UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

B.A.Moderatorship (Computer Science) **Trinity Term 2011**
Junior Freshman Examination

CS1023 & CS1024: Digital Logic Design I & II

Saturday 7th May 2011 **RDS Main Hall** **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. (a) State Huntington's Postulates. [5 marks]

(b) Use Huntington's Postulates to reduce to its simplest sum of products:

$F_1 = (A + B)(A + C)$ [5 marks]

(c) Use Huntington's Postulates to reduce to its simplest sum of products:

$F_2 = (A + B)(A' + C)$ [5 marks]

(d) Repeatedly apply de Morgan's Laws to simplify:

$F_3 = (AB(C+D') + A'C'(BD'+B'D))'$ [5 marks]

(e) Using de Morgan's Laws and the relationships between minterm and maxterm numbers, derive the canonical sum of products form for:

$F_4 = (AB)'(B+C'D)(A+D')$ [5 marks]

No marks will be given for a non-algebraic solutions.

Q2. An arithmetic and logic unit (ALU) that can perform the functions of three 1-bit input variables **X**, **Y** and **Z** shown in Table 1 has two outputs: (a) a primary output F_1, and (b) an output F_2 that is only used for carry/borrow. The desired function is selected using the mode inputs M_1 and M_0 as shown. Using the Quine-McCluskey method, derive the logic function for the primary output F_1.

M_1	M_0	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	0	OR	X + Y + Z	+ is Boolean OR

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

Q3. (a) Using Karnaugh Maps, and assuming a 4-bit binary-coded-decimal (BCD) input, derive the equations for a seven-segment decoder for decimal digits 0 through to 9, assuming the outputs are undefined for any other combinations. [15 marks]

(b) Draw the logic diagram of a PAL that implements the seven-segment decoder of Q3(a) above. [10 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**, x	x, x	**E**, x
B	**A**, x	**B**, **0**	**C**, x	x, x
C	x, x	**D**, x	**C**, **0**	**H**, x
D	**A**, x	**D**, **1**	x, x	x, x
E	**A**, x	x, x	**F**, x	**E**, **0**
F	x, x	**G**, x	**F**, **0**	**H**, x
G	**A**, x	**G**, **0**	x, x	x, x
H	**A**, x	x, x	x, x	**H**, **0**

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

Q5. Design a finite state machine (FSM) that implements a 3-bit universal counter/shifter with three mode request inputs, **M2**, **M1** and **M0**, three data inputs **D2**, **D1** and **D0**, and three data outputs **Y2**, **Y1** and **Y0**. At the next active clock edge the FSM performs the requested function shown in Table 2. Using D flip-flops, complete the design of the FSM.

M2	**M1**	**M0**	**Function**
0	0	0	No change
0	0	1	Load Y←D
0	1	0	Count up
0	1	1	Count down
1	0	0	Shift left
1	0	1	Shift right

Table 2: Function table for Question 5

[25 marks]

Q6. In the "Lamp Handball" laboratory experiment you implemented an electronic game of single-player handball using lamps to simulate a moving ball, and a toggle switch to simulate a bat. Starting with an "ASM Diagram", design an algorithmic state machine that implements a two-player game that behaves as follows. There are **8** lamps and the ball moves left-to-right or vice-versa at a rate of one lamp position per second. The ball is hit right by the left player **X** by activating a **HITX** switch, and is hit left by the right player **Y** by activating a **HITY** switch. The game is initialised by activating a **RESET** switch, which places the ball at the far right waiting for player **Y** to hit it. If a player hits the ball too early or too late, the ball disappears and that player loses. Use an 8-bit universal shift register in the datapath to simulate the moving ball. Use JK Flip-flops in the control path. [25 marks]