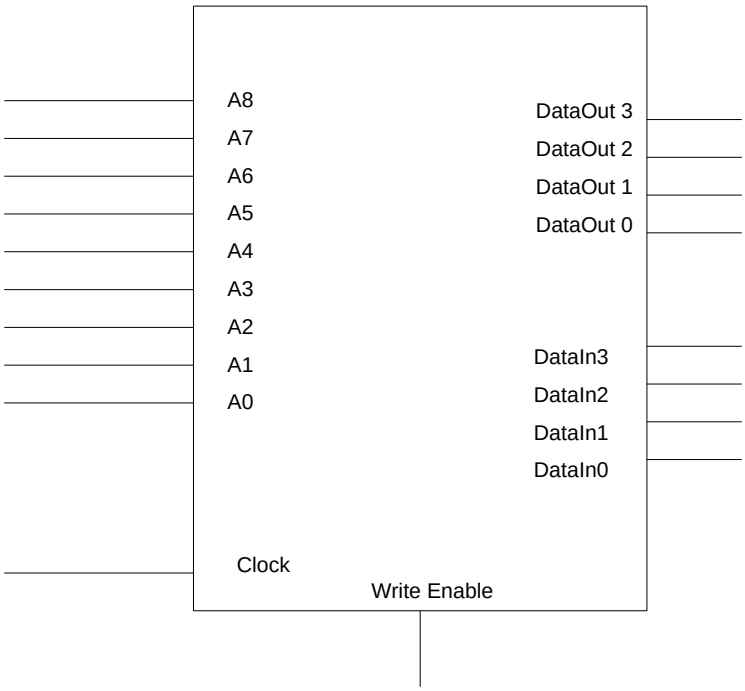


[2] **(iii)** Assume that each inverter has a delay of 0.5 ns. If this circuit is still to work correctly, what is the largest value that the hold time for the flip-flop labeled Flip Flop #2 can be?

SOLUTION: The largest value of T_H can be is Clock-to-Q – 2 x T_{inv} = 3ns – (2 x 0.5) = 2ns
2 marks for correct solution

Last Name_____ Student Number _____

[10] **Q4**. Consider the following digital memory.



[1] **(i)** How many word lines (rows) are there in this memory?

SOLUTION: $2^9 = 512$ [worth 1 mark]

[1] **(ii)** How many bit lines (columns) are there in this memory?

SOLUTION: 4 [worth 1 mark]

[1] **(iii)** How many bits does this memory contain?

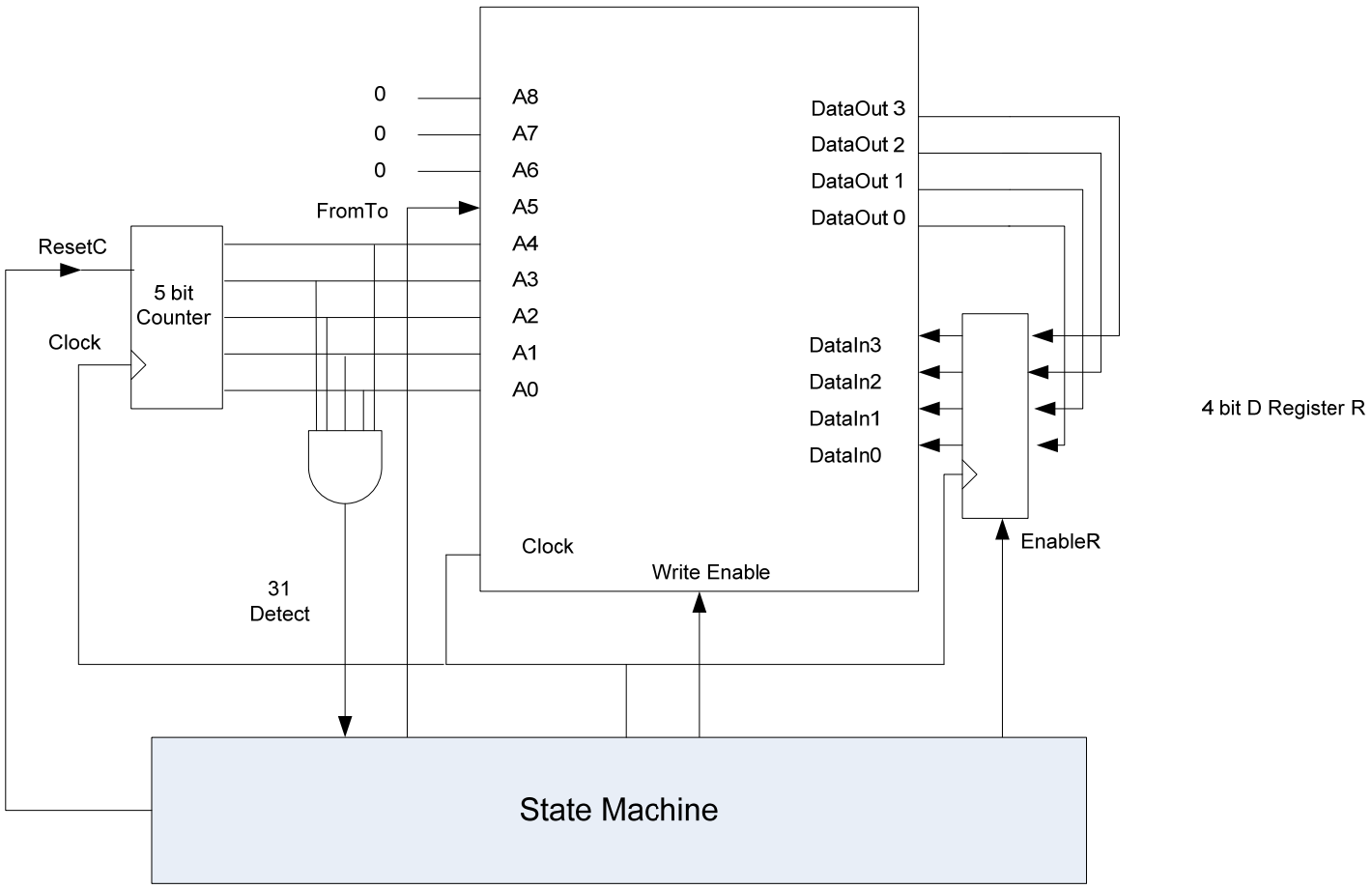
SOLUTION: $2^9 \times 4 = 512 \times 4 = 2048$ [worth 1 mark]

[7] **(iv)** On the next (blank) page, you are to give the block diagram of a digital circuit that copies the contents of memory locations 0 through 31 into memory locations 32 through 63. Your circuit should consist of a counter, a register, any other gates or constant signals you need, all controlled by a finite state machine. Assume that the address and Data In signals are registered within the memory and are clocked. **You do not have to design the state machine, you just have to show what its inputs and outputs are.**

Last Name_____ Student Number _____

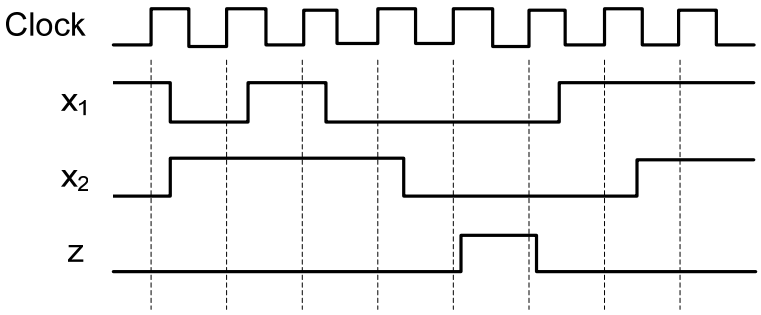
Q4 continued.

SOLUTION worth 7 marks:
[1 mark] Address Counter
[1] 31 Detector
[1] Register on DOUT ->DIN
[1] FromTo Signal
[0.5] 0's on A6, A7, A8
[1] WriteEnable
[0.5] Reset
[0.5] Register R Enable
[0.5] Counter Enable



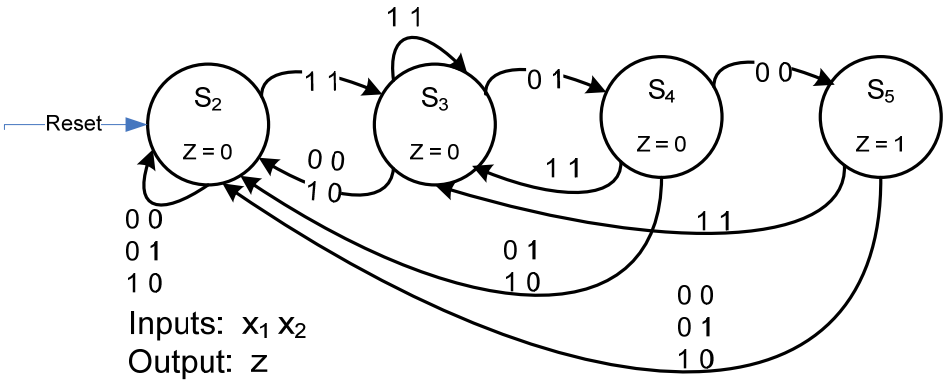
Last Name_____ Student Number _____

[8] **Q5.** A sequence recognizer is to be designed to check two incoming bit streams, $\mathbf{x_1}$ and $\mathbf{x_2}$. When the sequence $\mathbf{x_1}$ = **100** and $\mathbf{x_2}$ = **110** have appeared coincidentally on the input lines, the output signal, **z** is asserted (i.e. **z** = **1**) for exactly one clock cycle. An example timing diagram is given below.



You are asked to give a state diagram of the finite state machine for the system. Higher marks will be given to the designs with a fewer number of states.

Solution:

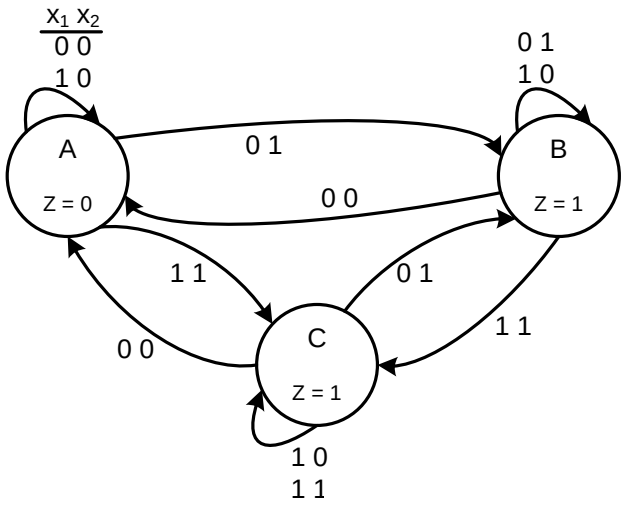


Marking scheme:

2 marks for each correct state with the correct state transitions
-0.5 mark for each wrong or missing transition
-1 mark for each extra state

Last Name_____ Student Number _____

[7] **Q6**.The following FSM has two data inputs, $\mathbf{x_1}$ and $\mathbf{x_2}$, and one output signal, **z**. We wish to implement this FSM using three flip flops and one-hot encoding. Let the inputs and outputs of the D flip-flops be D_A, D_B, D_C and Q_A, Q_B, Q_C respectively. Determine the next-state and output logic expressions using 1-hot encoding. Write the logic expressions in the minimal sum-of-product (SOP) forms.



Solutions:

$$D_A = Q_A(\overline{x_1}\,\overline{x_2} + x_1\overline{x_2}) + Q_B\overline{x_1}\,\overline{x_2} + Q_C\overline{x_1}\,\overline{x_2} = Q_A\overline{x_2} + (Q_B + Q_C)\overline{x_1}\,\overline{x_2}$$
$$= Q_A\overline{x_2} + \overline{Q}_A\overline{x_1}\,\overline{x_2} = (Q_A + \overline{x_1})\overline{x_2} = Q_A\overline{x_2} + \overline{x_1}\,\overline{x_2}$$

$$D_B = Q_A\overline{x_1}x_2 + Q_B(\overline{x_1}x_2 + x_1\overline{x_2}) + Q_C\overline{x_1}x_2 = Q_A\overline{x_1}x_2 + Q_B\overline{x_1}x_2 + Q_Bx_1\overline{x_2} + Q_C\overline{x_1}x_2 = \overline{x_1}x_2 + Q_Bx_1\overline{x_2}$$

$$D_C = Q_Ax_1x_2 + Q_Bx_1x_2 + Q_C(x_1\overline{x_2} + x_1x_2) = x_1x_2 + Q_Cx_1\overline{x_2} = x_1(x_2 + Q_C) = x_1x_2 + x_1Q_C$$

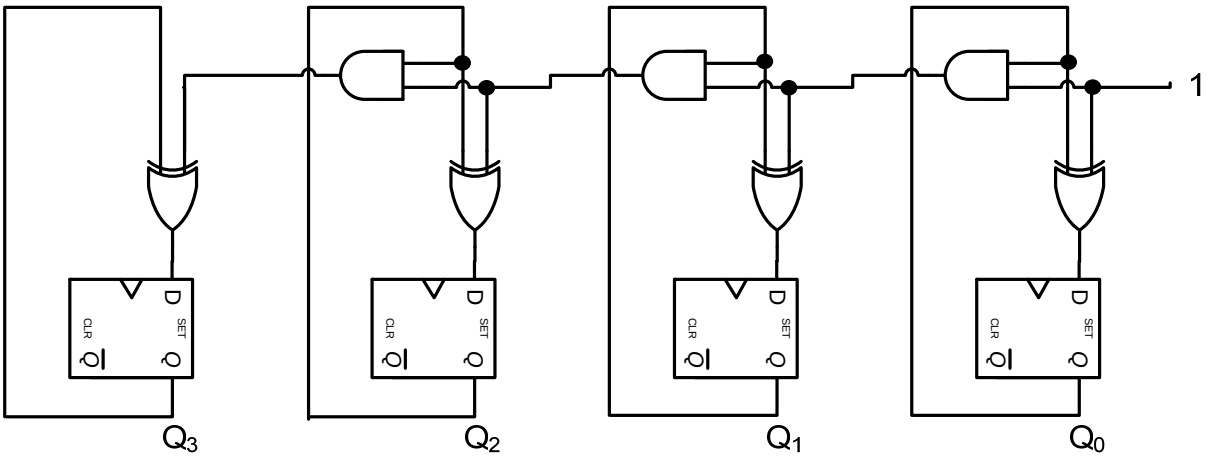
$$Z = Q_B + Q_C$$

Marking Scheme:
2 marks for each correct D_A, D_B and D_C expression in minimal SOP form
-0.5 mark for each correct non-minimal SOP form
-1 mark for each partially correct expression in minimal SOP form
-1.5 marks for each partially correct expression in non-minimal SOP form

1 mark for the correct Z expression

Last Name_____ Student Number _____

[8] **Q7**. The schematic below shows the design of a 4-bit binary counter. Assume that any gate delay can be calculated by the following formula: ***delay*** = 1 + 0.1 x (# of inputs) ns
Using this formula, the delay through an inverter would be 1.1 ns, and the delay through a 2-input gate would be 1.2 ns, and so on. Also assume that the setup time and the clock-to-Q delay for the D flip-flop are 0.3 ns and 0.7 ns respectively.



[3] **(i)** Determine the minimum amount of time that it takes for this counter to count from 0 (i.e. $Q_3Q_2Q_1Q_0 = 0000$) to 15 (i.e. $Q_3Q_2Q_1Q_0 = 1111$).

Solutions and Marking Scheme in []:

The longest delay path is from 1 to Q_3
= 4 x (2-input gate delay) + setup time + clock-to-Q delay
[0.5] [0.5] [0.5] [0.5]

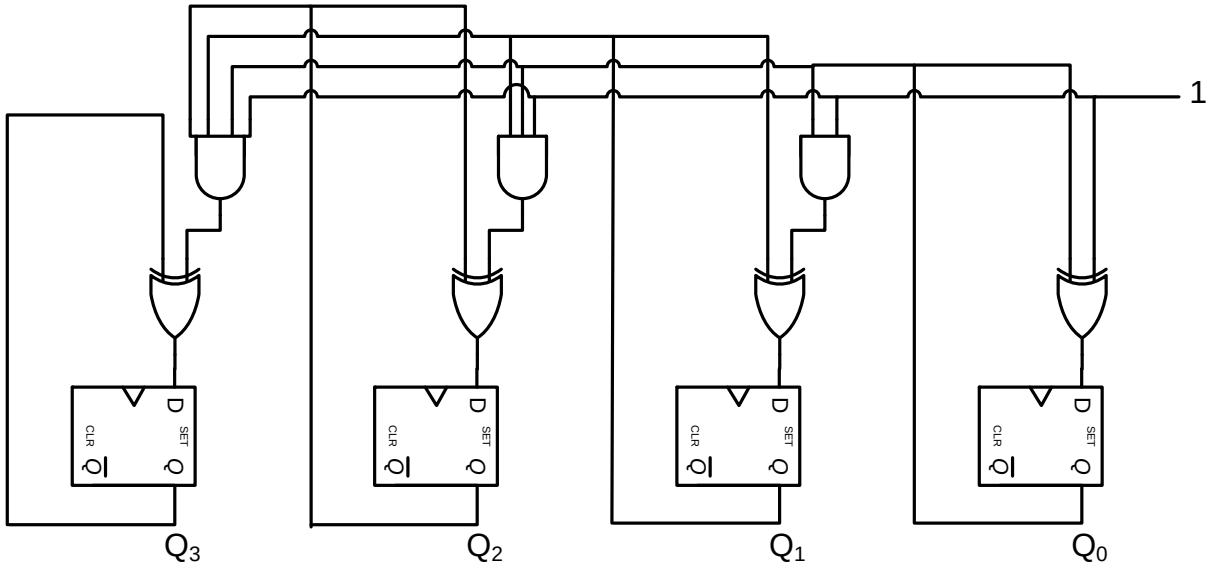
= 4 x 1.2 + 0.3 + 0.7
= 5.8 ns

[1]
T = 15 x 5.8 ns
= 87 ns

Last Name_____ Student Number _____

[5] **(ii)** Modify the circuit so that the counter can count at a faster rate. Show the circuit schematic of your design. Hint: Remember that the delay of a gate is given by the formula: ***delay*** = 1 + 0.1 x (# of inputs) ns. Determine the minimum amount of time for your design to count from 0 to 15.

Solutions and Marking Scheme in []:



[3 marks] for circuit

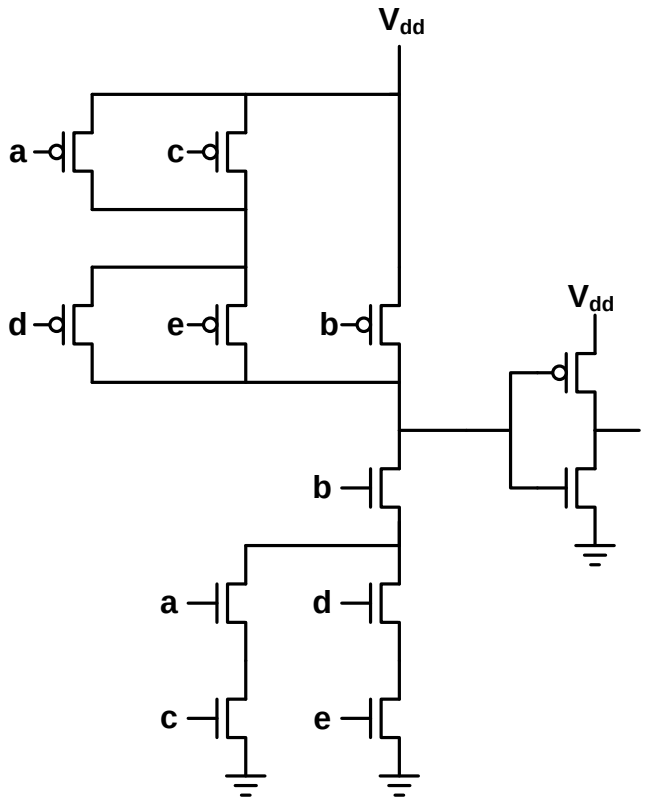
The critical path delay
= (4-input gate delay) + (2-input gate delay) + setup time + clock-to-Q delay
= 1.4 + 1.2 +0.3 +0.7
= 3.6 ns
[1.5]

T = 15 x 3.6 ns
= 54 ns
[0.5]

Last Name_____ Student Number _____

[3] **Q8 (i)** Draw the CMOS transistor circuit for the function **f=(abc+dbe)** using the fewest transistors

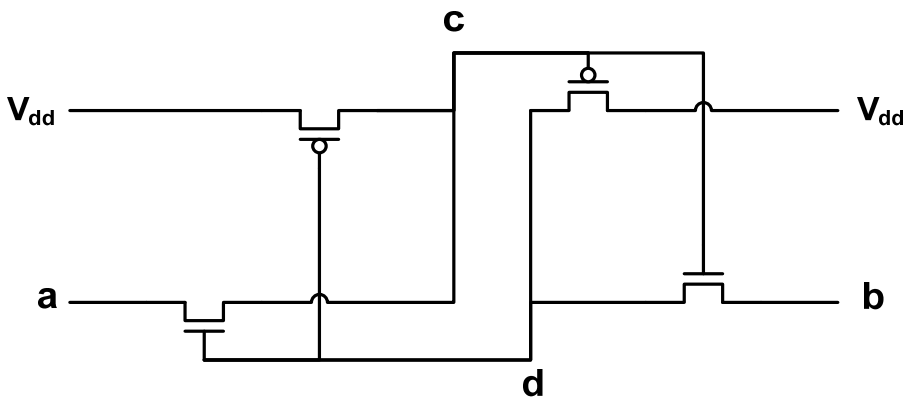
ANSWER: First factor out b. f= b(ac+de).



Marking Scheme:
3 marks for the circuit above.
2 marks for a circuit that does not factor out the b and uses 14 transisters
1 marks for any other correct solution

Last Name_____ Student Number _____

[2] **(ii)** Describe the function of the circuit below in the simplest way. **a** and **b** are inputs, while **c** and **d** are outputs. Assume that the initial value for the output **c** is 0 and **d** is V_{dd}.



ANSWER and Marking Scheme:
2 marks – the circuit implements an RS latch.
1 marks – describes something that resembles a latch or memory unit

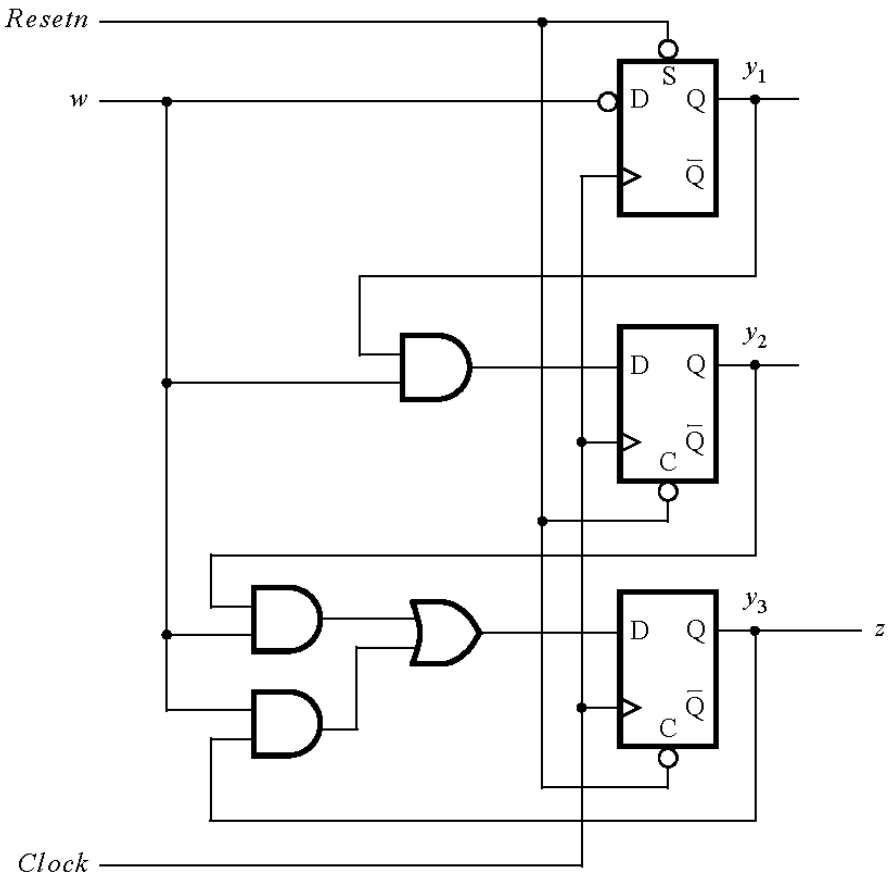
Last Name_____ Student Number _____

[8] **Q9.** The circuit below represents a finite state machine with input *w* and output *f*. The FSM has three states: S0, S1, S2, and implements the one-hot state assignment.

	y_3	y_2	y_1
S0	0	0	1
S1	0	1	0
S2	1	0	0

In the circuit diagram, the *C* input on a flip-flop serves as a reset input (i.e. set Q = 0), and the *S* input serves as a preset input (i.e. set Q = 1).

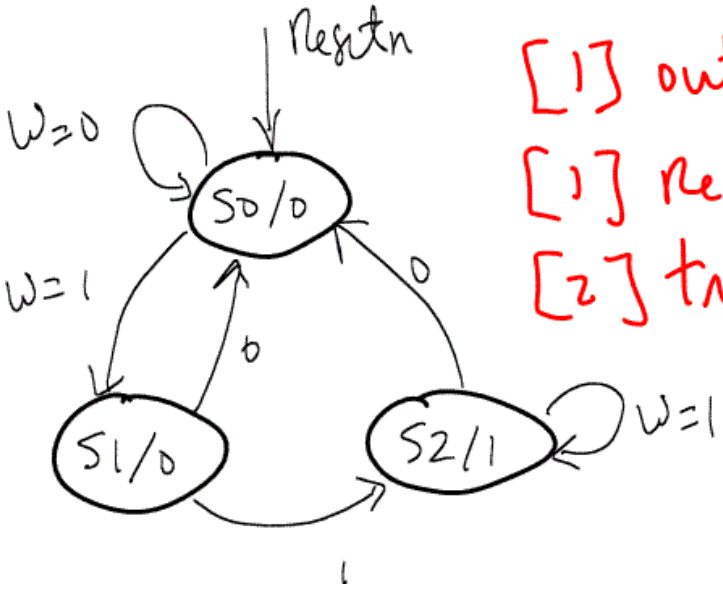
Circuit:



Last Name_____ Student Number _____

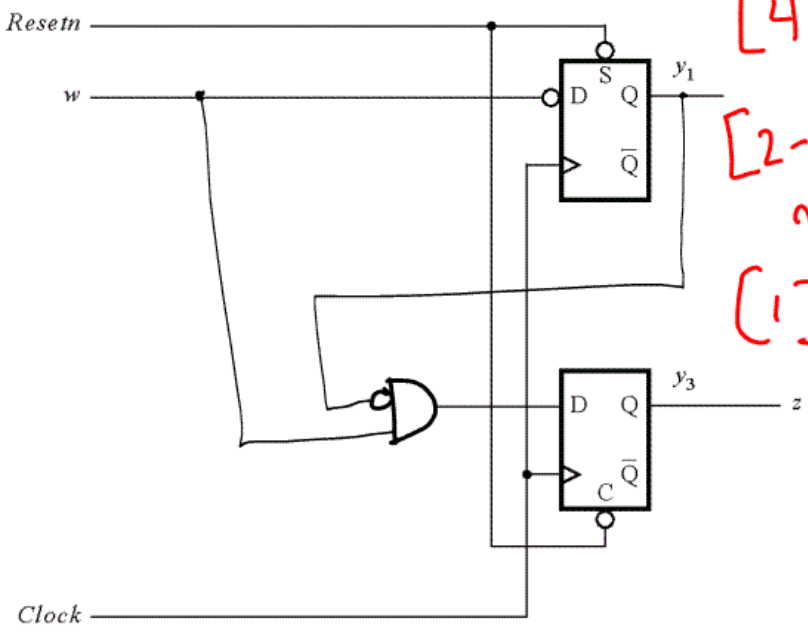
[4] **(i)** Draw a state diagram that corresponds to this circuit. Try to make the diagram as simple as possible.

ANSWER:



[1] output values
[1] reset
[2] transitions (-0.5 for each error)

[4] **(ii)** In the circuit below we have modified the FSM implementation by deleting the flip-flop that produces y_2. Complete the circuit schematic by drawing logic gates (as few as possible) that feed the input to flip-flop y_3, so that the circuit still works properly and produces the same output z as before.



[4] correct, AND gate
[2-3] correct, but multiple gates
[1] wrong, but good attempt

Last Name_____ Student Number _____

[9] **Q10**. Using inverters, 2-input AND, OR, XOR gates, and D flip-flop as needed draw the simplest logic circuit you can that corresponds to the Verilog code given below. Higher marks will be given for answers that use as few gates as possible.

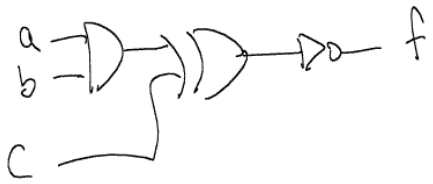
[3] **(i)**

```
module something(a, b, c, f);
        input a, b, c;
        output f;

        assign f = c ? (a & b ) : (~a + ~b);
endmodule;
```

ANSWER:

$f = c(ab) + \bar{c}(\bar{a}+\bar{b}) = c \odot (ab)$



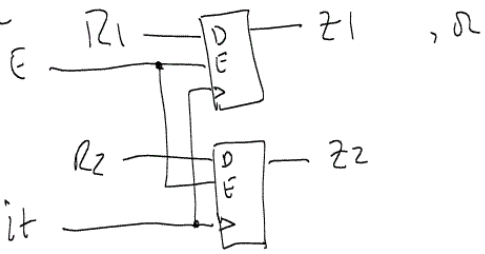
[3] this cct
[2] correct, up to 5 gates
[1] wrong, but good attempt

[3] **(ii)**

```
module something(input R1, R2, E, Do_it, output reg Z1, Z2);
        always @(posedge Do_it)
                if (E)
                        Z1 <= R1;
                        Z2 <= R2;
endmodule;
```

ANSWER:

[3] Full marks for "Syntax error", Ω
[3] Full marks for



, Ω

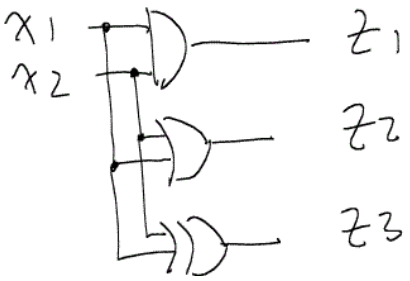
[3] Full marks for above, but showing muxes for Enable, Ω
[3] Full marks for above, but with no E for Z2.
[2] "close" to any Ω above
[1] Wrong, but good attempt

Last Name_____ Student Number _____

[3] **(iii)**

```
module something ( input [1:2] X, output [1:3] Z);
    assign  Z =   ({3{(X == 2'b00)}} & 3'b000) |
                  ({3{(X == 2'b01)}} & 3'b011) |
                  ({3{(X == 2'b10)}} & 3'b011) |
                  ({3{(X == 2'b11)}} & 3'b110) ;
endmodule
```

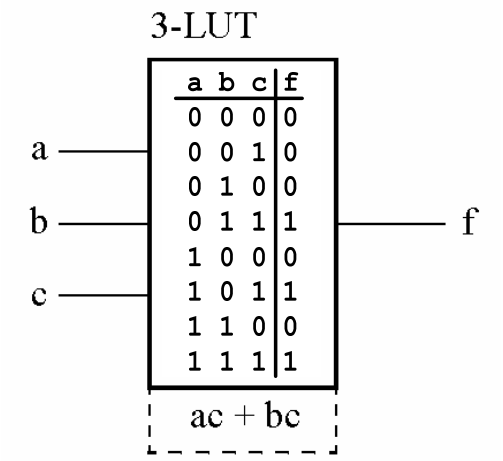
ANSWER:



[1] for each of the three correct outputs

Last Name_____ Student Number _____

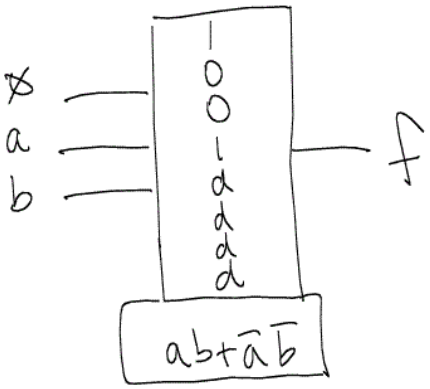
[7] **Q11**. The FPGA you used in the labs for this course comprises 4-input lookup tables (4-LUTs). For this question, assume that circuits are being targeted to an FPGA that has 3-input lookup tables (3-LUTs). For example, if we wish to implement the function f = c(a + b) then we can show how this function is implemented in a 3-LUT as follows:



Notice that this diagram shows the truth table implemented in the lookup table, and it shows a logic expression for the implemented function as a label on the bottom of the LUT. You are to draw similar diagrams for parts a. and b. below, using one or more 3-LUTs as needed to implement the given functions. For each 3-LUT be sure to show the truth table that corresponds to its inputs, and include the label on the bottom of the LUT that gives an expression for the implemented function. Use as few 3-LUTs as possible for each function. If any of the entries in your truth tables are don't cares, write their value as 'd'.

[3] **(i)**. $f = (a + \bar{b})(\bar{a} + b)$

ANSWER:



[3] for correct

-1 for not showing "d" entries

(either top or bottom half can be "d")

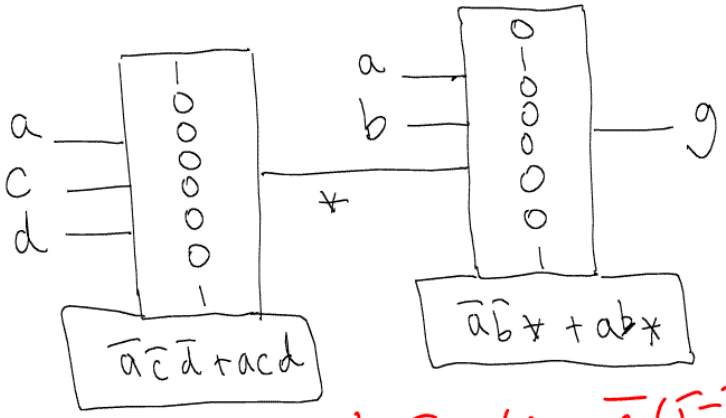
Last Name_____ Student Number _____

[4] **(ii)** $g = \bar{a}b\bar{c}\bar{d} + abcd$

ANSWER:

There are many correct answers, which all have the form:

$\bar{a}\bar{b}(\bar{a}\bar{c}\bar{d} + acd) + ab(acd + \bar{a}\bar{c}\bar{d})$



[4]

[3] for using three LUTs (e.g., $\bar{a}(\bar{b}\bar{c}\bar{d}) + a(bcd)$

THIS PAGE IS INTENTIONALLY BLANK

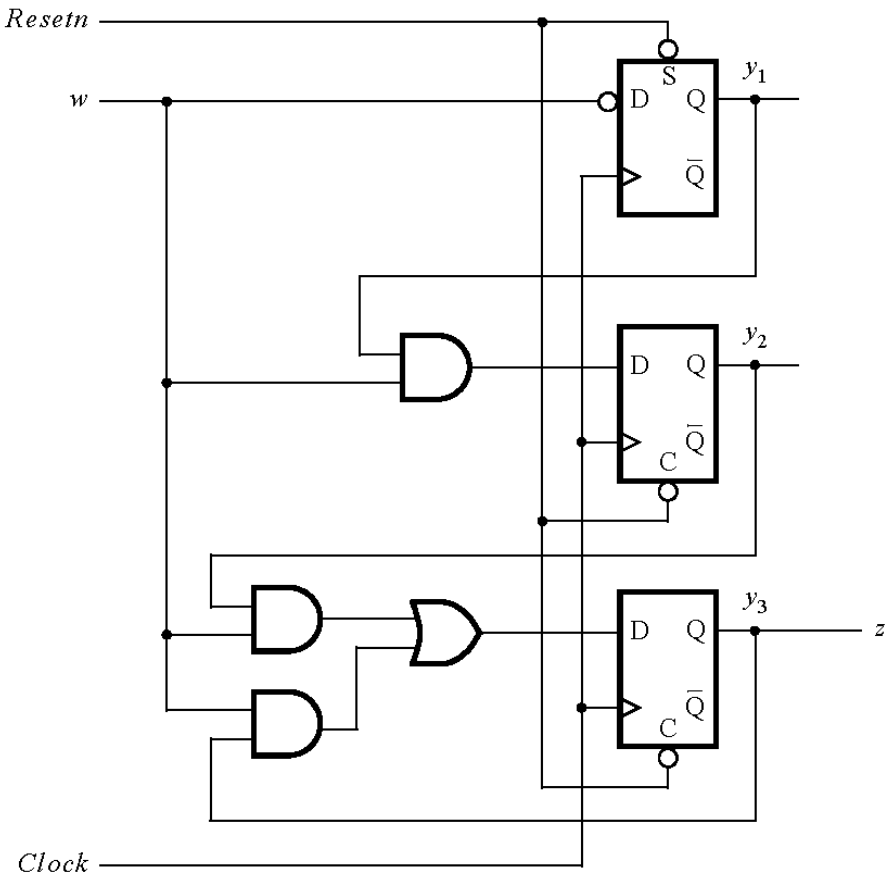
Last Name_____ Student Number _____

THIS PAGE IS INTENTIONALLY [8] **Q9.** The circuit below represents a finite state machine with input *w* and output *f*. The FSM has three states: S0, S1, S2, and implements the one-hot state assignment.

	y_3	y_2	y_1
S0	0	0	1
S1	0	1	0
S2	1	0	0

In the circuit diagram, the *C* input on a flip-flop serves as a reset input (i.e. set Q = 0), and the *S* input serves as a preset input (i.e. set Q = 1).

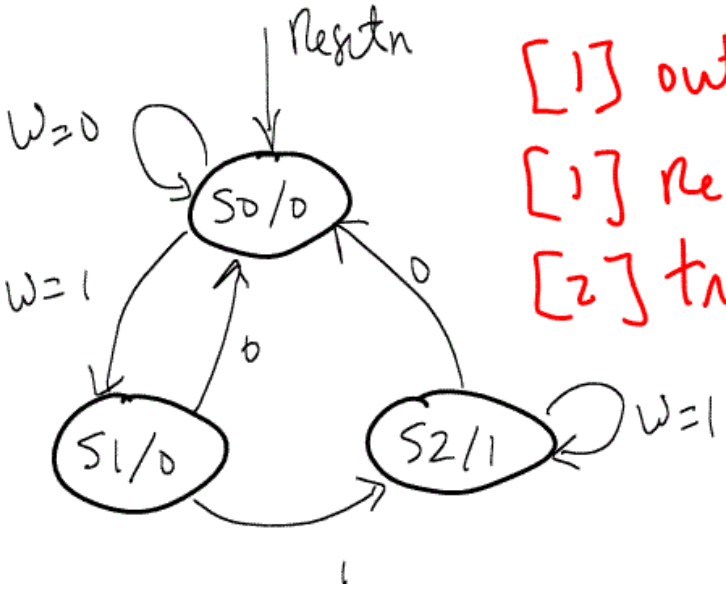
Circuit:



Last Name_____ Student Number _____

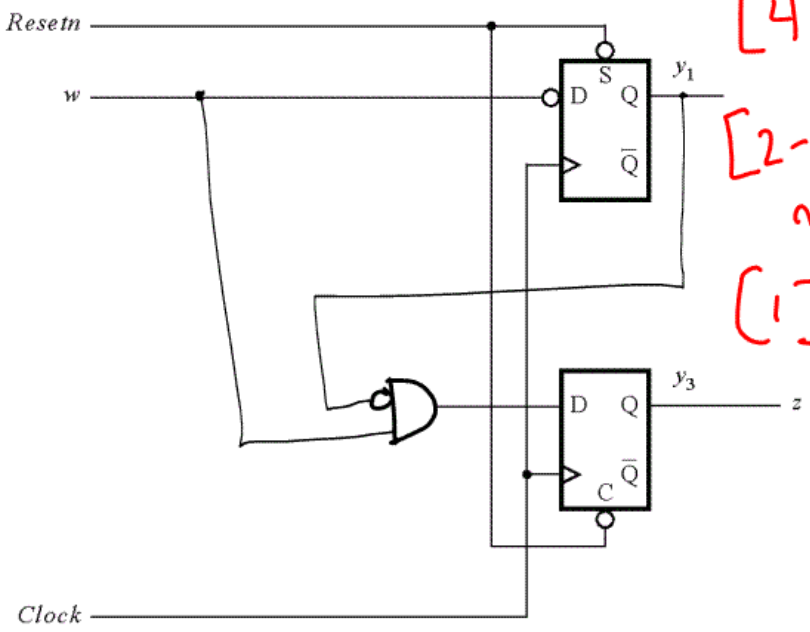
[4] **(i)** Draw a state diagram that corresponds to this circuit. Try to make the diagram as simple as possible.

ANSWER:



[1] output values
[1] reset
[2] transitions (-0.5 for each error)

[4] **(ii)** In the circuit below we have modified the FSM implementation by deleting the flip-flop that produces y_2. Complete the circuit schematic by drawing logic gates (as few as possible) that feed the input to flip-flop y_3, so that the circuit still works properly and produces the same output z as before.



[4] correct, AND gate
[2-3] correct, but multiple gates
[1] wrong, but good attempt

Last Name_____ Student Number _____

[9] **Q10**. Using inverters, 2-input AND, OR, XOR gates, and D flip-flop as needed draw the simplest logic circuit you can that corresponds to the Verilog code given below. Higher marks will be given for answers that use as few gates as possible.

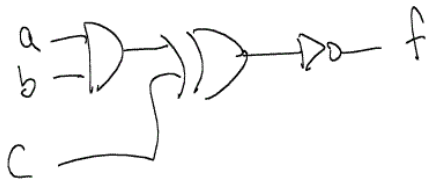
[3] **(i)**

```
module something(a, b, c, f);
        input a, b, c;
        output f;

        assign f = c ? (a & b ) : (~a + ~b);
endmodule;
```

ANSWER:

$f = c(ab) + \bar{c}(\bar{a}+\bar{b}) = c \odot (ab)$



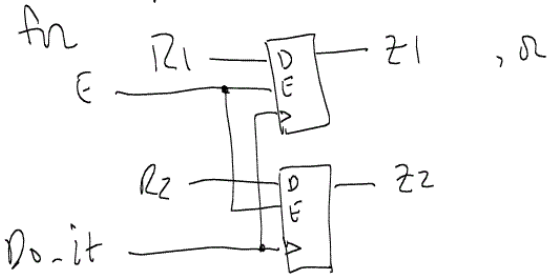
[3] this cct
[2] correct, up to 5 gates
[1] wrong, but good attempt

[3] **(ii)**

```
module something(input R1, R2, E, Do_it, output reg Z1, Z2);
        always @(posedge Do_it)
                if (E)
                        Z1 <= R1;
                        Z2 <= R2;
endmodule;
```

ANSWER:

[3] Full marks for "Syntax error", Ω
[3] Full marks for



, Ω

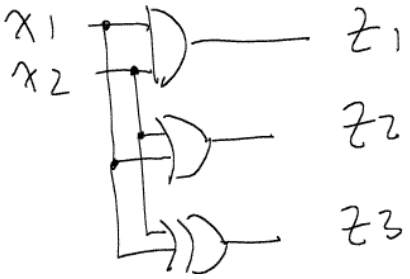
[3] Full marks for above, but showing muxes for Enable, Ω
[3] Full marks for above, but with no E for Z2.
[2] "close" to any Ω above
[1] Wrong, but good attempt

Last Name_____ Student Number _____

[3] **(iii)**

```
module something ( input [1:2] X, output [1:3] Z);
    assign  Z =   ({3{(X == 2'b00)}} & 3'b000) |
                  ({3{(X == 2'b01)}} & 3'b011) |
                  ({3{(X == 2'b10)}} & 3'b011) |
                  ({3{(X == 2'b11)}} & 3'b110) ;
endmodule
```

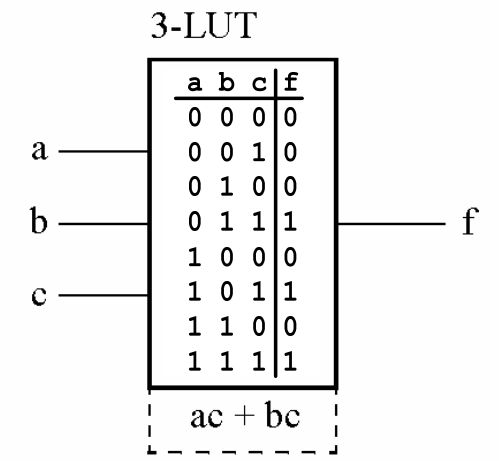
ANSWER:



[1] for each of the three correct outputs

Last Name_____ Student Number _____

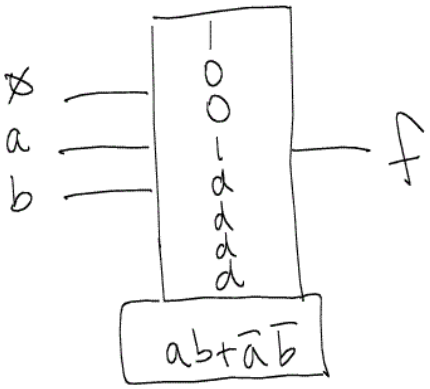
[7] **Q11**. The FPGA you used in the labs for this course comprises 4-input lookup tables (4-LUTs). For this question, assume that circuits are being targeted to an FPGA that has 3-input lookup tables (3-LUTs). For example, if we wish to implement the function f = c(a + b) then we can show how this function is implemented in a 3-LUT as follows:



Notice that this diagram shows the truth table implemented in the lookup table, and it shows a logic expression for the implemented function as a label on the bottom of the LUT. You are to draw similar diagrams for parts a. and b. below, using one or more 3-LUTs as needed to implement the given functions. For each 3-LUT be sure to show the truth table that corresponds to its inputs, and include the label on the bottom of the LUT that gives an expression for the implemented function. Use as few 3-LUTs as possible for each function. If any of the entries in your truth tables are don't cares, write their value as 'd'.

[3] **(i)**. $f = (a + \bar{b})(\bar{a} + b)$

ANSWER:



[3] for correct

-1 for not showing "d" entries

(either top or bottom half can be "d")

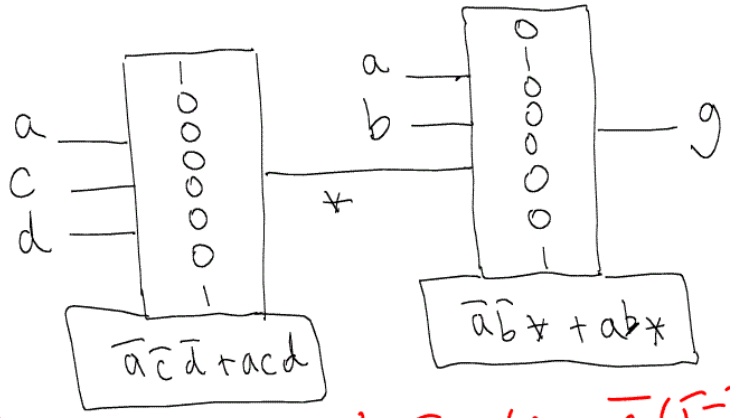
Last Name_____ Student Number _____

[4] **(ii)** $g = \bar{a}b\bar{c}\bar{d} + abcd$

ANSWER:

There are many correct answers, which all have the form:

$\bar{a}\bar{b}(\bar{a}\bar{c}\bar{d} + acd) + ab(acd + \bar{a}\bar{c}\bar{d})$



[4]

[3] for using three LUTs (e.g., $\bar{a}(\bar{b}\bar{c}\bar{d}) + a(bcd)$)

THIS PAGE IS INTENTIONALLY BLANK

THIS PAGE IS INTENTIONALLY