CSIE 2344 Midterm

□□□

TOTAL POINTS

84 / 100

QUESTION 1

1 Question 1 **12 / 12**

- ✓ + **12 pts** **Correct**
- + **11 pts** 1 mistake
- + **10 pts** 2 mistakes
- + **9 pts** 3 mistakes
- + **8 pts** 4 mistakes
- + **7 pts** 5 mistakes
- + **6 pts** 6 mistakes
- + **0 pts** No answer

QUESTION 2

2 Question 2 **8 / 8**

- ✓ + **8 pts** **Correct**
- + **6 pts** One minor mistake
- + **4 pts** One major mistake
- + **0 pts** No answer

QUESTION 3

3 Question 3 **8 / 8**

- ✓ + **8 pts** **Correct**
- + **6 pts** One minor mistake (in truth table)
- + **6 pts** One minor mistake (in K-map)
- + **6 pts** One minor mistake (no circuit)
- + **0 pts** No answer

QUESTION 4

4 Question 4-1 **8 / 8**

- ✓ + **8 pts** **Correct**
- + **6 pts** One minor mistake (correct K-map and four 2-literal PIs)
- + **6 pts** One minor mistake (one wrong minterm location)
- + **4 pts** Two minor mistakes (correct K-map and more than four 2-literal PIs)
- + **4 pts** One major mistake (multiple wrong minterm locations)
- + **4 pts** One major mistake (no "don't care")
- + **0 pts** No answer

QUESTION 5

5 Question 4-2 **8 / 8**

- ✓ + **8 pts** **Correct**
- + **6 pts** One mistake (same reason as Q4-1)
- + **6 pts** One minor mistake (almost correct)
- + **6 pts** One minor mistake (not minimum)
- + **4 pts** One major mistake (0 not covered)
- + **4 pts** One major mistake (all literals reversed)
- + **0 pts** No answer

QUESTION 6

6 Question 5-1 **2 / 8**

- + **8 pts** Correct
- + **6 pts** One minor mistake (not minimum)
- + **6 pts** One minor mistake (correct expression but wrong circuit)
- + **4 pts** One major mistake (static-1 hazard exists)
- + **4 pts** Two minor mistakes (wrong minterm locations in K-map and wrong circuit)
- ✓ + **2 pts** **One major mistake and one minor mistake (static-1 hazard exists and not minimum)**
- + **0 pts** No answer

QUESTION 7

7 Question 5-2 **6 / 8**

- + **8 pts** Correct
- + **6 pts** One minor mistake (with correct circling)
- ✓ + **6 pts** **One minor mistake (correct AND-OR gate circuit but not minimum)**
- + **4 pts** One major mistake (reasonable AND-OR gate circuit but incorrect function)

+ **4 pts** One major mistake (correct OR-AND gate circuit or correct AND-OR-NOT circuit)

+ **2 pts** One major mistake and one minor mistake (reasonable OR-AND gate circuit)

+ **2 pts** Some reasonable effort

+ **0 pts** No answer

QUESTION 8

8 Question 6-1 **8 / 8**

✓ + **8 pts** **Correct**

+ **7 pts** Correct with a 1-input AND gate (we did not define this)

+ **6 pts** One missing or unnecessary NOT gate

+ **4 pts** Two missing or unnecessary NOT gates

+ **4 pts** Correct function but not using gate equivalence

+ **2 pts** More missing or unnecessary NOT gates

+ **1 pts** Some reasonable effort

+ **0 pts** No answer

QUESTION 9

9 Question 6-2 **0 / 8**

+ **8 pts** Correct

+ **6 pts** Correct but missing circuit or not minimum

+ **4 pts** Correct function AND format with one minor mistake

+ **2 pts** Correct function AND format with two minor mistakes

+ **4 pts** Function with one minor mistake AND correct format

+ **2 pts** Function with two minor mistakes AND correct format

+ **2 pts** Some reasonable effort with NAND circuit

+ **1 pts** Some reasonable effort

✓ + **0 pts** **Incorrect or no answer**

QUESTION 10

10 Question 7 **8 / 8**

✓ + **8 pts** **Correct**

+ **7 pts** Missing the case n = 1

+ **6 pts** Good direction and almost completed

+ **4 pts** Good direction but not completed

+ **2 pts** Some effort

+ **0 pts** No answer

QUESTION 11

11 Question 8-1/8-2 **8 / 8**

✓ + **4 pts** **Q8-1 correct: 28**

✓ + **4 pts** **Q8-2 correct: 1, 4, 7, 5**

+ **2 pts** Some reasonable effort (close to 28)

+ **0 pts** Wrong or no asnwer

QUESTION 12

12 Question 8-3 **8 / 8**

✓ + **8 pts** **Correct**

+ **6 pts** Correct with a minor mistake

+ **6 pts** Correct with some missing explanation

+ **4 pts** Correct with insufficient explanation or inefficient

+ **4 pts** Heuristic (cannot guarantee to minimize the total cost)

+ **2 pts** Some reasonable effort

+ **1 pts** Some effort

+ **0 pts** No answer

+ **1 pts** Bonus for detailed design and analysis

QUESTION 13

13 Midterm for Schedule-Conflicting Students **0 / 0**

✓ + **0 pts** **April 8**

CSIE 2344, Spring 2019 — Midterm

Name: 陳約廷 SID: B04705001 Email: eopxd@ntu.im

1 True or False (12pts)

Determine whether the following statements are true or false by circling the correct choice. No explanation is required.

T [✓ F] 1. (1pt) Use 2's complement and 5-bit word, $-10_{10} = 10100_2$.

T [✓ F] 2. (1pt) Use 2's complement, 4-bit words, and a 4-bit parallel adder, assume A_3, B_3 are the first bits (*i.e.*, highest bits or most significant bits) of the two input words and S_3 is the first bit of the output sum, there is an overflow if and only if $(A_3 \equiv B_3) \oplus S_3 = 1$.

T [✓ F] 3. (1pt) Assume X, Y, Z are Boolean variables, $X \oplus YZ = (X \oplus Y)(X \oplus Z)$ is always true.

[✓ T] F 4. (1pt) Assume X, Y, Z are Boolean variables, $(X + Y)(X + Z) = X + YZ$ is always true.

[✓ T] F 5. (1pt) Follow the numbering method for minterms and maxterms in the lecture, $(m_i)' = M_i$.

[✓ T] F 6. (1pt) It is possible to reorder indexes (00-01-11-10 of both sides) in a Karnaugh map so that m_{15} is at the bottom-right corner and the Karnaugh map can still work.

T [✓ F] 7. (1pt) With three Boolean variables, $F = \sum m(0, 1, 2, 3)$ if and only if $F' = \prod M(4, 5, 6, 7)$.

T [✓ F] 8. (1pt) With three Boolean variables, $F_1 = \sum m(0, 1, 2, 3)$ and $F_2 = \sum m(0, 1, 4, 5)$ if and only if $F_1 \cdot F_2 = \sum m(0, 1)$.

T [✓ F] 9. (1pt) {XOR} is a functional complete set.

T [✓ F] 10. (1pt) {AND, XNOR} is a functional complete set.

[✓ T] F 11. (1pt) Given a Boolean function $F = A'B'C' + C$, $A'B'$ is a prime implicant.

[✓ T] F 12. (1pt) Given a Boolean function $F = A'B'C' + C$, $A'B'$ is an essential prime implicant.

2 Conversion to Base 7 (8pts)

Convert 71.75_{10} to base 7.

7 | 71 ⋯ 1
7 | 10 ⋯ 3
7 | 1 ⋯ 1
0

.75
× 7
.25 ⋯ 5
7
.75 ⋯ 1
7
.25 ⋯ 5

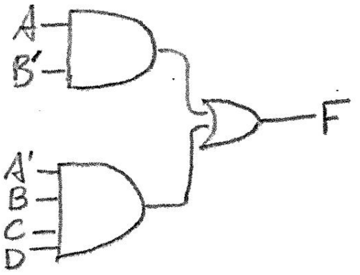
$$71.75_{10} = 131.\overline{51}_7$$

3 7-ELEVEn (8pts)

Design a combinational logic circuit which has one output F and a 4-bit input $ABCD$ representing a binary number ($ABCD = 0001$ represents 1_{10}). F should be 1 if and only if the input is at least 7 but no greater than 11 (*i.e.*, $7 \leq \text{input} \leq 11$). Use one OR gate and two AND gates.

CD \ AB	00	01	11	10
00	0	4	12	8 1
01	1	5	13	9 1
11	3	7 1	15	11 1
10	2	6	14	10 1

$F = AB' + A'BCD$



4 Karnaugh Maps (16pts)

Given $F(A,B,C,D) = \sum m(3,4,5,6,7,9,11,12,15) + \sum d(2,13)$.

1. (8pts) Find a minimum sum-of-products expression for F. Only the K-map and the final expression are required.

CD \ AB	00	01	11	10
00	0	4 1	12 1	8
01	1	5 1	13 X	9 1
11	3 1	7 1	15 1	11 1
10	2 X	6 1	14	10

(SOP) $F = A'C + BC' + AD$

2. (8pts) Find a minimum product-of-sums expression for F. Only the K-map and the final expression are required.

CD \ AB	00	01	11	10
00	0 0	4	12	8 0
01	1 0	5	13 X	9
11	3	7	15	11
10	2 X	6	14 0	10 0

(POS)

$$F' = B'D' + A'B'C' + ACD'$$

$$F = (B'D' + A'B'C' + ACD')'$$

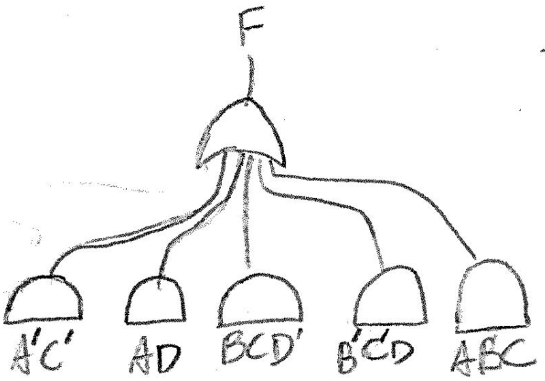
$$= (B+D)(A+B+C)(A'+C'+D)$$

(先AND再OR) = SOP

5 Static Hazards (16pts)

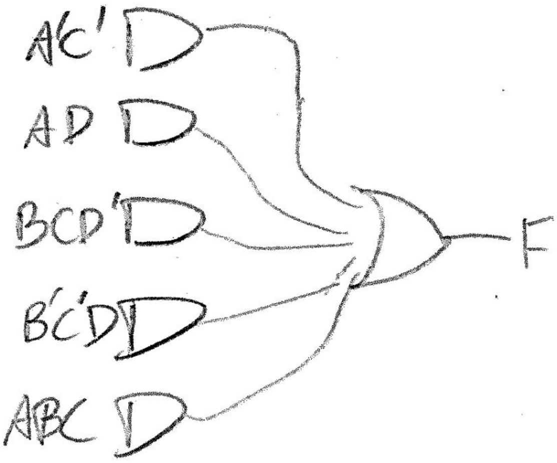
Given $F(A, B, C, D) = A'C' + AD + BCD'$.

1. (8pts) Draw a minimum two-level AND-OR gate circuit (AND at the second level, closer to the inputs, and OR at the first level, closer to the output) for F without static-1 hazards.



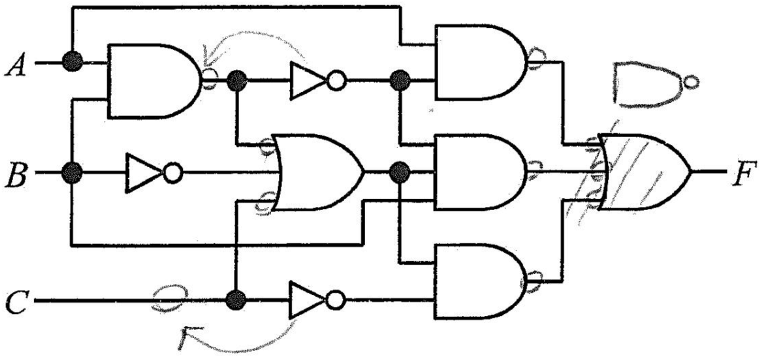
CD \ AB	00	01	11	10
00	1	1		
01	1	1	1	1
11			1	1
10		1	1	

2. (8pts) Draw a minimum two-level AND-OR gate circuit for F without static-0 hazards.

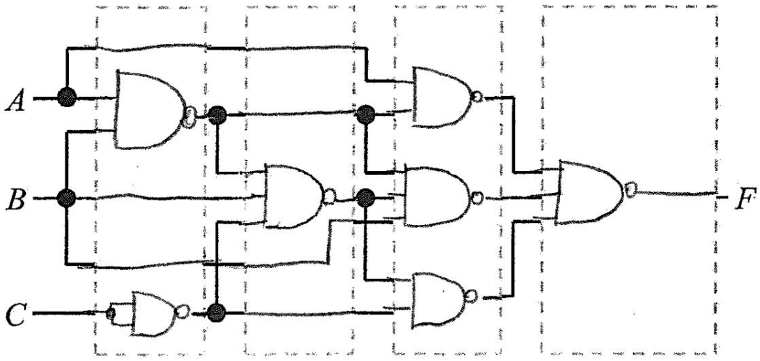


6 Circuit Conversion (16pts)

1. (8pts) Use gate equivalences to convert the following circuit into a four-level circuit containing only NAND gates (NOT gates are not allowed) and circuit inputs A, B, C (A', B', C' are not allowed as circuit inputs). To get the all points, the number of NAND gates should be 7. No explanation is required.



Draw Your Circuit Below



2. (8pts) Convert the circuit to a minimum three-level 2-input or 3-input NAND gate circuit where A', B', C' are ALLOWED as circuit inputs, but A, A' are not allowed as the inputs of third-level gates. **(Warning: this one may take some time!)**

7 Proving DeMorgan's Law (8pts)

Assume $X_1, X_2, \ldots, X_n$ are Boolean variables, prove $(X_1 + X_2 + \cdots + X_n)' = X_1'X_2' \ldots X_n'$ for $n \geq 1$. You should not assume that DeMorgan's Laws are true before proving them. (Hint: how to prove it for any positive integer n?)

For n variable AND-gate, $X_1X_2\cdots X_n = 1$ iff $X_1 = X_2 = \cdots = X_n = 1$ $\cdots$ $\textcircled{1}$

For n variable OR-gate, $(X_1+X_2+\cdots+X_n)' = 1$ iff $X_1 = X_2 = \cdots = X_n = 0$ $\cdots$ $\textcircled{2}$

$\textcircled{1}$: $(X_1X_2\cdots X_n) = 1 \Leftrightarrow X_1 = X_2 = \cdots = X_n = 1$

$\textcircled{2}$: $(X_1+X_2+\cdots+X_n)' = 1 \Leftrightarrow X_1 = X_2 = \cdots = X_n = 0$

Let transfer function $\phi(X_1, X_2, \cdots X_n) = (X_1', X_2', X_3', \cdots, X_n')$

$$(X_1 + X_2 + \cdots + X_n)' = 1 \overset{\textcircled{2}}{\Leftrightarrow} X_1 = X_2 = \cdots = X_n = 0$$

$$\Leftrightarrow \phi(Y_1, Y_2, \cdots, Y_n) \Leftrightarrow Y_1 = Y_2 = \cdots = Y_n = 1$$

$$\overset{\textcircled{1}}{\Leftrightarrow} Y_1Y_2\cdots Y_n = 1$$

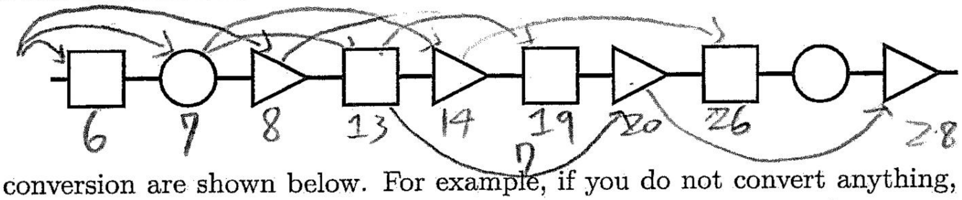
By transfer function, $X_i' = Y_i$,

We have $(X_1+X_2+\cdots+X_n)' = X_1'X_2'\cdots X_n'$ #

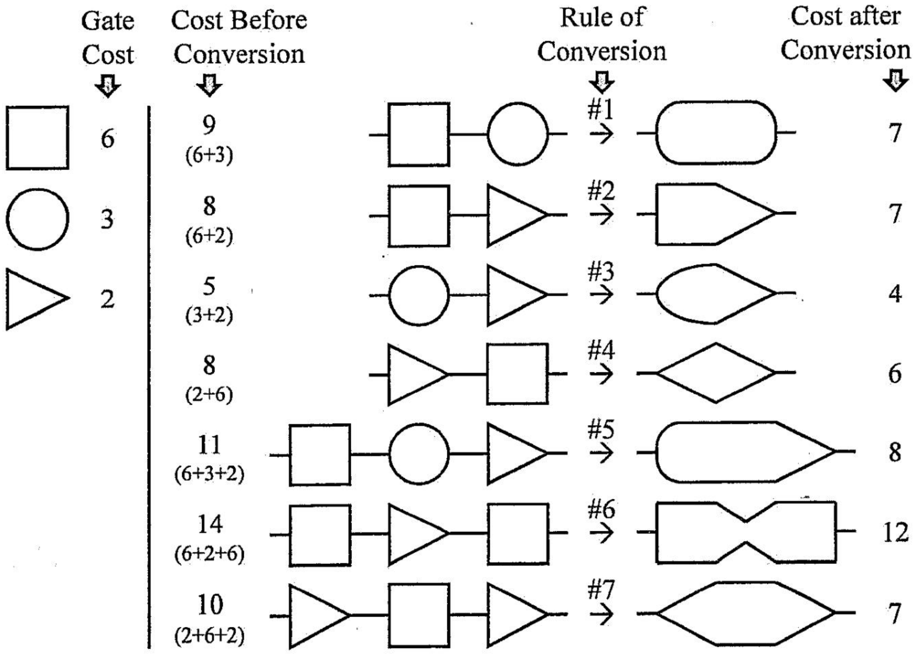
8 Logic Minimization (16pts)

6

Assume that each shape is one type of gates, and the input and output lines of each gate are 2019 bits. The goal of this problem is to convert the following sequence of gates into another sequence of gates so that the total cost is minimized.



The rules of conversion are shown below. For example, if you do not convert anything, the total cost is $6+3+2+6+2+6+2+6+3+2=38$. If you convert the last two gates (circle and triangle) by the rule #3, the total cost is $6+3+2+6+2+6+2+6+4=37$. Note that you can choose not to convert a gate if the total cost can be minimized. Also, a rule of conversion can be applied more than once.



1. (4pts) Write down the minimum total cost after converting the given sequence. No explanation is required. No partial credit will be given.

Cost = 二十八 = 28

2. (4pts) Write down the applied rules of conversion from left of the sequence to right of the sequence. No explanation is required. No partial credit will be given. **(Warning: one more question is on the next page!)**

#1, #4, #7, #5

7 + 6 + 7 + 8 = 28

3. (8pts) Explain(*e.g.*, write down Pseudocode) how to systematically and efficiently convert any sequence of gates (square, circle, and triangle) into another sequence of gates by the rules of conversion above and minimize the cost.

```
Gates = {Sequence of gates} = {$G_1 \cdots G_n$}   (n gates)
dp[0] = 0
dp[1] = dp[2] = ... = dp[n] = Infinity
for i = 0 ~ (n-1) {
    dp[i+1] = min (dp[i+1), dp[i] + Cost ($G_{i+1}$))
    if (i+2 ≤ n) and {$G_{i+1}$, $G_{i+2}$} in Rules :
        dp[i+2] = min ( dp[i+2], Cost (Rules({$G_{i+1}$, $G_{i+2}$})))
    if (i+3) ≤ N and {$G_{i+1}$, $G_{i+2}$, $G_{i+3}$} in Rules :
        dp[i+3] = min (dp[i+3], Cost (Rules({$G_{i+1}$, $G_{i+2}$, $G_{i+3}$})))
}
Min Cost = dp[n]
```