SOLUTIONS

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

ECE253F – Digital and Computer Systems
Midterm Examination

October 14, 2021 6:10pm - 7:40pm
Duration: 90 minutes

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

Please enter your name and student number in the spaces provided above as it appears on Quercus. It is important that your name exactly match the Quercus gradebook.

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. Your answer **MUST** be fully contained on the same page as the question. **Any material written on the back of each page will be ignored.**

Please state any assumptions you make when answering a question.

The number of marks for each question are indicated. The exam has **16 pages**, including this one.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total
10	6	7	6	8	10	8	55

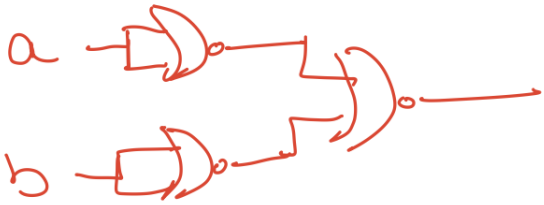
Question 1 [10 Marks]

[3 marks] (a) The Boolean operators AND, OR, NOT comprise a complete set, meaning that any Boolean function can be implemented with these operators. Prove that the NOR operator is by itself a complete set, i.e., that any Boolean function can be built with only NOR gates.

NOT can be implemented as: $\overline{(a + a)}$

OR can be implemented as $\overline{\overline{(a + b)}}$

AND can be implemented as $\overline{(\overline{a} + \overline{b})}$



(b) Derive the minimal sum of products (SoP) expression for the following Boolean expressions using only Boolean Algebra. Clearly indicate what property or rule you are using at each step. WARNING! Manage your time. If you are not converging on a solution, consider the number of marks allocated for these questions.

[4 marks]

i. $f = (ab + b\overline{c})(ad + \overline{a}bc + cd + \overline{a}\,\overline{c})$

$f = abad + ab\overline{a}bc + abcd + ab\overline{a}\,\overline{c} + b\overline{c}ad + b\overline{c}\,\overline{a}bc + b\overline{c}cd + b\overline{c}\,\overline{a}\,\overline{c}$ Distributive

$f = abd + a\overline{a}bc + abcd + ab\overline{a}\,\overline{c} + b\overline{c}ad + b\overline{c}\,\overline{a}c + b\overline{c}cd + b\overline{c}\,\overline{a}$ Rule 7a

$f = abd + abcd + b\overline{c}ad + b\overline{c}\,\overline{a}$ Rule 8a

$f = abd + \overline{a}b\overline{c}$ Combining

[3 marks]

ii. $f = (x + y)(w + \overline{x} + y)(\overline{wx})(w + \overline{y} + z)(w + z)$

$f = (x + y)(w + \overline{x} + y)(\overline{w} + \overline{x})(w + \overline{y} + z)(w + z)$ DeMorgan's

$f = (x + y)(w + \overline{x} + y)(\overline{w} + \overline{x})(w + z)$ covering

$f = (x + y)(\overline{x} + y)(\overline{w} + \overline{x})(w + z)$ combining

$f = y(\overline{w} + \overline{x})(w + z)$ combining

$f = y(\overline{w}w + \overline{w}z + \overline{x}w + \overline{x}z)$ Distributive

$f = y(\overline{w}z + \overline{x}w + \overline{x}z)$ Rule 8a

$f = y(\overline{w}z + \overline{x}w)$ Consensus

$f = \overline{w}yz + w\overline{x}y$ Distributive

Question 2 [6 Marks]

[2 marks] (a) Simplify the following logic expression using Boolean Algebra. For each step, indicate which rule was applied.

$f = ab + ac + \overline{(a+b)}\,c + ab\bar{c}$

$f = ab + ac + \bar{a}\bar{b}c + ab\bar{c}$ DeMorgan's

$f = ab + ac + \bar{a}\bar{b}c$ Absorption

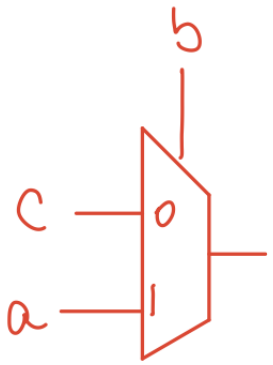
$f = ab + c(a + \bar{a}\bar{b})$ Distributive

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

$f = ab + ac + \bar{b}c)$ Distributive

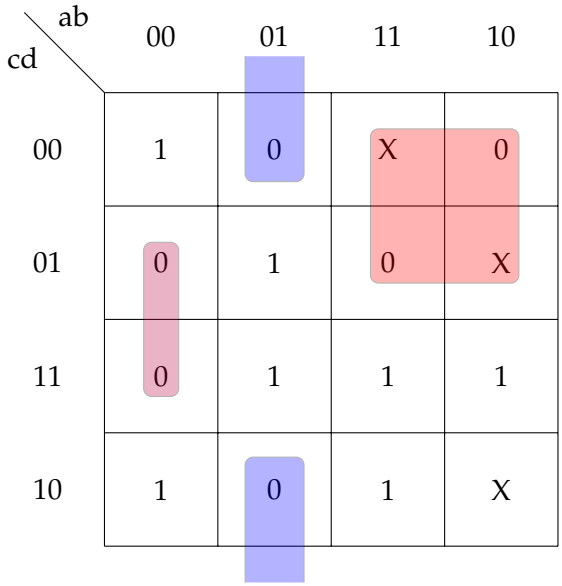
$f = ab + \bar{b}c$ Consensus

[4 marks] (b) Draw a schematic implementing your simplified expression from part (a) using only 2-to-1 multiplexers.



Question 3 [7 Marks]

[2 marks] (a) Fill in the k-map for function $f(a,b,c,d) = \Sigma m(0,2,5,7,11,14,15) + D(9,10,12)$, where D() indicates minterms that are specified as 'Don't Cares'.



[3 marks] (b) Derive a minimal product-of-sums (PoS) expression for the function in part (a).

$f = (\overline{a} + c)(a + b + \overline{d})(a + \overline{b} + d)$

[2 marks] (c) Calculate the cost of your PoS expression for part (b) using the definition of cost from Brown and Vranesic's textbook and given in lecture. You may assume that inverters are "free" in your cost calculation; i.e. signals are freely available in true and complemented form.

cost = 4 gates + 11 inputs = 15

Question 4 [6 Marks]

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

9-bit binary	decimal	hexadecimal
011101110		
	495	
		17F

9-bit binary	decimal	hexadecimal
011101110	255-16-1 = 238	0EE
1 1110 1111	495	1EF
1011 1111	511 - 128 = 383	17F

Question 5 [8 Marks]

Consider the following logic function, expressed in sum-of-products (SoP) form: $f = \overline{a} + bc + \overline{d}$.

[4 marks] (a) Write the function f in product-of-sums (PoS) form. Show your work for complete marks.

$$\overline{f} = \overline{(\overline{a} + bc + \overline{d})}$$

$$= (\overline{\overline{a}})(\overline{bc})(\overline{\overline{d}})$$

$$= a(\overline{b} + \overline{c})(d)$$

$$= ad\overline{b} + ad\overline{c}$$

$$\overline{\overline{f}} = \overline{ad\overline{b} + ad\overline{c}}$$

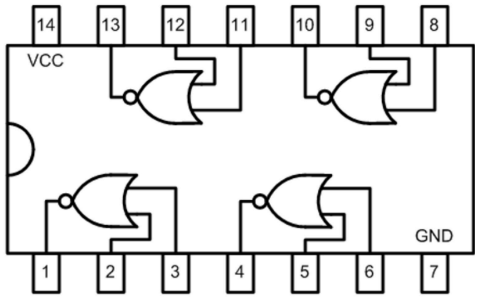
$$= \overline{(ad\overline{b})}\,\overline{(ad\overline{c})}$$

$$= (\overline{a} + \overline{d} + b)(\overline{a} + \overline{d} + c)$$

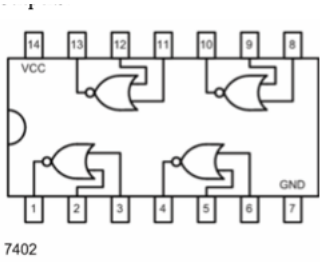
in PoS form

Question 5 continued …

[4 marks] (b) In Lab 1, you built logic circuits using 7400-series chips on the breadboard. Assuming you have only 7402 chips available (quad 2-input NOR shown below), build the logic function given in part (a). Draw a schematic of connected 7402 chips. Carefully label the inputs and outputs.



7402



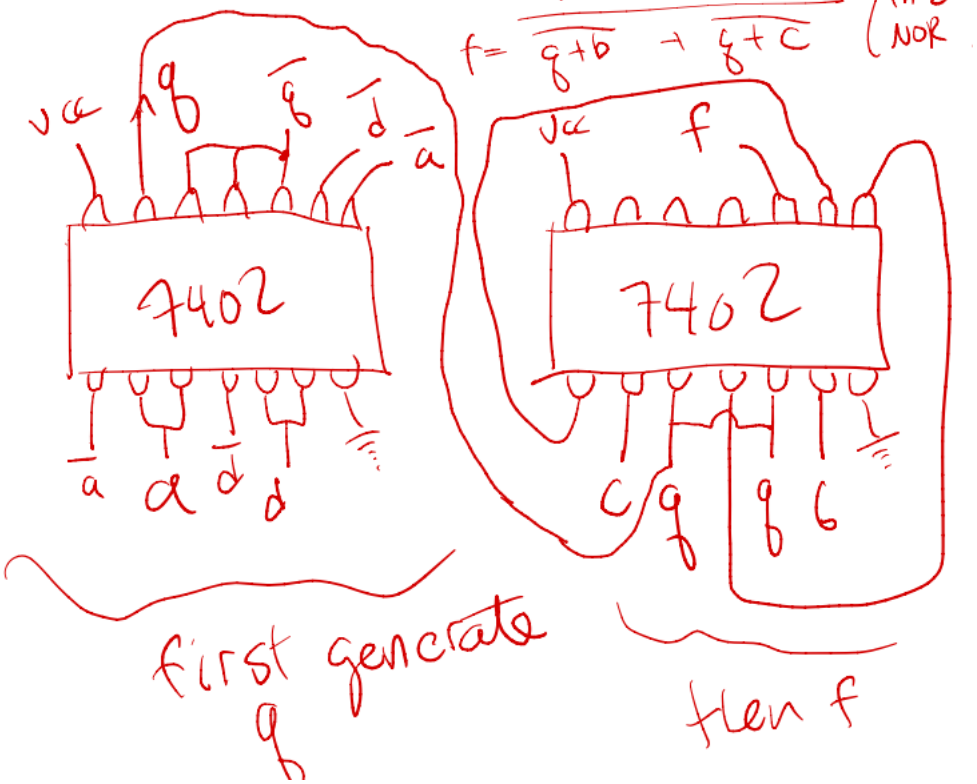
7402

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

let $g = \bar{a}+\bar{d} = \overline{\bar{a}+\bar{d}}$ (in 2-NOR)

$f = (g+b)(g+c)$

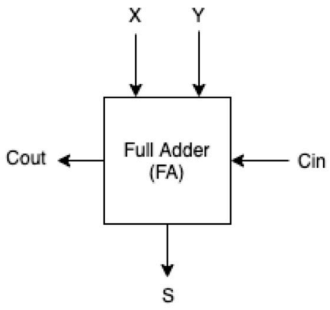
$f = \overline{\overline{g+b} + \overline{g+c}}$ (in 2-NOR)

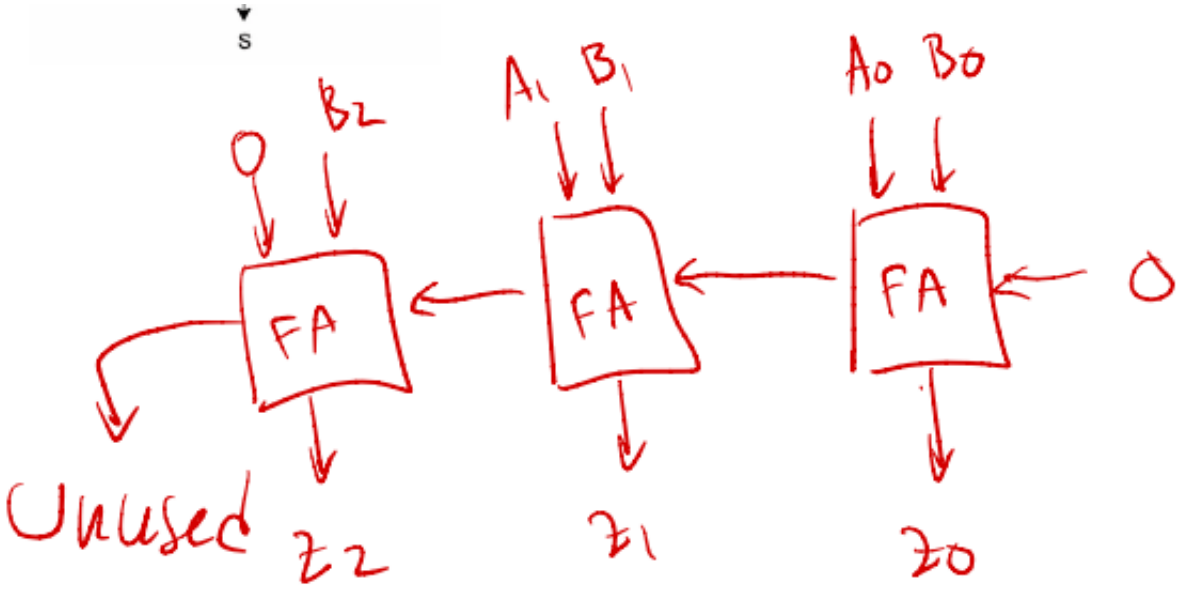


Question 5 continued …

Question 6 [10 Marks]

[4 marks] (a) A_{1-0} is a 2-bit-wide signal representing a binary number. B_{2-0} is a 3-bit-wide signal representing a second binary number. The range of A is restricted, in decimal, to [0:2] (i.e., 3 will never occur); the range of B is restricted to [0:5] (i.e., 6 and 7 will never occur). Design a circuit that computes $Z = A + B$ using full-adder blocks, as shown below. Z_{2-0} is a 3-bit-wide output signal.





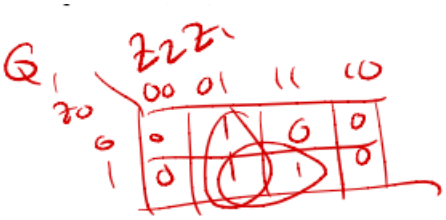
Question 6 continued ...

[6 marks] (b) Given the 3-bit-wide signal Z_{2-0} as computed in part (a). Design a circuit that computes $Q = Z \bmod 5$, where *mod* is the modulus operator (% operator in the C language). Q_{2-0} is a 3-bit-wide signal representing a binary number. Since $Q = Z \bmod 5$, Q is restricted to the range [0:4] (in decimal).

(i) Fill in the truth table for Q_{2-0} as functions of Z_{2-0}.

Z_2	Z_1	Z_0	Q_2	Q_1	Q_0
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

(ii) Use K-maps to find the minimum-cost implementations of Q_2, Q_1, and Q_0 as functions of Z_{2-0}. Show the three functions in sum-of-products (SoP) form.



$Q_0 = \bar{z}_2 z_0 + \bar{z}_0 z_2 z_1$

$Q_1 = \bar{z}_2 z_1 + z_1 z_0$

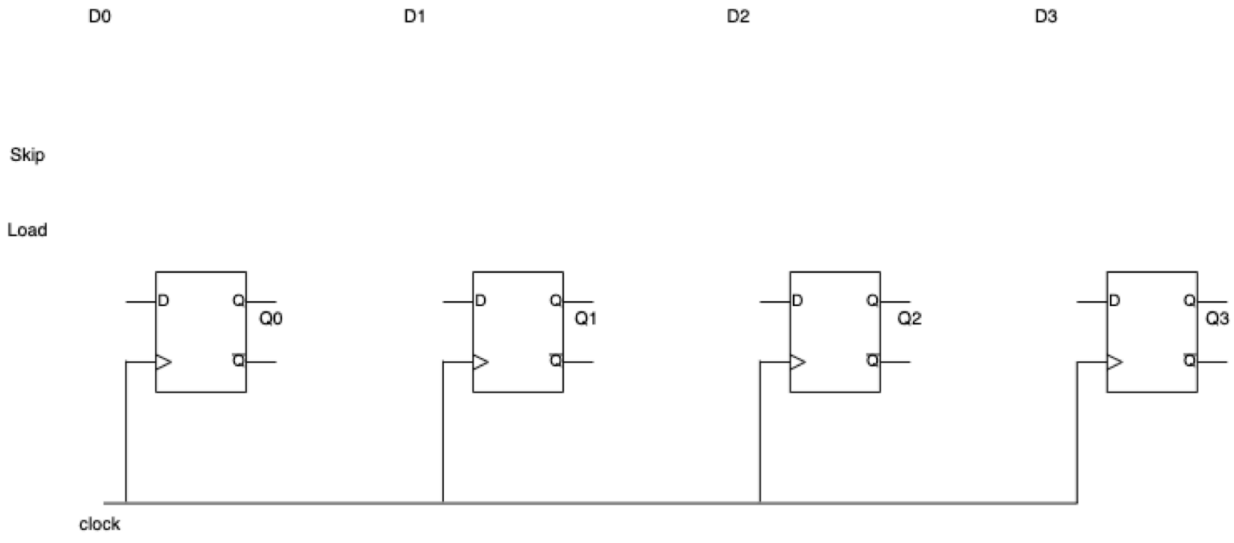
$Q_2 = z_2 \bar{z}_1 \bar{z}_0$

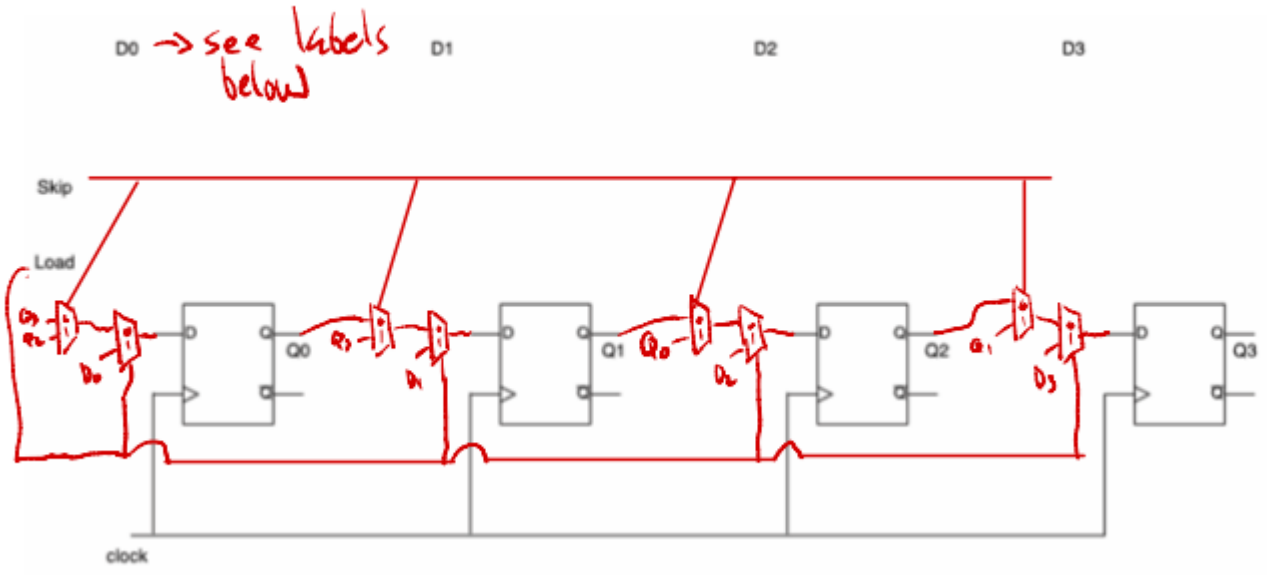
Question 7 [8 Marks]

[4 marks] (a) You must design a circular shift register that contains four D-type flip-flops and behaves in the following way: The Q outputs on the four flip-flops are called $Q0$-$Q3$. There are four inputs $D0$-$D3$, and also inputs $Skip$ and $Load$. When $Load$ is 1, the values on inputs $D0$-$D3$ are loaded into the flip-flops at the next rising clock edge (parallel load). Meaning that when $Load$ is 1, at the rising clock edge, $D0$ is transferred to $Q0$, $D1$ is transferred to $Q1$, and so on.

When $Load$ is 0, there are two possible behaviours: 1) When $Skip$ is 0, the values in the register shift right one position at each rising clock edge. That is, the value on $Q0$ is transferred to $Q1$, the value on $Q1$ is transferred to $Q2$, and so on, with the value of $Q3$ being transferred to $Q0$ (circular shift). 2) When $Skip$ is 1, the values in the register shift right by *two* positions at each rising clock edge – they "skip" a flip-flop. That is, the value on $Q0$ is transferred to $Q2$, the value on $Q1$ is transferred to $Q3$, and so on, with the value of $Q3$ being transferred to $Q1$.

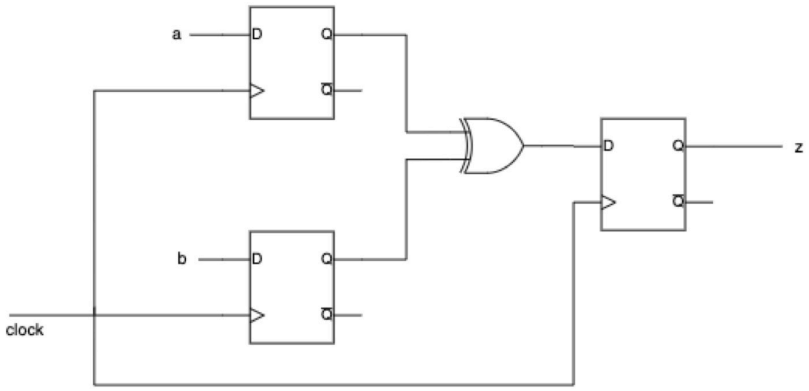
Complete the circuit diagram to achieve the behaviour described above. You may use any gates or multiplexers that you have learned about in class.



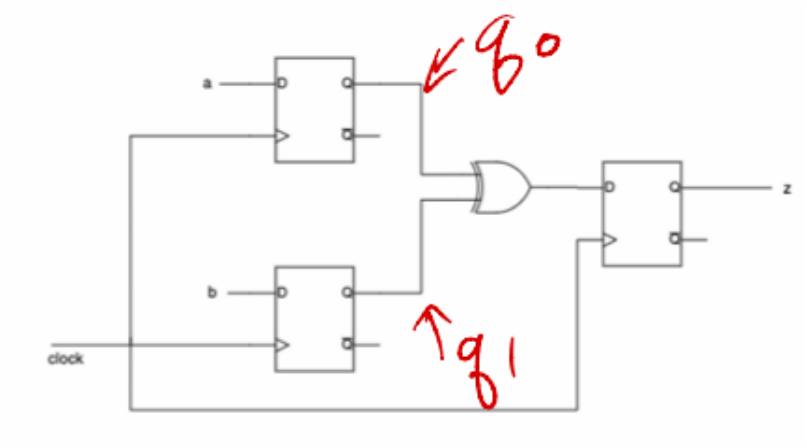

D0 → see labels below
D1
D2
D3
Skip
Load
Q0
Q1
Q2
Q3
clock

Question 7 continued ...

[4 marks] (b) Complete the Verilog to describe the behaviour of the circuit shown below.




```
module FFcircuit(a,b,z,clock);
    input a,b,clock;
    output reg z;
```




```
module FFcircuit(a,b,z,clock);
    input a,b,clock;
    output reg z;
```

reg q0, q1;
always @ (posedge clock)
begin
z <= q0 ^ q1;
q0 <= a;
q1 <= b;
end
endmodule

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