

# Chapter 3

# Gate-Level Minimization

J.J. Shann

1



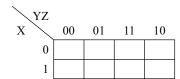
# §3-2

- 1. Simplify the following Boolean function into sum-of-products by using Karnaugh maps (K-maps), and calculate the gate input counts of *F* before and after the optimization:
  - (a) i.  $F(X, Y, Z) = \Sigma m(1, 3, 4, 5, 6, 7)$

ii. 
$$F(X, Y, Z) = XY + XZ + X'YZ'$$

(b) 
$$F(X, Y, Z) = XYZ' + (X' + Z)(Y + Z')$$

(a) XZ 00 01 11 10 0 1 1 1 10



EX-2

2



#### §3-3

- 2. Optimize the following Boolean function into sum-of-products by using K-map. Find all prime implicants and essential prime implicants, and apply the selection rule:
  - (a)  $F(A, B, C, D) = \Sigma m(1,5,6,7,11,12,13,14,15)$
  - (b) F(A, B, C, D) = A'BCD + A'BCD' + A'C'D + AB'D + ACD' + ABC + (A + B' + C + D)'

(a)	CD AB	00	01	11	10
	00				
	01				
	11				
	10				

EX-3

3



### §3-4

- 3. Optimize the following function into (i) sum-of-products and (ii) product-of-sums forms by using K-map, and calculate the gate input count for each form:
  - (a)  $F(A, B, C, D) = \Sigma m(1,5,6,7,11,12,13,14,15)$
  - (b) F(A, B, C, D) = A'BCD + A'BCD' + A'C'D + AB'D + ACD' + ABC + (A + B' + C + D)'

(a)	CD AB	00	01	11	10
	00				
	01				
	11				
	10				

CD				
AB	00	01	11	10
00				
01				
11				
10				

EX-4

1



#### §3-5

- 4. Optimize the following Boolean function F together with don't-care condition d into (i) sum-of-products and (ii) product-of-sums forms by using K-map. Express each simplified function in "Σm" notation, and determine the gate input count of it.
  - (a)  $F(W, X, Y, Z) = \Sigma m(5,7,11,14,15)$ ,
    - $d(W, X, Y, Z) = \Sigma m(1,6,12,13)$
  - (b)  $F(W, X, Y, Z) = \Sigma m(0,2,5,8,14,15),$ 
    - $d(W, X, Y, Z) = \Sigma m(4,7,10,13)$

YZ WX	00	01	11	10
00				
01				
11				
10				

EX-5

5



## Quine-McCluskey Method

- 5. Optimize the following Boolean function *F* together with don't-care condition *d* by using Quine-McCluskey method.
  - (a)  $F(X, Y, Z) = \Sigma m(1,5,6)$ ,  $d(X, Y, Z) = \Sigma m(2,7)$
  - (b) i.  $F(W, X, Y, Z) = \Sigma m(5,7,11,14,15),$ 
    - $d(W, X, Y, Z) = \Sigma m(1,6,12,13)$
    - ii.  $F(W, X, Y, Z) = \Sigma m(0,2,5,8,14,15),$ 
      - $d(W, X, Y, Z) = \Sigma m(4,7,10,13)$

(a)

Implication Table				
Column 1	Column 2	Column 3		

**Implication Chart** 





# Multiple-Level Circuit Optimization

- 6. Use decomposition to find a minimum gate input count, multiple-level implementation for each of the following functions using AND and OR gates and inverters. Draw the logic diagram and calculate the gate input counts of *F* before and after the decomposition.
  - (a) F(A, B, C, D) = ABC' + A'BC + A'CD + AC'D
  - (b) F(A, B, C, D) = AC + BC + A'BD + AB'D

EX-7

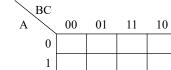
7

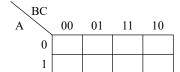


#### §3-6

- 7. Simplify the Boolean function and implement it
  - (a)  $F(A, B, C) = \Sigma m(1, 3, 4, 5, 6)$ 
    - i. by 2-level NAND gates ii. by 2-level NOR gates
  - (b)  $F(A, B, C, D) = \Sigma m(1,5,6,7,11,12,13,14,15)$ 
    - i. by 2-level NAND gates ii. by 2-level NOR gates

(a)



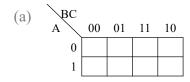


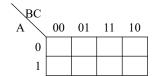
EX-8

8



- 8. Simplify the following functions with AND-NOR, NAND-AND, OR-NAND, NOR-OR 2-level forms:
  - (a)  $F(A, B, C) = \Sigma m(1, 3, 4, 5, 6)$
  - (b)  $F(A, B, C, D) = \Sigma m(1,5,6,7,11,12,13,14,15)$





EX-9

9



### §3-8

9. Consider a 3-bit message to be transmitted together w/ an odd parity bit. Design the odd-parity generation function *P* for the transmitter and the odd-parity checking function *C* for the receiver. Assume that the output of the checker is equal to 1 if error occurs.



# Brief Answers of the Exercises (1/4)

- $$\begin{split} 1. \;\; &(a) \;\; i. \;\; X+Z \;, \;\; GIC_{after} = 2 \\ &\quad ii. \;\; XZ+YZ' \;, \;\; GIC_{before} = 12, \;\; GIC_{after} = 7 \\ &(b) \;\; Y+X'Z' \;, \;\; GIC_{before} = 13, \;\; GIC_{after} = 6 \end{split}$$
- 2. (a) PIs: BD, AB, BC, ACD, A'C'D ; EPIs: AB, BC, ACD, A'C'D  $F{=}AB+BC+ACD+A'C'D$ 
  - (b) PIs: A'B, AC, BC, A'C'D, AB'D, B'C'D ; EPIs: A'B, AC  $F = A'B + AC + B'C'D \label{eq:F}$
- 3. (a) F = AB + BC + ACD + A'C'D, GIC = 16; F = (B+D)(A'+B+C)(A+B+C')(A+C+D), GIC = 17
  - (b) F = F = A'B + AC + B'C'D, GIC = 13F = (A' + B' + C) (A + B + C') (B + C + D), GIC = 15

EX-11

11



### Brief Answers of the Exercises (2/4)

(a) 
$$F = Y'Z + \begin{cases} YZ' \\ XY \end{cases}$$
  
(b) i.  $F = WYZ + XZ + \begin{cases} XY \\ WX \end{cases}$   
ii.  $F = X'Z' + XZ + \begin{cases} WXY \\ WYZ' \end{cases}$ 

# Brief Answers of the Exercises (3/4)

6. (a) F = (B + D) (AC' + A'C),  $GIC_{before} = 18$ ,  $GIC_{after} = 12$ 

```
(b) F = (A + B) (C + D (A' + B')) or
        = (A + B) (C + D) (C + A' + B')
      GIC_{before} = 16, GIC_{after} = 12
7. (a) i. F = A'C + AC' + AB'
                          B'C
          = ( (A'C)' (AC')' \lceil (AB')' \rceil )'
                            (B'C)'
      ii. F = (A+C)(A'+B'+C')
           = ((A+C)' + (A'+B'+C')')'
  (b) i. F = AB + BC + ACD + A'C'D
           = ( (AB)'(BC)'(ACD)'(A'C'D)' )'
      ii. F = (B+D) (A'+B+C) (A+B+C') (A+C+D)
           =((B+D)'+(A'+B+C)'+(A+B+C')'+(A+C+D)')'
```

EX-13

13

# Brief Answers of the Exercises (4/4)

```
8. (a) AND-NOR, NAND-AND: AOI
        F'=A'C'+ABC
        F = (A'C')' (ABC)'
     OR-NAND, NOR-OR: OAI
        F' = (A'+C)(A+C') (A'+B)^{-1}
                          (B+C')
        F = (A'+C)' + (A+C')' + (A'+B)'
                                (B+C')'
  (b) AND-NOR, NAND-AND: AOI
        F' = B'D' + AB'C' + A'B'C + A'C'D'
        F = (B'D')' (AB'C')' (A'B'C)' (A'C'D')'
    OR-NAND, NOR-OR: OAI
        F' = (A'+B') (B'+C') (A'+C'+D') (A+C+D')
        F = (A'+B')' + (B'+C')' + (A'+C'+D')' + (A+C+D')'
9. P = (x \oplus y \oplus z)', C = (x \oplus y \oplus z \oplus P)'
```