University of Toronto
Faculty of Applied Science and Engineering

Midterm Test
October 22, 2015

ECE253 – Digital and Computer Systems

Examiner – Prof. Stephen Brown

Print:

First Name . . . . Last Name . . . .

Student Number . . . .

1. There are **5** questions and **16** pages. Do **all** questions. The duration of the test is 1.5 hours.
2. **ALL WORK IS TO BE DONE ON THESE SHEETS.** THERE IS EXTRA SPACE ON PAGE 12 FOR ANY QUESTION IF YOU NEED TO USE IT.

   THERE ARE BLANK PAGES AT THE END OF THE EXAM THAT YOU CAN TEAR OFF TO USE FOR ROUGH WORK. YOU DON'T NEED TO HAND IN THESE EXTRA PAGES.
3. Closed book. No aids are permitted.
4. No calculators are permitted.

1 [16]	
2 [8]	
3 [10]	
4 [9]	
5 [10]	
Total [53]	

[16 marks] 1. Short answers:

[4 marks] (a) Perform the following number conversions.

i. 16’hFF03 to octal

Answer 16'o177403 1_111_111_100_000_011

ii. 12’b110100011101 to hexadecimal

Answer 12'hD1D 1101_0001_1101

iii. 9’o765 to hexadecimal

Answer 9'h1F5 1_1111_0101

iv. 16’d12800 to binary

Answer 16'b0011_0010_0000_0000 $(25*2^9)$

[2 marks] (b) Perform the following additions of binary numbers.

```
i.     0 0 1 0 1 1 0 0        ii.    1 1 1 1 1 1 1 1
       0 0 1 0 0 1 0 0               1 1 1 1 1 1 1 1
       0 0 1 0 1 0 1 1               1 1 1 1 1 1 1 1
     + 0 0 1 0 1 0 1 0             + 0 1 0 1 0 1 0 1
       10100101                    1101010010
```

[1 mark] (c) In Verilog, what is a *sensitivity list*?
Answer:

List of signals that directly affect the output of always block

or

List of signals that trigger always block execution

-0.5 for no always or incorrect mention of variable behaviour

[1 mark] (d) In Verilog when do we use the keyword *reg*?
Answer:

Variables that are assigned a value in an always block

-0.5 for for implying it's used as a sole output of always block

[1 mark] (e) In Verilog how do you write code that will *infer* a flip-flop(s)?

Answer:

Posedge/negedge clk in sensitivity list

-0.5 for not mentioning sensitivity list

-1 for showing wrong sensitivity list e.g., (posedge clk, D, Q)

[1 mark] (f) On the lab boards, is a light in a 7-segment display illuminated by driving it to 0 or 1?

Answer: 0

Is a red LED illuminated by driving it to a 0 or 1?

Answer: 1

[1 mark] (g) How many selector bits are needed for a 13-to-1 multiplexer?

Answer: 4

[1 mark] (h) What does the term *BCD* mean, and what is it used for?

Answer:

Binary **C**oded **D**ecimal. Represent decimal **digits** in 4-bit binary

+0.5 for name

+0.5 for description

[1 mark] (i) What does *active-low* mean?

Answer:

Circuit or element executes its function on a logical 0 input. Example: reset**n**

[1 mark] (j) Consider the snippet of Verilog code below:

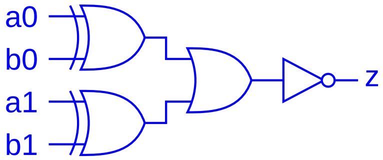
```
wire c;
wire [1:0] A, B;
assign A = 2'b11;
assign B = A & c;
```

What will be assigned to B[1] and B[0]?

Answer: B[1] = ... 0 . . . . . . , B[0] = c, 0, or 1'bx

[2 marks] (k) Draw a circuit that has two 2-bit inputs $A = a_1a_0$ and $B = b_1b_0$, and produces an output z. The value of z should be 1 if A = B, otherwise z should be 0. Draw the circuit in the space below, using logic gates. Make your circuit as simple as possible, using any of the AND, OR, NOT, or XOR gates.

Answer:



b1	b0	a1	a0	z
0	0	0	0	1
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	1
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	1
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	1

-1 for not simplified (5 gates or less)
-0.5 for invalid gate (e.g., NAND, NOR)

[8 marks] 2. Boolean Algebra:

[2 marks] (a) Prove the following Boolean relation using algebraic manipulation in two steps, using exactly two identities. Show your work and specify which identity is used in each of your two steps. The identities are listed at the end of this test.

Identity $(x + xy)z + x\overline{z} = x$

Absorption 13a ____________________ -1 incorrect rule or boolean logic

Combining 14a ____________________

[2 marks] (b) The following Boolean relation can be proved using algebraic manipulation in one step, using exactly one identity. Show the identity below.

Identity $((\overline{w \oplus x}) + \overline{y}) \cdot ((\overline{w \oplus x}) + z) = (\overline{w \oplus x}) + \overline{y}z$

Distributive 12a ____________________ -1 incorrect rule or boolean logic

[2 marks] (c) Prove the following Boolean relation using algebraic manipulation in two steps, using exactly two identities. Show your work and specify which identity is used in each of your two steps.

Identity $xz + yz + x + y = x + y$

Distributive 12a/Absorption 13a ____________________ -1 incorrect rule or boolean logic

Absorption 13a ____________________

[2 marks] (d) Prove the following Boolean relation using algebraic manipulation in three steps, using exactly three identities. Show your work and specify which identity is used in each of your three steps.

Identity $wy + xy + yz + (\overline{w} \cdot \overline{x})z = (w + x) \cdot y + (\overline{w + x}) \cdot z$

Consensus 17 ____________________ -0.5 incorrect rule or boolean logic

Distributive 12 ____________________

DeMorgan 15 ____________________

[10 marks] 3. Karnaugh maps:

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	1	1	0
01	0	1	1	1
11	1	0	1	1
10	1	0	1	1

(i)

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	1	1	0	1
01	0	0	0	1
11	1	0	0	0
10	1	0	1	1

(ii)

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	d	1	0	d
01	0	1	1	0
11	0	0	1	1
10	d	0	1	1

(iii)

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	1	1	0
01	0	0	1	1
11	0	1	1	0
10	0	0	1	1

(iv)

(a) For the function in Karnaugh map (i) above list **all minimal sum-of-products** solutions:

$x_2\overline{x_3} + x_1x_4 + \overline{x_2}x_3 \mathbf{+ x_1x_2}$

$x_2\overline{x_3} + x_1x_4 + \overline{x_2}x_3 \mathbf{+ x_1x_3}$

1.5 for each correct function
-0.5 each incorrect term

(b) For the function in Karnaugh map (ii) above list **all minimal product-of-sums** solutions:

$(\overline{x_1}+\overline{x_2}+x_3)(x_1+\overline{x_2}+\overline{x_3})(x_1+x_3+\overline{x_4})(\overline{x_1}+\overline{x_3}+x_4)$

-1 for including $(\overline{x_2}+\overline{x_4})$
-0.5 incorrect term
-1 all terms complemented

(c) For the function depicted in Karnaugh map (iii) above list **all prime implicants**:

$\overline{x_1}x_2\overline{x_3}$	$\overline{x_2}\,\overline{x_4}$
$x_1x_2x_4$	$\overline{x_1}\,\overline{x_3}\,\overline{x_4}$
x_1x_3	$x_2\overline{x_3}x_4$

-0.5 each missing term
-1 each incorrect term

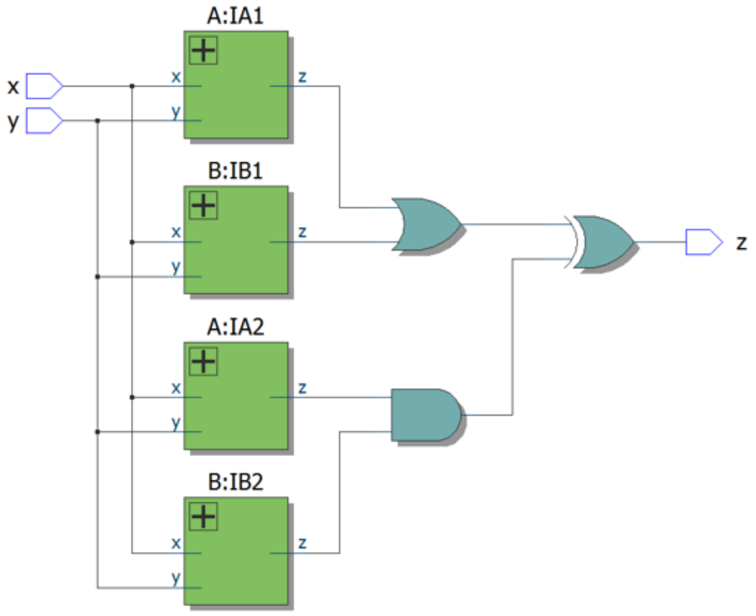
(d) For the function depicted in Karnaugh map (iv) above, let $g = x_3 \oplus x_4$. Fill in the logic expression below. Make the simplest expression you can, using g as indicated.

$f =$ x_1 $\cdot(g) +$ x_2 $\cdot(\overline{g})$

-0.5 each suboptimal term
-1 for swapping the two answers

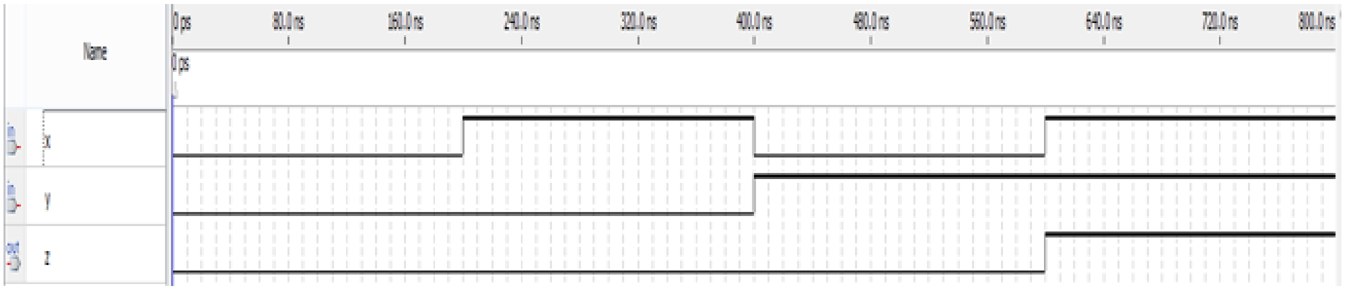
[9 marks] 4. Debugging question:

You and your lab partner, *Biff*, have created a hierarchical design. Biff stayed home to watch baseball, and so you have to debug the circuit on your own. A top-level view of your design is shown below. Biff created the submodules A and B, and you have not looked at his/her source code. You compile the top-level design and download it onto the lab board—it does *not* work properly.

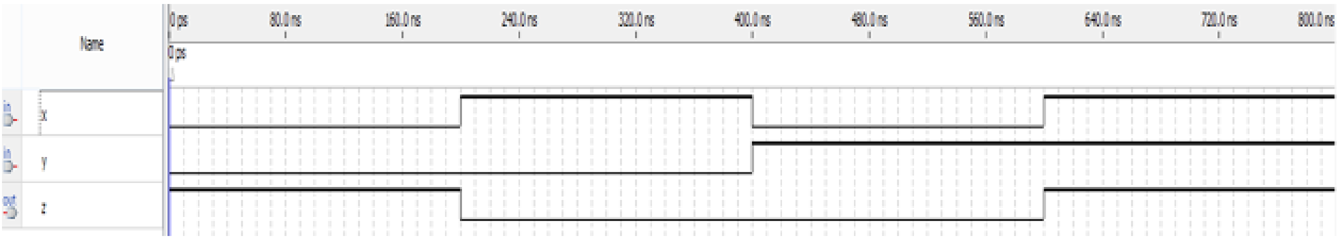


According to Biff, module A is supposed to implement the function $z = (x \oplus y)\ \&\ x$ and module B is supposed to implement the function $z = (x\ \&\ y) \oplus (x \mid y) \oplus (1'b1)$. Biff has provided you with simulation output waveforms for modules A and B, shown below. (In case it is unclear in the diagram, x is the top waveform, y is in the middle, and z is on the bottom.)

Simulation waveforms for A:



Simulation waveforms for B:



[3 marks] Of course, Biff has not bothered to check if these waveforms show the correct behavior, so you need to check them yourself. Do the simulation results show a problem in one or both of the submodules? Describe any problem(s) you see in the space below (be specific):

Answer:

A implemented incorrectly +1

x=1, y=1 should be z=0 +1

x=1, y=0 should be z=1 +1

-0.5 if B is stated as incorrect

[2 marks] Being clever, or at least more so than Biff, you notice that the expressions for A and B can be optimized.

Give an optimized logic expression for A. You do not need to show your work in generating this expression.

+0.5 correct

+0.5 optimal

Answer: ...... $z = x\bar{y}$ ......

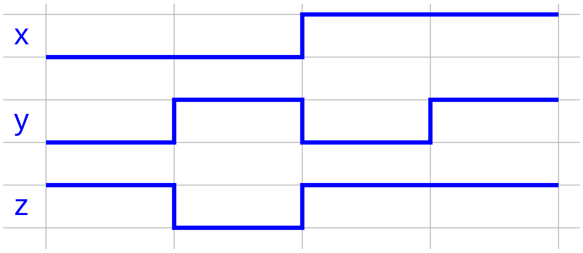
Give an optimized logic expression for B. You do not need to show your work in generating this expression.

+0.5 correct
+0.5 optimal

Answer: $z = \overline{x \oplus y}$; $z = x \odot y$; $z=(x+\bar{y})(\bar{x}+y)$; $z=xy+\bar{x}\bar{y}$

[2 marks] In the space below, show correct waveforms for the output of the top-level design, for all combinations of x, y. Show only the primary inputs x and y in your answer, and the primary output z.

Answer:



+0.5 for each output of z

[2 marks] Show a simplified logic expression for the final output z of the top-level design. Show the simplest expression that you can.

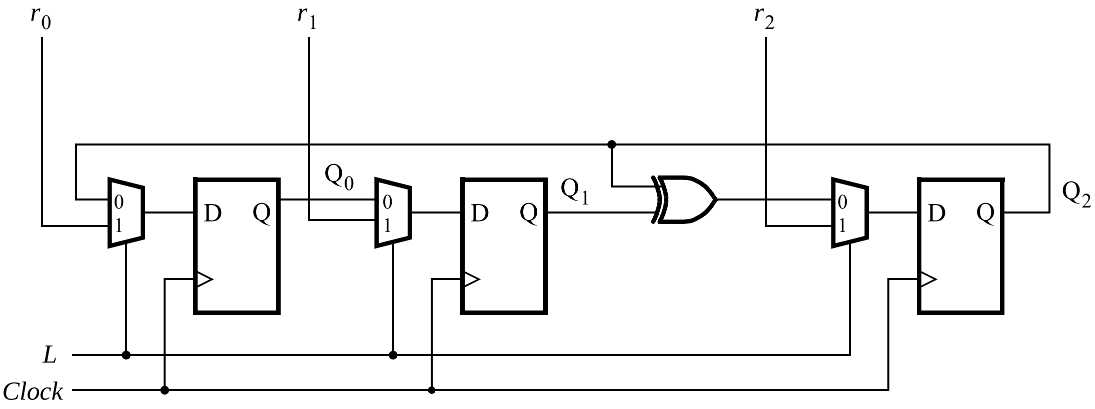
Answer:

$z = x + \bar{y}$

+1 for correct
+1 for optimal
Considered correct if it implements above waveform

[10 marks] 5. Verilog Question:

Consider the sequential circuit below:



[4 marks] (a) Assume that you want to implement hierarchical Verilog code for this circuit, using three instantiations of a submodule that has a flip-flop and multiplexer in it. Write a Verilog module named *MUXDFF* for this submodule.

Answer:

```
module MUXDFF (
   input clock,
   input L,
   input a,
   input b,
   output reg Q);

  always @(posedge clock)
    Q <= L ? b : a;

endmodule
```

+2 for the flip-flop
+2 for the multiplexer
~ -1 per error

... continued on the next page

[4 marks] (b) Write a top-level Verilog module for the sequential circuit. Assume that you are going to implement the circuit on the DE1-SoC board. Connect the R inputs to the *SW* switches, connect *Clock* to *KEY*$_0$, and *L* to *KEY*$_1$. Connect the Q outputs to the red lights *LEDR*.

Answer:

```
module top (
   input [2:0] SW,      // R
   input [1:0] KEY,     // L and clock
  output [2:0] LEDR);   // Q

   wire [2:0] Q;
   assign LEDR = Q;

   MUXDFF m0 (KEY[0], KEY[1],   Q[2]   , SW[0], Q[0]);
   MUXDFF m1 (KEY[0], KEY[1],   Q[0]   , SW[1], Q[1]);
   MUXDFF m2 (KEY[0], KEY[1], Q[1]^Q[2], SW[2], Q[2]);

endmodule
```

+1 for instantiating an XOR
+1 for 3 instances of MUXDFF
+2 for connecting everything correctly

[2 marks] (c) When $L = 0$, this sequential circuit counts through a particular sequence. Assuming that the flip-flops originally store $Q_2Q_1Q_0 = 100$, what is the counting sequence of this circuit?

Answer:

100, 101, 111, 011, 110, 001, 010, [100, ...]

+1 first 4 terms correct
+1 next 3 terms (and recognizing the repeat)
-0.5 if bit ordering reversed, but otherwise correct:

$Q_0Q_1Q_2$ = 100, 010, 001, 101, 111, 110, 011, [100, ...]

Extra answer space for any question on the test, if needed:

Boolean Identities

12a.	$x \cdot (y + z) = x \cdot y + x \cdot z$	*Distributive*
13a.	$x + x \cdot y = x$	*Absorption*
14a.	$x \cdot y + x \cdot \overline{y} = x$	*Combining*
15a.	$\overline{x \cdot y} = \overline{x} + \overline{y}$	*DeMorgan's theorem*
16a.	$x + \overline{x} \cdot y = x + y$	
17a.	$x \cdot y + y \cdot z + \overline{x} \cdot z = x \cdot y + \overline{x} \cdot z$	*Consensus*