UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

JF B.A.Moderatorship (Computer Science) **Trinity Term 2010**

Digital Logic Design I & II

Friday 30th April 2010 **RDS-Main** **14.00 – 17.00**

Dr. B. A. Coghlan

Please answer FOUR questions in total. All questions carry equal marks.

Please use separate answer books for each question.

This entire question paper must be handed up at the end of the examination.

Log tables and graph paper are available from the invigilators, if required.

Non-programmable calculators are permitted for this examination – please indicate the make and model of your calculator on each answer book used.

You may not start this examination, until you are instructed to do so by the Invigilator.

Q1. Imagine that you are debugging software. All the errors you encounter involve variables or constants. In the first version there are some errors involving string variables. There are also some syntax errors and some errors involving constants, both groups of which are in function calls. In the second version there are some errors that do not involve syntax and some that involve constants, neither groups of which are in function calls - there are also some syntax errors involving variables, and some syntax errors in function calls that do not involve strings. Algebraically reduce (to their simplest form) the Boolean expressions that describe:
(a) the union of the two versions [15 marks]
(b) the intersection of the two versions [10 marks]
No marks will be given for a non-algebraic solution.

Q2. Using Karnaugh maps, design an arithmetic and logic unit (ALU) that can perform functions of three 1-bit variables X, Y and Z, as shown in Table 1:

M1	M0	Function	Result	Comment
0	0	Add	X + Y + Z	+ is arithmetic add
0	1	Subtract	X – Y – Z	– is arithmetic minus
1	0	AND	X · Y · Z	· is Boolean AND
1	1	OR	X + Y + Z	+ is Boolean OR

Table 1: Function table for Question 2 [25 marks]

Q3. A driving licence tester evaluates candidates on five criteria, where the licence is granted if the candidate passes four or five of the criteria A, B, C, D and E. Using the Quine-McCluskey method, derive the logic function Y that decides whether a candidate receives their licence. [25 marks]

Q4. Using implication tables and merger diagrams, minimize the primitive flow table shown in Table 2:

	00	01	11	10
A	**A, 0**	**B,** ×	×, ×	**E,** ×
B	**A,** ×	**B, 0**	**C,** ×	×, ×
C	×, ×	**D,** ×	**C, 0**	**H,** ×
D	**A,** ×	**D, 1**	×, ×	×, ×
E	**A,** ×	×, ×	**F,** ×	**E, 0**
F	×, ×	**G,** ×	**F, 0**	**H,** ×
G	**A,** ×	**G, 0**	×, ×	×, ×
H	**A,** ×	×, ×	×, ×	**H, 0**

Table 2: Flow table for Question 4 [25 marks]

Q5. Assume the state equations for a control path Finite State Machine (FSM) are:

$A \leftarrow \sum (1, 3, 5, 7, 10, 11, 14, 15)$

$B \leftarrow \sum (0\text{-}8, 10, 12, 14)$

$C \leftarrow \sum (0\text{-}5, 8\text{-}13)$

Using JK flip-flops, complete the design of the FSM. [25 marks]

Q6. Design an algorithmic state machine that behaves as a 3-level push-down stack, with two 1-bit control inputs PUSH and POP, a 1-bit data input X, a 1-bit data output Y, and three 1-bit data registers A, B and C. When PUSH is at logic 1 the stack pushes down $A \leftarrow X$, $B \leftarrow A$ and $C \leftarrow B$. When POP is at logic 1 the stack pops up $A \leftarrow B$, $B \leftarrow C$ and $C \leftarrow 0$. If PUSH and POP are both at logic 1 the stack is unchanged. Assume the output Y shows the present value of A. Design the control path finite state machine using set assignment (one flip-flop per state), and the data path with D flip-flops. [25 marks]