※ 注意：請於試卷上「非選擇題作答區」內依序作答，並應註明作答之大題及其題號。

You are graded on the correctness and clarity of your answers. Note that, to receive the full credit, you have to clearly explain how you derive the answer for each problem.

Problem 1. (12%)

a. Which of the following is a midterm of function $f(a, b, c)$?
  (A) M_7
  (B) $a + b + c$
  (C) $a'b'c'$
  (D) m_8

b. Which of the following is logically equivalent to function $f(a, b, c) = m_3$?
  (A) Dual of M_3
  (B) Dual of M_4
  (C) Complement of M_5
  (D) Complement of M_6

c. A digital circuit is described by its negative logic function f_N. How would one find its positive logic function f_P?
  (A) f_P = dual{f_N}
  (B) $f_P = f_N$
  (C) f_P = complement{f_N}
  (D) $f_P = -f_N$

d. Which of the following is logically equivalent to $(A \oplus B) + AB$?
  (A) AB
  (B) $A \oplus B$
  (C) $A + B$
  (D) $(A \oplus AB) + (B \oplus AB)$

Problem 2. (12%)

Implement a 2-input XOR gate using only 2-input NAND gates. Use as few 2-input NAND gates as possible.

Problem 3 (10%)

Use the 4-bit 2-to-1 MUX in Fig. 1 to implement the device described as follows. Add as few logic gates as possible.

1. It has a 4-bit input $X = x_3x_2x_1x_0$ and a control input S.
2. It has a 5-bit output $Y = y_4y_3y_2y_1y_0$.
3. Both X and Y are 2's complement numbers.
4. If $S = 1$, Y should be equal to $2X$.
5. If $S = 0$, Y should be equal to $X/2$ (round down, i.e., Y is the largest integer that does not exceed $X/2$).

The following are some examples.

```
X = 0010    S = 1 ==>    Y = 00100
X = 1111    S = 0 ==>    Y = 11111
X = 1111    S = 1 ==>    Y = 11110
X = 0011    S = 0 ==>    Y = 00001
```

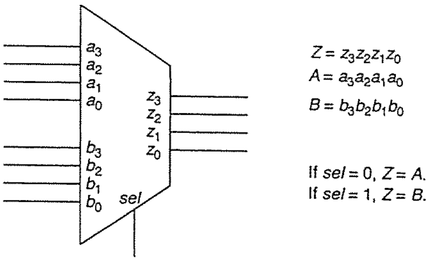

Fig. 1 A 4-bit 2-to-1 multiplexer (Problem 3).

見背面

Problem 4. (16%)

Design a finite state machine that takes an input bit stream and filters all strings of consecutive 1's into a single 1. (Use a Moore design.) For example, if the input stream is

    0001000111100011 1000...,

then the output should be

    00001000000100000100....

a. (8%) Draw the state transition diagram (with as few states as possible).

b. (8%) Derive the logic for the state machine. Show the Boolean functions (in minimized SOP form) for the inputs to each flip-flop and the output. (Do not draw the logic schematic.)

Problem 5. (14%)

You are to design the controller of a self-controlled car that is guided by the tape on the ground. The specifications are as follows.

1. The controller has two inputs S_L and S_R that correspond to the outputs of the left and right sensors, respectively. S_L equals 1 if the left sensor detects the tape; otherwise, S_L equals 0. Similarly, S_R equals 1 if the right sensor detects the tape; otherwise, S_R equals 0.
2. The controller has two outputs T_R and T_L. If T_RT_L = 00, the car will go straight. If T_RT_L = 10, the car should turn right. If T_RT_L = 01, the car should turn left. Note that T_RT_L = 11 should never be used.
3. If both sensors detect the tape, the car should go straight.
4. If the left sensor detects the tape and the right sensor does not, the car should turn left.
5. If the right sensor detects the tape and the left sensor does not, the car should turn right.
6. If neither sensor detects the tape, the car should go left if it was last going left; otherwise, it should go right.

Design the state transition diagram of the controller with as few states as possible.

Problem 6. (16%)

Clock-gating is a popular low power design technique. If the present and next states of a flip-flip are the same, we can safely disable its clock, e.g., by keeping it zero, without affecting the circuit's operation. As shown in Fig. 2, when *enable* equals 0, the clock gater, which consists of just one AND gate in this example, will disable the clock signal to flip-flop F_2. As a result, Q_2 will remain at its present value.

a. (8%) Show how the timing of the *enable* signal and the delay of the AND gate (of the clock gater) affect the clock signal to F_2 and how this may affect the circuit operation.

b. (8%) Modify the clock gater to solve this problem. Show how your clock gater works.

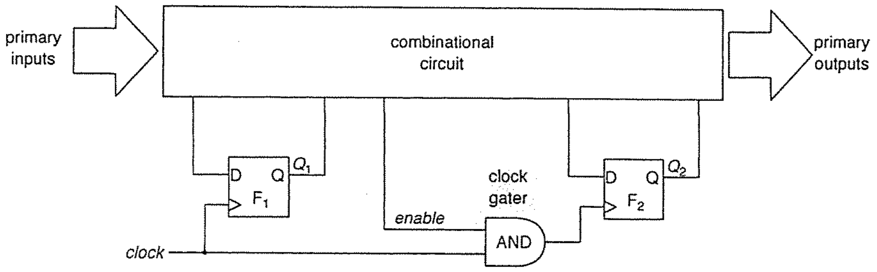

Fig. 2 A clock gating example (Problem 6).

接次頁

Problem 7. (20%)

Consider the sequential circuit X in Fig. 3. "addrGen" is a 4-bit binary counter which, when "*en*" equals 1, repeatedly generates the sequence $S_1S_2S_1S_2$…, where S_1 and S_2 are defined as follow.

S_1: $c_3c_2c_1c_0$ = 0000, 0001, 0010, …, 1110, 1111.

S_2: $c_3c_2c_1c_0$ = 0000, 0000, 0001, 0001, …, 1110, 1110, 1111, 1111.

If "*en*" equals 0, the counter is reset to $c_3c_2c_1c_0$ = 0000. The "*max*" output becomes one when $c_3c_2c_1c_0$ = 1111. (Fig. 4 shows the timing diagram of "addrGen.")

Complete the timing diagram of X in Fig. 5 till the output "*end*" first becomes 1. Properly simplify the timing diagram by skipping repeated events and showing only the important ones. Mark clearly the cycle indices (above the clock waveform) to indicate when the events occur.

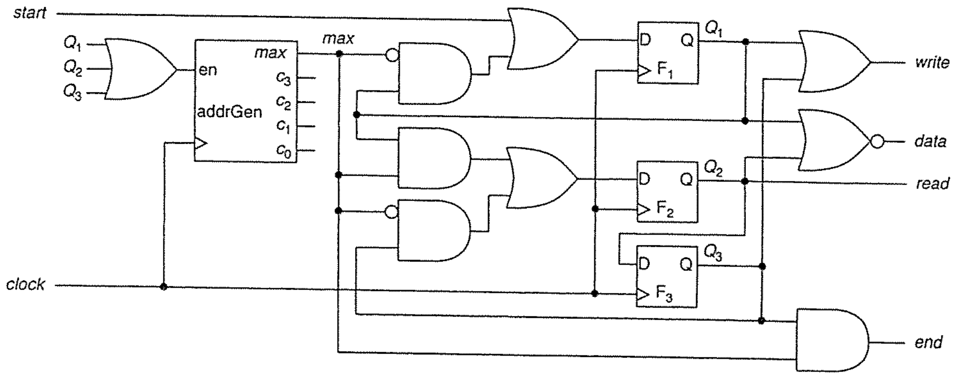

Fig. 3 The sequential circuit X (Problem 7).

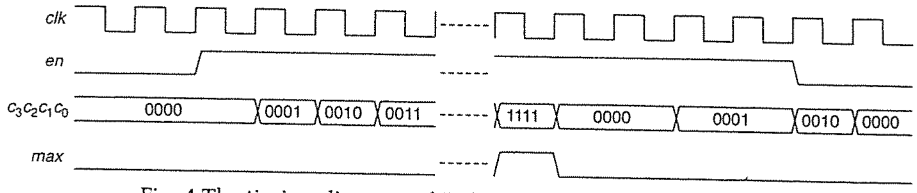

Fig. 4 The timing diagram of "addrGen" in Fig. 3 (Problem 7).

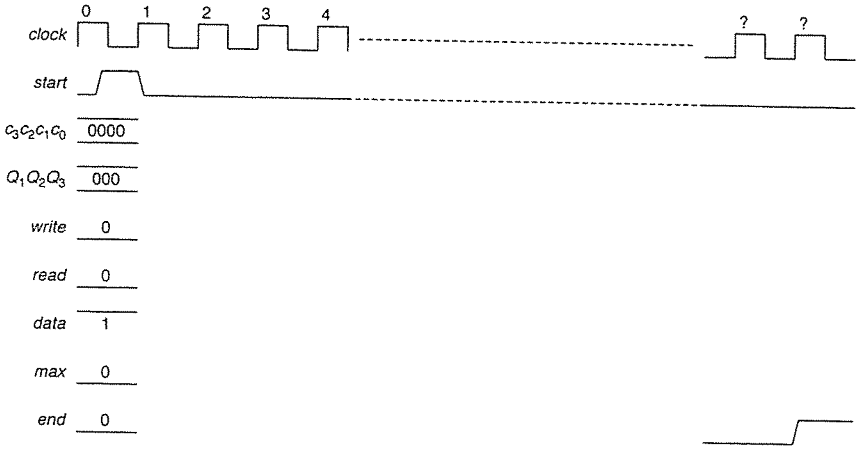

Fig. 5 The timing diagram of circuit X (Problem 7).

試題隨卷繳回