XCS10231

Digital Logic Design I & II

UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

Integrated Computer Science Programme **Trinity Term 2012**
Junior Freshman Examination

CS1023 & CS1024: Digital Logic Design I & II

Friday 11 May 2012 **Luce Lower** **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.

Digital Logic Design I & II

Q1. (a) What are the relationships and differences between Boolean Algebra, Huntington's Postulates, and Shannon's Switching Algebra? [5 marks]

(b) State Huntington's Postulates. [5 marks]

(c) Using truth tables, prove distributivity for 3 variables. [5 marks]

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

F(A,B,C,D,E,F) = (A + B)(A' + C) + (D + E)(D' + F) [5 marks]

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

F(A,B,C,D)= (AB(C+D') + A'C'(BD'+B'D))' [5 marks]

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

Q2. Some early decimal computers used excess-3 coding, which represents negative numbers "-3" to "-1" as 0000 to 0010, and positive decimal numbers "0" to "9" as 0011 to 1100. Adding "5" (1000) to "6" (1001) yields a result of "10" (10011). The advantage is that if a standard 4-bit binary adder is used, the result is carry = 1, sum = 0000, so the adder's carry (1) represents the decimal carry "1" (i.e. the most significant digit of the decimal result). The problem is that the adder's sum (0000) is not correct – in fact the sum is always wrong. Design a converter that corrects the sum. [25 marks]

Q3. Use the Quine-McCluskey method to simplify the expression:

F(A,B,C,D,E) = $\sum$(0, 4, 5, 7, 9, 12-15, 23, 31)

where the minterms 3, 6, 10, 16, 20 are don't-cares. [25 marks]

Q4. Derive the primitive flow table for a gated D latch. [25 marks]

Q5. Design a simple machine that implements a 32-bit program counter (PC) for a computer. The PC must be able to increment to the next instruction address, unconditionally and conditionally branch to a specified address, and handle interrupts. The PC has instruction inputs bits **M1** and **M0** to define the desired PC operation, a condition input **C**, a 32-bit branch address input **B[31..0]**, an interrupt input **X**, a 32-bit internal "stack" **Z[31..0]**, and a 32-bit address output **A[31..0]**. Each instruction cycle has two steps: the machine first performs the

desired PC operation shown in Table 2, then if there is an interrupt (i.e. **X**=1), the newly created address is saved to the stack (i.e. **Z**←**A**), and the PC jumps to the interrupt handler address (i.e. **A**←**11...11**). Whoever writes the interrupt handler software must end it with a "Return from interrupt" that pops those stack contents back to **A**. Design the machine as a finite state machine that controls the two 32-bit registers **A** and **Z**. Use D flip-flops.

M1	M0	Function
0	0	Increment to next instruction: **A**←**A**+1
0	1	Unconditionally branch: **A**←**B**
1	0	If **C**=1 then branch: **A**←**B**, otherwise increment: **A**←**A**+1
1	1	Return from interrupt: **A**←**Z**

Table 2: Function table for Question 5

[25 marks]

Q6. In the "Lamp Handball" lab experiment you implemented a single-player handball game using 8 lamps to show a moving ball, and a switch as a bat, where you were given the circuit diagram.

Starting with an "ASM Diagram", do a formal design for the game.

Remember, the game is initialised by activating a **RESET** switch, which places the ball at the far right waiting for the player to hit it. Once hit, the ball moves left at a rate of one lamp position per second, then bounces back rightwards at the same rate. If the player then hits the rebounded ball too early or too late, the ball disappears and the player loses. An 8-bit universal shift register is used in the datapath that simulates the moving ball, and JK Flip-flops are used in the control path. [25 marks]