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)

X \ YZ	00	01	11	10
0				
1				

X \ YZ	00	01	11	10
0				
1				

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)

AB \ CD	00	01	11	10
00				
01				
11				
10				

EX-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)

AB \ CD	00	01	11	10
00				
01				
11				
10				

AB \ CD	00	01	11	10
00				
01				
11				
10				

EX-4

§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 “Σ*m*” notation, and determine the gate input count of it.

   (a) F(W, X, Y, Z) = Σm(5,7,11,14,15),
   d(W, X, Y, Z) = Σm(1,6,12,13)

   (b) F(W, X, Y, Z) = Σm(0,2,5,8,14,15),
   d(W, X, Y, Z) = Σm(4,7,10,13)

(a)

WX \ YZ	00	01	11	10
00				
01				
11				
10				

WX \ YZ	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) = Σm(1,5,6), d(X, Y, Z) = Σm(2,7)

   (b) i. F(W, X, Y, Z) = Σm(5,7,11,14,15),
   d(W, X, Y, Z) = Σm(1,6,12,13)

   ii. F(W, X, Y, Z) = Σm(0,2,5,8,14,15),
   d(W, X, Y, Z) = Σm(4,7,10,13)

(a)

Implication Table		
Column 1	Column 2	Column 3

Implication Chart

EX-6

6

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) $F(A, B, C, D) = AC + BC + A'BD + AB'D$

§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)

A \ BC	00	01	11	10
0				
1				

A \ BC	00	01	11	10
0				
1				

§3-7

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) $F(A, B, C, D) = \Sigma m(1,5,6,7,11,12,13,14,15)$

(a)

A \ BC	00	01	11	10
0				
1				

A \ BC	00	01	11	10
0				
1				

§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)

1. (a) i. $X + Z$, $GIC_{after} = 2$

   ii. $XZ + YZ'$, $GIC_{before} = 12$, $GIC_{after} = 7$

   (b) $Y + X'Z'$, $GIC_{before} = 13$, $GIC_{after} = 6$

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

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

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) $F = F=A'B + AC + B'C'D$, $GIC = 13$

   $F=(A'+B'+C) (A+B+C') (B+C+D)$, $GIC = 15$

EX-11

Brief Answers of the Exercises (2/4)

4. (a) i. $F = WYZ + XZ + \left\{\begin{matrix} WX & = \Sigma m(5, 7, 11, 12, 13, 14, 15),\ GIC = 10 \\ XY & = \Sigma m(5, 6, 7, 11, 13, 14, 15),\ GIC = 10 \end{matrix}\right.$

   ii. $F = (X+Z) (W+X) \left\{\begin{matrix} (Y+Z) \\ (W+Z) \end{matrix}\right\} \left\{\begin{matrix} (W'+Y) \\ (X+Y) \end{matrix}\right\}$ * 4 possible solutions

   (b) i. $F = X'Z' + XZ + \left\{\begin{matrix} WXY & = \Sigma m(0, 2, 5, 7, 8, 10, 13, 14, 15),\ GIC = 12 \\ WYZ' & = \Sigma m(0, 2, 5, 7, 8, 10, 13, 14, 15),\ GIC = 12 \end{matrix}\right.$

   ii. $F = (X+ Z') \left\{\begin{matrix} (W'+X'+Y) \\ (X'+Y+Z) \end{matrix}\right\} \left\{\begin{matrix} (W+X'+Z) \\ (W+X'+Y') \end{matrix}\right\}$ * 4 possible solutions

5. (a) $F = Y'Z + \left\{\begin{matrix} YZ' \\ XY \end{matrix}\right\}$

   (b) i. $F = WYZ + XZ + \left\{\begin{matrix} XY \\ WX \end{matrix}\right\}$

   ii. $F = X'Z' + XZ + \left\{\begin{matrix} WXY \\ WYZ' \end{matrix}\right\}$

EX-12

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. $F = A'C + AC' + \left\{\begin{matrix} AB' \\ B'C \end{matrix}\right\}$

   $= (\,(A'C)'\,(AC')'\,\left\{\begin{matrix} (AB')' \\ (B'C)' \end{matrix}\right\}\,)'$

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

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

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

   $F'= (A'+C)(A+C')\left\{\begin{matrix} (A'+B) \\ (B+C') \end{matrix}\right\}$

   $F = (A'+C)' + (A+C')' + \left\{\begin{matrix} (A'+B)' \\ (B+C')' \end{matrix}\right\}$

   (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. $P = (x \oplus y \oplus z)'$, $C = (x \oplus y \oplus z \oplus P)'$