NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2011-2012

CE1005/CZ1005 – DIGITAL LOGIC

CPE104 – LOGIC DESIGN

April/May 2012 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 5 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) Perform the following number conversions, showing each step clearly.

   (i) Convert BEF_{16} to decimal

   (ii) Convert 87.3125_{10} to binary

   (iii) Convert 256_7 to octal

   (9 marks)

   (b) (i) Obtain the 8-bit 2's complement representation of the decimal numbers, -61 and 31. Show each step clearly.

   (ii) Hence, obtain the Boolean expression of a logic circuit whose output Y goes high only when its 8-bit input X (in 2's complement representation) is in the following range.

$$-61_{10} < X < 31_{10}$$

   (7 marks)

Note: Question No. 1 continues on Page 2

(c) Perform the following arithmetic operations in 8-bit 2's complement. In each case, state whether or not overflow has occurred.

(i) 10101010 + 11000111

(ii) 00110011 - 01100101

(iii) 11001100 + 11001001

(9 marks)

2. (a) Using Boolean algebraic manipulations, minimize the following logic expression to Sum-of-Products (SOP) form.

F(w, x, y, z) = (w' + xy + z')(w'x' + (xy + z)')'

(6 marks)

(b) A cooling control circuit has four logic inputs: A, B, C and D from four separate temperature sensors. It has two logic outputs: X1 and X0, that select one of the following settings on the cooling system: Off, Low, Medium or High. Table Q2 shows the sensor input conditions required for each setting. Set up the truth table for this circuit. State any assumptions that you have made.

Table Q2

Sensor input condition	Cooling system setting
All 4 sensor inputs are low	Off
Exactly 1 sensor input is high	Low
Exactly 2 sensor inputs are high	Medium
3 or more sensor inputs are high	High

(7 marks)

(c) Given the following canonical logic expression and "don't care" expression d, obtain the minimum-cost Product-of-Sums (POS) expression for F using the Karnaugh map technique. All loops must be clearly shown.

F(v, w, x, y, z) = ∑m(0, 1, 5, 8, 10, 14, 15, 24, 29, 30, 31)
d(v, w, x, y, z) = ∑m(4, 7, 12, 13, 20, 22, 27, 28)

(12 marks)

3. A circuit has two 2-bit inputs, *x*[1:0] and *y*[1:0], and a single 1-bit output, *gte*, that is asserted when the binary number on the *x* input is greater than or equal to the binary number on the *y* input.

(a) Find a minimal Sum-of-Products (SOP) expression for the *gte* output.

(4 marks)

(b) Hence, draw the corresponding circuit and write a structural gate-level Verilog module implementation of the circuit.

(5 marks)

(c) Write a concise alternative Verilog implementation of the same function, that does not require you to determine a gate-level circuit.

(5 marks)

(d) How does using this alternative representation impact the efficiency of the final implementation on the FPGA, and why?

(5 marks)

(e) Hence, write a Verilog module that has two 8-bit inputs, *x* and *y*, and an 8-bit output, *mx*, that is set to the larger of the two input values (or either input if the values are equal). Use only **assign** statements.

(6 marks)

4. (a) Write a Verilog module that implements a 6-bit counter. The counter should have an *en* input that enables counting when asserted, and a *dn* input that indicates the counter should count down rather than up when asserted.

(5 marks)

(b) You are required to implement a finite state machine (FSM) that has three states: *idle*, *up*, and *down*; two 1-bit inputs, *start* and *reverse*; and two 1-bit outputs, *count_en* and *dwn*:

- The FSM starts in the *idle* state, and moves to that state when reset is asserted.
- In the *idle* state, when *start* is asserted, the FSM moves to the *up* state.
- In the *up* state, if *reverse* is asserted, the FSM moves to the *down* state. Similarly, in the *down* state, if *reverse* is asserted, the FSM moves to the *up* state.
- If *start* is asserted in either the *up* or *down* state, the FSM moves to the *idle* state.
- The *count_en* output is asserted in the *up* and *down* states. The *dwn* output is asserted only in the *down* state.

(i) Draw a state transition diagram that captures the above behavior.

(4 marks)

(ii) Implement the state machine in Verilog. Use a combinational **always** block for the state transitions, and a separate **always** block for the register process.

(7 marks)

(iii) Write a Verilog module that connects an instance of your counter from Q4(a), and the FSM in Q(4)(b)(ii) so that the FSM outputs control the counter.

(4 marks)

(iv) Draw a timing diagram of the state transitions and FSM outputs in your top-level module given the inputs shown in Figure Q4. Hence, add the count sequence output to the timing diagram.

(5 marks)

Note: Figure Q4 is on Page 5

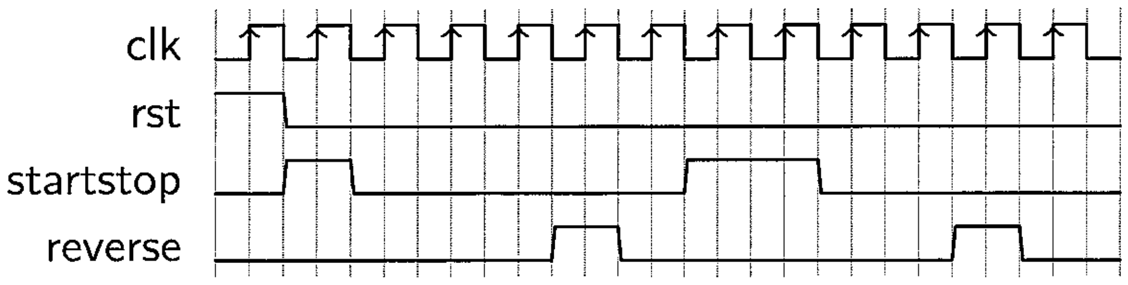

Figure Q4

END OF PAPER

CE1005 DIGITAL LOGIC
CPE104 LOGIC DESIGN
CZ1005 DIGITAL LOGIC

Please read the following instructions carefully:

1. **Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**

2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.

3. Please write your Matriculation Number on the front of the answer book.

4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.