Chapter 2

Logic minimization

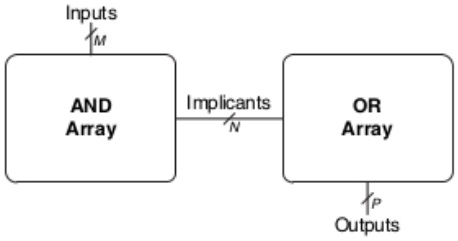
2.1 Logic minimization

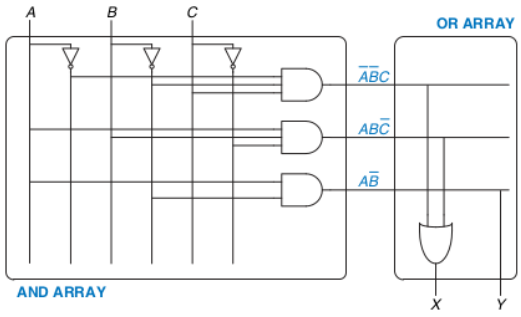
A general optimization criteria for multi-level logic are to Minimize some combination of:

1. Area occupied by the logic gates and interconnect;
2. the Critical Path Delay of the longest path through the logic;
3. the Degree of Testability of the circuit, measured in terms of the percentage of faults covered by a specified set of test vectors, for an appropriate fault model (Eg., single stuck faults, multiple stuck faults, etc.);
4. Power consumed by the logic gates.

In this course, we will start with two-level multi-input circuits and a criteria based on the number of gates/transistors/diodes.

2.2 Programmable Logic Arrays





2.3 Two-level circuits

The cost that we are going to consider in this class depend upon:

1. Number of gates.
2. Number of input to the gates.

More gates need more transistors, more area on the chip. More-inputs the gate need more transistors within each gate. Number of gate inputs can be considered secondary criterion to the number of gates.

Example 2.1. *Find the cost of the following Boolean expression* $X = \bar{A}\bar{B}C + AB\bar{C} + A\bar{B}$.

Problem 2.1 (5 marks)**.** *Find the cost of the following Boolean expression* $X = A\bar{B}C + \bar{A}B\bar{C} + \bar{B}C$.

2.4 Terminology for K-maps

Running Example: $f = \sum m(0, 1, 2, 3, 7) = \bar{x}_1 + x_1x_2x_3$.

Literal A single variable or its complement. Example: $\bar{x}, x_1, x_2, x_3$

Implicant A product term which is true for a function. All minterms are implicants. Example: $x_1x_2x_3$, $\bar{x}_1$, $m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$, $\bar{x}_1x_3$, $\bar{x}_1\bar{x}_3$.

Prime Implicant An implicant that cannot be combined into fewer literals. Example: $\bar{x}_1, x_2x_3$.

Essential Prime Implicant An implicant that cannot be combined into fewer literals. Example: x_2x_3.

Cover : List of Prime Implicants that account for all $f = 1$.

Cost : Number of gates (excluding not gate on literals) and number of inputs to each gate.

Example 2.2. *Find minimum cost expression for the function* $f(x_1, x_2, x_3) = \prod M(4, 5, 6)$

Problem 2.2 (10 marks)**.** *Find minimum cost expression for the function* $f(x_1, x_2, x_3) = \prod M(2, 5, 6)$

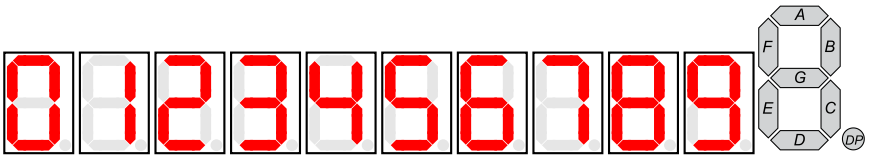


Figure 2.1: 7 Segment Representations of Each Integer

2.4.1 Incompletely specified functions or Don't cares

BCD Value				LED Segment
D_3	D_2	D_1	D_0	E
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	d
1	0	1	1	d
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d

Example 2.3. *Find minimum cost expression for the function*

$$f(x_1, \ldots, x_4) = \sum m(2, 4, 5, 6, 10) + D(12, 13, 14, 15)$$

Problem 2.3 (10 marks)**.** *Find minimum cost expression for the function*

$$f(x_1, \ldots, x_4) = \sum m(0, 2, 4, 6, 7, 8, 9, 13) + D(1, 12, 15)$$

2.5 A few more Boolean problems

Example 2.4. *Simplify the following Boolean expression:*

$$f = x_1\bar{x}_3\bar{x}_4 + x_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3$$

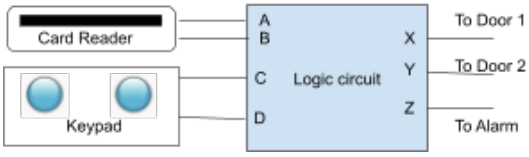
Problem 2.4 (20 marks)**.** *A simple security system for two doors consists of a card reader and a keypad.*

A person may open a particular door if he or she has a card containing the corresponding code and enters an authorized keypad code for that card. Note that card-code and keypad-code are different. The outputs from the card reader are given in the table below.

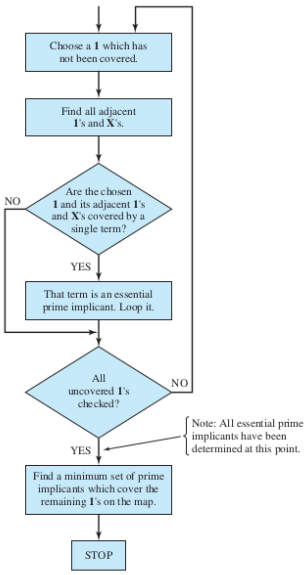
To unlock a door, a person must hold down the proper keys on the keypad and, then, insert the card in the reader. The authorized keypad code for door 1 is 10, and the authorized keypad code for door 2 is 11. If the card has an invalid code or if the wrong keypad code is entered, the alarm will ring when the card is inserted. If the correct keypad code is entered, the corresponding door will be unlocked when the card is inserted.

Design the logic circuit for this simple security system. Your circuit's inputs will consist of a card code AB, and a keypad code CD. The circuit will have three outputs XYZ (if X is 1, door 1 will be opened; if Y is 1, door 2 will be opened; if Z 1, the alarm will sound).

Find the minimal cost two-level circuit using K-maps for X, Y, Z. Provide the minimal cost. (It can be either of SOP/POS forms)



	A	B
No card inserted	0	0
Valid card-code for door 1	0	1
Valid card-code for door 2	1	1
Invalid card code	1	0



Row	x_1	x_2	x_3	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Table 2.1: Truth table for a 3-way light switch

Example 2.5.

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

Problem 2.5 (10 marks)**.** *Find the minimum SOP (sum of products) and POS (product of sum) expression for the function* $f(a,b,c,d) = \prod M(5,7,13,14,15) \cdot \prod D(1,2,3,9)$

Problem 2.6 (10 marks)**.** *If the Sum of Products (SOP) form for* $\bar{f} = AB\bar{C} + \bar{A}\bar{B}$*, then give the Product of Sums (POS) form for* f*.*

Problem 2.7 (10 marks)**.** *Use DeMorgan's Theorem to find* f *if* $\bar{f} = (A + \bar{B}C)D + EF$*.*

Example 2.6. *Implement the function in Table 2.1 using only NAND gates.*

Problem 2.8 (10 marks)**.** *Implement the function in Table 2.1 using only NOR gates.*

Problem 2.9 (10 marks)**.** *Find the minimum-cost Sum of Products (SOP) and Product of Sums (POS) forms for the function* $f(x_1,x_2,x_3) = m(1,3,4,5)$*. Chose the minimum-cost expression by comparing Product of Sums (POS) and Sum of Products (SOP) forms.*

Problem 2.10 (10 marks)**.** *Find the minimum-cost Sum of Products (SOP) and Product of Sums (POS) forms for the function* $f(x_1,x_2,x_3) = \sum m(1,5,7) + D(2,4)$*.*

Problem 2.11 (10 marks)**.** *Find the minimum-cost Sum of Products (SOP) and Product of Sums (POS) forms for the function* $f(x_1,x_2,x_3,x_4) = \prod M(1,2,4,5,7,8,9,10,12,14,15)$*. Chose the minimum-cost expression by comparing Product of Sums (POS) and Sum of Products (SOP) forms.*

Problem 2.12 (10 marks)**.** *Derive a minimum-cost realization of the four-variable function that is equal to 1 if exactly two or exactly three of its variables are equal to 1; otherwise it is equal to 0.*

Bibliography

[1] Sarah L Harris and David Harris. *Digital design and computer architecture*. Morgan Kaufmann, 2022.

[2] Brown Stephen and Vranesic Zvonko. *Fundamentals of digital Logic with Verilog design*. McGraw Hill, 2022.