NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

TERM TEST #1
AY2014/5 Semester 2

CS2100 — COMPUTER ORGANISATION

7 March 2015 Time Allowed: **1 hour 15 minutes**

INSTRUCTIONS

1. This question paper contains **TEN (10)** questions (excluding the bonus question) and comprises **SEVEN (7)** printed pages.
2. The last two pages are for your rough work. They contain blank truth tables, K-maps and state table for your use.
3. An **Answer Sheet**, comprising **TWO (2)** printed page, is provided for you.
4. Write your **Name**, **Matriculation Number** and **Tutorial Group Number** on the Answer Sheet with a **PEN**.
5. You may write your answers in pencil (at least 2B).
6. Answer **ALL** questions within the space provided on the Answer Sheet.
7. Submit only the Answer Sheet at the end of the test. You may keep the question paper.
8. Maximum score is **30 marks**.
9. This is a **CLOSED BOOK** test. However, a single-sheet double-sided handwritten A4 reference sheet is allowed.
10. Calculators and computing devices such as laptops and PDAs are not allowed.

——— END OF INSTRUCTIONS ———

Bonus question:

0. This is the bonus question which is worth 1 mark. The mark of this question will only be added if the total mark scored is less than 30.

   The photograph on the right shows the demonstration on essential prime implicants in one of the CS2100 lectures this semester.

   Write down the colours of <u>two</u> of the three umbrellas used. (You must get both colours correct to be awarded the bonus mark. No mark will be awarded if you write three or more colours and one of them is wrong.)

   Optionally, you may also fill in the speech bubble what you want to say to the setter and grader of this paper. ☺

Note the following acronyms used in this paper:

- **SOP**: Sum-of-Products
- **POS**: Product-of-Sums

Questions 1 – 5: Each multiple-choice-question has only <u>one</u> correct answer. Write your answers in the boxes on the **Answer Sheet**. Two marks are awarded for each correct answer and no penalty for wrong answer.

1. Given the following set of codes, what is its efficiency?

   { 101110, 010111, 110101, 000000, 111000 }

   A. Efficiency = 1/6
   B. Efficiency = 1/5
   C. Efficiency = 1/3
   D. Efficiency = 1/2
   E. Efficiency = 5/6

2. Given the same set of codes in question 1 above, what is its Hamming distance?

   A. 1
   B. 2
   C. 3
   D. 4
   E. 5

3. Given the following Boolean expression of $F(A,B,C)$:

$$F(A,B,C) = A' + B + C'$$

Which of the following statements is/are true about the above expression?

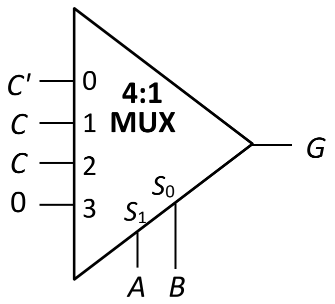
(i) It is an SOP expression
(ii) It is a POS expression
(iii) It is a sum-of-minterms expression
(iv) It is a product-of-maxterms expression

A. Only (i)
B. Only (i) and (ii)
C. Only (i) and (iii)
D. Only (ii) and (iv)
E. Only (i), (ii) and (iv)

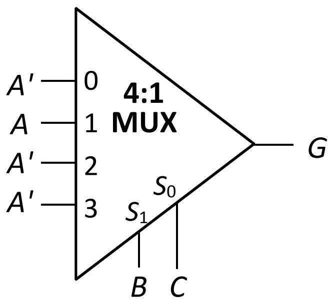
4. Given the following Boolean function $G(A,B,C)$ where X denotes don't-care:

$$G(A,B,C) = \Pi M(1, 4, 6, 7) \cdot X(3)$$

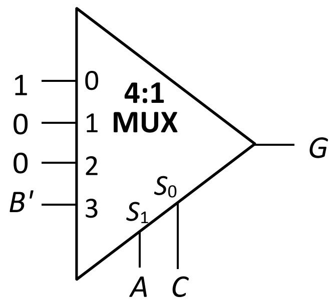
Which of the following implementations of G using a 4:1 multiplexer are correct?



(i)



(ii)

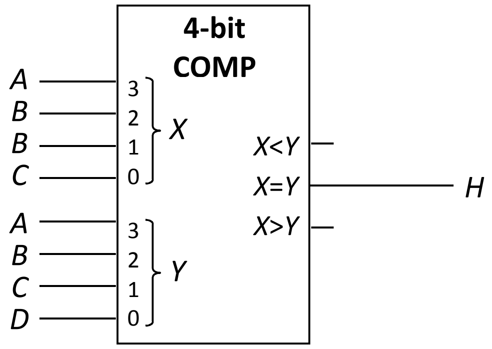


(iii)

A. None of (i), (ii) or (iii)
B. Only (i) and (ii)
C. Only (i) and (iii)
D. Only (ii) and (iii)
E. All of (i), (ii) and (iii)

5. The Boolean function $H(A,B,C,D)$ is implemented using a 4-bit magnitude comparator as shown below. What is function $H(A,B,C,D)$?

   A. $\Sigma m(0, 7, 8, 15)$
   B. $\Sigma m(3, 6, 9, 12)$
   C. $\Sigma m(5, 8, 11, 14)$
   D. $\Sigma m(12, 13, 14, 15)$
   E. None of the above.



Questions 6 – 10: Write your answer in the space provided on the **Answer Sheet**. You do not need to show workings, unless otherwise stated.

6. [4 marks]
   (a) Convert the decimal value **26.37** into base 3, correct to 3 places as accurately as possible. [2 marks]
   (b) Convert binary value **10111** into its equivalent standard Gray code value. [1 mark]
   (c) Simplify $\boldsymbol{(A \oplus B \oplus C) \oplus (A \oplus B \oplus C)'}$ [1 mark]

7. Given the following hexadecimal representation in IEEE 754 single-precision floating-point number system:

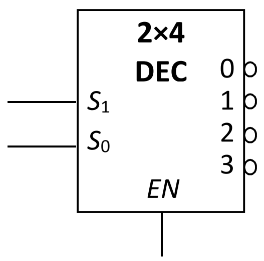
   C 2 1 2 8 0 0 0

   Fill in the first 20 bits of the representation and write out the decimal value it represents. [3 marks]

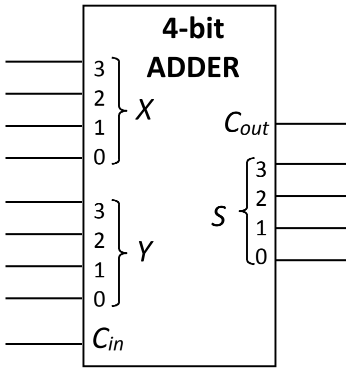
8. What is the simplified SOP expression for Boolean function Z as shown below?

   $$\boldsymbol{Z = A' \cdot B \cdot C' \cdot D \cdot E + (D'+B)' + B \cdot D \cdot E}$$

   Can the function Z be implemented using a single 2×4 decoder with 1-enable and active low outputs as shown below, <u>without</u> any additional logic gate? Explain. Note that complemented literals are <u>not</u> available. [4 marks]



9. You are to take a 2-bit unsigned number *ab* and calculate the 5-bit unsigned number *cdefg* which is *ab* × 6 + 3. For example, if *ab* = 10 (value 2), then *cdefg* = 01111 (value 15). You are to implement this using a single 4-bit parallel adder as shown below, with no additional logic gate. Complemented literals are not available. [3 marks]

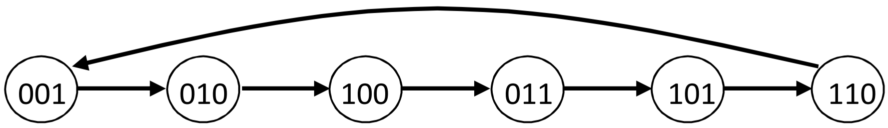


10. [6 marks] The following are the binary codes for the digits in a base 6 system:

0 = 001; 1 = 010; 2 = 100; 3 = 011; 4 = 101; 5 = 110

(a) Is this set of codes self-complementing? [1 mark]

(b) Suppose the above codes are used in a sequential circuit as shown below. The state values are represented by Boolean variables *A*, *B* and *C*. Two T flip-flops are used for *A* and *B* while a JK flip-flop is used for *C*.



Write the simplified SOP expressions for the flip-flop inputs *TA*, *JC* and *KC*. The simplified expression for *TB* has been worked out for you: *TB* = 1. [3 marks]

(c) For each of the two unused states 000 and 111, determine the next state. [2 marks]

——— END OF PAPER ———

(Blank truth tables, K-maps and state table are provided in the next two pages.)

This page is for your rough work.

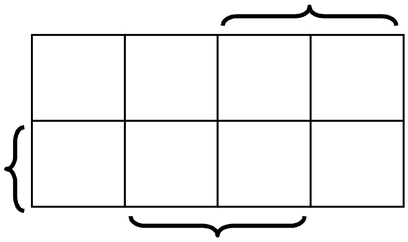
A	*B*	*C*	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

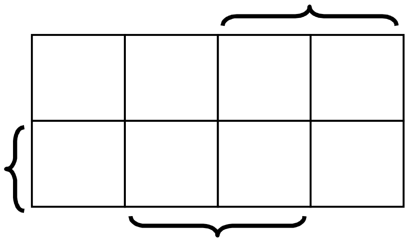
A	*B*	*C*	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

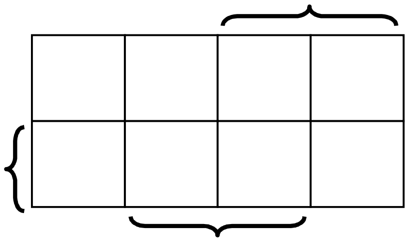
A	*B*	*C*	*D*	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

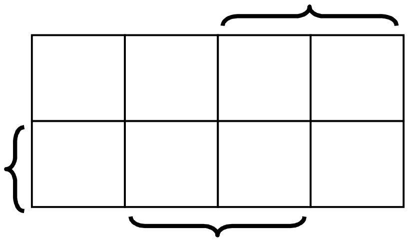
A	*B*	*C*	*D*	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

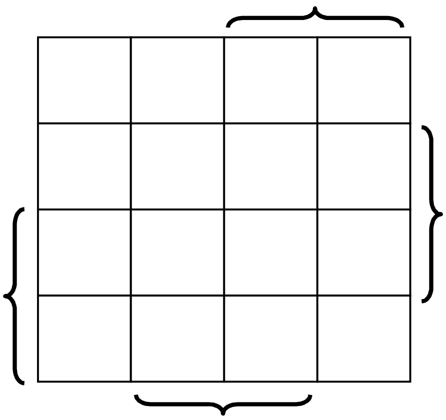
This page is for your rough work.

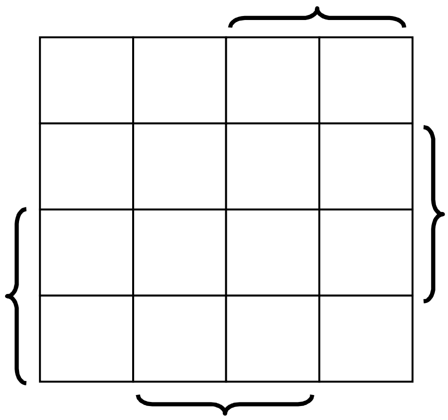












Present state			Next state						
A	***B***	***C***	A^+	B^+	C^+				