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.

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

(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

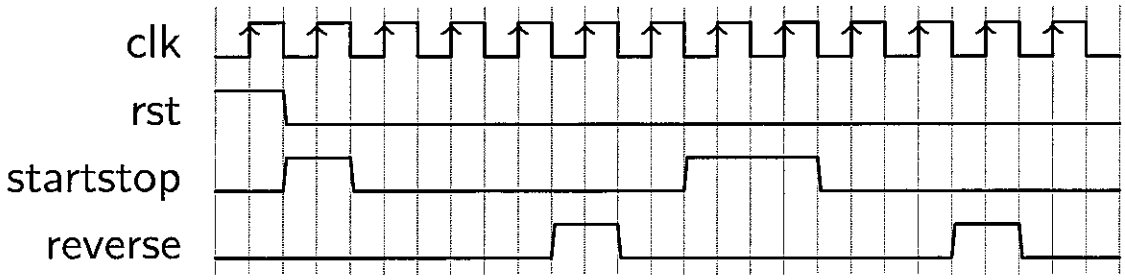

Figure Q4

END OF PAPER

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2012-2013

CE1005/CZ1005 – DIGITAL LOGIC

CPE104/CSC104 – LOGIC DESIGN

Nov/Dec 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. Show all steps clearly.

   (i) Convert decimal 74 to binary.

   (ii) Convert hexadecimal DFA to decimal.

   (iii) Convert octal 1654 to hexadecimal.

   (iv) Convert decimal 8.4375 to binary.

   (10 marks)

   (b) Obtain the minimum-cost sum-of-product (SOP) expression of the following Boolean function using algebraic manipulations. Show all steps clearly.

   $$F(a, b, c, d) = (a + d)(a' b + c' d)(a c + b d)'$$

   (8 marks)

Note: Question No. 1 continues on Page 2

(c) Implement the following logic function using NAND gates only. Illustrate with a clearly-labelled logic circuit diagram.

$$F(x, y, z) = x' y' z' + x y' z + x y z'$$

(7 marks)

2. (a) Figure Q2 shows a CMOS logic circuit with inputs A, B, C, D and output Z. Obtain its truth table.

(6 marks)

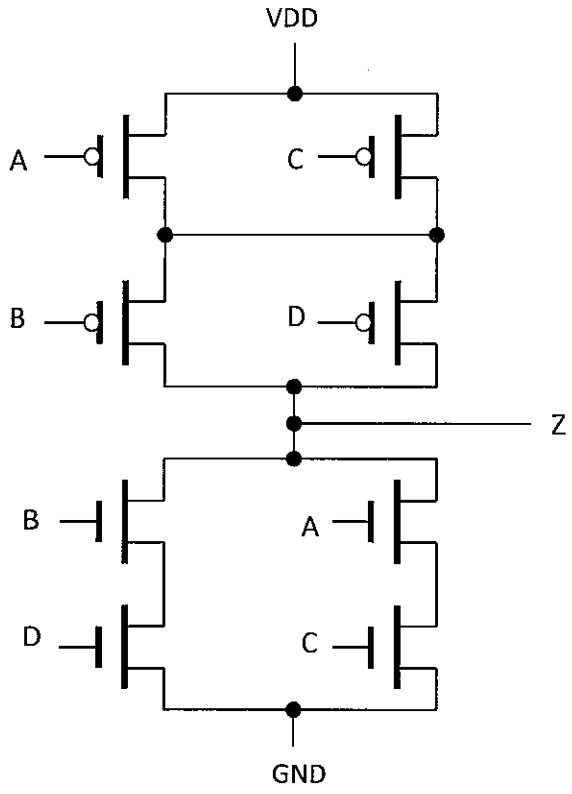

Figure Q2

(b) Showing all steps clearly, represent the signed decimal value **-51** using

(i) 8-bit sign-magnitude representation.

(ii) 8-bit two's complement representation.

(4 marks)

Note: Question No. 2 continues on Page 3

(c) Perform the following operations using 8-bit two's complement arithmetic. Show all the steps clearly. In each case, state whether or not arithmetic overflow has occurred.

(i) 01010101 + 00111101

(ii) 11001100 - 00111101

(6 marks)

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

F (a, b, c, d) = $\prod$ M (0, 2, 5, 10, 13)

d (a, b, c, d) = $\prod$ M (7, 8, 11, 12, 14)

(9 marks)

3. (a) A 1-bit full adder is a combinational circuit with three inputs (*Ain*, *Bin* and *Cin*) and two outputs (*Sum* and *Cout*), where *Sum* represents the 1-bit summation of the three inputs and *Cout* is the 1-bit carry. A minimised expression for each of the full adder outputs is:

$$Sum = Ain \oplus Bin \oplus Cin$$

$$Cout = Ain.Bin + Ain.Cin + Bin.Cin$$

(i) Draw a circuit for the full adder using only **2-input gates**.

(4 marks)

(ii) Write a structural gate-level Verilog module which implements the full adder described by the circuit given in Q3(a)(i).

(6 marks)

Note: Question No. 3 continues on Page 4

(b) You are required to implement a circuit which produces a 5-bit output (*Y*). If the select input (*Sel*) is a logic '0', *Y* is the sum of two 4-bit values (*Xin* and *Yin*). If the select signal is a logic '1', *Y* is the negative of *Xin*.

(i) Sketch a schematic diagram to implement the required circuit using four 1-bit full adder blocks and any other necessary components.

(7 marks)

(ii) Briefly explain why you would **not** implement the circuit using structural Verilog as per the schematic diagram given in Q3(b)(i).

(2 marks)

(iii) Write a Verilog module which implements the required circuit and uses just a single **assign** statement.

(6 marks)

4. (a) Describe the behaviour of the following sequential components, and the difference between them:

- Transparent D-type Latch
- D-type flip-flop

Which is preferred in modern digital design and why?

(6 marks)

(b) You are required to implement a finite state machine (FSM) that implements an alarmed locking mechanism. It has two 1-bit inputs, *correct* and *wrong*, and two 1-bit outputs, *lock* and *alarm*:

- The FSM starts in the *idle* state, and moves to that state whenever reset is asserted.
- In the *idle* state, if *correct* is asserted, the FSM moves to the *unlock* state, where it remains.
- In the *idle* state, if *wrong* is asserted, the FSM moves to a *failone* state.
- In *failone*, if *wrong* is asserted once more, the FSM moves to the *failtwo* state; if *correct* is asserted, it moves to the *unlock* state.
- In *failtwo*, if *wrong* is asserted the FSM moves to the *failalarm* state, where the *alarm* output is asserted, and it stays in that state until *correct* is asserted, at which point it returns to the *idle* state.
- In *failtwo*, if *correct* is asserted, the FSM enters the *unlock* state.

Note: Question No. 4 continues on Page 5

- In the *unlock* state, the *lock* output is low; it is high in all other states.
- In the *failalarm* state, the *alarm* output is asserted. It is deasserted in all other states
- You should assume that *correct* and *wrong* are not asserted at the same time.

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

(6 marks)

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

(8 marks)

(iii) Draw a timing diagram of the state transitions and FSM outputs of your top-level module given the input sequence in Figure Q4.

(5 marks)

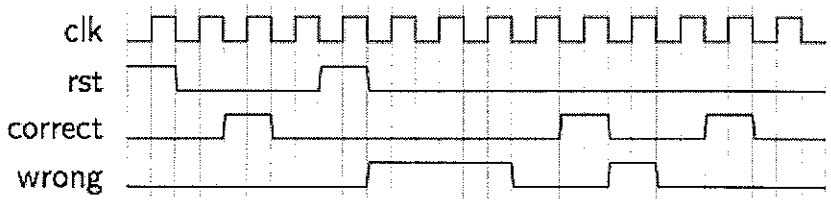

Figure Q4

END OF PAPER

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2012-2013

CE1005/CZ1005 – DIGITAL LOGIC

CPE104/CSC104 – LOGIC DESIGN

April/May 2013 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 6 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) Convert the decimal value **3.172** to binary (to 8 significant bits). All steps must be shown clearly.

(4 marks)

(b) Convert the hexadecimal value **EF** to Binary-coded decimal (BCD). All steps must be shown clearly.

(3 marks)

(c) Simplify the following Boolean expression using algebraic manipulations to obtain the minimum cost sum-of-products (SOP) expression.

$$F = [\ a'\ b' + d + b\ (c' + a)'\ ]\ [\ (d\ c')' + a\ c\ ]'$$

(5 marks)

Note: Question No. 1 continues on Page 2

(d) Implement the following Boolean expression using a minimum number of 2-input NOR gates only. Illustrate with a clearly labeled logic circuit diagram.

$$F(w, x, y) = (x + y)(w + y)(x' + y')$$

(6 marks)

(e) Figure Q1 shows a CMOS logic circuit with inputs A, B, C, D and output Z. Obtain its truth table. Give also the Boolean expression of Z. You are not required to minimise the expression.

(7 marks)

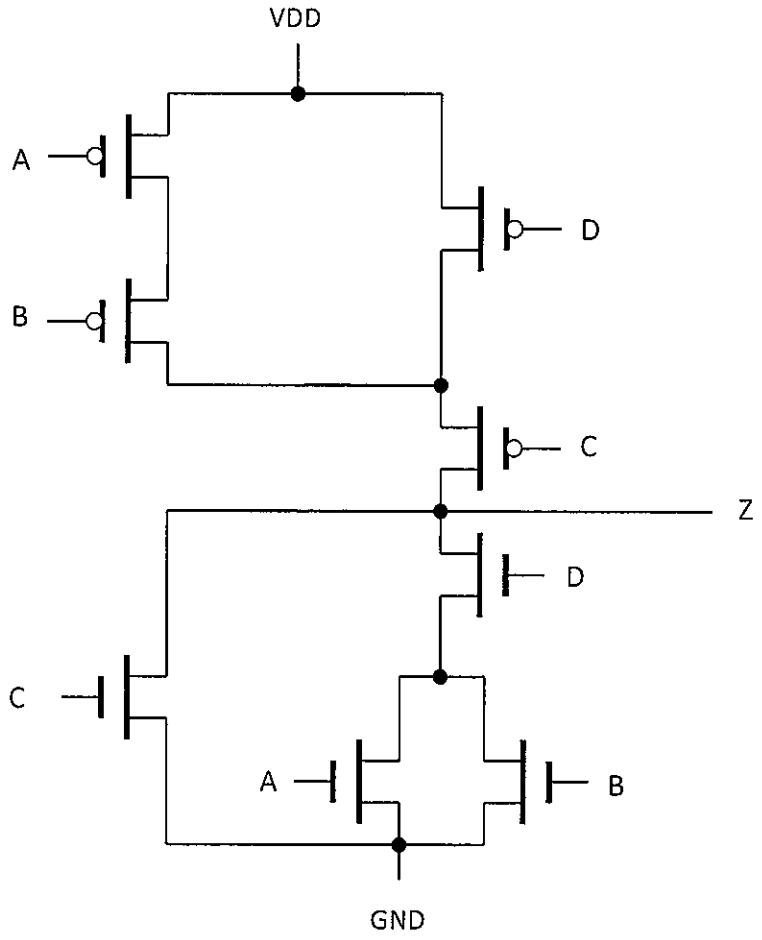

Figure Q1

2. (a) Perform the following signed two's complement multiplication. All steps must be shown clearly. Give also the decimal equivalent of the multiplicands and result.

$$101011 \times 110101$$

(7 marks)

(b) Table Q2 shows the functional truth table of a combinational logic circuit that adds or subtracts two signed numbers, X and Y in 4-bit two's complement representation. The circuit produces logic 1 at the output OVF when there is an arithmetic overflow.

Table Q2

Input	**Outputs**	
ADD*/SUB	Z	OVF
0	Z = X + Y	1 if there is arithmetic overflow. 0 otherwise.
1	Z = X - Y	1 if there is arithmetic overflow. 0 otherwise.

(i) Obtain the Boolean expression of OVF in terms of X, Y and Z when ADD*/SUB = 0.

(ii) Using the result from Q2(b)(ii), obtain a Boolean expression of OVF in terms of X, Y, Z and ADD*/SUB inputs.

(8 marks)

(c) A combinational logic circuit has four inputs A, B, C, D and two outputs P and Q. The canonical expressions for P, Q 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. All loops must be clearly shown.

(i) Give the sum-of-products (SOP) expression for P.

$P(A, B, C, D) = \sum m(0, 3, 7, 9, 10)$
"don't cares" $X(A, B, C, D) = \sum m(2, 6, 8, 14)$

(ii) Give the product-of-sums (POS) expression for Q.

$Q(A, B, C, D) = \prod M(0, 5, 8, 11, 15)$
"don't cares" $Y(A, B, C, D) = \prod M(6, 7, 10, 12)$

(10 marks)

3. (a) Draw the gate level circuit described by the Verilog code in Figure Q3a.

(4 marks)

```
module mod1 (input a, b, c, d, output f);

    not n1 (na, a);
    not n2 (nc, c);
    and a1 (w1, a, b);
    or  o1 (w2, nc, d);
    and a2 (w3, w1, w2);
    and a3 (w4, a, c);
    and a4 (w5, na, nc, d);
    or  o2 (f, w3, w4, w5);
endmodule
```

Figure Q3a

(b) Determine the minimal sum-of-products (SOP) representation for the logic function in Q3(a).

(5 marks)

(c) Hence, write a Verilog module that implements the minimal circuit determined in Q3(b) using a single **assign** statement.

(4 marks)

(d) Briefly explain why this minimisation step is not necessary when using modern design tools and targeting FPGAs.

(3 marks)

(e) Briefly state two advantages of using **assign** statements like the one in Q3(c) as opposed to a gate-level circuit as in Figure Q3a.

(4 marks)

(f) Determine, with explanation, whether the circuit in Figure Q3b implements the same function.

(5 marks)

Note: Question No. 3 continues on Page 5

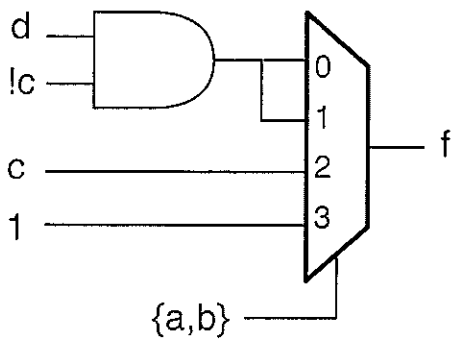

Figure Q3b

4. (a) Identify the three errors in the Verilog code for a counter in Figure Q4a.

(4 marks)

```
module counter (input clk, rst, output [4:0] count);
    always @ *
    begin
        count = count + 1'b1;
    end
endmodule
```

Figure Q4a

(b) Write a correct implementation of an enabled 4-bit up-counter in Verilog. The counter should include an additional *skip* input which, when high, makes the counter increase by 3 rather than increment by 1. Assume all signals are active high.

(6 marks)

(c) If the waveforms in Figure Q4b are used to drive the counter, show the resulting output sequence.

(5 marks)

Note: Question No. 4 continues on Page 6

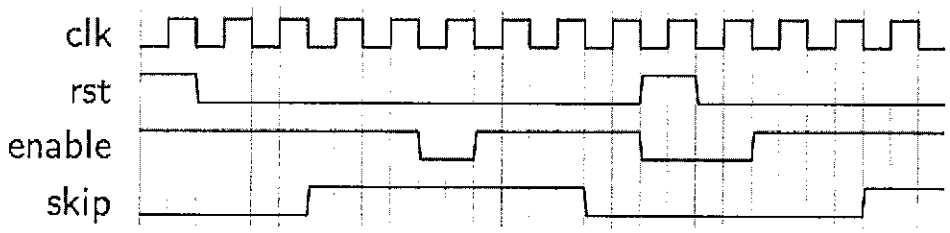

Figure Q4b

(d) Draw the circuit described by the Verilog code fragment in Figure Q4c, labeling all signals accordingly, assuming they are all declared as 1-bit signals. What is this type of circuit called?

(5 marks)

```
always @(posedge clk)
begin
    a <= x;
    s <= m;
    m <= a;
    x <= e;
end
```

Figure Q4c

(e) Briefly discuss the difference between flip-flops and latches, and why flip-flops are preferred in modern design.

(5 marks)

END OF PAPER

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2013-2014

CE1005/CZ1005 – DIGITAL LOGIC

Nov/Dec 2013 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 6 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) Perform the following unsigned number conversions. All steps must be shown clearly.

   (i) Convert 9B (hexadecimal) to octal.

   (ii) Convert 10101.101 (binary) to decimal.

   (iii) Convert 2.4 (decimal) to binary (with 8 significant bits).

   (7 marks)

   (b) The operands (enclosed in pairs of square brackets) in the following arithmetic operations are given in signed decimal. In each case, express the operands in 8-bit two's complement representation and perform the operation using two's complement arithmetic. All steps must be shown clearly. In each case, state whether or not arithmetic overflow has occurred.

   (i) [-87] + [110]

   (ii) [93] – [-54]

   (6 marks)

Note: Question No. 1 continues on Page 2

(c) Using Boolean algebra, minimize the following logic expression to SOP (sum-of-product) form. All steps must be shown clearly.

F (a, b, c, d) = a' (c' + d)' (a c)' [a' c d + (b' + c)']'

(6 marks)

(d) Determine the minimum number of bits required to represent the number specified in each case below. All steps must be shown clearly.

(i) unsigned decimal **99**

(ii) any signed 20-digit decimal number

(6 marks)

2. (a) A facility is fitted with four sensors T, M, H and L, each of which produces a logic output independently. Depending on the logic outputs of the sensors, two independent controllers W and X, are to be activated as specified in Table Q2.

Table Q2

Outputs of sensors	**Controllers to be activated**
T, M, H and L are all 0	None
H or/and L are 1; T and M are both 0	W only
H and L are 0; T or M (but not both) is 1	W only
H or/and L are 1; T or M (but not both) is 1	X only
T and M are both 1	Both W and X

Design a combinational logic circuit that takes the four sensor logic signals as inputs, and produces two outputs W and X, such that they are logic 1 when the corresponding controllers are to be activated.

(i) Determine the truth table of the circuit.

(ii) Obtain the simplified Boolean expressions for W and X in SOP (sum-of-product) form with the aid of Karnaugh maps. All loops must be clearly shown.

(15 marks)

Note: Question No. 2 continues on Page 3

(b) Figure Q2 shows a CMOS logic circuit with inputs A, B, C, D and output F. Determine its truth table and logic expression.

(6 marks)

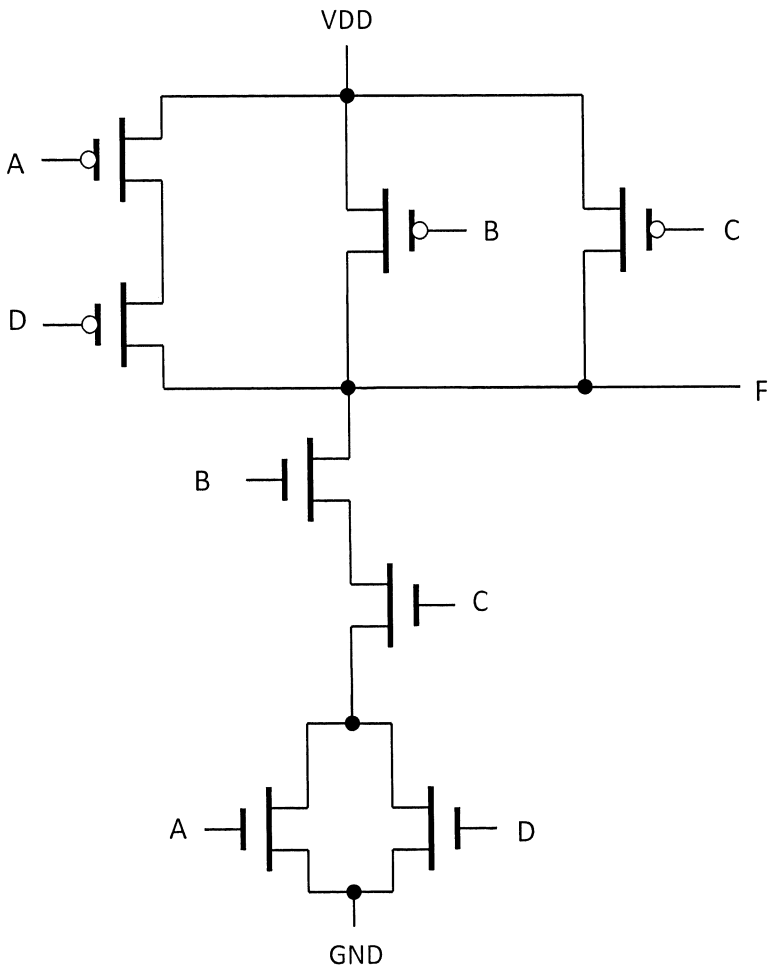

Figure Q2

(c) Briefly describe each of the following terms and how they may affect the performance of a logic circuit.

(i) Propagation delay

(ii) Noise margin

(4 marks)

3. A pressure gauge measures tyre pressure relative to atmospheric pressure, referred to as the relative pressure (*Rp*). The pressure gauge produces a relative pressure output between 0 kPa and 400 kPa. The absolute pressure is the relative pressure plus the atmospheric pressure, which at sea level is approximately 100 kPa.

(a) Sketch a simple circuit using a single unsigned adder and a single multiplexer which will output (as *Pout*) the relative pressure if a 1-bit *Disp* signal is a logic '0', and the absolute pressure if *Disp* is a logic '1'. Clearly indicate the size (in bits) of all inputs and outputs. Note that the circuit should only have two inputs and one output (*Pout*).

(6 marks)

(b) Write a structural Verilog module to implement the circuit in Q3(a) by instantiating the adder module declared below. You are not required to implement the adder.

```
module adder #(parameter SIZE=4)(
        input [SIZE-1:0] A, B, output Cout, output [SIZE-1:0] Sum);
```

(11 marks)

(c) Write a behavioural Verilog module to implement the circuit in Q3(a).

(8 marks)

4. (a) Figure Q4a shows a negative edge-triggered flip-flop, a level sensitive latch, and a positive edge-triggered flip-flop. Complete the timing diagram in Figure Q4b by filling in the waveforms for *Out1*, *Out2* and *Out3*.

(6 marks)

(b) Figure Q4c shows a circuit with two inputs (*Clk* and *Rst*), and a 4-bit output bus (*Q[3:0]*). When *Rst* is HIGH, *Q[3:0]* = 0000. When *Rst* is LOW, the chain of flip-flops behaves like a shift register.

Note: Question No. 4 continues on Page 5

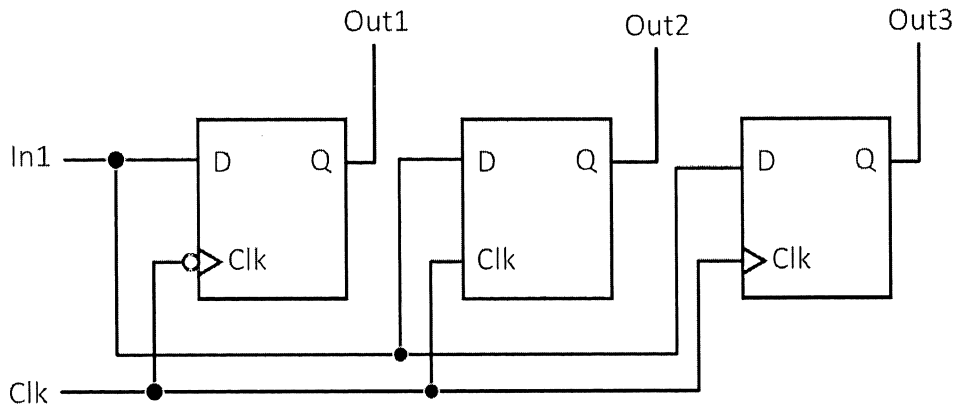

Figure Q4a

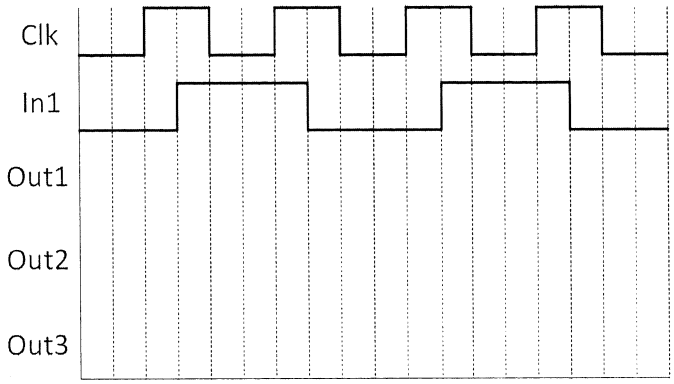

Figure Q4b

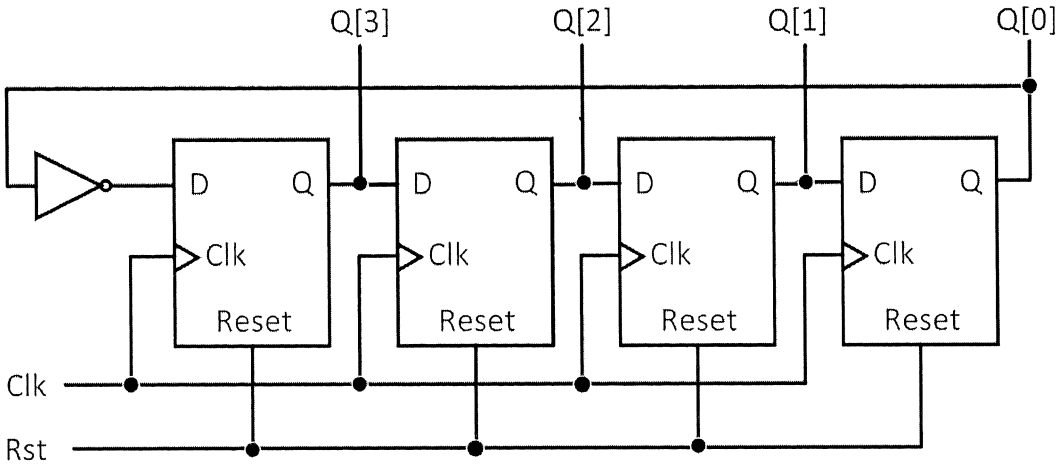

Figure Q4c

Note: Question No. 4 continues on Page 6

(i) Draw the state transition diagram of the circuit in Figure Q4c. Each state must be labeled with the output values (*Q[3:0]*). Show how the input *Rst* affects the state transitions.

(6 marks)

(ii) Write a Verilog module for the circuit in Figure Q4c.

(6 marks)

(c) A factory uses a sensor to detect cans on a conveyor belt, and increments the number of cans detected and the number of missing cans. The finite state machine (FSM) has three inputs: *start*, *stop*, *detected*; and two outputs: *inc_cans*, *inc_missing*:

- The FSM starts in the *idle* state and moves to that state when *stop* is asserted.
- In the *idle* state, if *start* is asserted, the FSM moves to the *check_sensor* state.
- In the *check_sensor* state, if *detected* is asserted, the FSM moves to the *detected* state.
- In the *check_sensor* state, if *detected* is de-asserted, the FSM moves to the *missing* state.
- In the *detected* state, the *inc_cans* output is HIGH and the FSM moves to the *check_sensor* state at the rising edge of the clock. The *inc_cans* output is LOW in other states.
- In the *missing* state, the *inc_missing* output is HIGH and the FSM moves to the *check_sensor* state at the rising edge of the clock. The *inc_missing* output is LOW in other states.

Draw a state transition diagram that captures the above behaviour.

(7 marks)

END OF PAPER

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)

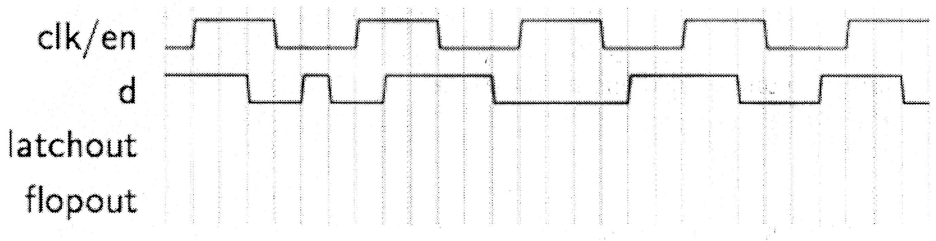

Figure Q4a

(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

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2014-2015

CE1005/CZ1005 – DIGITAL LOGIC

Nov/Dec 2014 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 6 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. All steps must be shown clearly.

(i) Convert 3.91 (decimal) to binary. Give your answer in 8 significant figures.

(3 marks)

(ii) Convert FADE (hexadecimal) to octal.

(3 marks)

(b) Minimise the following logic expression algebraically. Give your answer in Sum-of-Products form. Show all steps clearly.

$$F(a, b, c, d) = (a + c' d') [b' c + (a' d + b c d')']$$

(8 marks)

Note: Question No. 1 continues on Page 2

(c) Figure Q1a shows a CMOS logic circuit with inputs A, B, C and D and output F. Obtain its truth table and the Boolean expression of F. You are not required to simplify the expression.

(7 marks)

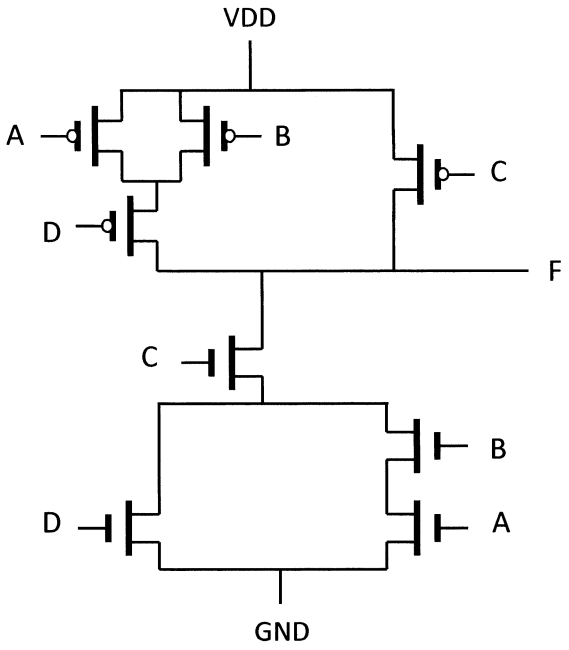

Figure Q1a

(d) The signal shown in Figure Q1b is applied to the input of a Schmitt-trigger inverter with threshold voltages VT- = 0.7 V and VT+ = 1.6 V. Sketch the inverter's output waveform. Show its relation with the input signal clearly.

(4 marks)

Figure Q1b

2. (a) Figure Q2 shows the block diagram of a logic circuit that performs the arithmetic addition X + Y, where X and Y are both 4-bit unsigned numbers stored in memory. The sequence of operation is controlled by the logic signals CLR, LOAD and TRANSFER.

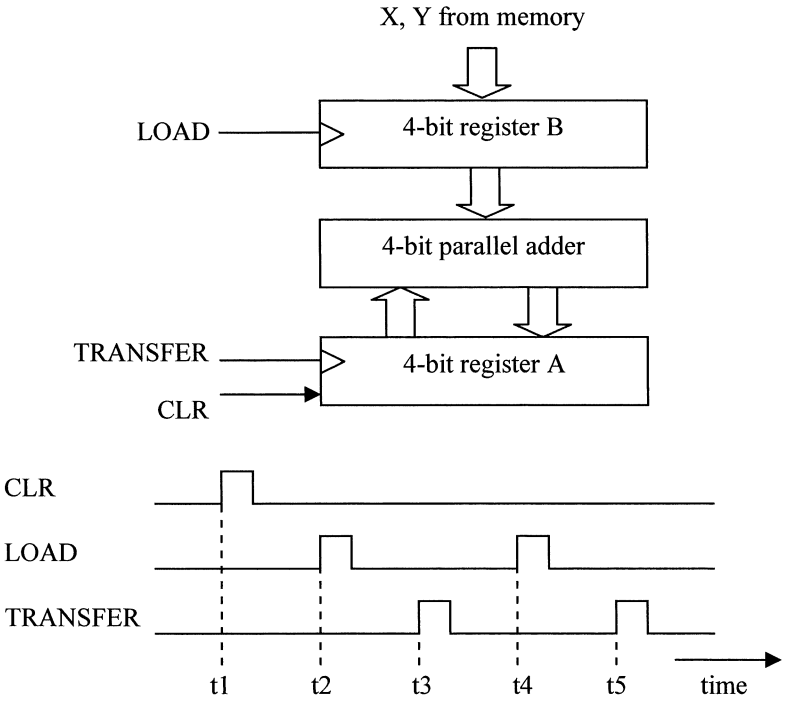

Figure Q2

(i) State the purpose accomplished by the signals CLR, LOAD and TRANSFER for each of the time instances t1, t2, t3, t4 and t5 in Figure Q2.

(5 marks)

(ii) What operation takes place in the circuit between time t4 and t5? Will the circuit still function correctly if the time difference between t4 and t5 is reduced to zero? Explain your answer.

(3 marks)

(b) Perform the arithmetic operation $\mathbf{-92_{10} - 38_{10}}$. Express each signed decimal number in 8-bit two's complement representation and then perform the operation. Show all steps clearly. State whether or not there is overflow.

(5 marks)

Note: Question No. 2 continues on Page 4

(c) A combinational logic circuit's input is a 4-bit unsigned binary number represented by x3 (MSB), x2, x1 and x0 (LSB). Its output y is logic 1 if the input is a prime number. For all other cases, the output y is logic 0. The relevant prime numbers are 2, 3, 5, 7, 11 and 13 (in decimal).

(i) Construct a truth table for the logic circuit.

(6 marks)

(ii) Using Karnaugh map, obtain a minimized Sum-of-Products expression for the output. Show all loops clearly.

(6 marks)

3. You are required to implement a combinational circuit that determines the number of days in a month.

(a) The combinational circuit has a 4-bit input bus (*M[3:0]*) and 2-bit output bus (*D[1:0]*). The binary encoding for **Month** (*M[3:0]*) and **Days** (*D[1:0]*) are shown in Table Q3. When there are no valid inputs for **Month**, *D[1:0]* = 00.

(i) Find the minimal Boolean expressions (Sum-of-Products or Product-of-Sums) for *D[1]* and *D[0]*.

(4 marks)

(ii) Write the Verilog **assign** statements to implement the minimal circuit determined in Q3(a)(i).

(4 marks)

Table Q3

Month	M[3:0]	Days	D[1:0]
January	0001	31	11
February	0010	28	01
March	0011	31	11
April	0100	30	10
May	0101	31	11
June	0110	30	10
July	0111	31	11
August	1000	31	11
September	1001	30	10
October	1010	31	11
November	1011	30	10
December	1100	31	11

Note: Question No. 3 continues on Page 5

(b) The combinational circuit in Q3(a) is modified to include an additional active high input (*leap_year*) and an additional output (*D[2]*). When *leap_year* is asserted and the input **Month** is February, *D[2]* = 1, otherwise *D[2]* = 0. The function for *D[1:0]* is the same as described in Q3(a).

(i) Implement the function for *D[2]* using structural gate-level primitives in Verilog.

(5 marks)

(ii) Write a single Verilog **assign** statement to implement the function for *D[2]*.

(3 marks)

(iii) Using only behavioral description, write the Verilog module for the modified combinational circuit with inputs (*M[3:0]*, *leap_year*) and output (*D[2:0]*).

(9 marks)

4. (a) Figure Q4a shows a 2-bit counter with synchronous reset. Complete the timing diagram for the counter in Figure Q4b by filling in the waveform for *cnt[1:0]*.

(4 marks)

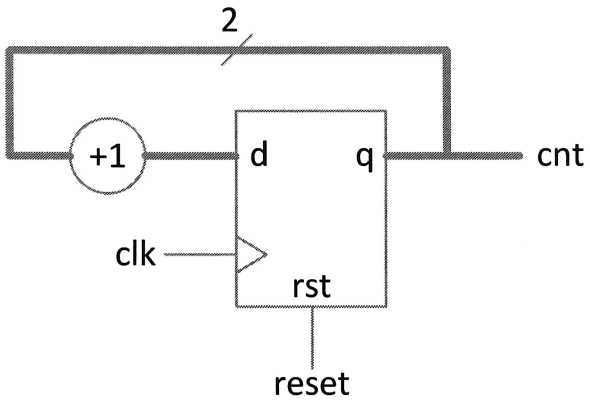

Figure Q4a

Note: Question No. 4 continues on Page 6

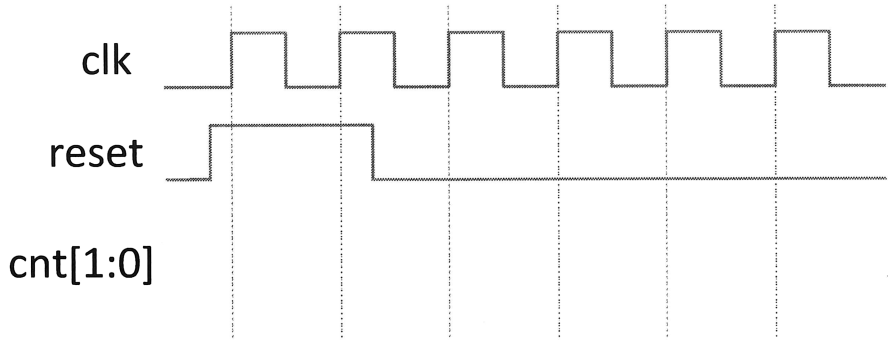

Figure Q4b

(b) A driver alert system monitors the eyes of the driver and alerts the driver when the eyes are closed for a period of time. The finite state machine (FSM) has five active high inputs: *start*, *stop*, *open, close, timeout*; and two outputs: *on_alarm*, *rst_timer*:

- The FSM starts in the *idle* state.
- In the *idle* state, if *start* is asserted, the FSM moves to the *check_eye_status* state.
- In the *check_eye_status* state, if both *close* and *timeout* are asserted, the FSM moves to the *alert_driver* state.
- In the *check_eye_status* state, if *open* is asserted, the FSM moves to the *reset_timer* state.
- In the *check_eye_status* state, if *stop* is asserted, the FSM moves to the *idle* state.
- In the *alert_driver* state, the *on_alarm* output is HIGH and the FSM moves to the *reset_timer* state at the rising edge of the clock. The *on_alarm* output is LOW in other states.
- In the *reset_timer* state, the *rst_timer* output is HIGH and the FSM moves to the *check_eye_status* state at the rising edge of the clock. The *rst_timer* output is LOW in other states.

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

(7 marks)

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

(14 marks)

END OF PAPER

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2014-2015

CE1005/CZ1005 – DIGITAL LOGIC

Apr/May 2015 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. Show all steps clearly.

   (i) Convert 2A.B (in hexadecimal) to decimal. (2 marks)

   (ii) Convert 745.37 (in octal) to hexadecimal. (2 marks)

   (iii) Convert 1.136 (in decimal) to binary. Give your answer in 8 significant digits. (3 marks)

   (b) Determine the number of bits required to represent an unsigned 15-digit decimal number in each format below. Show all steps clearly.

   (i) BCD (binary coded decimal) (2 marks)

   (ii) Binary (4 marks)

Note: Question No. 1 continues on Page 2

(c) Using algebraic manipulations, obtain the minimum-cost SOP (Sum of Product) expression for the following logic function. Show all steps clearly.

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

(6 marks)

(d) Implement the following logic function using a minimum number of 2-input NOR gates only. Illustrate with a clearly labelled circuit diagram.

$$F(a, b, c) = (b + c) (a' + b' + c')$$

(6 marks)

2. (a) (i) Briefly describe the following parameters of a logic device:

- VOL(max)
- VOH(min)
- VIL(max)
- VIH(min)

(4 marks)

(ii) State the relation between the parameters in Q2(a)(i) and the device's noise margin.

(2 marks)

(b) Each operand enclosed within a pair of square brackets is a signed decimal number. Represent each operand in 8-bit 2's complement format and perform the arithmetic operations. Show all steps clearly. State clearly whether or not there is overflow in each case.

(i) [-28] + [101]

(4 marks)

(ii) [-61] – [73]

(4 marks)

Note: Question No. 2 continues on Page 3

(c) (i) A 4-input combinational logic circuit with active low output F* is described by the following canonical expression. Use a Karnaugh map to obtain the minimum-cost POS (Product of Sum) expression for F*. Show all loops clearly on the Karnaugh map.

$$F^*(A, B, C, D) = \prod M(1, 3, 4, 6, 9, 11, 14)$$

(6 marks)

(ii) The inputs A and D are active low. Replace the variable names A and D with A* and D* respectively in the POS expression you have obtained in Q2(c)(i).

(1 mark)

(iii) Using suitable logic symbols with matched bubbles, draw a clearly labelled logic circuit diagram to illustrate the relation between F* and the four inputs A*, B, C and D*.

(4 marks)

3. (a) A full adder performs a single bit addition by taking inputs *a*, *b* and *cin*, and produces outputs *sum* and *cout*.

(i) Write down the truth table for a full adder.

(3 marks)

(ii) Draw a gate-level circuit for the full adder, using only 2-input gates.

(4 marks)

(b) You are provided with a Verilog implementation of the full adder described in Q3(a) with the following module declaration.

```
module full_add(input a, b, cin, output sum, cout);
```

(i) Implement a Verilog module that adds two 3-bit numbers, *x[2:0]* and *y[2:0],* and returns a 3-bit answer *total[2:0]* and an overflow output *oflow*. You are required to use the full_add module in your implementation. There should be no carry in. You can hard-wire a signal by setting it to an appropriate logic value.

(10 marks)

Note: Question No. 3 continues on Page 4

(ii) Briefly describe how you can change the implementation in Q3(b)(i) to realize a 3-bit adder/subtractor. You do not need to implement the full module.

(3 marks)

(iii) Explain why we do not generally build arithmetic circuits in this manner. What is the preferred approach in modern design?

(2 marks)

(c) Briefly describe the functionality of a multiplexer. How can you create a 2-by-1 multiplexer using a single Verilog **assign** statement?

(3 marks)

4. (a) Write a Verilog module of an enabled 5-bit counter. The module should also include an additional *jump* input that, when asserted, increases the counter by 5 rather than increment by 1. Assume all inputs are active high.

(7 marks)

(b) The waveform in Figure Q4a is used to drive the counter. Show the resulting output sequence of the counter.

(4 marks)

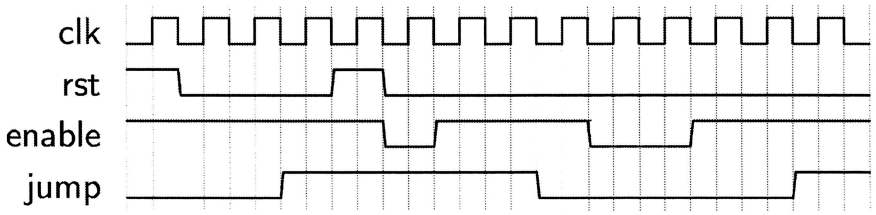

Figure Q4a

(c) Figure Q4b shows a finite state machine (FSM) state transition diagram. Implement the FSM in Verilog using an **always** block for the state transition logic and a separate **always** block for the state register. Assume missing transitions are self-transitions.

(8 marks)

Note: Question No. 4 continues on Page 5

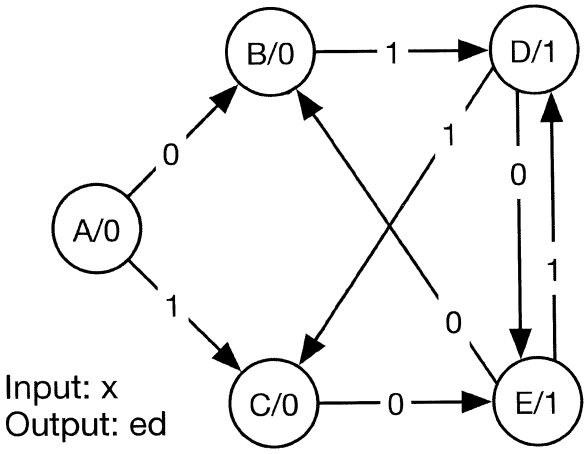

Figure Q4b

(d) Deduce the function of the FSM in Q4(c).

(3 marks)

(e) Briefly explain the difference between a latch and a flip-flop and why one is preferred over the other in modern digital design.

(3 marks)

END OF PAPER

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2015-2016

CE1005/CZ1005 – DIGITAL LOGIC

Nov/Dec 2015 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 6 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. Show all the steps clearly.

(i) Convert 672.15 (in octal) to hexadecimal.

(2 marks)

(ii) Convert C1.FD (in hexadecimal) to decimal. Give your answer in 6 significant digits.

(2 marks)

(iii) Convert 6.48 (in decimal) to binary. Give your answer in 8 significant bits.

(3 marks)

(b) A digital system is used to record the number of people visiting a facility. Assume that the number of visitors in any single day does not exceed three thousand. Determine the number of bits required to represent the total number of visitors in one year (assume 365 days) in unsigned binary. Show each step clearly.

(5 marks)

Note: Question No. 1 continues on Page 2

(c) Simplify the following Boolean expression algebraically. Show each step clearly and give your answer in Sum-of-Product (SOP) form.

$$F(a, b, c, d) = (a' c' + d')(b d + c')' (a' d + (b' c)')' + (c + d')'$$

(7 marks)

(d) A circuit has the following canonical Boolean expression:

$$F^* = (A + B + C^*)(A + B' + C^{*'})(A' + B' + C^*)$$

Draw a logic circuit diagram with suitable symbols and matched bubbles to clearly illustrate the three maxterms that assert the active Low output F*. The inputs A and B are both active High while the input C* is active Low. Do not expand or simplify the expression.

(6 marks)

2. (a) There are four judges in a contest: one chief judge and three assistant judges. All the judges make independent decisions. A contestant is successful when

- all three assistant judges pass him/her, or
- both the chief judge and at least one assistant judge pass him/her.

Otherwise, the contestant is unsuccessful.

A combinational logic circuit takes the four judges' individual decisions as inputs to produce the success status for a contestant.

(i) Construct a truth table for the success status output SU (1 means Successful, 0 means Unsuccessful). Take A, B, C and D as individual decision inputs (1 means Pass, 0 means Fail) from the chief judge and assistant judges.

(6 marks)

(ii) By using a Karnaugh map, find the minimum-cost Sum-of-Product expression for the output SU. Show all the loops clearly.

(5 marks)

Note: Question No. 2 continues on Page 3

(b) (i) Represent each of the following signed decimal values in 8-bit 2's complement format. Show all the steps clearly.

- -72
- +67

(4 marks)

(ii) A circuit produces a logic High output when its input, a signed decimal value falls between the range of -72 and +67 (both values included). Otherwise the output is Low. Assume that the input is represented in 8-bit 2's complement format. Determine the Boolean expression of the circuit output. You are not required to simplify the expression.

(4 marks)

(c) Figure Q2 shows a logic circuit with inputs, EN and A, and output X. Describe how the circuit works and determine its truth table.

(6 marks)

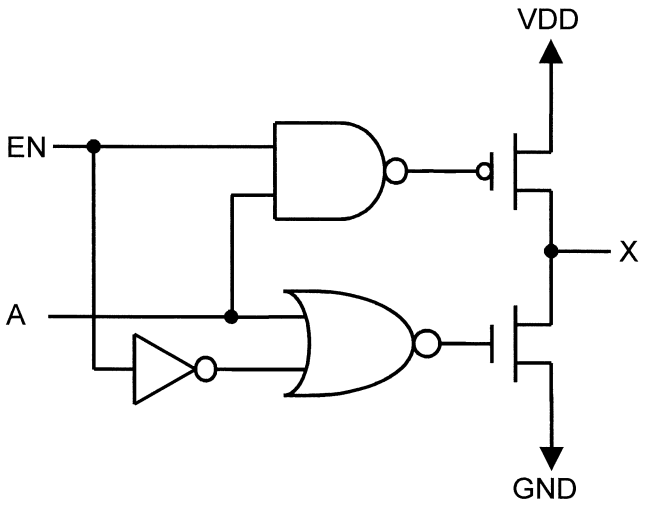

Figure Q2

3. A majority gate is a combinational circuit in which the output is a 1 if the majority of the inputs are 1's, otherwise the output is a 0. A 3-input majority gate has an input bus *A[2:0]* and an output *Z*.

(a) You are required to implement the 3-input majority gate.

(i) Draw a Karnaugh map for the 3-input majority gate and find the minimized SOP (Sum-of-Products) expression.

(5 marks)

(ii) Draw a logic circuit that implements the minimized SOP expression.

(4 marks)

(iii) Write the Verilog module for the logic circuit by using structural gate-level primitives in Verilog.

(5 marks)

(b) The 3-input majority gate can be implemented using a 4-to-1 multiplexer.

(i) Complete the 3-input majority gate in Figure Q3 by indicating the appropriate inputs *A[2:0]* at respective '?'.

(5 marks)

(ii) Write the combinational **always** block in Verilog for the 3-input majority gate in Q3b(i).

(6 marks)

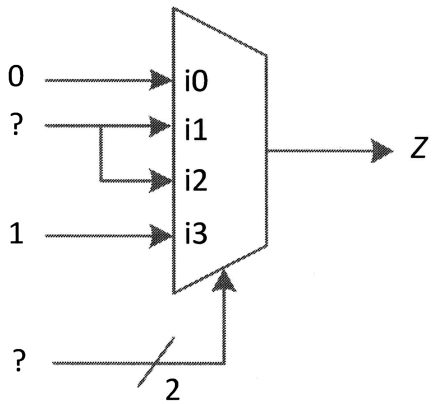

Figure Q3

4. (a) Figure Q4a shows a circuit with two inputs (*Clk* and *Rst*) and one 2-bit output bus (*Q[1:0]*).

(i) Complete the timing diagram shown in Figure Q4b for the circuit in Figure Q4a by drawing the waveforms for *Q[1]* and *Q[0]*. Assume that *Rst = 0*.

(8 marks)

(ii) Write the Verilog module for the circuit in Figure Q4a with synchronous reset.

(8 marks)

(iii) Draw the state transition diagram of the circuit in Figure Q4a. Each state must be labelled with the output *Q[1:0]*.

(4 marks)

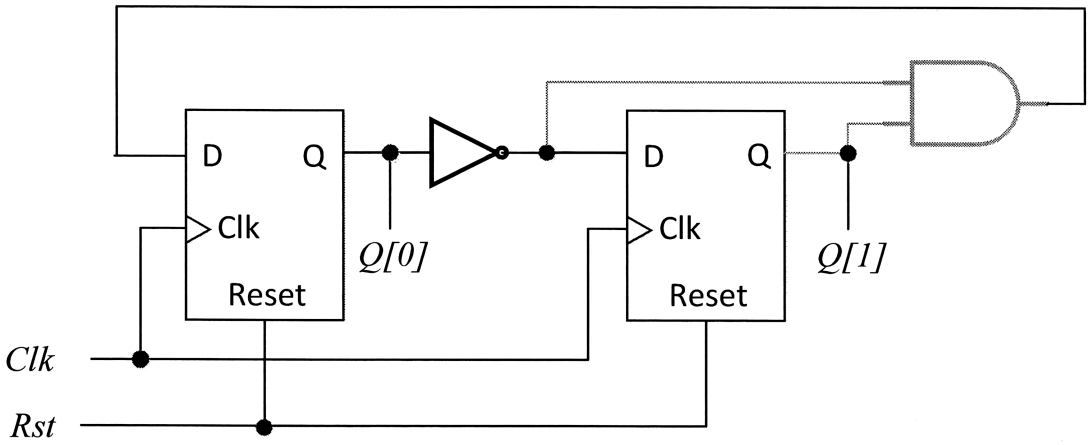

Figure Q4a

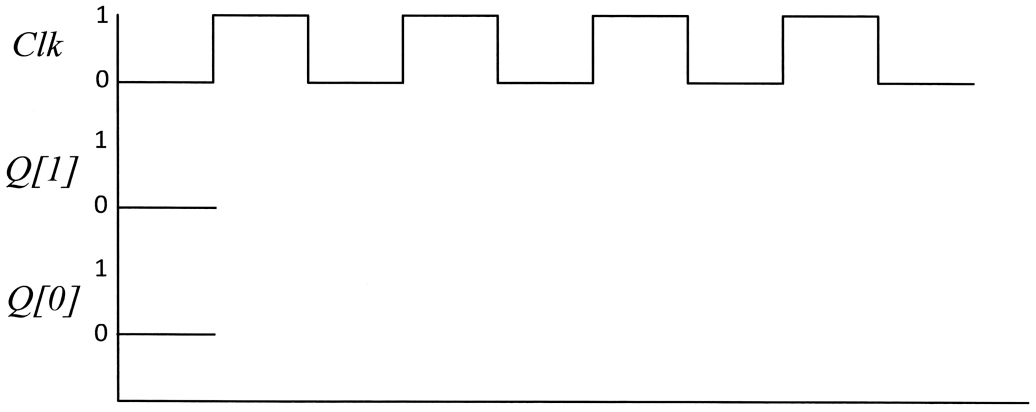

Figure Q4b

Note: Question No. 4 continues on Page 6

(b) A mobile robot detects the presence of obstacles by using front and side sensors, and navigates by traversing along the obstacle walls. The finite state machine (FSM) for the mobile robot should have three active High inputs: *start*, *front_obstacle* and *end_wall*; and one output: *on_buzzer*:

- The FSM starts in the *idle* state.
- In the *idle* state, if *start* is asserted, the FSM moves to the *forward* state.
- In the *forward* state, if *front_obstacle* is asserted, the FSM moves to the *turn_left* state.
- In the *turn_left* state, the *on_buzzer* output is HIGH and the FSM moves to the *follow_wall* state at the rising edge of the clock. The *on_buzzer* output is LOW in other states.
- In the *follow_wall* state, if *end_wall* is asserted, the FSM moves to the *clear_wall* state.
- In the *clear_wall* state, if *front_obstacle* is asserted, the FSM moves to the *turn_left* state. Otherwise, the FSM moves to the *turn_right* state at the rising edge of the clock.
- In the *turn_right* state, the FSM moves to the *forward* state at the rising edge of the clock.

Draw the state transition diagram that captures the above behavior of the robot.

(5 marks)

END OF PAPER

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2015-2016

CE1005/CZ1005 – DIGITAL LOGIC

Apr/May 2016 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 6 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) A 16-bit integer **9A6D** (hexadecimal) is stored in a digital system. For each of the following representations of the integer, determine the decimal value, and sign where applicable. Show all steps clearly.

(i) unsigned binary

(2 marks)

(ii) sign-magnitude

(2 marks)

(iii) two's complement

(3 marks)

(b) (i) Represent the unsigned decimal value **25.904** in 16-bit fixed-point format, with 8 bits each allocated to the integer and the fractional parts. Show all steps clearly.

(3 marks)

(ii) Briefly describe how the value in Q1(b)(i) may be represented in a 16-bit floating-point format.

(3 marks)

Note: Question No. 1 continues on Page 2

(c) Use algebraic manipulations to obtain a minimum cost Sum-of-product (SOP) expression for the following Boolean function. Show all steps clearly.

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

(6 marks)

(d) A digital circuit has logic inputs Enable*, A, B*, C and output G*, where * denotes an active Low signal. When the circuit is disabled, the output G* is negated. When the circuit is enabled, the output G* is asserted if-and-only-if exactly 2 out of the 3 inputs A, B*, C are asserted. Sketch a clearly-labelled logic circuit diagram using suitable symbols to illustrate the circuit implementation. You are not required to simplify the circuit.

(6 marks)

2. (a) Sketch a timing diagram to illustrate each of the following parameters of an inverter:

- Rise time, tr
- Fall time, tf
- Propagation delay, tPHL
- Propagation delay, tPLH

(6 marks)

(b) Briefly describe the following types of digital circuit output:

(i) Tristate

(3 marks)

(ii) Open-drain

(3 marks)

(c) Perform the following 5-bit 2's complement signed multiplication. Show all steps clearly. Give the answer in decimal value.

11001 x 10010

(5 marks)

Note: Question No. 2 continues on Page 3

(d) The output F of a 4-input logic circuit has the following canonical expression along with the don't care inputs specified by D. Use a Karnaugh map to obtain a minimum cost Product-of-sum (POS) expression for the output F. Show all loops clearly.

$F(w, x, y, z) = \sum m(1, 6, 11, 12)$
$D(w, x, y, z) = \sum m(7, 10, 13, 14, 15)$

(8 marks)

3. (a) State whether each of the following statements is TRUE or FALSE.

- An N-input decoder has 2N outputs.
- An n-input multiplexer requires $\log_2(n)$ select bits.
- When using structural gate-level primitives in Verilog module, order of statements is important.
- Identifiers in Verilog can start with letters, numbers, underscore (_) and dollar ($).
- Sensitivity list must contain all signals that affect output.

(5 marks)

(b) Sketch each of the following logic gates which is implemented using only 2:1 multiplexer:

- NOT gate
- AND gate
- OR gate

(6 marks)

Note: Question No. 3 continues on Page 4

(c) Determine the function of the schematic shown in Figure Q3 and write a Verilog module using ***if-else*** statement. The module will have two inputs (A, B) and one output (Y).

(6 marks)

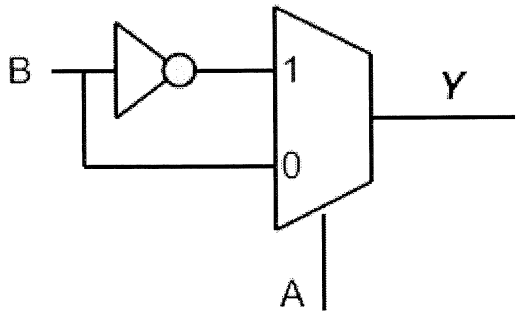

Figure Q3

(d) There are four doors in an auditorium. In order to best manage the crowd, only one door is opened at any one time. You are to design a circuit based on a 2x4 decoder which will guide the guests to the opened door. The input to the circuit is a 2-bit binary number. The circuit has four outputs. Each output controls a light bulb that is placed above each door. You are required to write a Verilog module to implement this circuit using ***case*** statements.

(8 marks)

4. (a) Determine the value of a and b for the following Verilog code segments and briefly explain the reasons for your answer. Assume initial value of a = 10.

(i)
```
always@(posedge clk)
  begin
    a = a + 1;
    b = a + 1;
  end
```

(ii)
```
always@(posedge clk)
  begin
    a <= a + 1;
    b <= a + 1;
  end
```

(6 marks)

Note: Question No. 4 continues on Page 5

(b) Figure Q4a shows a 3-bit binary counter with two inputs (***CLK*** & ***RESET***) and one 3-bit output (***Q[2:0]***). The initial state of the counter is ***Q[0] = Q[1] = Q[2] = 0***. Based on the schematic shown in Figure Q4a:

(i) Complete the timing diagram shown in Figure Q4b. Assume the counter is at initial state and the reset signal is low.

(7 marks)

(ii) Determine whether the counter shown in Figure Q4a is an up or a down counter.

(2 marks)

(iii) Write a Verilog module for the 3-bit counter that you have determined in Q4(b)(ii).

(6 marks)

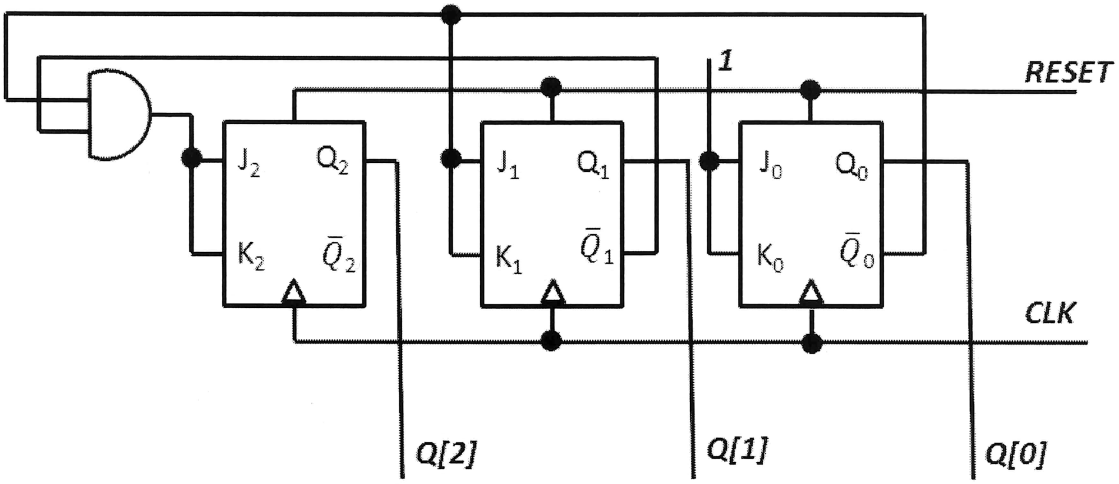

Figure Q4a

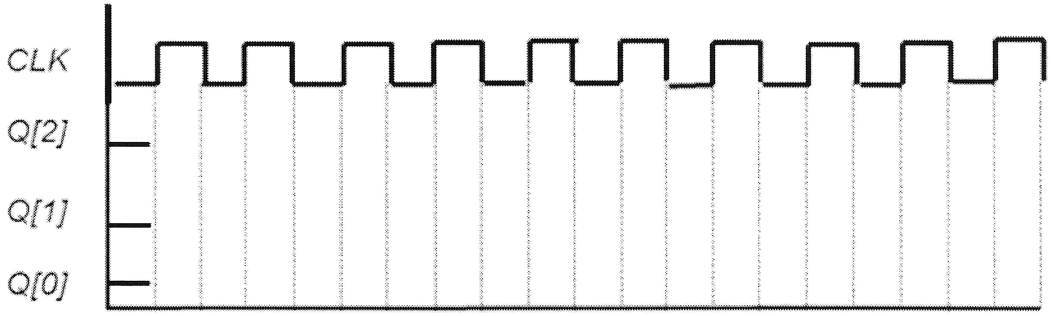

Figure Q4b

Note: Question No. 4 continues on Page 6

(c) Figure Q4c shows a FSM which detects a particular bit sequence. Determine the output sequence for the following sequence of inputs:

- 0,0,1,1,1,0,1,1,1,1,0,1

(4 marks)

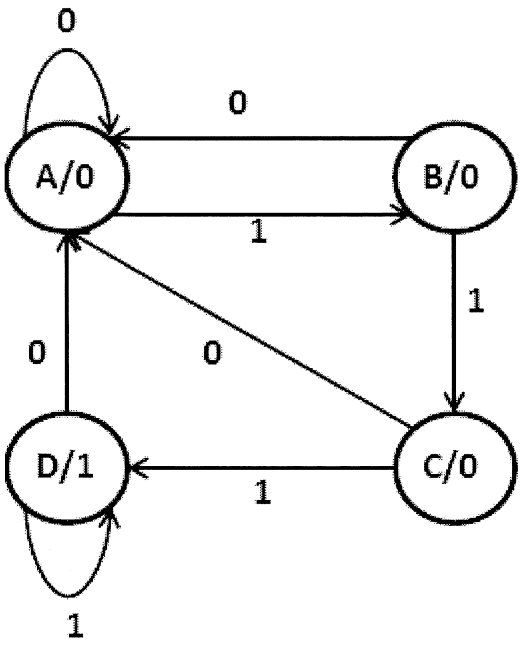

Figure Q4c

END OF PAPER

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*.

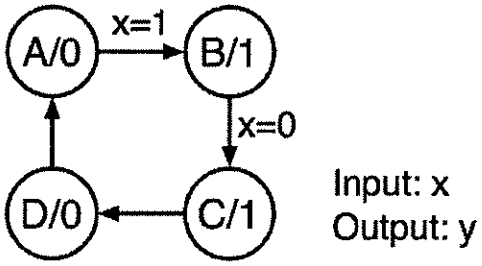

Figure Q4

(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