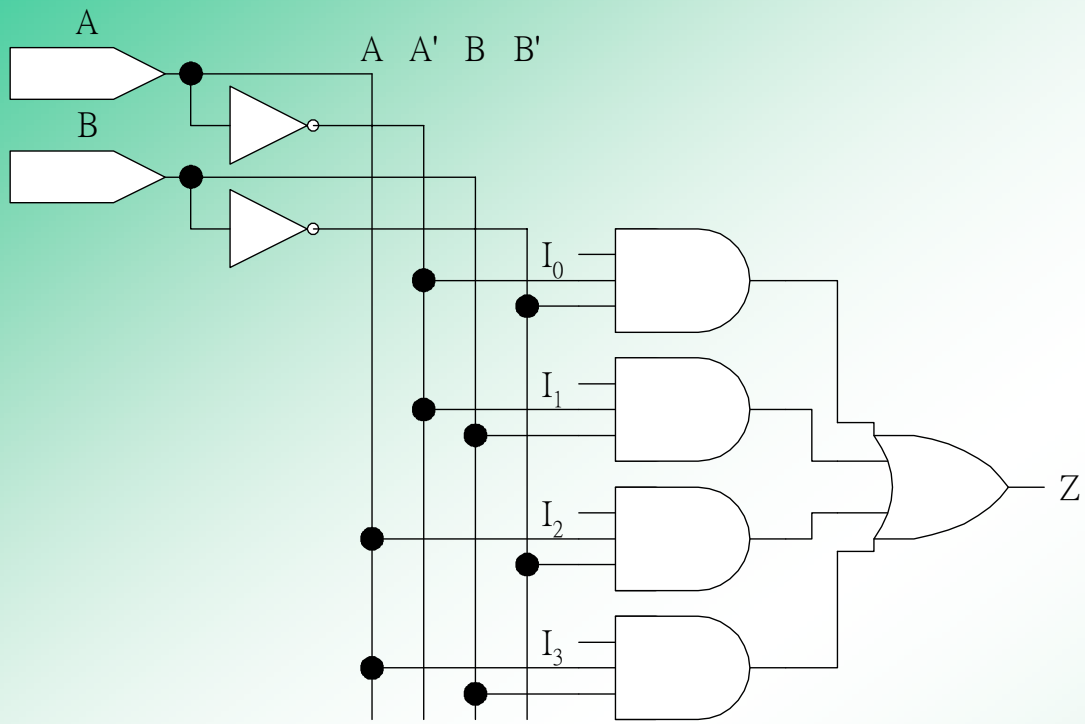
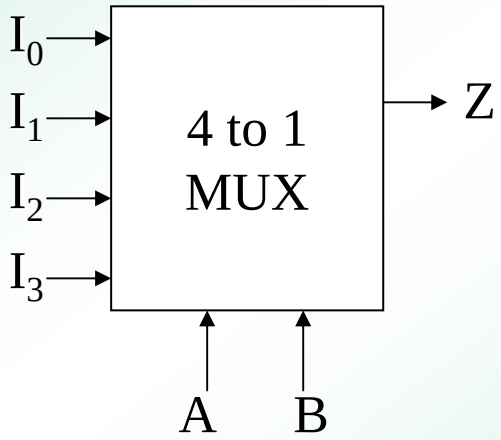
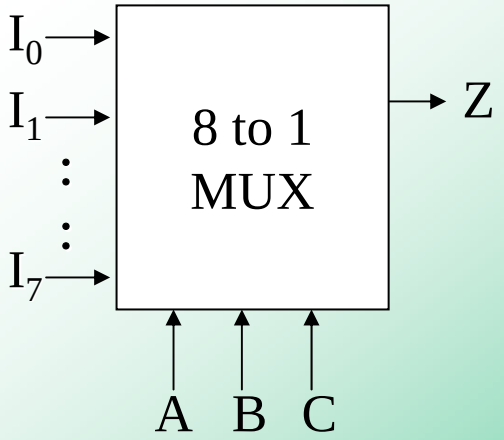
Unit 9.

Multiplexers, Decoders, and Programmable Logic Devices

Multiplexers (Data Selectors)



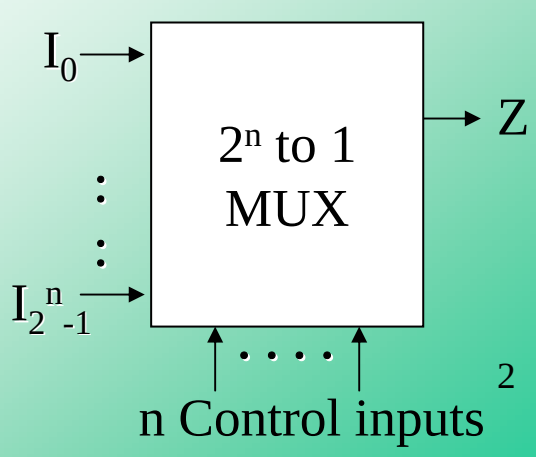




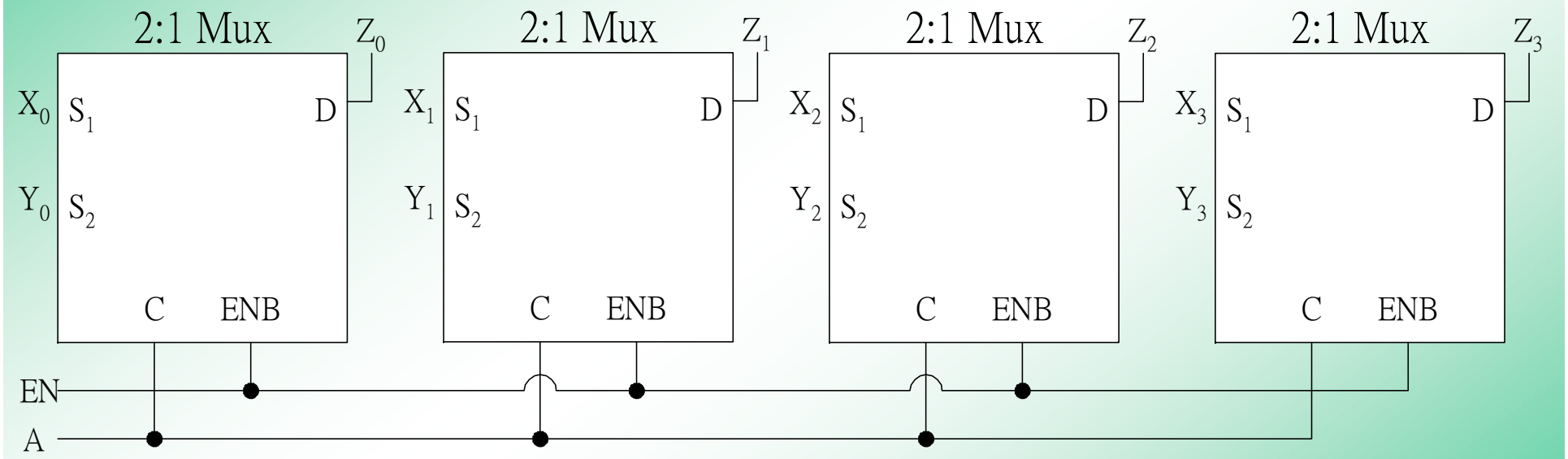
$$4-1: Z = A'B'I_0 + A'BI_1 + AB'I_2 + ABI_3$$

$$8-1: Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 + AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$$

$$2^n-1: Z = \sum_{k=0}^{2^n-1} m_k I_k$$



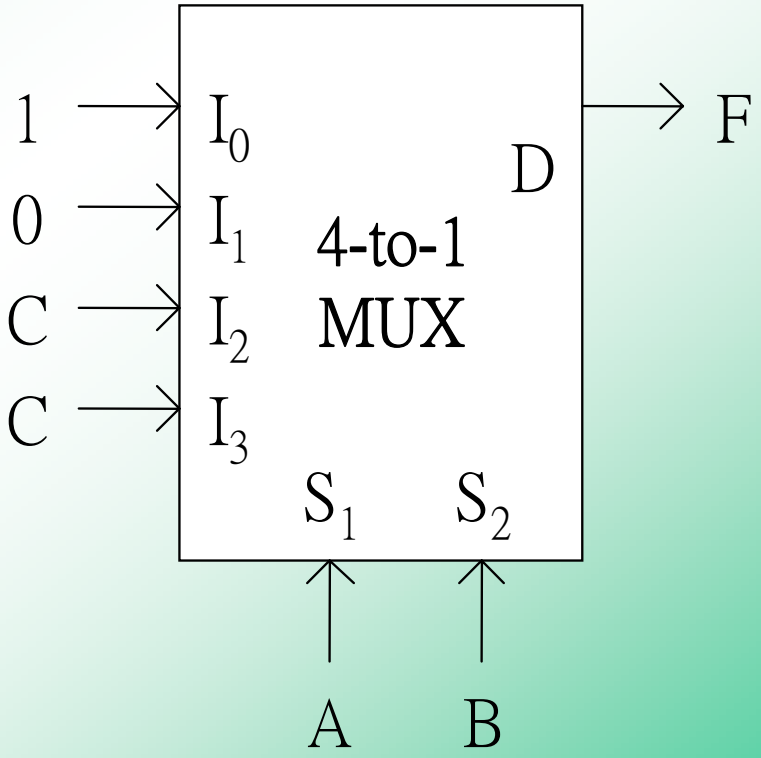
Quad Multiplexer to Select Data



4-to-1 MUX to realize 3-variable function

Ex :

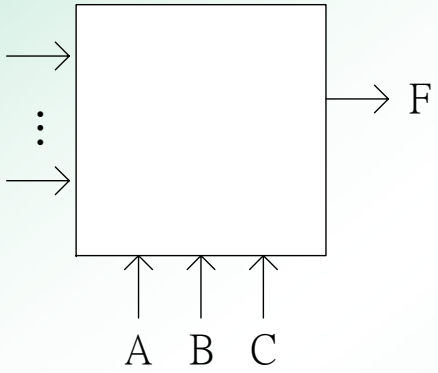
$$F(A,B,C) = A'B' + AC$$
$$= A'B' + AC(B + B')$$
$$= A'B' \cdot 1 + AB'C + ABC$$



8-to-1 MUX to realize 4-variable function

F = A’B’C’ + A’BC +
ABD’ + AC’D’ +
B’CD

= ***A’B’C’(D+D’)***
+ A’BC (D+D’)+
AB(C+C’)D’ +
A(B+B’)C’D’ +
(A+A’)B’CD



$A'B'C' = 1$

$A'B'C = D$

$A'BC' = 0$

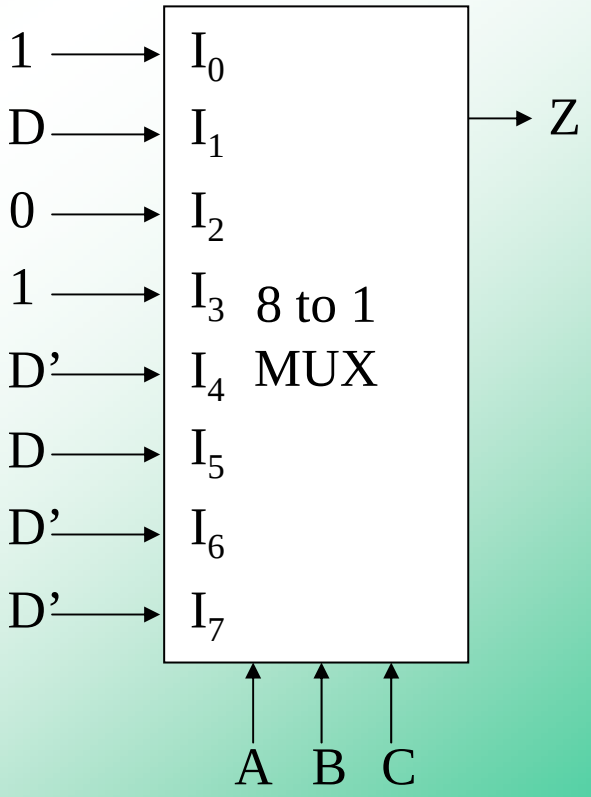
$A'BC = 1$

$AB'C' = D'$

$AB'C = D$

$ABC' = D'$

$ABC = D'$



	A	B	C	D	F	
I_0	0	0	0	0	1	1
	0	0	0	1	1	
I_1	0	0	1	0	0	D
	0	0	1	1	1	
I_2	0	1	0	0	0	0
	0	1	0	1	0	
I_3	0	1	1	0	1	1
	0	1	1	1	1	
I_4	1	0	0	0	1	$\overline{D}$
	1	0	0	1	0	
I_5	1	0	1	0	0	D
	1	0	1	1	1	
I_6	1	1	0	0	1	$\overline{D}$
	1	1	0	1	0	
I_7	1	1	1	0	1	$\overline{D}$
	1	1	1	1	0	

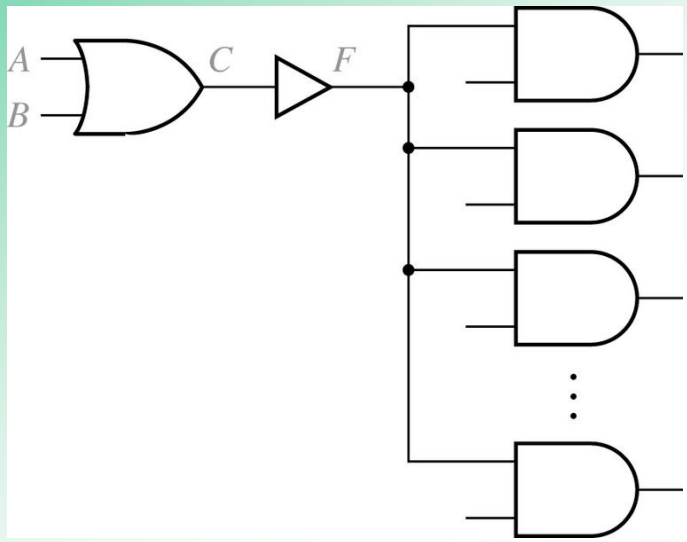
F=A'B'C'(D+D') + A'BC (D+D')+ AB(C+C')D' + A(B+B')C'D' + (A+A')B'CD

$I_0 I_1$	I_2	$\cdots$	I_{n-1}	F			
10	0	$\cdots$	0	0	0	1	1
11	0	$\cdots$	1	0	1	0	1
				$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
				0	I_n	$\overline{I_n}$	1

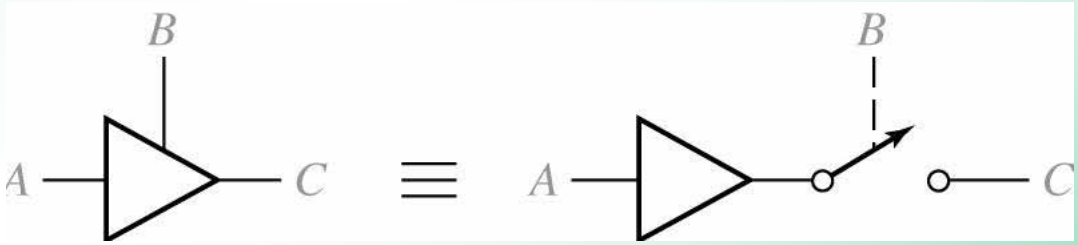
$$F = A'B'C' + A'B'CD + A'BC + AB'C'D' + AB'CD + ABC'D' + ABCD'$$
$$= A'B'C' + A'BC + ABD' + AC'D' + B'CD$$

§ Three - state buffers

Buffers: to increase the driving capability of a gate output



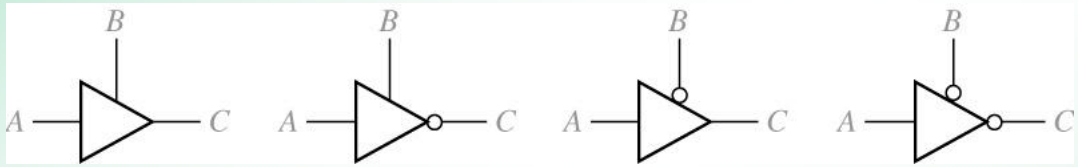
Tri-state Buffers: permits gate outputs to be connected together



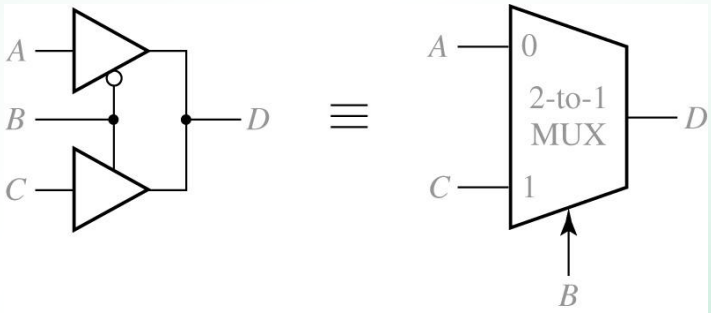
Truth tables

B	*A*	*C*
0	0	*Z*
0	1	*Z*
1	0	0
1	1	1

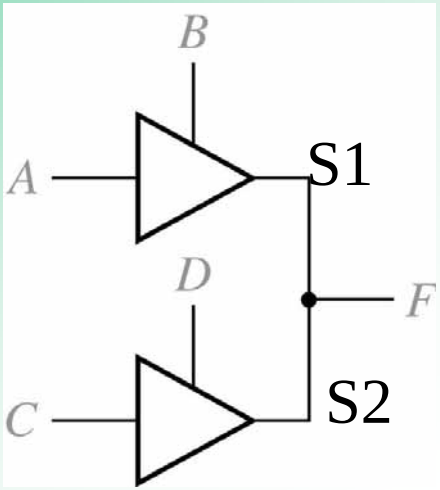
Truth Table



Data selection using three-state buffers



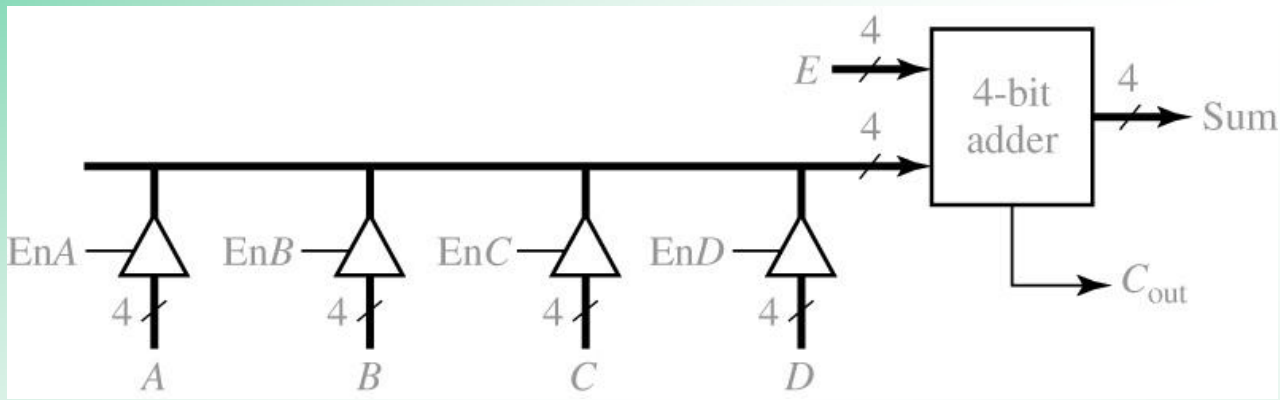
Circuits with two three state buffers



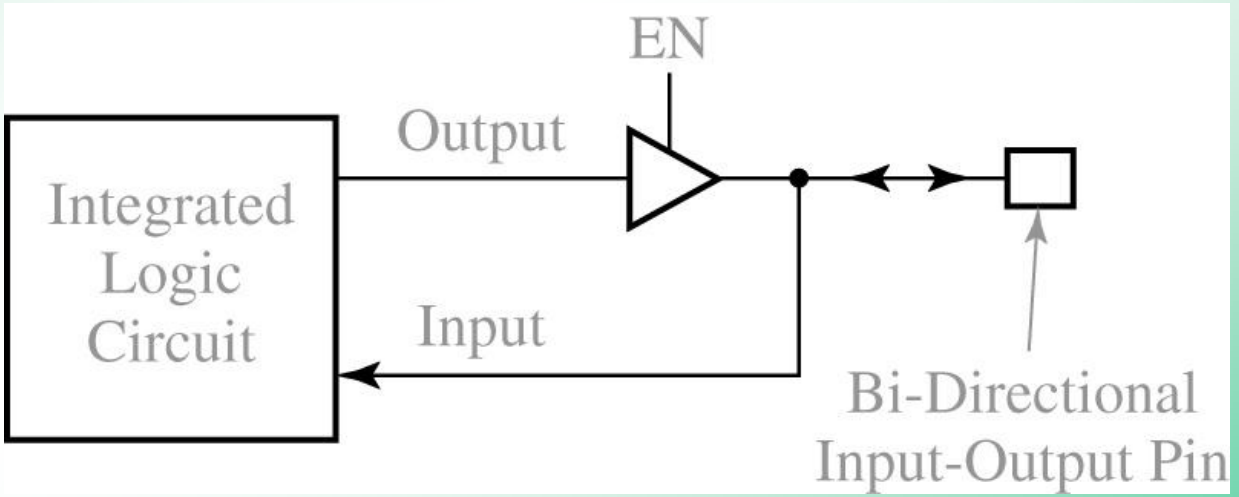
S1	S2			
	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

Application of three-state buffer

1. Bus

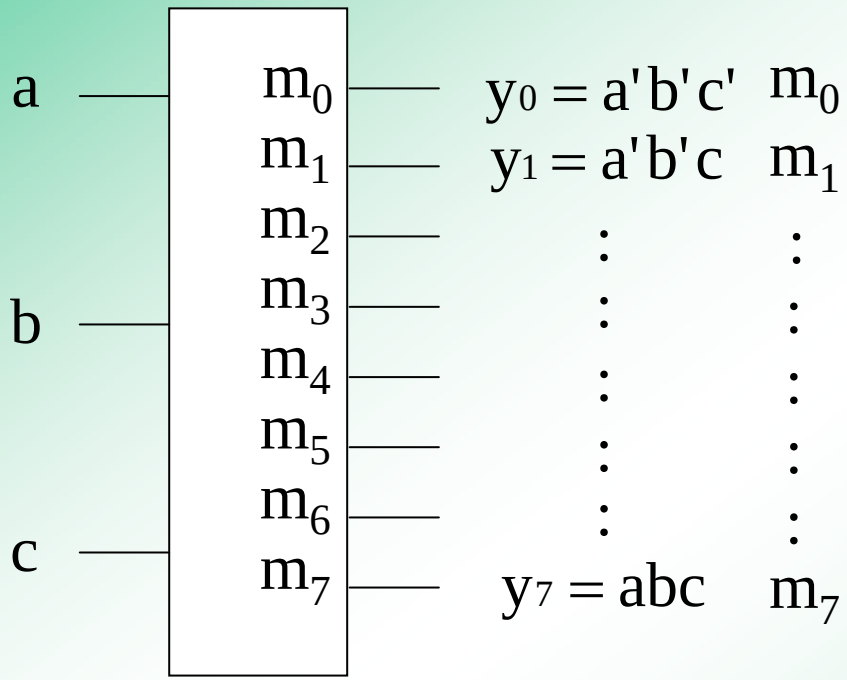


2. Chip I/O

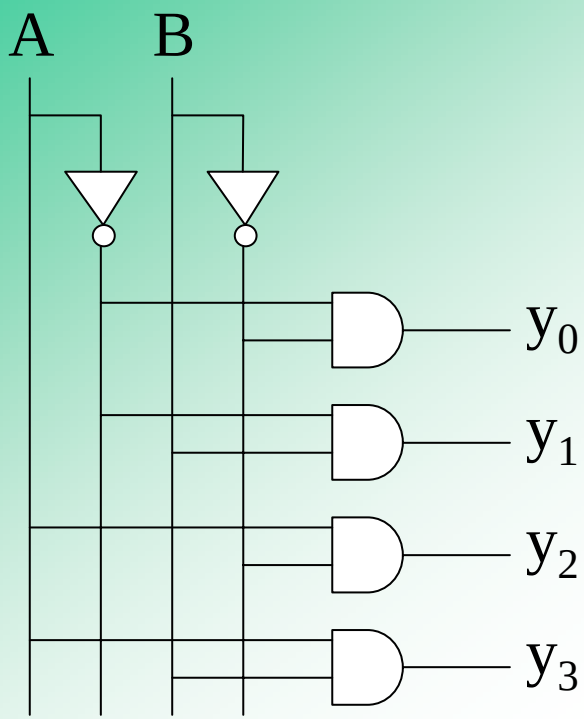


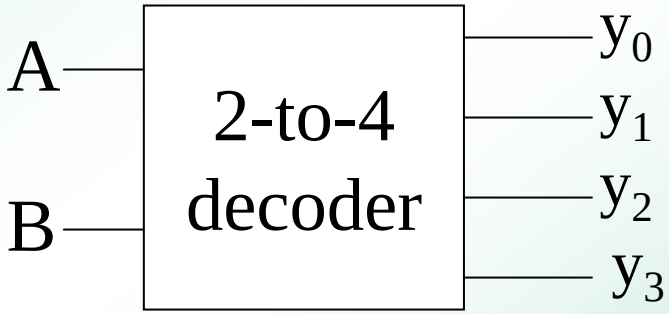
§ Decoders

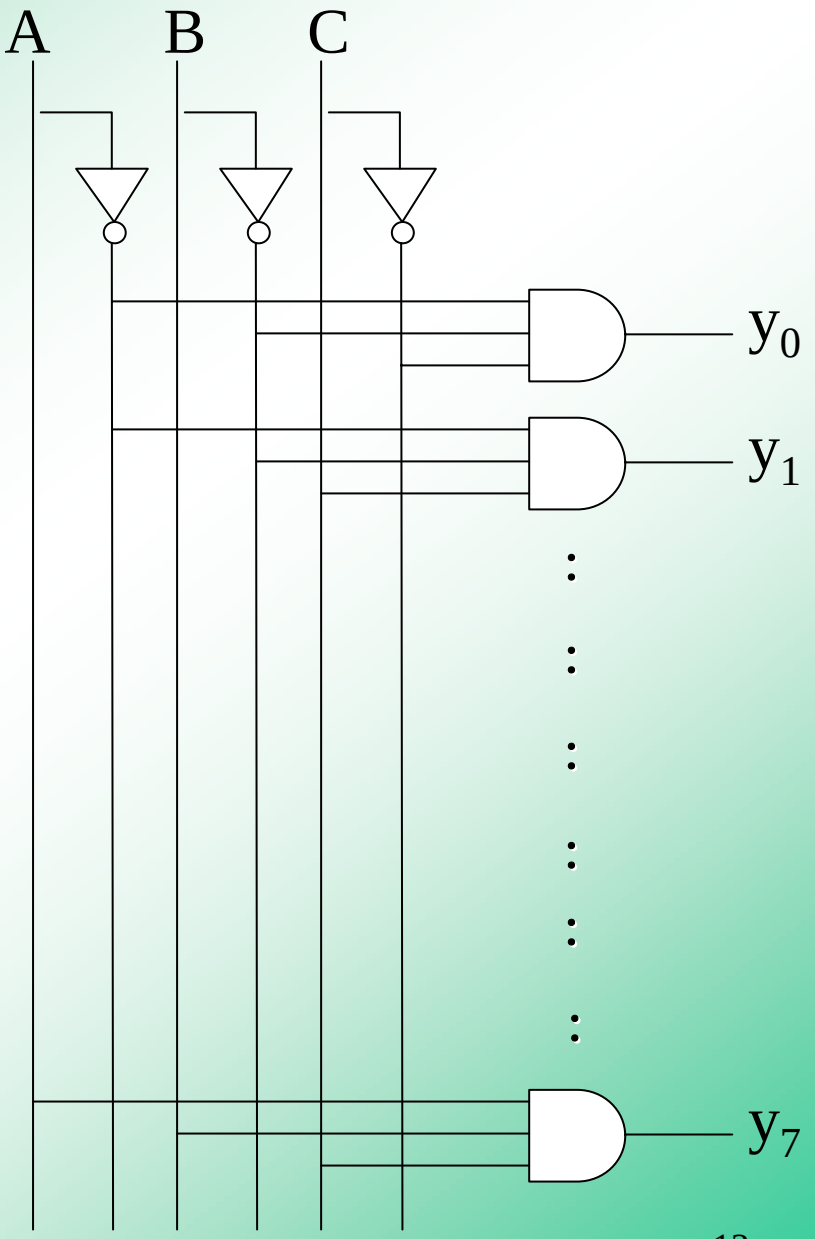
Generate all of the minterms of inputs

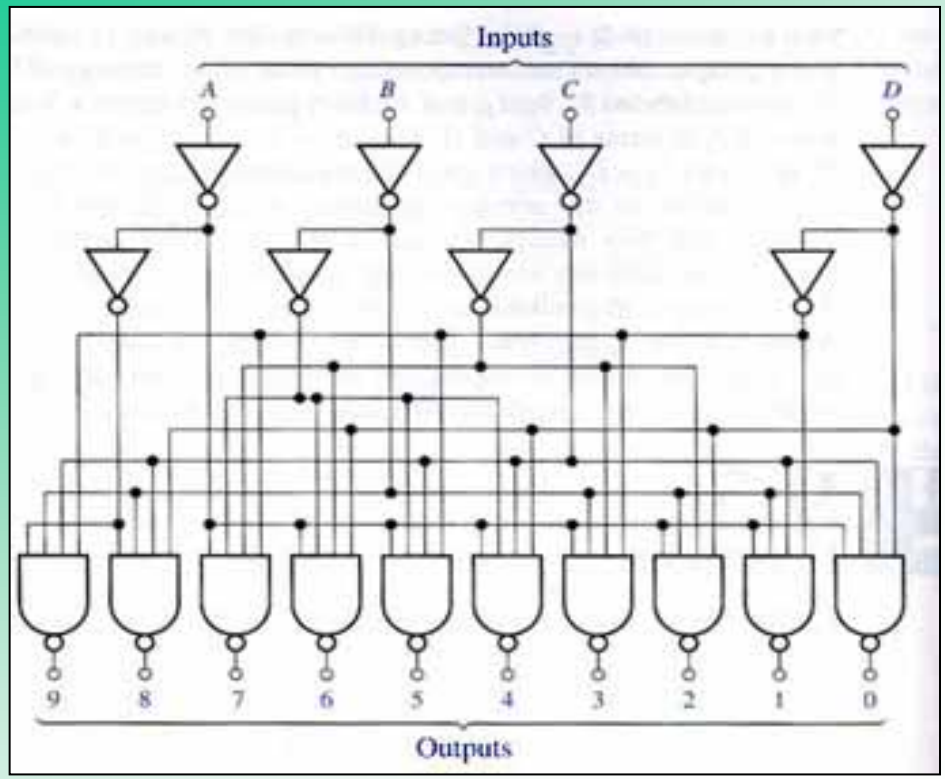


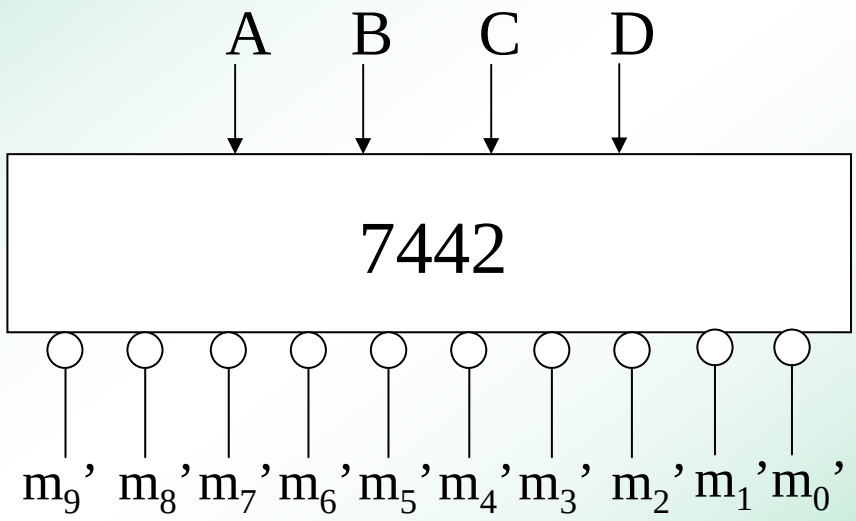
a	b	c	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	..	..	..	..	..	0
0	0	1	0	1						:
0	1	0	:		1					:
0	1	1	:			1				:
1	0	0	:				1			:
1	0	1	:					1		:
1	1	0	:						1	0
1	1	1	0	..	..	..	..	..	0	1


A
B
y_0
y_1
y_2
y_3


A
B
2-to-4
decoder
y_0
y_1
y_2
y_3

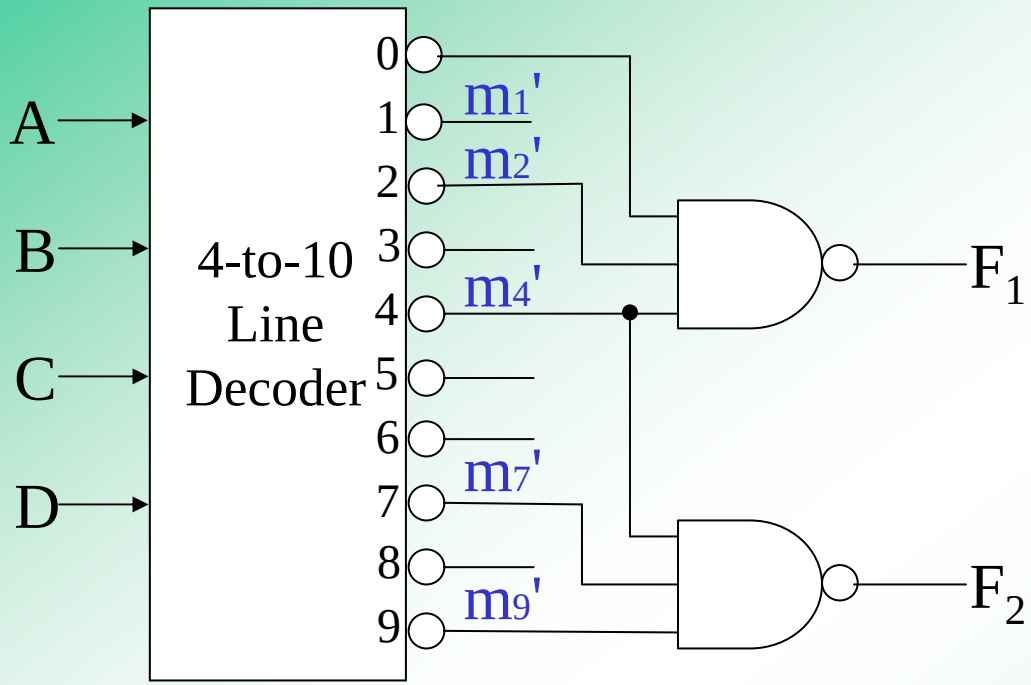

A
B
C
y_0
y_1
y_7





BCD input				Decimal Output									
A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Realization of a multiple output circuit using a decoder

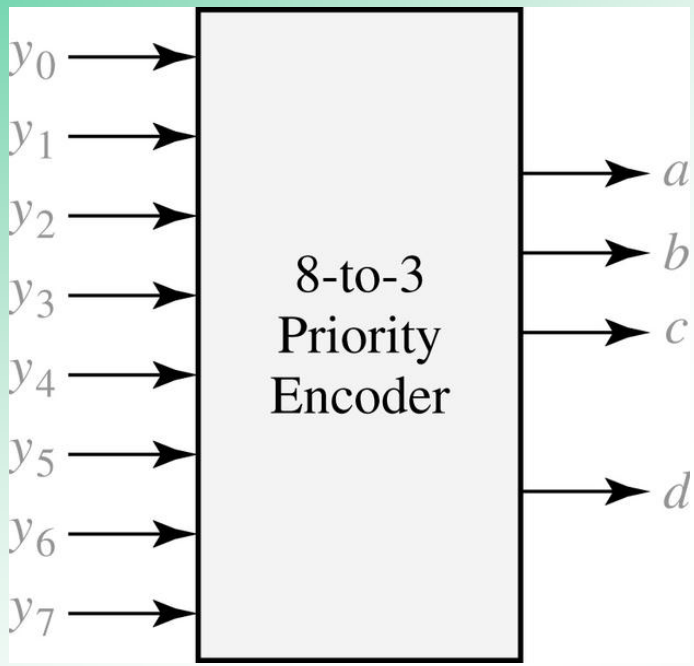


又 $F_1 = m_1 + m_2 + m_4 \quad F_2 = m_4 + m_7 + m_9$

可以 4-to-10 Decoder 來 realize

By DeMorgan's law

§ Encoders

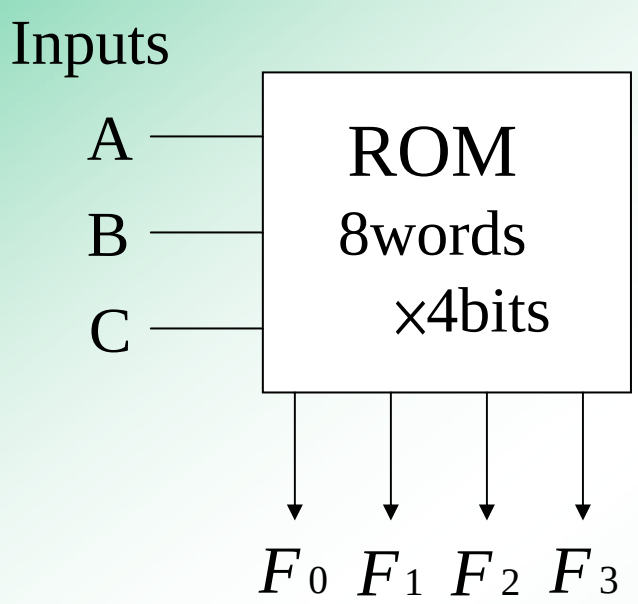


priority

y0	y1	y2	y3	y4	y5	y6	y7	a	b	c	d
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
X	1	0	0	0	0	0	0	0	0	1	1
X	x	1	0	0	0	0	0	0	1	0	1
X	X	X	1	0	0	0	0	0	1	1	1
X	X	X	X	1	0	0	0	1	0	0	1
X	X	X	X	X	1	0	0	1	0	1	1
X	X	X	X	X	x	1	0	1	1	0	1
X	X	X	X	X	X	X	1	1	1	1	1

§ Read Only Memory

An LSI circuit to realize multiple output Boolean function(s)



A	B	C	F_0	F_1	F_2	F_3
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	1	1	1	1	1

Stored in ROM (2^3 words of 4 bits each)

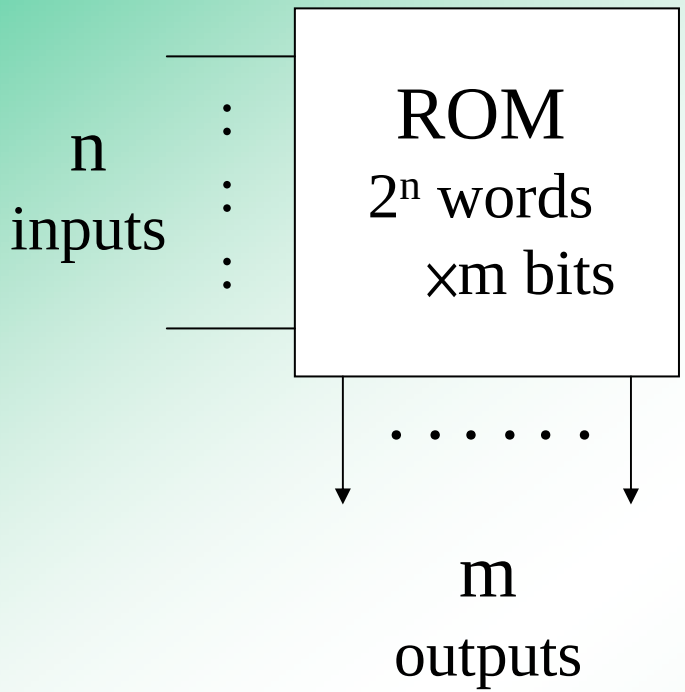
$$F_0 = m_0 + m_1 + m_5 + m_7$$

$$F_1 = m_2 + m_3 + m_4 + m_5 + m_7$$

$$F_2 = m_0 + m_1 + m_3 + m_4 + m_5 + m_6 + m_7$$

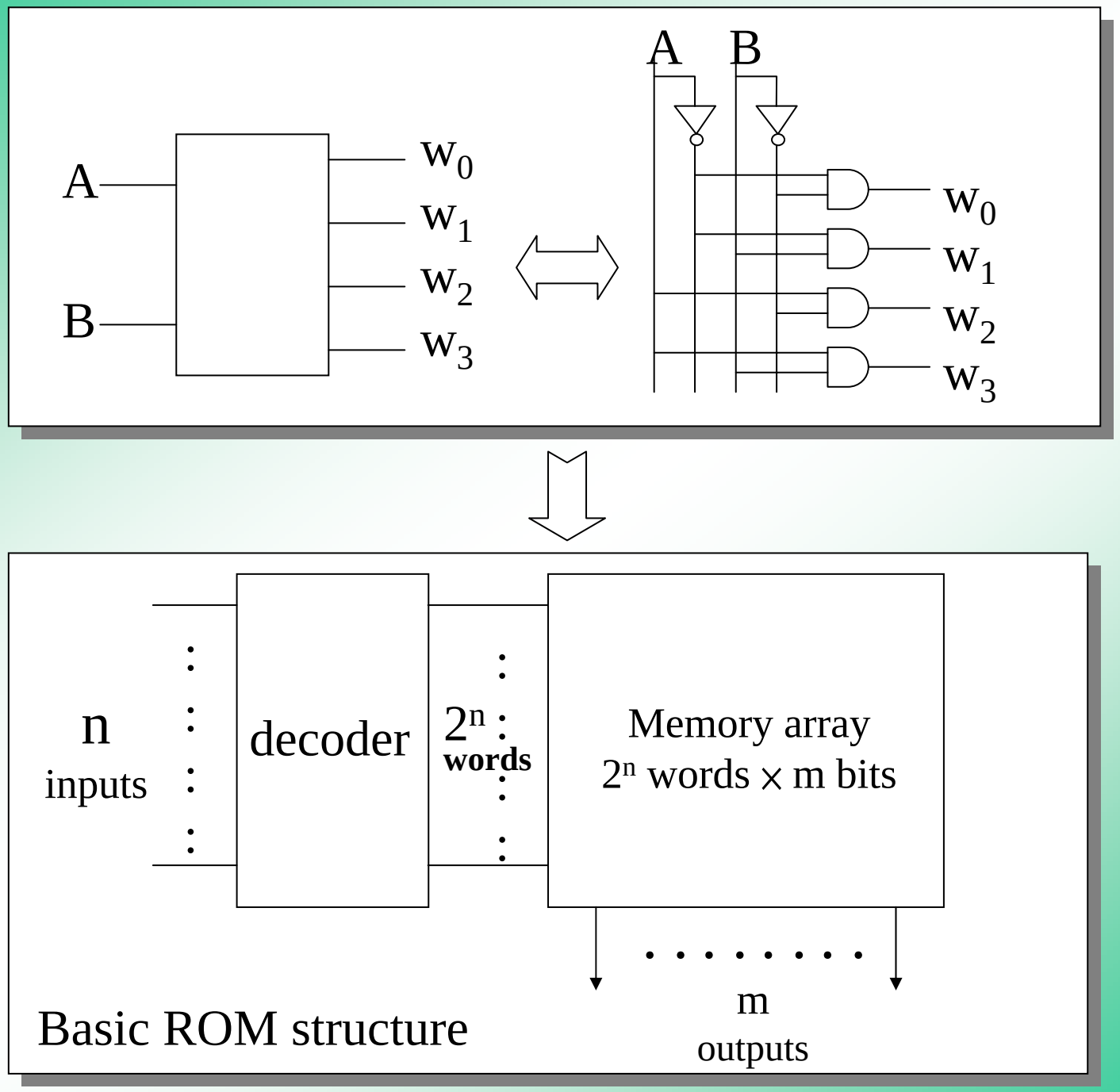
$$F_3 = m_6 + m_7$$

General Form



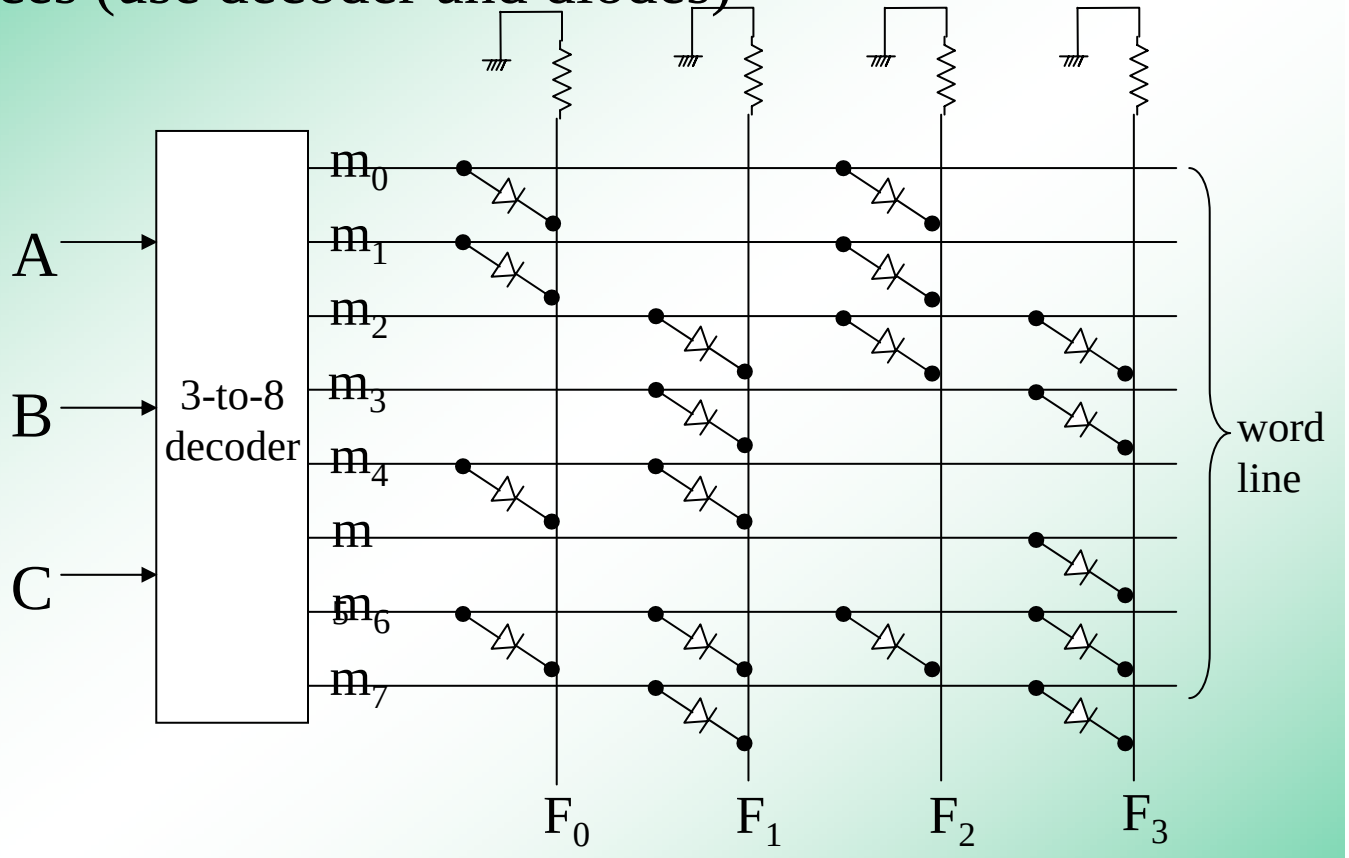
n inputs				m outputs				
				F_0	⋯	⋯	⋯	F_{m-1}
0	⋯	0	0	0	0	⋯	1	0
0	⋯	0	1	:				:
:			:	:				:
:			:	:				:
:			:	:				:
:			:	:				:
1	⋯	1	1	1	0	⋯	1	0

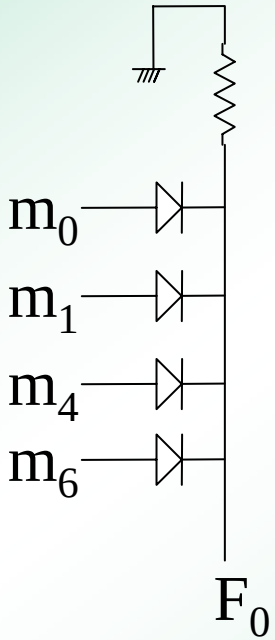
2^n words

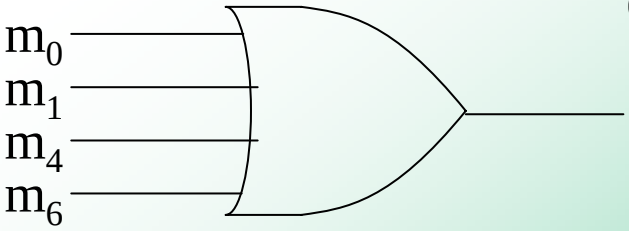

A
B
w_0
w_1
w_2
w_3
A B
w_0
w_1
w_2
w_3
n
inputs
decoder
2^n
words
Memory array
2^n words × m bits
m
outputs
Basic ROM structure

ROM as logic devices (use decoder and diodes)

A	B	C	F_0	F_1	F_2	F_3
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	1	1
0	1	1	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	1	0	1	1	1	1
1	1	1	0	1	0	1





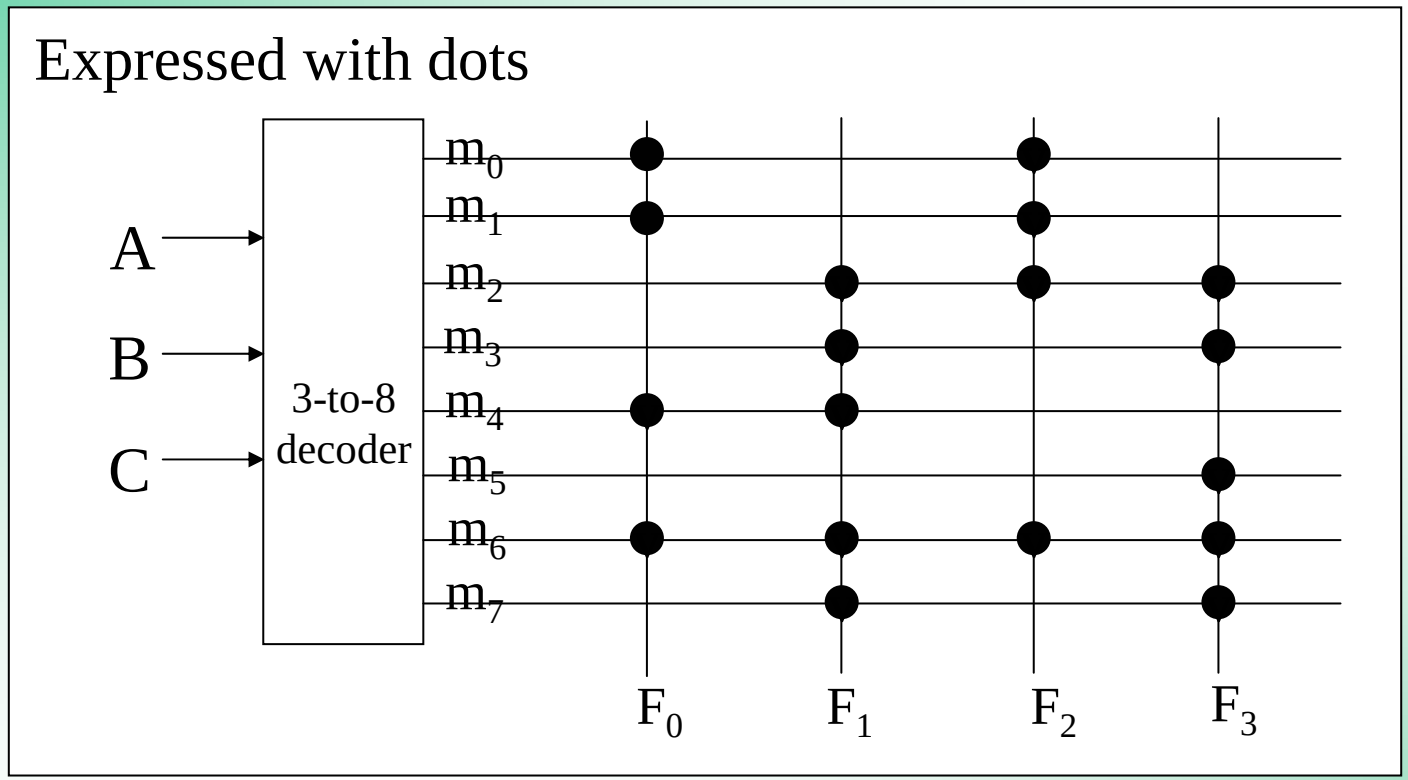


$$F_0 = m_0 + m_1 + m_4 + m_6$$

$F_0 = \sum m(0,1,4,6)$ $F_1 = \sum m(2,3,4,6,7)$

$F_2 = \sum m(0,1,2,6)$ $F_3 = \sum m(2,3,5,6,7)$

Expressed with dots



(1)Use "mask" to program ROM

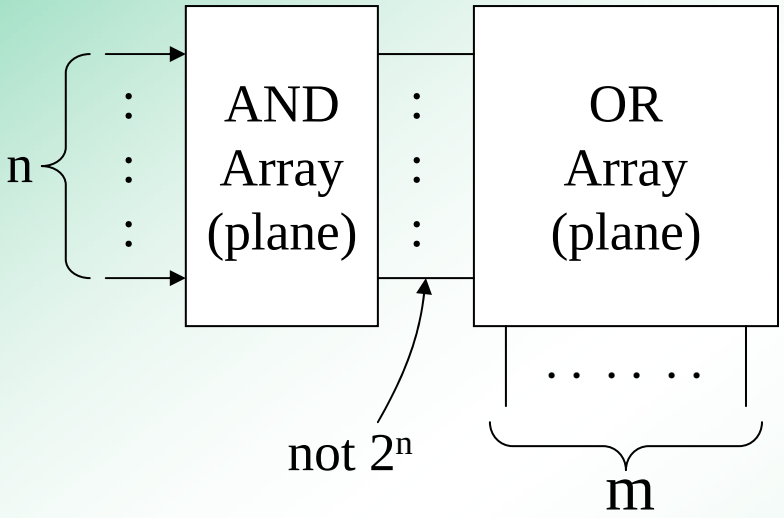
(2)EPROM electrically program UV light erase

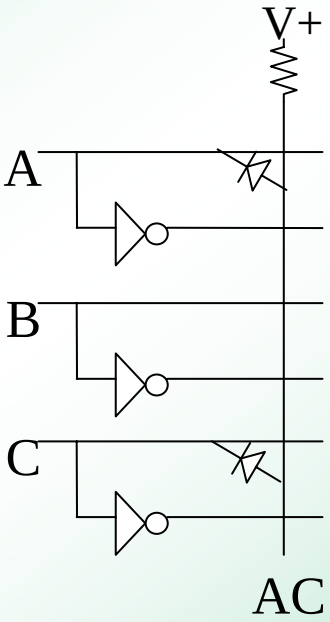
(3)EEPROM

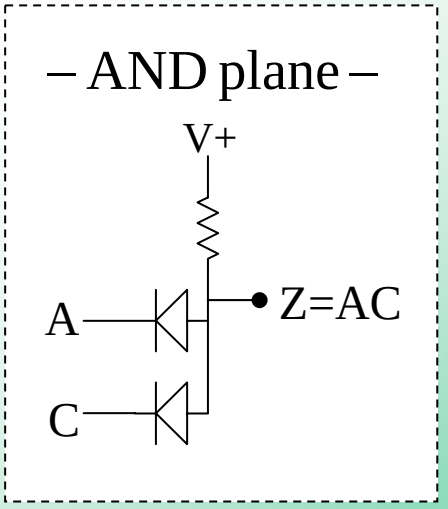
§ Programmable Logic Devices

Various kinds : PLA , PAL , EPLD , PEEL , GAL

※ PLAs : n × m realizes m functions with n inputs.







2 - level SOP implementation

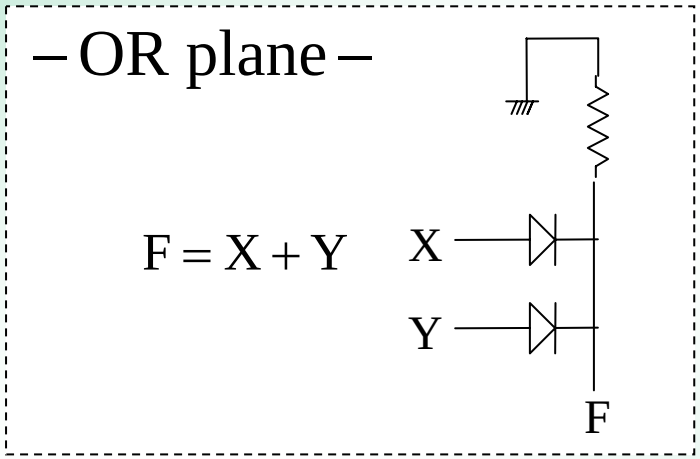
ROM directly implement truth table

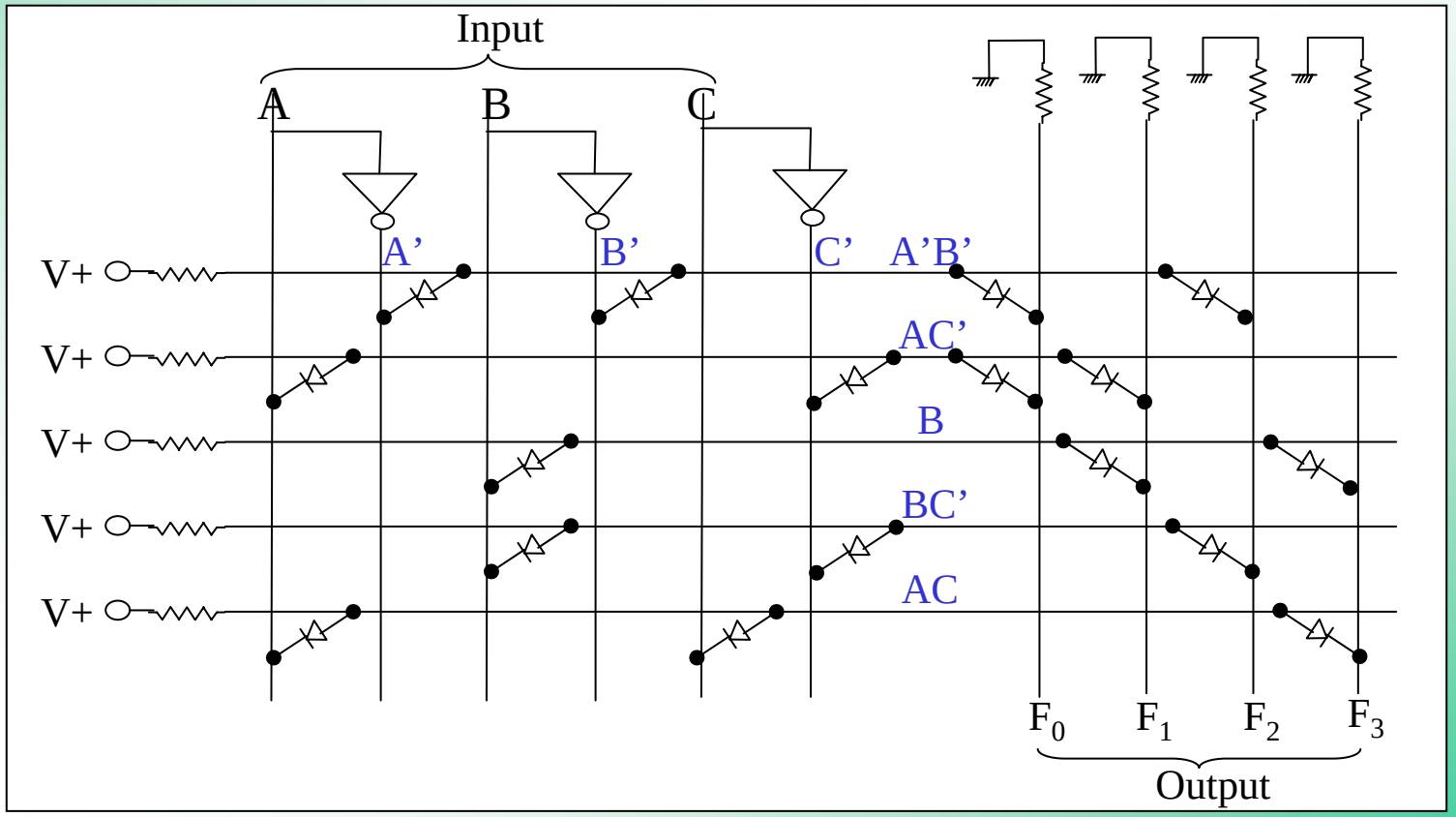
If any of A , C not activiated ,

$A = C = GND$

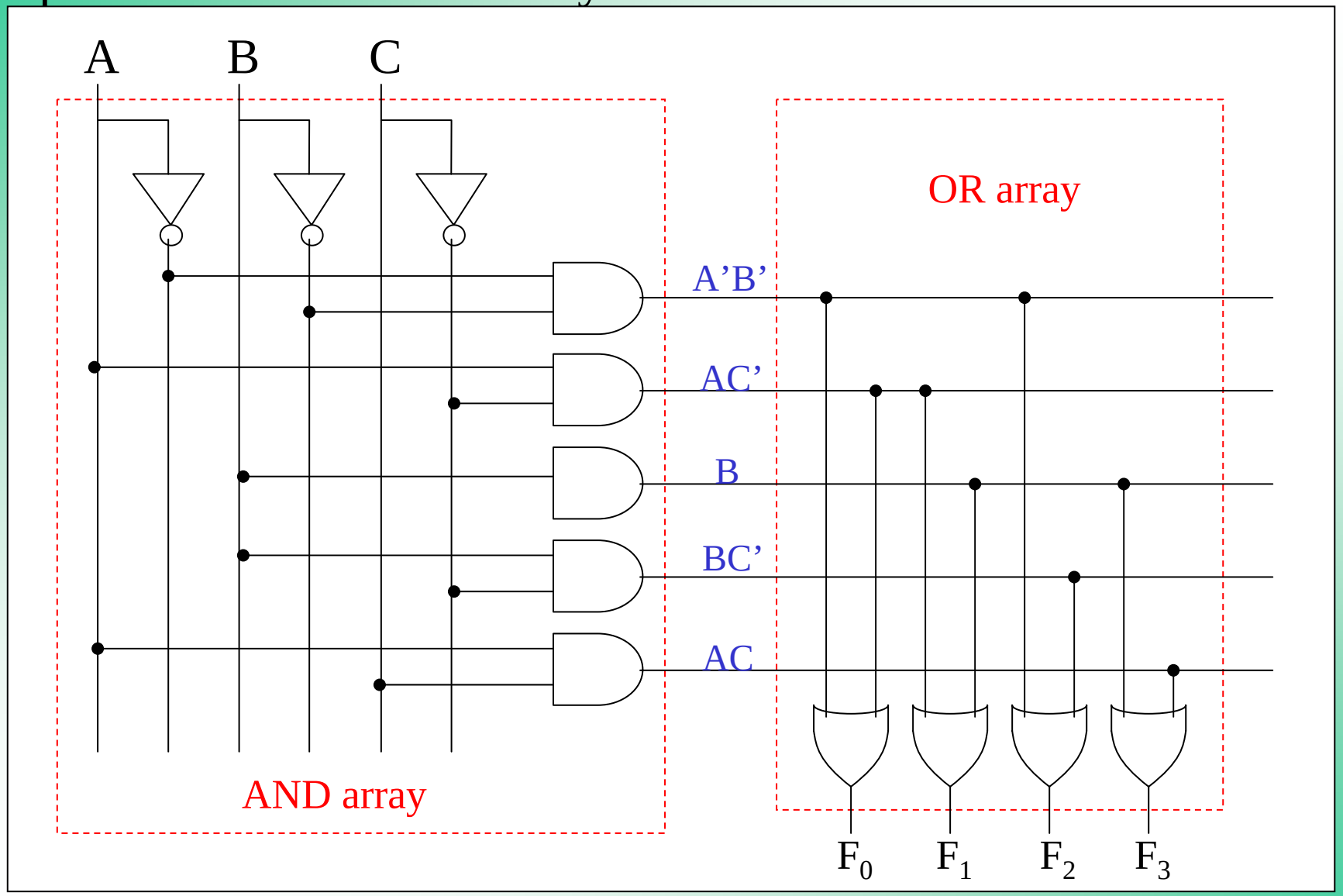
i.e. $A = 0$ or $C = 0$

	A	B	C	F_0	F_1	F_2	F_3
A'B'	0	0	–	1	0	1	0
AC'	1	–	0	1	1	0	0
B	–	1	–	0	1	0	1
BC'	–	1	0	0	0	1	0
AC	1	–	1	0	0	0	1





Equivalent AND - OR Array



Example

$$F_1 = \sum m(2,3,5,7,8,9,10,11,13,15)$$

$$F_2 = \sum m(2,3,5,6,7,10,11,14,15)$$

$$F_1 = \sum m(6,7,8,9,13,14,15)$$

Minimized multiple output expressions (using K - map)

$$F_1 = a'bd + abd + ab'c' + b'c$$

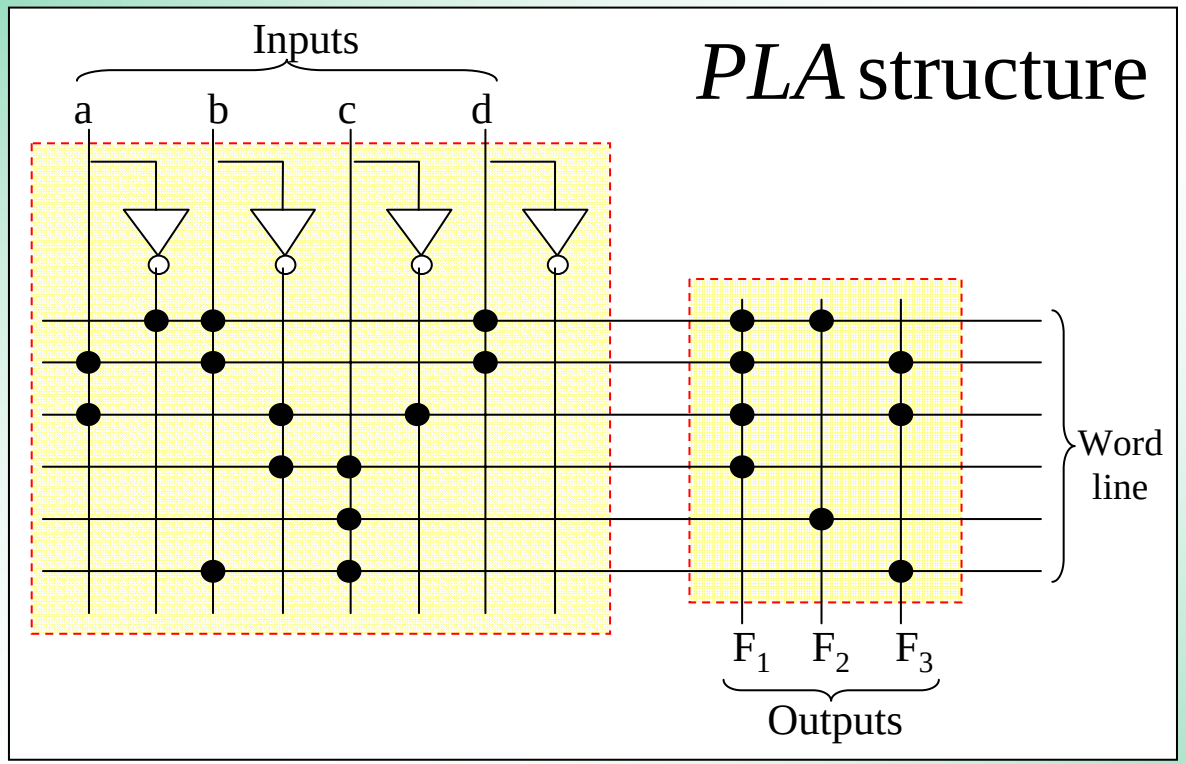
$$F_2 = c + a'bd$$

$$F_3 = bc + ab'c' + abd$$

PLA table

a	b	c	d	F_1	F_2	F_3
0	1	–	1	1	1	0
1	1	–	1	1	0	1
1	0	0	–	1	0	1
–	0	1	–	1	0	0
–	–	1	–	0	1	0
–	1	1	–	0	0	1

a	b	c	d	F_1	F_2	F_3
0	1	–	1	1	1	0
1	1	–	1	1	0	1
1	0	0	–	1	0	1
–	0	1	–	1	0	0
–	–	1	–	0	1	0
–	1	1	–	0	0	1



a b c d = 0 11 1 1,5,6 rows selected , $\Rightarrow F_1 = 1(\text{1st row}) + 0(\text{5th}) + 0(\text{6th}) = 1$

$F_2 = 1 + 1 + 0 = 1$

$F_3 = 0 + 0 + 1 = 1$

= 0 0 0 1 no rows are selected, => $F_1F_2F_3 = 000$

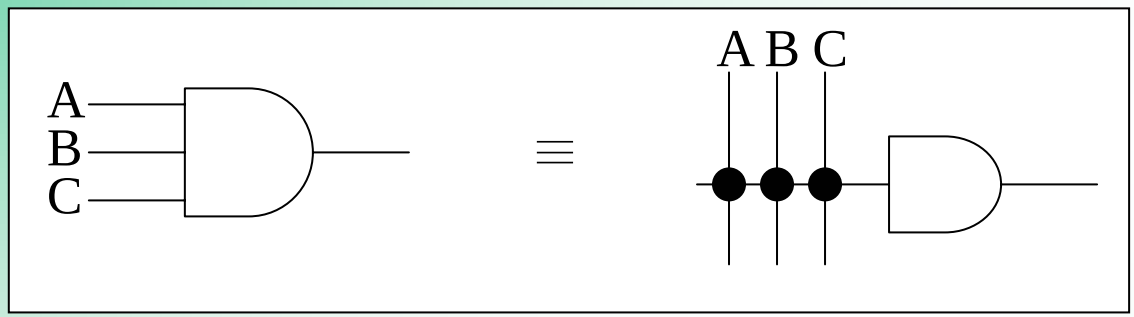
= 1 00 1 3rd rows selected, => $F_1F_2F_3 = 101$

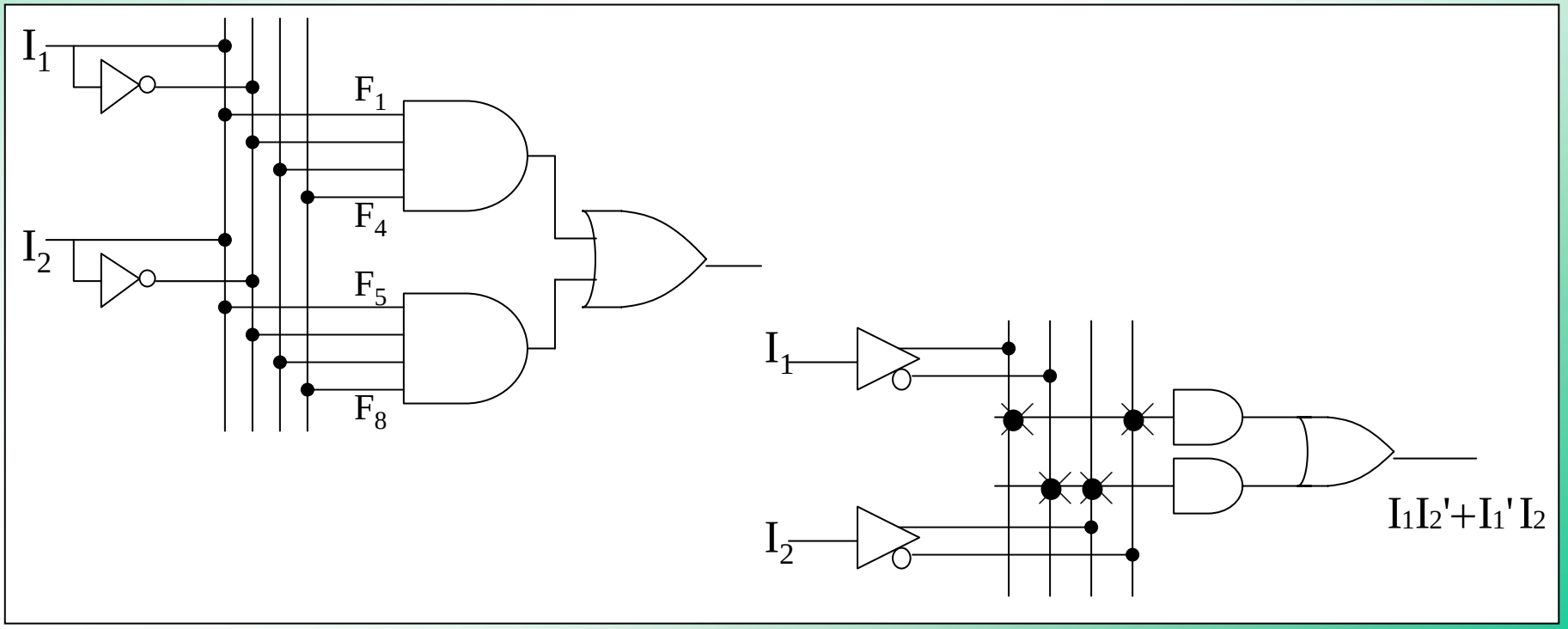
◎ Mask programmable PLA

◎ Field programmable PLA

※ PALs : n × m realizes m functions with n inputs.

A special case of PLA . (AND array programmble
OR array NOT programmble)



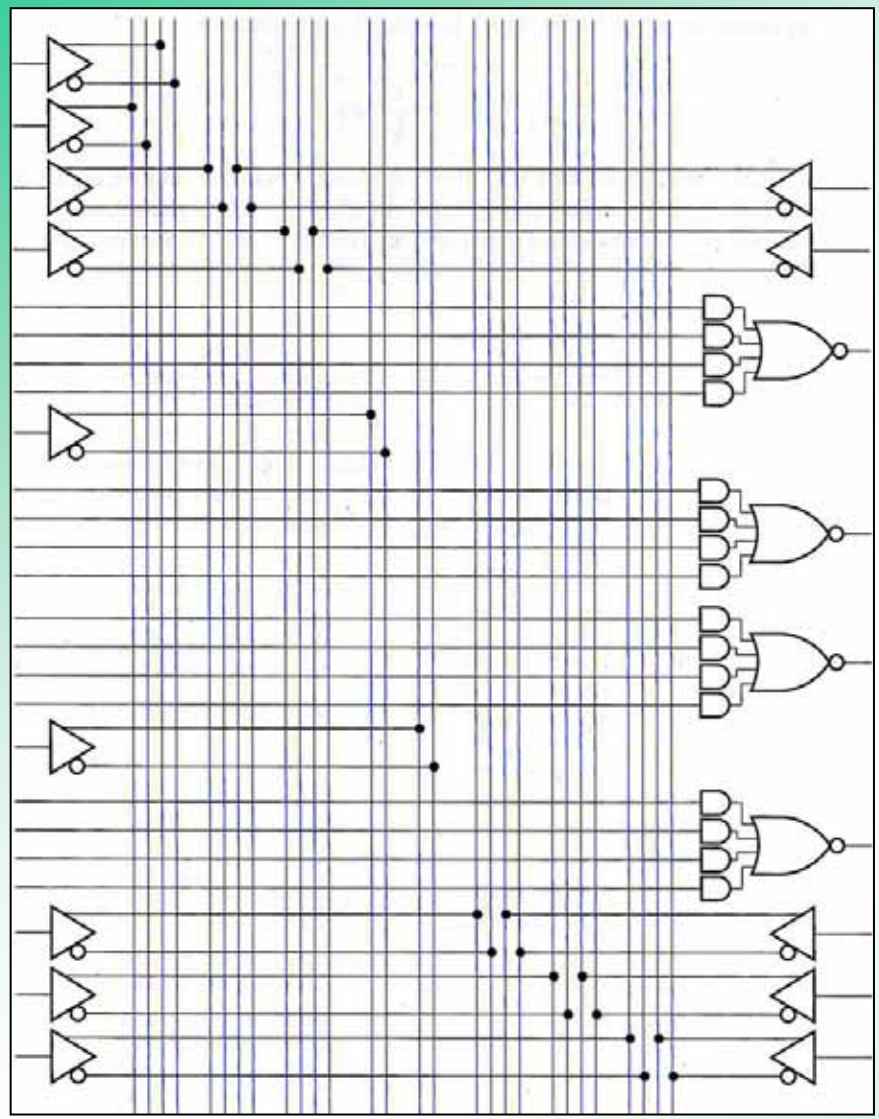


Some typical available PALs

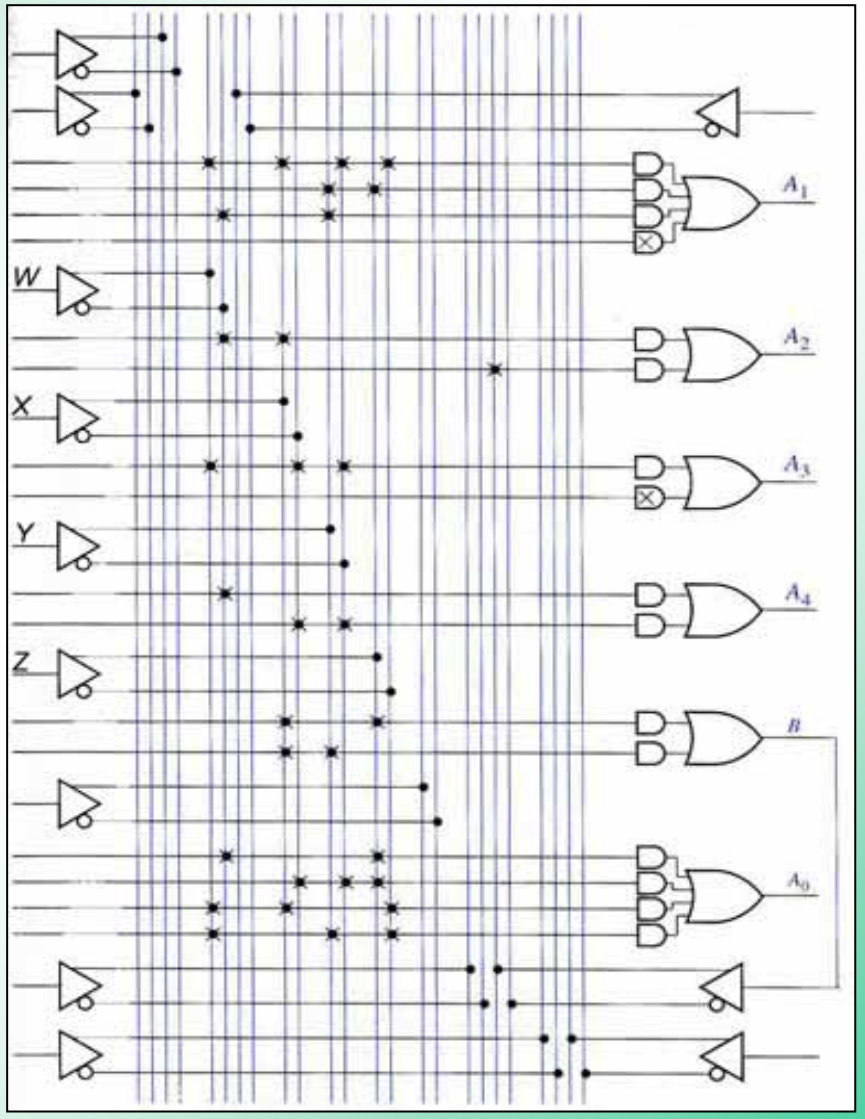
Type No.*	No. of Inputs	No.of Outputs	Gate Configuration	No. of Inputs per OR GATE
10H8	10	8	AND - OR	2
12H6	12	6	AND - OR	4,2,2,2,2,4
14H4	14	4	AND - OR	4
16H2	16	2	AND - OR	8
16C1	16	1	AND - OR/NOR	16
20C1	20	1	AND - OR/NOR	16
10L8	10	8	AND - NOR	2
12L6	12	6	AND - NOR	4,2,2,2,2,4
14L4	14	4	AND - NOR	4
16L2	16	2	AND - NOR	8
12L10	12	10	AND - NOR	2
14L8	14	8	AND - NOR	4,2,2,2,2,2,4
16L6	16	6	AND - NOR	4,4,2,2,2,4
18L4	18	4	AND - NOR	4
20L2	20	2	AND - NOR	8
16L8	16	8	AND - NOR	8
20L8	20	8	AND - NOR	8
20L10	20	10	AND - NOR	4

H	active H
L	active L
C	both active H and L

Internal Logic diagram



14L4



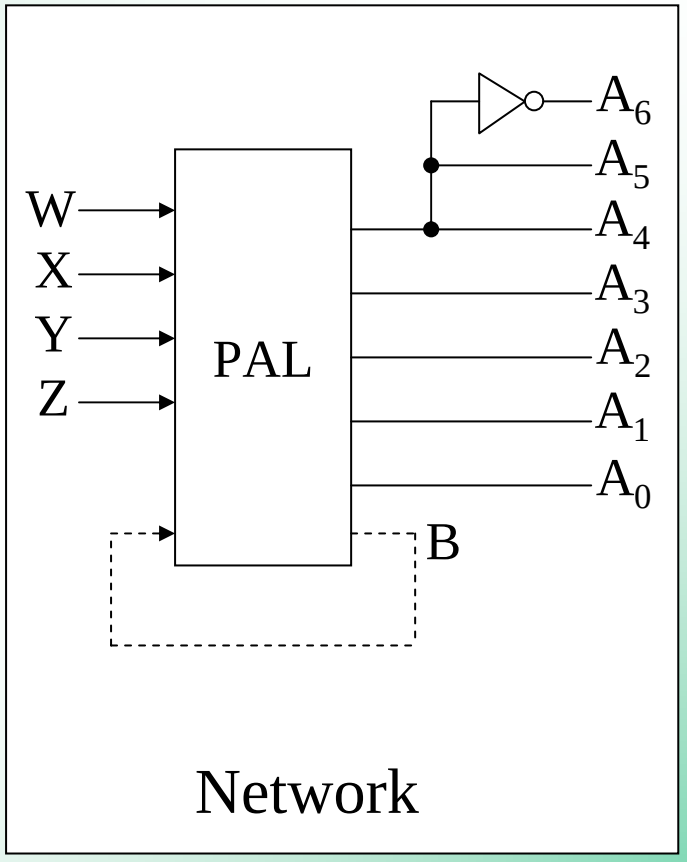
12H6

Number of inputs to an OR gate fixed and limited.

Example Hexadecimal Digit : Binary to ASCII Converter

Input				Hex	ASCII Code for Hex Digit						
W	X	Y	Z	Digit	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	1	1	0	0	0	1
0	0	1	0	2	0	1	1	0	0	1	0
0	0	1	1	3	0	1	1	0	0	1	1
0	1	0	0	4	0	1	1	0	1	0	0
0	1	0	1	5	0	1	1	0	1	0	1
0	1	1	0	6	0	1	1	0	1	1	0
0	1	1	1	7	0	1	1	0	1	1	1
1	0	0	0	8	0	1	1	1	0	0	0
1	0	0	1	9	0	1	1	1	0	0	1
1	0	1	0	A	1	0	0	0	0	0	1
1	0	1	1	B	1	0	0	0	0	1	0
1	1	0	0	C	1	0	0	0	0	1	1
1	1	0	1	D	1	0	0	0	1	0	0
1	1	1	0	E	1	0	0	0	1	0	1
1	0	1	1	F	1	0	0	0	1	1	0

Truth table



Network

$A_4 = A_5 \quad A_6 = \overline{A_5}$

$\therefore$ 4 input variables
5 output functions
$\Rightarrow$ 12H6 or 12L6

Use K - map method :

$\Rightarrow A_4 = W'+X'Y' \qquad A_3 = WX'Y' \qquad A_2 = W'X + XZ + XY$

$A_4' = WX + WY \quad A_3' = \underline{\underline{W'}} + \underline{\underline{X}} + \underline{\underline{Y}} \quad A_2' = X'+WY'Z'$

$A_1 = WXY'Z'+YZ+W'Y \qquad A_0 = \underline{\underline{W'Z}} + \underline{\underline{X'Y'Z}} + \underline{\underline{WXZ'}} + \underline{\underline{WYZ'}}$

$A_1' = W'Y'+Y'Z + X'Y'+WYZ' \quad A_0' = W'Z'+WYZ+WX'Z+X'Y'Z$

A_0 : $\left.\begin{array}{l} 1\times 4 \text{ inputs AND to OR} \\ 2\times 3 \text{ inputs AND to OR} \end{array}\right\}$ 12H6 not available

但 $A_2 = W'X + (XZ + XY) = W'X + B$

B connected back to one input

$\therefore$ <u>12H6結果可用</u>

CAD programs are available.

PLD Summary

	AND Array	OR Array
PLA	Programmable	Programmable
PAL	Programmable	Fixed
ROM	Fixed	Programmable
Not Programmable	Fixed	Fixed

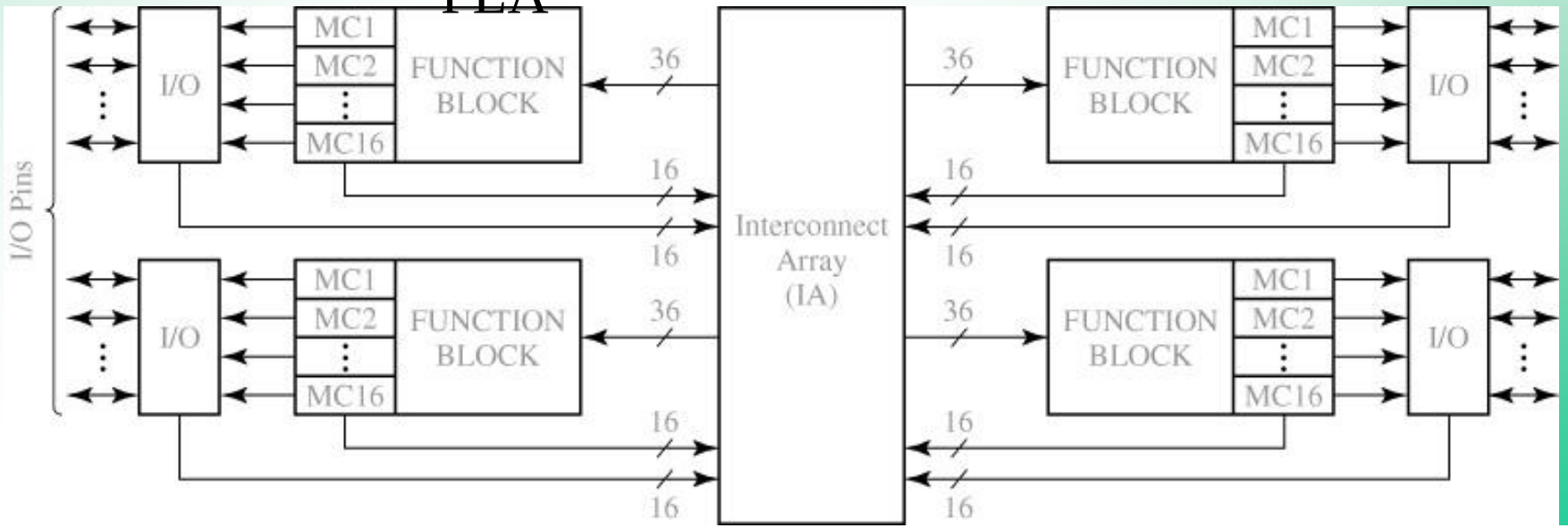
§ Complex Programmable Logic Devices (CPLD)

Integrate and interconnect many PALs and PLAs on a single chip

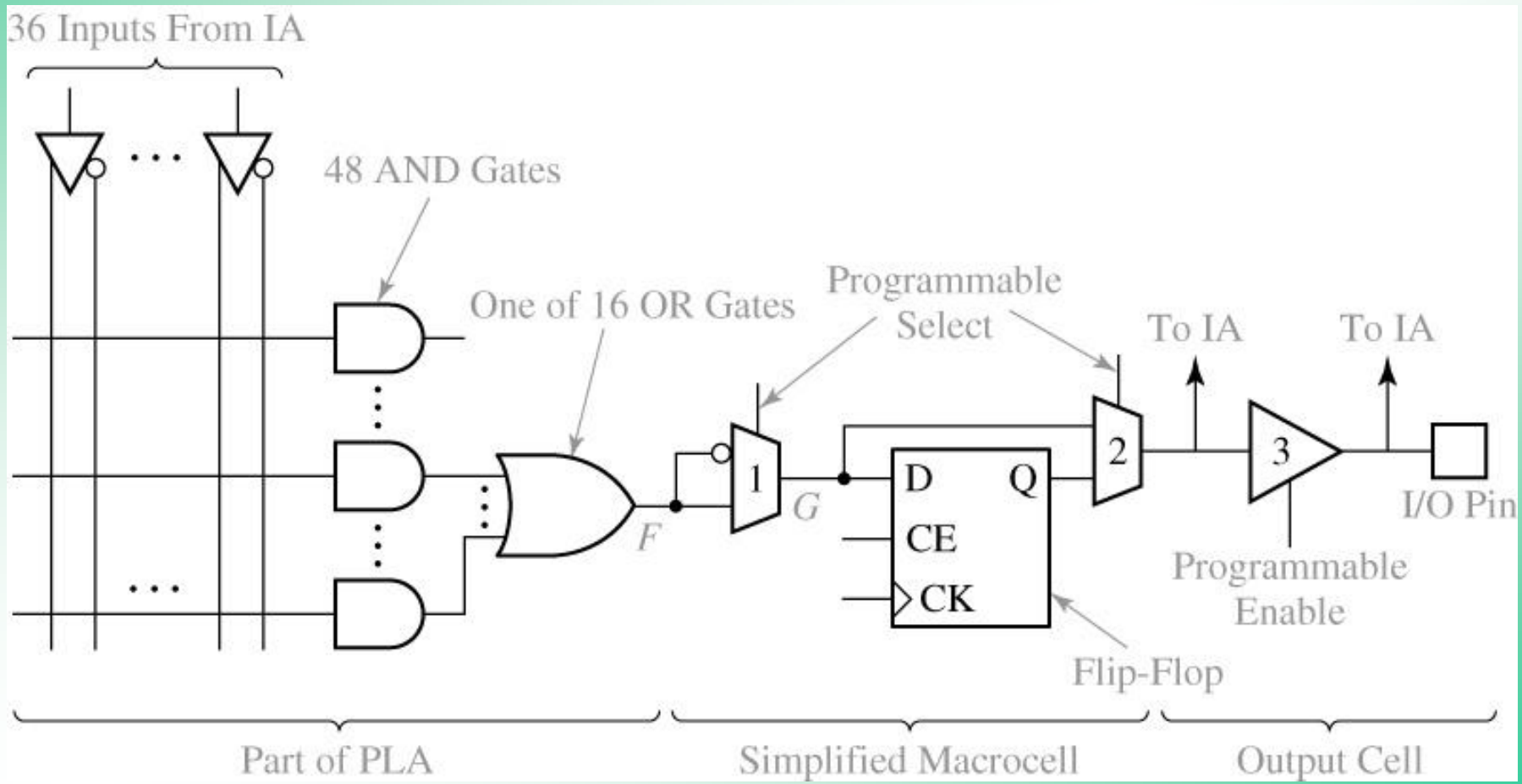
- *PLA to form combinational logic*
- *Macro Cell*
  - *Register for Sequential Logic*
  - *Pass Through for Combinational Logi*
- *I/O Cell*
  - *Three State Buffer for Bidirectional I/C*
  - *Input Latch to synchronize inputs*
- *Floating Gate Programmable*

Xilinx XCR3054XL CPLD

PLA



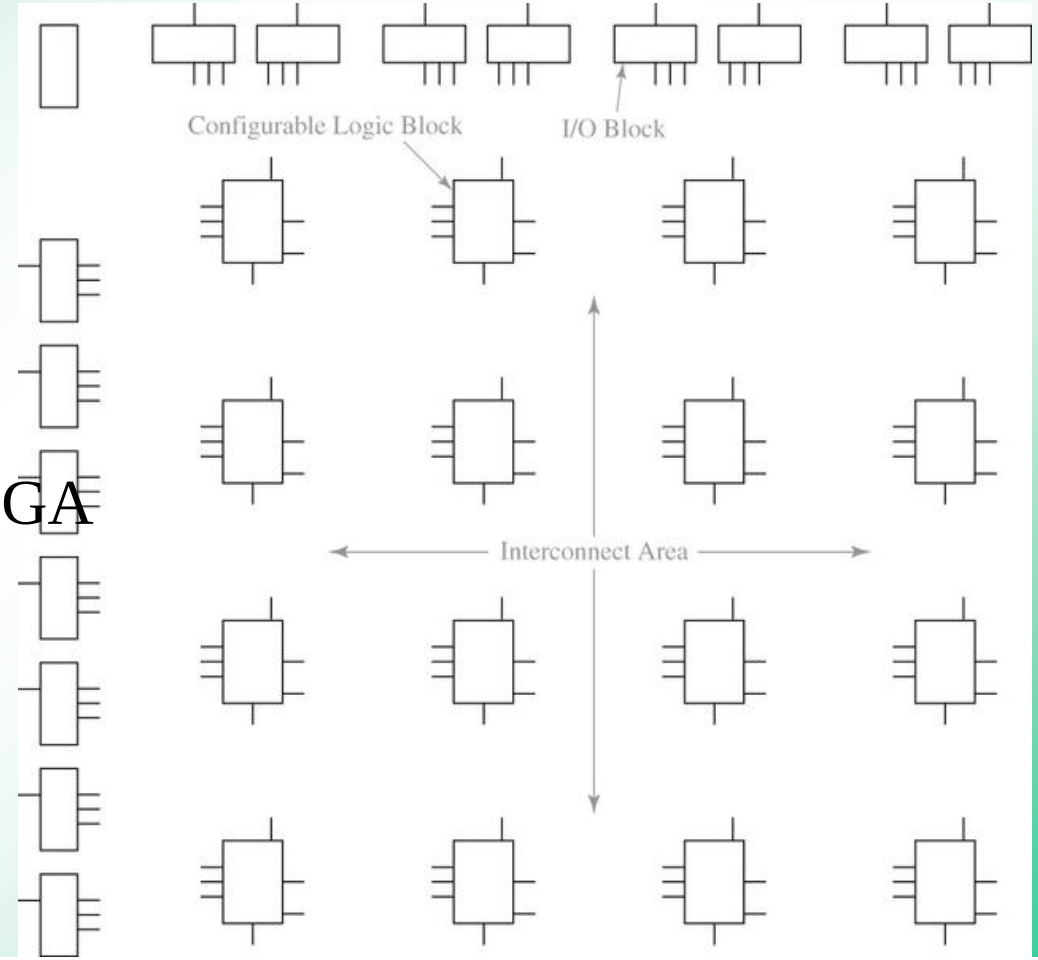
CPLD Function block an macrocell



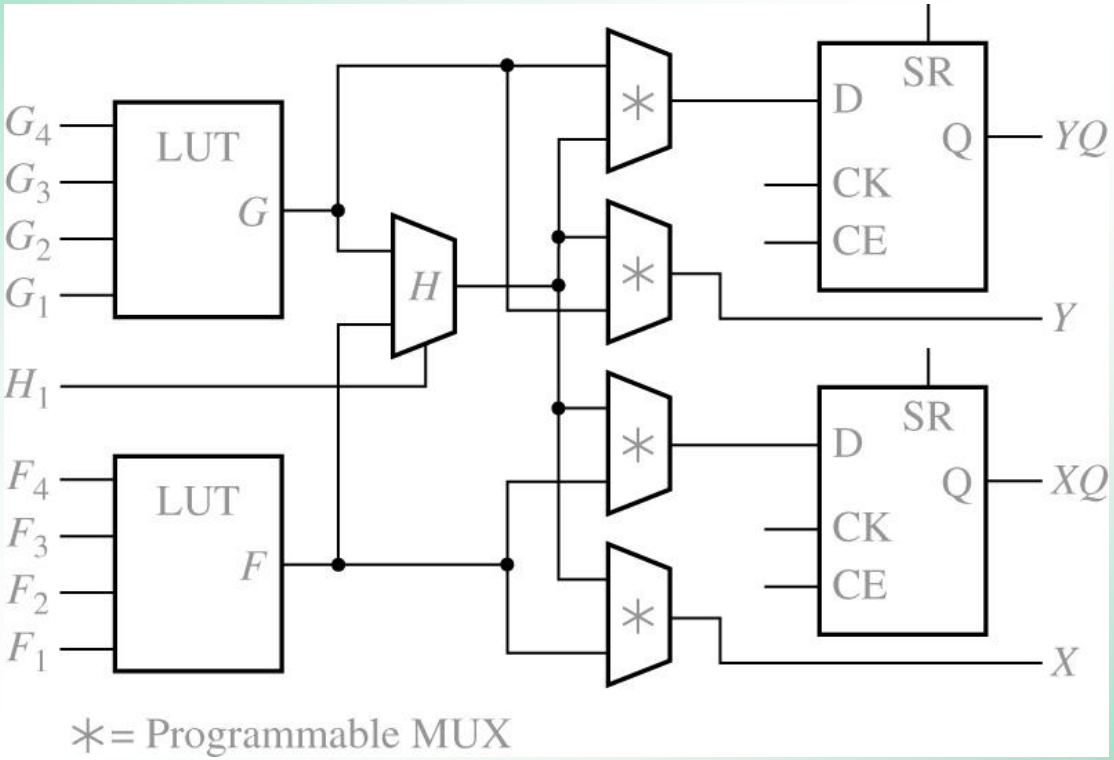
§ Field Programmable Gate Array (FPGA)

An IC that contains an array of identical logic cells with programmable interconnections

Layout of a typical FPGA



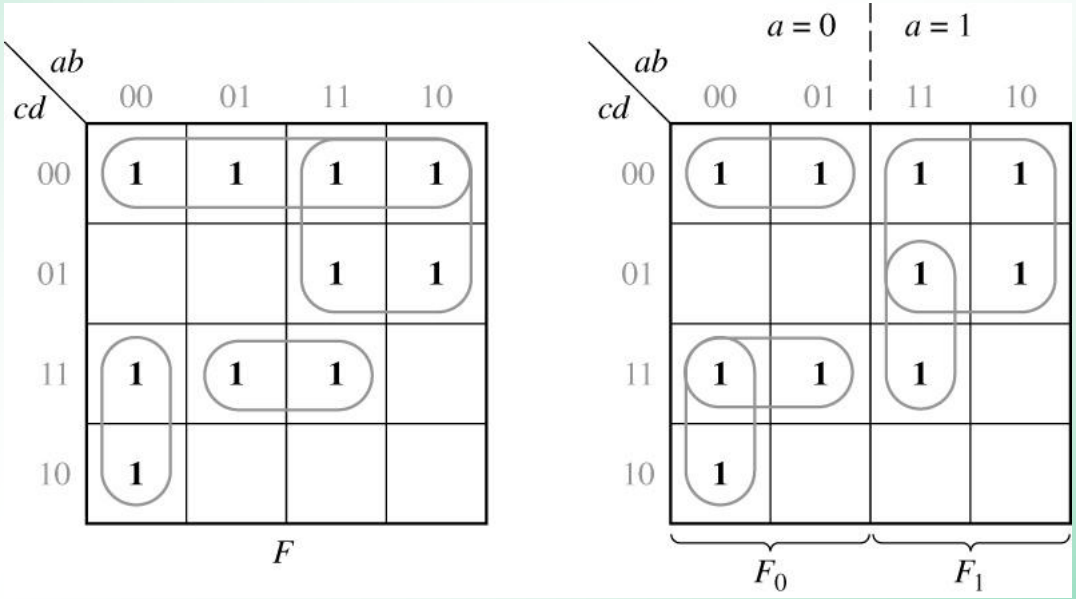
CLB: configurable logic block



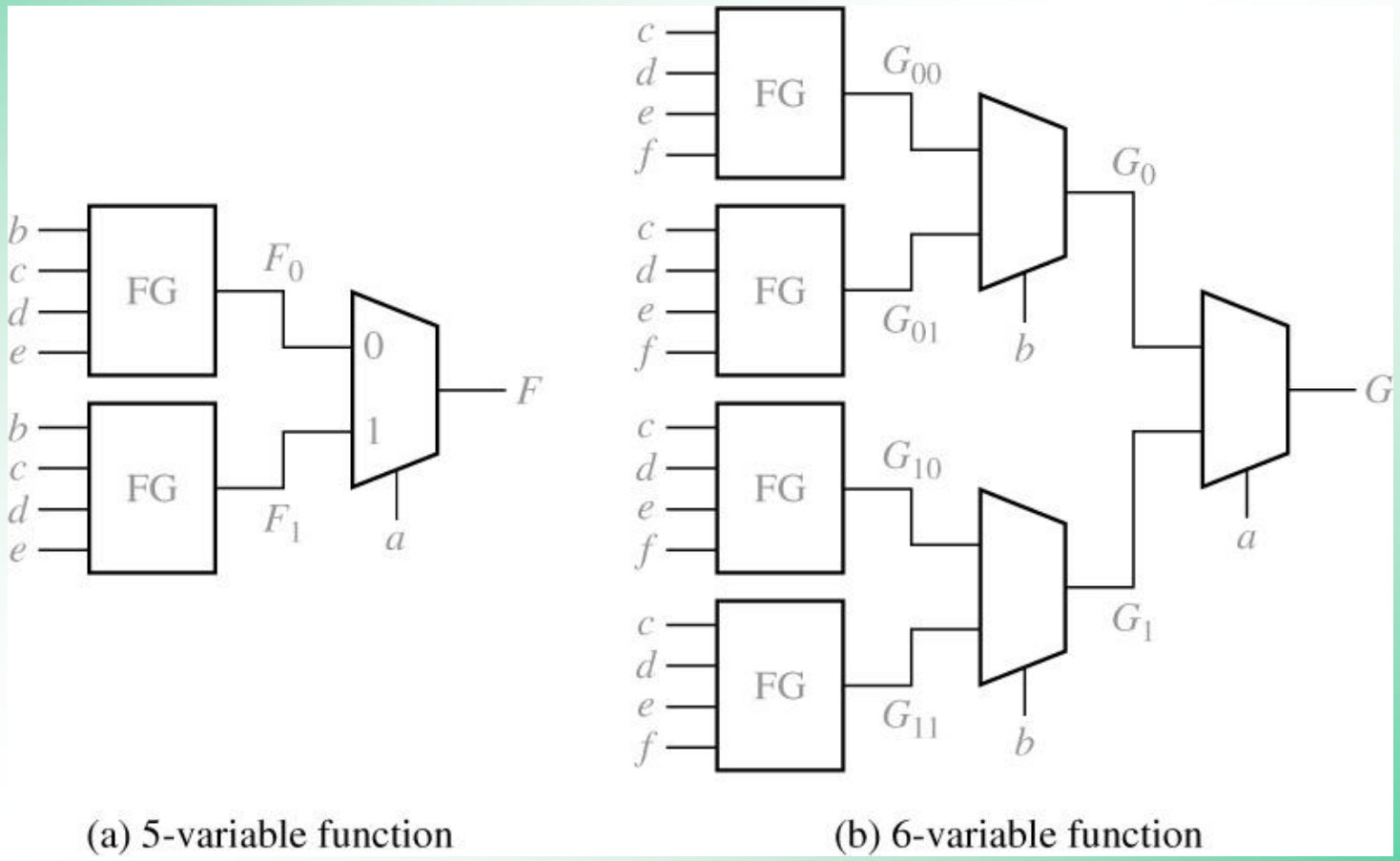
Decomposition of switching functions

Shannon's expansion theorem

$F(x_1, x_2, \ldots x_{i-1}, x_i, x_{i+1}, \ldots)$
$= x_i' f_0 + x_i f_1$



Realization of 5- and 6-variables functions with function generators



(a) 5-variable function (b) 6-variable function

Homework of Chap.9

12.(a)
15.
21.
23.
26.
30.