CPSC 121 Midterm 1
Tuesday, February 12th, 2013

[1] 1. Do you want your tutorial attendance to count towards your grade?

If you answered 'YES', then 1% of your final course mark will be based on your tutorial attendance (# attended divided by total number minus 2, but no more than full credit). Online quizzes will be worth 5%. If you answered 'NO', then tutorial attendance is worth nothing in your mark, and online quizzes count for 6% of your final course mark.

[7] 2. Determine the validity of the following arguments *using rules of inference*. You may use inference rules from class or one of the texts (Epp or Rosen), or you may use a logical equivalence from class or the texts. You must state what inference rule or logical equivalence you use at each step.

1. $\sim p \rightarrow (u \vee v)$
2. $s \rightarrow \sim q$
3. $q \wedge (p \rightarrow r)$
4. $\sim u$
5. $\sim r \vee s$

$\therefore v$

Solution:

6.	q	specialization from (3)
7.	$\sim\sim q$	double negative law from (6)
8.	$\sim s$	modus tollens from (2) and (7)
9.	$\sim r$	elimination from (5) and (8)
10.	$p \rightarrow r$	specialization from (3)
11.	$\sim p$	modus tollens from (9) and (10)
12.	$u \vee v$	modus ponens from (1) and (11)
13.	v	elimination from (4) and (12)

[4] 3. What is the decimal value of 1001010 if the sequence of bit is interpreted as

[2] a. A 7 bit, unsigned integer?

Solution: The value is $2^6 + 2^3 + 2^1 = 64 + 8 + 2 = 74$.

[2] b. A 7 bit, signed (two's complement) integer?

Solution: The value is $-2^7 + 74 = -128 + 74 = -54$. Alternatively, we could compute the negation of 1001010 using two's complement to get 0110110, convert it to decimal which gives $2^5 + 2^4 + 2^2 + 2^1 = 32 + 16 + 4 + 2 = 54$, and then stick a minus sign in front of the answer once again getting -54.

[6] 4. Given the following definitions:

- S: the set of all canadian senators.
- T: the set of all canadian province and territories.
- $C(x)$: senator x has been charged with a crime.
- $B(x, y)$: senator x was born in province or territory y.
- $P(x, y)$: senator x belongs to political party y.

translate each of the following predicate logic statements into English. To obtain full marks, your English translation will need to sound reasonably natural.

[3] a. $\forall x \in S, (C(x) \wedge P(x, \text{"Conservative"})) \rightarrow \sim B(x, \text{"Yukon"})$

Solution : No conservative senator who has been charged with a crime was born in Yukon.

[3] b. $\forall x \in S, \forall y \in S, (C(x) \wedge C(y)) \rightarrow \forall t \in T, \sim B(x, t) \vee \sim B(y, t)$

Solution : No two senators who have been charged with a crime come from the same province.

[9] 5. Given the same definitions as for the previous question (rewritten here for convenience):

- S: the set of all canadian senators.
- T: the set of all canadian province and territories.
- $C(x)$: senator x has been charged with a crime.
- $B(x, y)$: senator x was born in province or territory y.
- $P(x, y)$: senator x belongs to political party y.

translate each of the following English sentences into predicate logic.

[3] a. No liberal senator was born in Alberta.

Solution : $\forall x \in S,\ P(x, \text{"Liberals"}) \rightarrow \sim B(x, \text{"Alberta"})$

[3] b. Only senators from Québec can belong to the Bloc Québécois.

Solution : $\forall x \in S,\ P(x, \text{"Block Québécois"}) \rightarrow B(x, \text{"Québec"})$

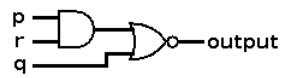
[3] c. Senators from at least two different provinces or territories have been charged with a crime.

Solution : Note that we do not need to specify that $x \neq y$ in the following solution, because a senator can only be born in one province[1]. The answer is thus

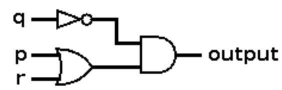
$$\exists x \in S, \exists y \in S, \exists p \in T, \exists q \in T,\ p \neq q \wedge C(x) \wedge C(y) \wedge B(x, p) \wedge B(y, q)$$

[1] we will assume none of their mothers was unlucky enough to give birth just as the car was crossing a provincial boundary on the way to a hospital

[9] 6. Here are five propositions or circuits:



A)



B)

C)	$((p \oplus q) \vee r)$
D)	$\exists x \in D, \forall y \in D, K(x,y) \wedge \sim(R(x) \vee S(x))$
E)	$\exists x \in D, \exists y \in D, \sim R(x) \wedge \sim S(x) \wedge K(x,y)$

Each of the propositions below is logically equivalent to one of propositions or circuits (A) through (E). In each case, state which of (A) through (E) it is logically equivalent to, and then prove the logical equivalence.

[5] a. $(p \to q) \to (r \wedge \sim q)$

Solution :

$$\begin{aligned}
(p \to q) \to (r \wedge \sim q) &\equiv (\sim p \vee q) \to (r \wedge \sim q) && \text{by the definition of } \to \\
&\equiv \sim(\sim p \vee q) \vee (r \wedge \sim q) && \text{by the definition of } \to \\
&\equiv (\sim\sim p \wedge \sim q) \vee (r \wedge \sim q) && \text{by the DeMorgan's law} \\
&\equiv (p \wedge \sim q) \vee (r \wedge \sim q) && \text{by the double negative law} \\
&\equiv (p \vee r) \wedge \sim q && \text{by the distributive law}
\end{aligned}$$

which is the output of circuit B.

[5] b. $\sim\forall x \in D,\ (\exists y \in D,\ K(x,y)) \to (R(x) \vee S(x))$

Solution :

$$\begin{aligned}
&\sim\forall x \in D,\ (\exists y \in D,\ K(x,y)) \to (R(x) \vee S(x)) \\
&\quad\equiv \exists x \in D,\ \sim((\exists y \in D,\ K(x,y)) \to (R(x) \vee S(x))) && \text{by the generalized De Morgan's law} \\
&\quad\equiv \exists x \in D,\ \sim(\sim(\exists y \in D,\ K(x,y)) \vee (R(x) \vee S(x))) && \text{by the definition of } \to \\
&\quad\equiv \exists x \in D,\ \sim\sim(\exists y \in D,\ K(x,y)) \wedge \sim(R(x) \vee S(x))) && \text{by the deMorgan's law} \\
&\quad\equiv \exists x \in D,\ (\exists y \in D,\ K(x,y)) \wedge \sim(R(x) \vee S(x))) && \text{by the double negative law} \\
&\quad\equiv \exists x \in D,\ (\exists y \in D,\ K(x,y)) \wedge (\sim R(x) \wedge \sim S(x)) && \text{by the deMorgan's law} \\
&\quad\equiv \exists x \in D,\ \exists y \in D,\ (K(x,y) \wedge (\sim R(x) \wedge \sim S(x))) && \text{by the distributive law} \\
&\quad\equiv \exists x \in D,\ \exists y \in D,\ \sim R(x) \wedge \sim S(x) \wedge K(x,y) && \text{by the ass. and comm. laws}
\end{aligned}$$

which is the same as E.

[6] 7. Short Answers

[3] a. To determine whether or not a 64-bit binary integer n is 0, a computer needs to OR all of n's bits together. If the output of the OR gate is false, then the integer is 0. If only two-input and three-input OR gates are available, many of them will need to be connected to deal with n. Should we connect them as a chain or as a tree? Justify your answer briefly.

Solution : We should connect them as a tree: the length of the longest path in the tree is about $\log_2 b$ where b is the number of bits, and so the output will become available in approximately 6 gate delays. With a chain, it would take roughly 63 gate delays before the output becomes available.

[3] b. Suppose that you want to prove that two propositions with variables p, q, r, s and t are **not** logically equivalent. What would the shortest (and simplest) such proof look like?

Solution : You could provide values of p, q, r, s and t for which one proposition is true, and the other proposition is false.

[8] 8. Design a circuit that takes as input two two-bit unsigned integers a_1a_0 and b_1b_0, and outputs the minimum x_1x_0 of the two inputs. For instance, if the first input is 2 ($a_1 = 1$ and $a_0 = 0$) and the second input is 1 ($b_1 = 0$ and $b_0 = 1$) then the output should be 01. Hints:

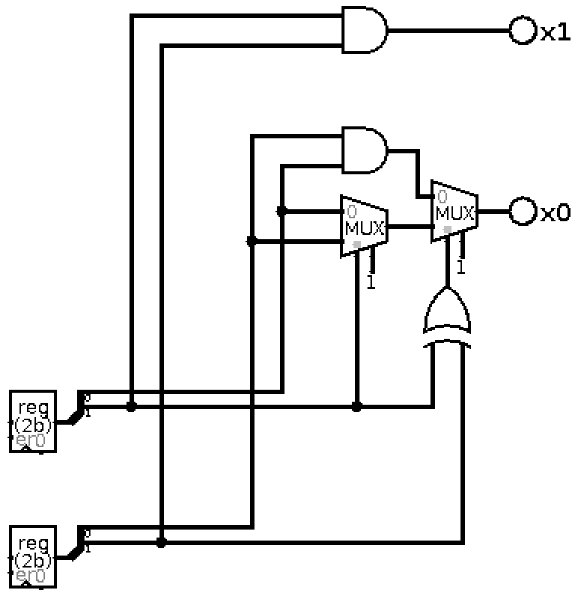
- One of the two outputs x_1, x_0 is very easy to compute (only one gate is needed).
- To compute the second output, you will need to consider two cases; deal with each case separately and then combine them at the end.

You may use AND, NAND, NOR, NOT, OR and XOR gates, as well as multiplexers.

Solution : We can compute x_1 by looking only at a_1 and b_1: x_1 will be the smallest of the two, which we can achieve using a single AND gate. We will divide the computation of x_0 into two cases (using a multiplexer):

- If $b_1 = a_1$ then we use the smallest of a_0, b_0 (another AND gate).
- If $b_1 \neq a_1$ then we want to select that of a_0, b_0 that corresponds to whichever one of a_1, b_1 that was equal to 0; this can also be done using a multiplexer.

We therefore get the following circuit:


x1
x0
MUX
MUX
reg
(2b)
reg
(2b)