NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2011-2012

CE1005/CZ1005 – DIGITAL LOGIC

CPE104/CSC104 – LOGIC DESIGN

November/December 2011 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 3 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) Perform the following number system conversions. Show full working.

   (i) Convert the decimal number, **-100**, to 8-bit two's complement binary.

   (ii) Convert the two's complement binary number, **1100 0001**, to decimal.

   (6 marks)

   (b) Perform the following 8-bit two's complement additions. Show full working.

   (i) Add **1010 0000** to **0111 0001**.

   (ii) Add **1110 1110** to **1111 0001**.

   (iii) Add **1010 1110** to **1001 0001**.

   (9 marks)

   (c) Perform the following 5-bit two's complement multiplications. Express your answer as a 10-bit two's complement number. Show full working.

   (i) Multiply **10100** by **01101**.

   (ii) Multiply **10101** by **10010**.

   (10 marks)

2. (a) Show using Boolean algebraic manipulation that:

$$\Sigma_{XYZ}\,(0, 1, 4, 5) = \prod_{XYZ}\,(2, 3, 6, 7)$$

(7 marks)

(b) A logic function, F, is defined by the following expression:

$$F = \Sigma_{WXYZ}\,(2, 7, 8, 10, 15) + D_{WXYZ}\,(11, 12, 14)$$

where D represents the *don't care* input combinations.

(i) Determine the minimum-cost sum-of-products (SOP) expression for the logic function F.

(9 marks)

(ii) Determine the minimum-cost product-of-sums (POS) expression for the logic function F.

(9 marks)

3. You are to design a circuit that takes a 4-bit unsigned binary input $x[3:0]$ and produces a 1 on its output, *div25*, whenever the corresponding input is exactly divisible by 2 or 5.

(a) Draw a Karnaugh map for the desired function and find a minimised expression.

(7 marks)

(b) Draw a logic circuit that implements the minimal expression.

(5 marks)

(c) Hence, write a Verilog module that implements the minimal expression using only gate-level primitives.

(7 marks)

(d) Rewrite the Verilog module using only a single assign statement.

(4 marks)

(e) In the context of FPGA design, explain why using an un-minimised representation *or* the code in Q3(c) or Q3(d) would result in the same implementation.

(2 marks)

4. A finite state machine (FSM) transition diagram is shown in Figure Q4. Each node shows the state name and the output, *y*, in that state. The input, *x*, controls state transitions as shown in the diagram. Unlabelled transitions always occur. The states are encoded using *s1* and *s0*.

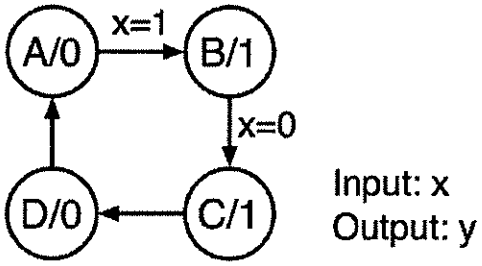

Figure Q4

(a) Write a state transition table that captures the FSM's behaviour. Encode the states as follows (*s1, s0*): A=00, B=01, C=10, and D=11, and use (*ns1, ns0*) for the next state variables.

(6 marks)

(b) Using the state transition table, derive minimised Boolean expressions for each of *ns1*, *ns0* and the output *y*.

(6 marks)

(c) Implement the FSM in Verilog. The module should have inputs *x*, *clk*, and *rst*, and output *y*. Use assign statements for *ns1*, *ns0* and *y*. Implement a state register for *s1* and *s0*. Ensure the FSM starts in state A.

(9 marks)

(d) Draw a timing diagram to show how the state machine reacts to the series of inputs given below on subsequent clock cycles (assuming it starts in state A). In addition, show the corresponding outputs.

1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0

(4 marks)

END OF PAPER