Exam No

The University of British Columbia

Computer Science 121

Midterm 1

February 9, 2012

Time: 70 minutes

Total marks: 65

Instructors (Circle one): Section 202 - Patrice Belleville

Section 203 & BCS - George Tsiknis

Name____Student No______________
(PRINT) (Last) (First)

Signature____

This examination has 7 pages.

Check that you have a complete paper.

This is a closed book exam, but you may use a sheet of 8.5x11 inches paper with your notes.

Answer all the questions on this paper.

Give very **short but precise** answers. Always use point form where it is appropriate.

Work fast and do the easy questions first. Leave some time to review your exam at the end.

The marks for each question are given in []. Use this to manage your time.

Good Luck

MARKS	
1	
2	
3	
4	
5	
6	
Total	

Question 1. [9] Logical forms

a. [3] Write a short propositional logic expression that corresponds to the following truth table:

p	q	r	s
F	F	F	T
F	F	T	T
F	T	F	F
F	T	T	T
T	F	F	T
T	F	T	F
T	T	F	T
T	T	T	T

Expression:

i. **(p <--> q) ∨ (q <--> r)**
OR
ii. **~((~p ∧ q ∧ ~r) ∨ (p ∧ ~q ∧ r))**

b. **[3]** Determine whether the following statement is a tautology a contradiction or neither, using a method of your choice:

i. **((p → q) ∧ (p → ~q)) → ~p**

truth table:
OR
a proof like:

(p -> q) ^ (p -> ~q) === (~p v q) ^ (~p v ~q)
=== ~p v (q ^ ~q)
=== ~p v F
=== ~p

c. [3] If p and q are statements , then
p unless q means **if ~q then p** .
Use this definition to translate the following sentence to propositional logic:
The safe will not open unless you provide the code and if you don't provide the code, I'll call the police.
No variable is allowed to correspond to a proposition that contains logical operators (for instance, you can not write that the answer is "p", where "p" corresponds to the whole sentence). You may circle portions of the printed statement and write the variable name beside the circle to reduce writing time.

s = safe will open
c = provide code
p = I'll call the police

Sentence is: (~c → ~s) ∧ (~c → p)

Question 2. [12] Logical Equivalences and Digital Circuits

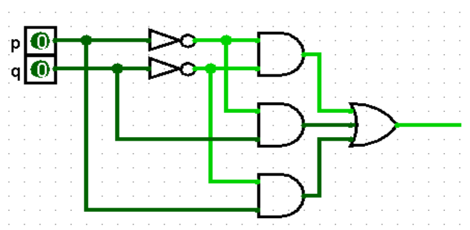
a. [6] Prove the following equivalence. Use a formal logical equivalence proof showing the rule that is applied at each step, and start your proof from the left-hand side of the equivalence.

$$\sim(p \to q) \vee (p \wedge q) \equiv p$$

Proof:

$\sim(p \to q) \vee (p \wedge q)$	**given**
$\sim(\sim p \vee q) \vee (p \wedge q)$	**$\to$ definition**
$(\sim\sim p \wedge \sim q) \vee (p \wedge q)$	**de Morgan**
$(p \wedge \sim q) \vee (p \wedge q)$	**double negation**
$p \wedge (\sim q \vee q)$	**absorption**
$p \wedge T$	**negation**
p	**identity**

b. [6] Consider the following digital circuit. The circuit is input/output equivalent to another one that uses at most two logical gates with at most two inputs. Show the simpler circuit and justify your answer.



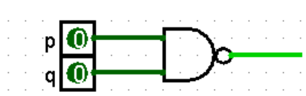
The give circuit can be represented by the proposition:

$$(\sim p \wedge \sim q) \vee (\sim p \wedge q) \vee (p \wedge \sim q)$$

which is equivalent to

$(\sim p \wedge (\sim q \vee q)) \vee (p \wedge \sim q)$	**distribution**
$(\sim p \wedge T) \vee (p \wedge \sim q)$	**negation**
$\sim p \vee (p \wedge \sim q)$	**identity**
$(\sim p \vee p) \wedge (\sim p \vee \sim q)$	**distribution**
$T \wedge (\sim p \vee \sim q)$	**negation**
$(\sim p \vee \sim q)$	**identity**
$\sim (p \wedge q)$	**identity**

The circuit can be implemented with a NAND gate or AND and a NOT gate:



Question 3. [8] Number representation

The following table shows the first few powers of 2 which may be helpful for this set of questions.

power	value
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32

power	value
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1,024
2^{11}	2,048

power	value
2^{12}	4,096
2^{13}	8,192
2^{14}	16,384
2^{15}	32,768
2^{16}	65,536
2^{20}	1,048,576

a. [2] What is the binary representation of the decimal value 109 in 8 bits?

01101101

b. [2] What decimal value is represented by the 6-bit **unsigned** binary number 110010?

50

c. [2] What decimal value is represented by the 6-bit **signed** binary number 110010?

-14

d. [2] What is the binary representation of the decimal value -15 in 8 bits?

```
      00001111     15
      11110000
+1    00000001
      11110001    -15
```

<u>Question 4.</u> [10] Using logic to model and solve problems

Design a circuit that accepts as input a 4-bit binary number and produces a two bit output that indicates the highest bit of the input number that is 1. For instance

input 0110 will produce 10 meaning that the highest 1-bit is bit 2
" 1111 " " 11 " " " " " " 3
" 0001 " " 00 " " " " " " 0

You may assume that we never input 0000 to this system (imagine that we have another circuit that tests for 0000 before it passes the inputs to this system.)

a. [3] What is the formula for the first (higher) bit of the output? (Hint: don't try to create a long truth table for this; you should be able to directly determine the formula.)

Let the input be `b3, b2, b1, b0:`

The formula for the highest bit of the output is:

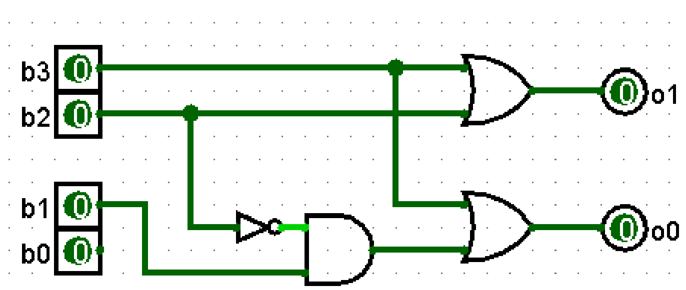
b3 ∨ b2

b. [3] What is the formula for the second (lower) bit of the output? (Hint: again, you should determine the formula directly).

The formula for the lowest bit of the output is:

b3 ∨ (~b2 ∧ b1)

c. [4] Now design the circuit for this problem.



Question 5. [12] Propositional Proofs

a. [6] Consider the following propositional logic proof. At each step indicate the missing part which is either the conclusion or the rule that is used for the step.

1.	**$p \vee q$**	**premise**
2.	**$q \to s$**	**premise**
3.	**$p \wedge r \to w$**	**premise**
4.	**$\sim s$**	**premise**
5.	**$\sim q \to m \wedge r$**	**premise**
6.	**$\sim q$**	**2, 4, modus tollens**
7.	**p**	**1, 6, elimination**
8.	**$m \wedge r$**	**5, 6, modus ponens**
9.	**r**	**8, specialization**
10.	**$p \wedge r$**	**7, 9, conjunction**
11.	**w**	**3,10, modus ponens**

b. [6] Determine whether the following argument is valid or not. If you think it is invalid provide a truth value assignments that proves your claim. Otherwise provide a proof for it.

1. $p \to q$
2. $m \vee s$
3. $\sim s \to \sim r$
4. $\sim q \vee s$
5. $\sim s$
6. $\sim p \wedge m \to u$

$\therefore$ $\sim u$

The argument is not valid. Here is an assignment that makes the premises true and the conclusion false:

s = F
q = F
p = F
m = T
r = F
u =T

<u>Question 6.</u> [14] Predicate Logic
Databases usually store information about entities(things) and relationships between the entities. For instance, a company's database will stores information about the employees, the departments they are in, their salaries, etc. To reason about the data stored in the database we can express this information in logic and use it to derive the properties we need.
Let's consider the employee database of some company. We use the following predicates and functions to represent the basic information in this database:

- `E` is the domain of all employees
- `D` is the domain of all departments in the company
- `Works(e, d)` : it is true iff employee e works in department d
- `Soccer(e)` : it is true iff employee e is in the company's soccer team
- `(salary e)` : a function that returns the salary of employee e

Translate each of the following statements to Predicate Logic:

a) [4] **"Every employee works in some department and every department has some employee working in it"**

(∀e∈E, ∃d∈D, W(e, d)) ∧ (∀d∈D, ∃e∈E, W(e, d))

b) [5] **"Every employee works in exactly one department"**

∀e∈E, ∃d∈D, W(e, d) ∧ ~∃d1∈D, W(e, d1) ∧ ~ d=d1
OR
∀e∈E, ∃d∈D, W(e, d) ∧ ∀d1∈D, W(e, d1) → d=d1

c) [5] **"The soccer players in the finance department earn more that $100,000 each"**

∀e∈E, W(e, finance) ∧ S(e) → (salary e) > 100000