NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2013-2014

CE1005/CZ1005 – DIGITAL LOGIC

Apr/May 2014 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) A certain serial communication system uses odd parity for error detection in the transmission of 8-bit data.

(i) Briefly describe how the parity error detection method works.

(3 marks)

(ii) Give an example to explain why this method is not reliable when there are multiple bit errors.

(3 marks)

(b) Perform the following number conversions. All steps must be shown clearly.

(i) Convert 39 (base 12) to base 16.

(3 marks)

(ii) Convert 6.407 (base 10) to base 2. Give your answer in 8 bits.

(3 marks)

Note: Question No. 1 continues on Page 2

(c) A logic circuit has two active-high inputs BUF and RD, and two active-low inputs EN* and MM*. The circuit has an active-high output GET which is asserted only when at least one of these two conditions is true:

(i) Either BUF or MM* (but not both) is asserted, and both EN* and RD are asserted.

(ii) Both MM* and EN* are asserted, and RD is negated. BUF may be asserted or negated.

Sketch a clearly-labelled logic circuit diagram to illustrate the above input-output relation. You may use standard and alternate logic symbols.

(6 marks)

(d) Minimise the following logic expression algebraically. Show every step clearly. Give your answer in the sum-of-product (SOP) form.

$$F(w, x, y, z) = (w' x + y z') [(x z + w') (y + x z')]'$$

(7 marks)

2. (a) Some logic outputs are tristate, i.e. low, high and high-impedence (Hi-Z). Briefly describe how two or more tristate outputs may be connected together.

(4 marks)

(b) A pair of 4-bit signed numbers in two's complement representation can be added or subtracted using a circuit that comprises four full adders and other logic gates. Sketch a clearly-labelled logic circuit diagram to illustrate this adder/subtractor. Briefly describe the operation of the circuit.

(6 marks)

(c) (i) Express these two signed decimal numbers, -19 and -15, in 6-bit two's complement representation. Show all steps clearly.

(2 marks)

(ii) Perform (-19) × (-15) using 6-bit signed two's complement multiplication. Give your answer in 12 bits. Show all steps clearly.

(3 marks)

Note: Question No. 2 continues on Page 3

(d) A combinational logic circuit has four inputs a, b, c, d and two outputs w and y. The canonical expressions for w, y and the respective "don't care" inputs are given below. In each case, obtain the minimum cost Boolean expression with the use of a Karnaugh map. Show all loops clearly.

(i) Give the sum-of-product (SOP) expression for w.

$w(a, b, c, d) = \sum m(1, 4, 13, 14)$
"don't cares" $u(a, b, c, d) = \sum m(2, 5, 7, 9, 11)$

(5 marks)

(ii) Give the product-of-sum (POS) expression for y.

$y(a, b, c, d) = \prod M(2, 6, 7, 8, 10, 13)$
"don't cares" $v(a, b, c, d) = \prod M(0, 5, 11)$

(5 marks)

3. A control system in an industrial plant monitors the temperature of a particular process to ensure safe levels are maintained. It compares the current temperature with a reference temperature and alerts the supervisors if there is a significant difference.

(a) The ***alert* module** has 4 inputs, *high*, *extreme*, *ovr1*, and *ovr2*. It produces 3 outputs, *alarm*, *cool*, and *malf* as follows:

- When the *high* input is asserted, the module asserts *alarm*. The supervisors are able to override this behaviour by asserting either *ovr1* or *ovr2*, in which case, *alarm* will not be asserted.
- If the temperature has been detected as dangerously high, both the *high* and *extreme* inputs will be high, and in this case, *alarm* will be asserted unless both the *ovr1* and *ovr2* inputs are asserted.
- The *cool* output will be asserted whenever *high* and *extreme* are asserted.
- The *malf* output is asserted if the *extreme* input is asserted while the *high* input is deasserted.

(i) Deduce the logic equations for each output of the ***alert* module**, and hence draw a gate-level diagram that implements the function.

(7 marks)

(ii) Implement your function from Q3(a)(i) using structural gate-level primitives in Verilog.

(6 marks)

Note: Question No. 3 continues on Page 4

(b) The *high* and *extreme* inputs to the ***alert*** **module** are calculated by comparing the current temperature with a reference value in the ***tempcomp*** **module**. If the difference is greater than 10, the *high* output is asserted. If the difference is greater than 20, the *extreme* input is asserted.

(i) Implement the ***tempcomp*** **module**, assuming 6-bit inputs *currtemp* and *reftemp*, and 1-bit outputs *high* and *extreme*. You may assume that *currtemp* is always greater than or equal to *reftemp*. Use a single **assign** statement to generate each output.

(7 marks)

(ii) Hence, briefly explain why the use of **assign** statements for implementing combinational logic is preferable to using gate-level structural descriptions.

(5 marks)

4. (a) The inputs in Figure Q4a are applied separately to a negative level-sensitive D-latch producing *latchout* and a positive edge-triggered D flip-flop producing *flopout*. Complete the diagram (in your answer book) to show the resulting behaviour of the primitives.

(6 marks)

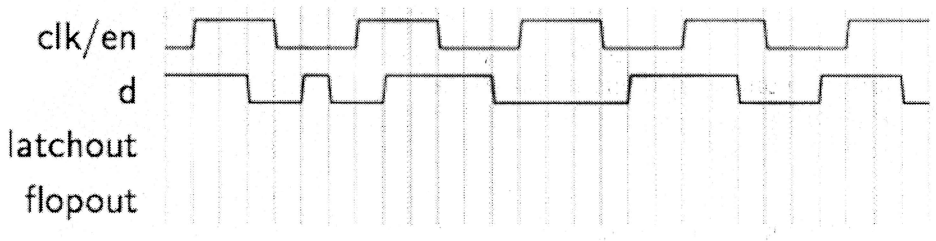

Figure Q4a

(b) Write the Verilog code for an enabled 6-bit down counter. When *reset* is asserted, the counter should take the value on the 6-bit *cnt_in* input. When the *en* input is asserted the counter should count down. In addition to the count output, there should be a *last* output that is asserted in the cycle in which the count is zero.

(7 marks)

Note: Question No. 4 continues on Page 5

(c) A sequence lock is implemented using a finite state machine (FSM). There are four 1-bit inputs *a*, *b*, *c*, and *d*, a 1-bit *press* input, along with a 1-bit *cancel* input. When any of the four letter inputs is asserted, *press* is also high. The 1-bit *lock* output starts asserted in the initial *idle* state. When *cancel* is asserted in any other intermediate state, the FSM returns to the *idle* state. (Assume *cancel* and *press* are never high at the same time.) When inputs are asserted in the sequence *b, a, d, c*, the *lock* output should be de-asserted and the state machine should stay in the final state. If at any point an incorrect input is entered, the FSM enters the *wrong* state. Any letter key pressed in this state will enter the terminal *alarm* state where the 1-bit *alarm* output is asserted.

(i) Draw a state transition diagram to represent this behaviour. Ensure you label all transitions and make clear the outputs in each state. You should ensure that multiple key presses in the same cycle do not circumvent the intended behavior.

(6 marks)

(ii) Given the code in Figure Q4b, write the state transition **always** block, using the signal names given. You may assume further intermediate states if necessary.

(6 marks)

```
module lockfsm (input clk, rst, a, b, c, d, cancel, output lock, alarm);

        parameter st_idle=3'b000, st_1=3'b001, st_2=3'b010, st_3=3'b011,
                st_4=3'b100, st_unlock=3'b101, st_wrong=3'b110,
                st_alarm=3'b111;

        reg [2:0] st, nst;

        always@(*)
        begin

        // state transitions here

        end

        always@(posedge clk)
        if(rst)
                st<=st_idle;
        else
                st <= nst;
endmodule
```

Figure Q4b

END OF PAPER

CE1005 DIGITAL LOGIC
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.