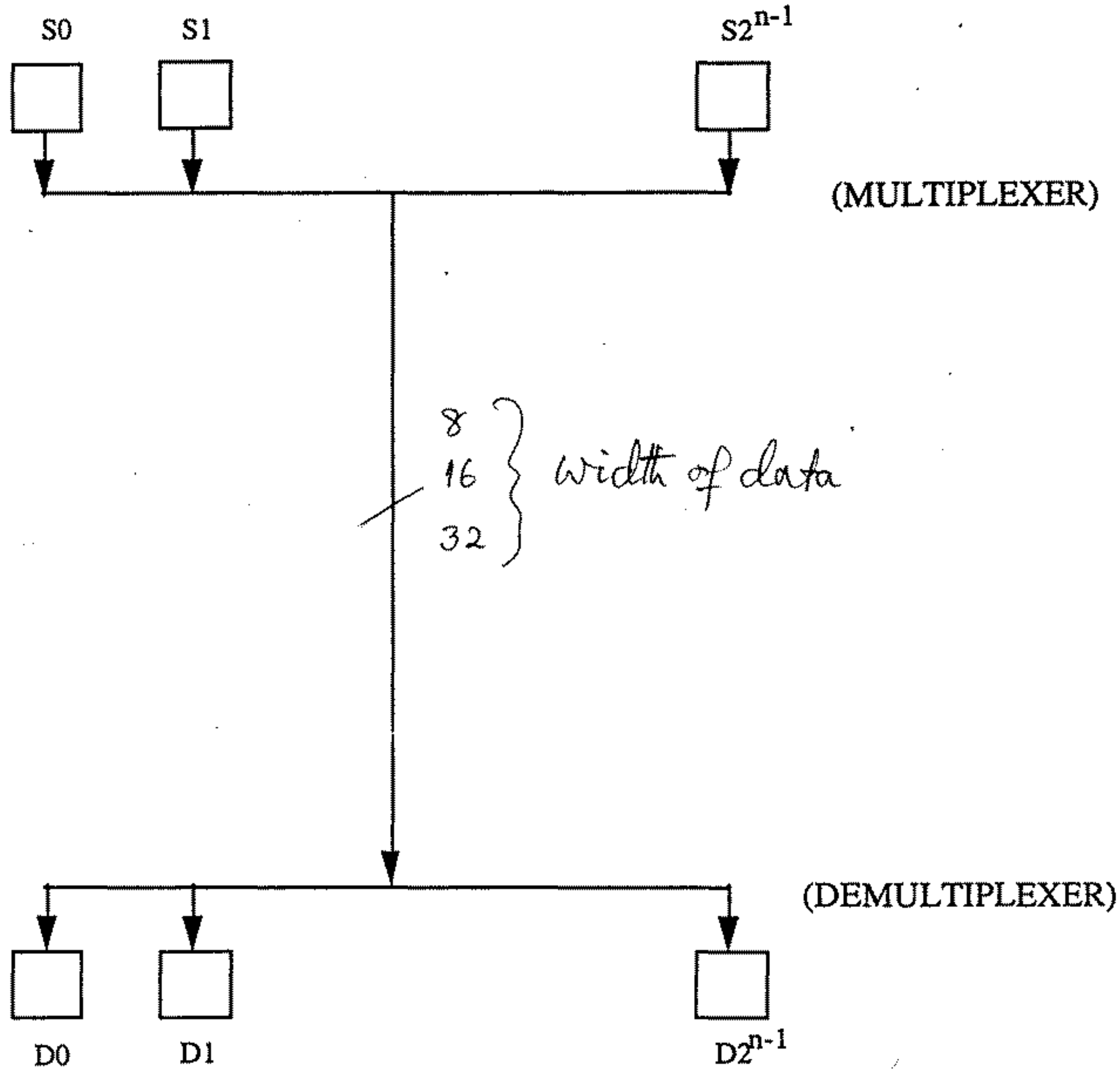
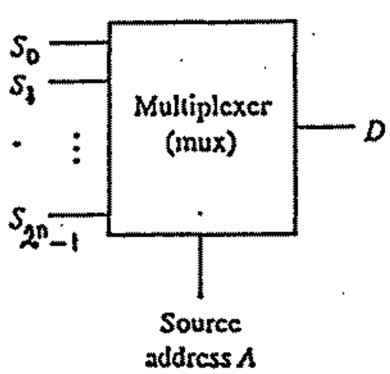
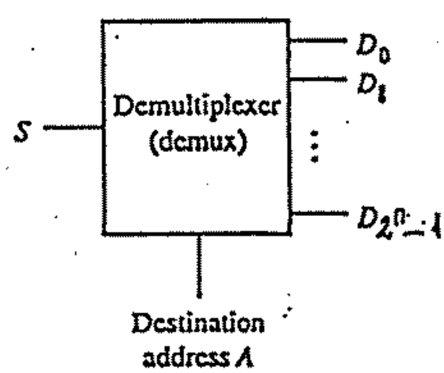
DATA TRANSFER



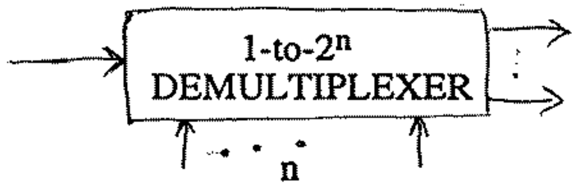




DEMULTIPLEXER (DATA DISTRIBUTOR)

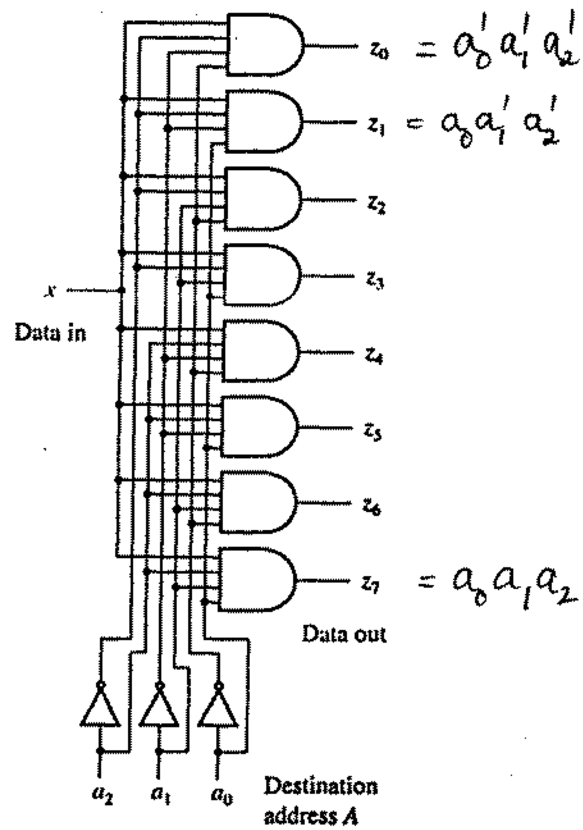
DIRECTS THE INPUT TO ONE OUT OF 2^n OUTPUTS

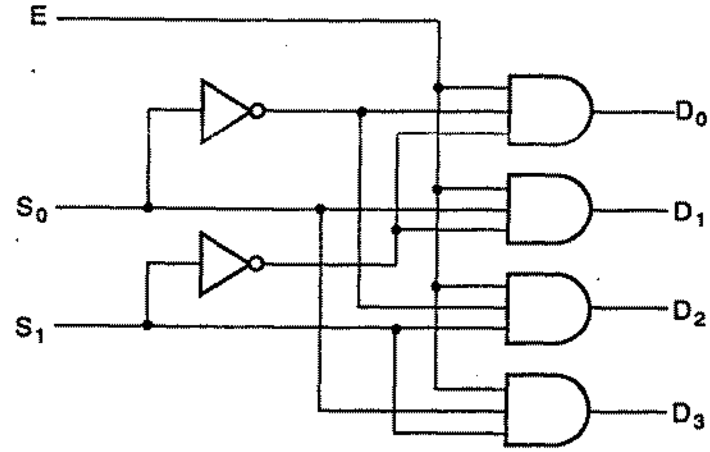
INPUT



2^n OUTPUTS

DESTINATION SELECTION LINES
(WHOSE BIT COMBINATION DETERMINES WHICH OUTPUT IS SELECTED)





a_2	a_1	a_0	z
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

S_1	S_0	D
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

IN GENERAL:
1-to-2^n DEMULTIPLEXER = n-to-2^n DECODER

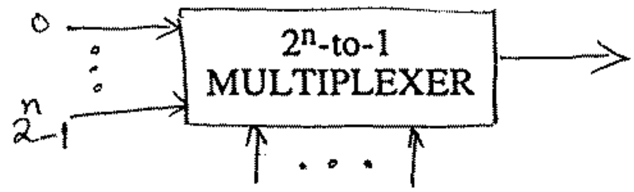
with enable input

MULTIPLEXER (DATA SELECTOR)

Section 3.7

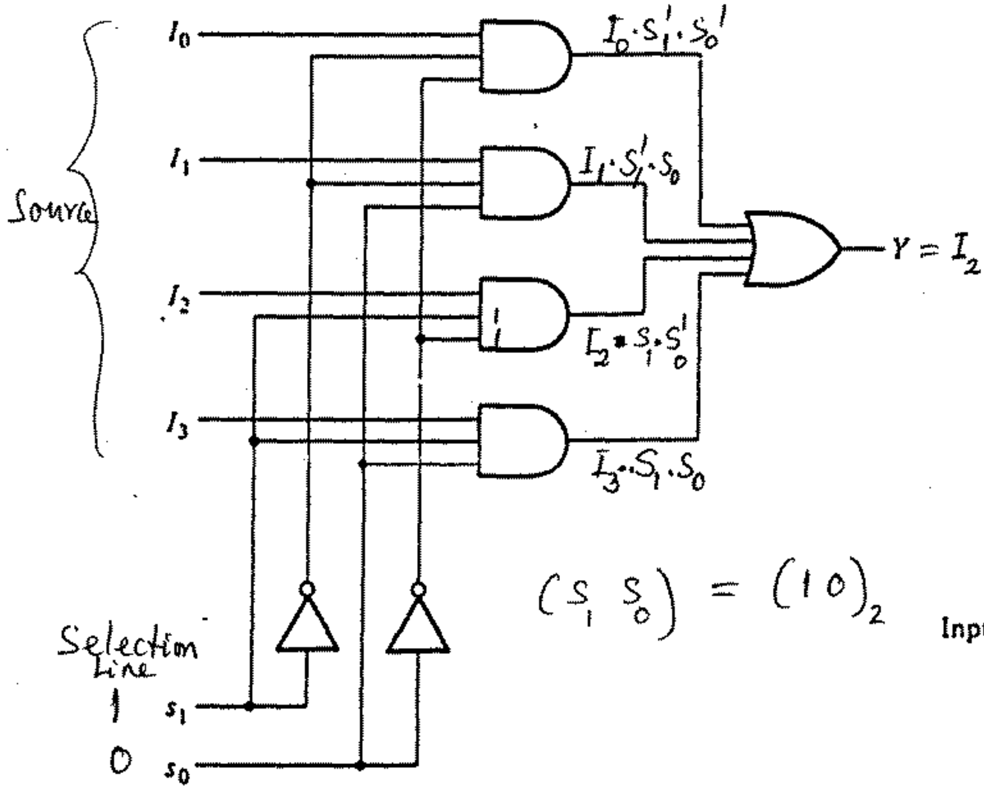
SELECTS ONE OUT OF 2^n INPUTS AND DIRECTS IT TO THE OUTPUT

2^n INPUTS



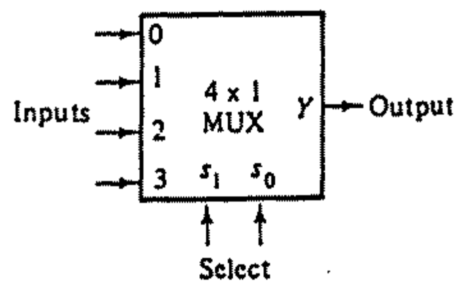
OUTPUT

n
SOURCE SELECTION LINES
(WHOSE BIT COMBINATION DETERMINES WHICH INPUT IS SELECTED)



s_1	s_0	$Y = f(s_0, s_1)$
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table



A 4-to-1-line multiplexer (a) Logic diagram (c) Block diagram

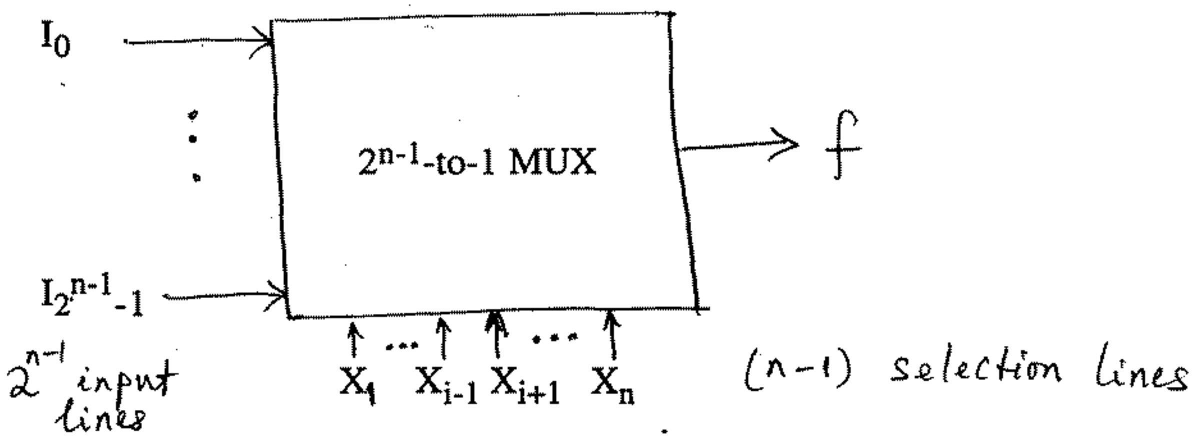
IN GENERAL:
2^n-to-1 MULTIPLEXER = n-to-2^n DECODER
+
2^n INPUT LINES (ONE TO EACH AND GATE)
+
OR GATE (COLLECTING OUTPUTS OF AND GATES)

∴ Any boolean function $f(x_1, ..., x_n)$ can be implemented by a 2^{n-1}-to-1 MULTIPLEXER

GIVEN A BOOLEAN FUNCTION $f(x_1, ..., x_n)$
IMPLEMENT IT USING A 2^{n-1}-to-1 MULTIPLEXER

PROCEDURE:

1. GIVEN $f(x_1, ..., x_n)$ in CSOP $= \sum m(\cdots)$



(NOTE THAT X_i IS NOT USED AS A SELECTION LINE)

2. FIND THE VALUE OF I_j , for $j = 0, 1, 2, ..., 2^{n-1}-1$
BY USING THE TABLE WHOSE FORMAT SHOWN BELOW

	I_0 I_1 I_2 ....$I_{2^{n-1}-1}$
X'_i	minterms where X_i is primed
X_i	minterms where X_i is unprimed
⇒	VALUES OF I_j

A) PUT MINTERMS IN THE TABLE

B) CIRCLE THOSE MINTERMS IN CSOP

C) INSPECT COLUMN I_j , $j = 0, 1, 2, ..., 2^{n-1}-1$

if both minterms are NOT circled; $I_j \leftarrow 0$

if both minterms are circled; $I_j \leftarrow 1$

if top minterm is circled only; $I_j \leftarrow X'_i$

if bottom minterm is circled only; $I_j \leftarrow X_i$

GIVEN f(A, B, C) in CSOP,
PUT MINTERMS IN THE TABLE
DEPENDING ON THE VARIABLE NOT USED AS A SELECTION LINE.

f(A, B, C)		A	B	C	
	m0	0	0	0	A′B′C′
	m1	0	0	1	A′B′C
	m2	0	1	0	A′B C′
	m3	0	1	1	A′B C
	m4	1	0	0	A B′C′
	m5	1	0	1	A B′C
	m6	1	1	0	A B C′
	m7	1	1	1	A B C

$2^2 = 4$ address can be located.

VARIABLE NOT USED AS A SELECTION LINE IS:

1) A

	I_0	I_1	I_2	I_3
A′	0	1	2	3
A	4	5	6	7

2) B

	I_0	I_1	I_2	I_3
B′	0	1	4	5
B	2	3	6	7

3) C

	I_0	I_1	I_2	I_3
C′	0	2	4	6
C	1	3	5	7

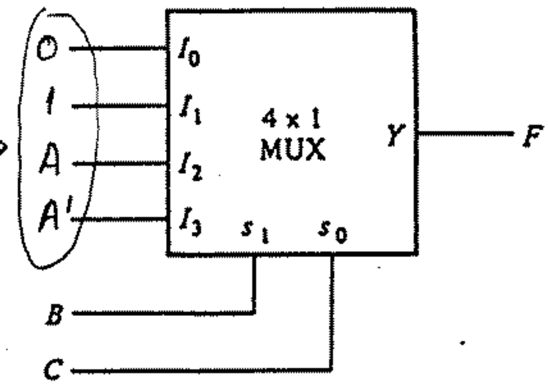
Given F, find a multiplexer implementation.

Minterm	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

(b) Truth table

	I_0	I_1	I_2	I_3
A'	0	①	2	③
A	4	⑤	⑥	7
	0	1	A	A'

(c) Implementation table

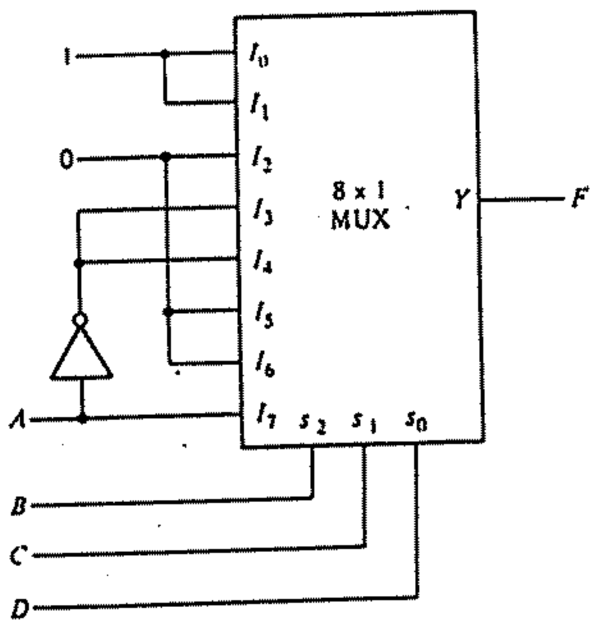


(a) Multiplexer implementation

Implementing $F(A, B, C) = \sum (1, 3, 5, 6)$ with a multiplexer

---

Given a multiplexer implementation, find F.

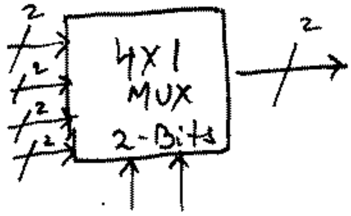


⇓

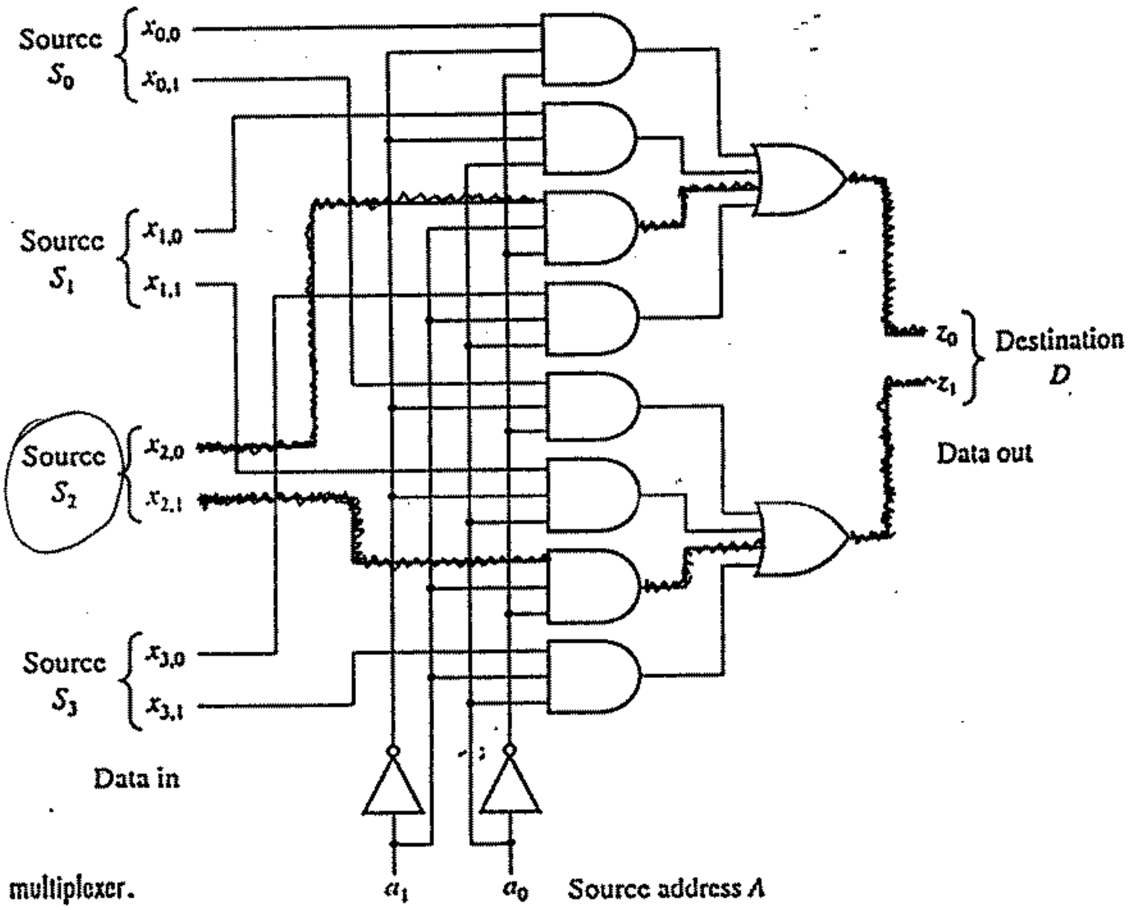
	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	⓪	①	2	③	④	5	6	7
A	⑧	⑨	10	11	12	13	14	⑮
	1	1	0	A'	A'	0	0	A

Implementing $F(A, B, C, D) = \sum m(0, 1, 3, 4, 8, 9, 15)$

Must each source be 1 bit? No!



$$z_0 = \bar{a}_1\bar{a}_0 x_{0,0} + \bar{a}_1 a_0 x_{1,0} + a_1\bar{a}_0 x_{2,0} + a_1 a_0 x_{3,0}$$
$$z_1 = \bar{a}_1\bar{a}_0 x_{0,1} + \bar{a}_1 a_0 x_{1,1} + a_1\bar{a}_0 x_{2,1} + a_1 a_0 x_{3,1}$$

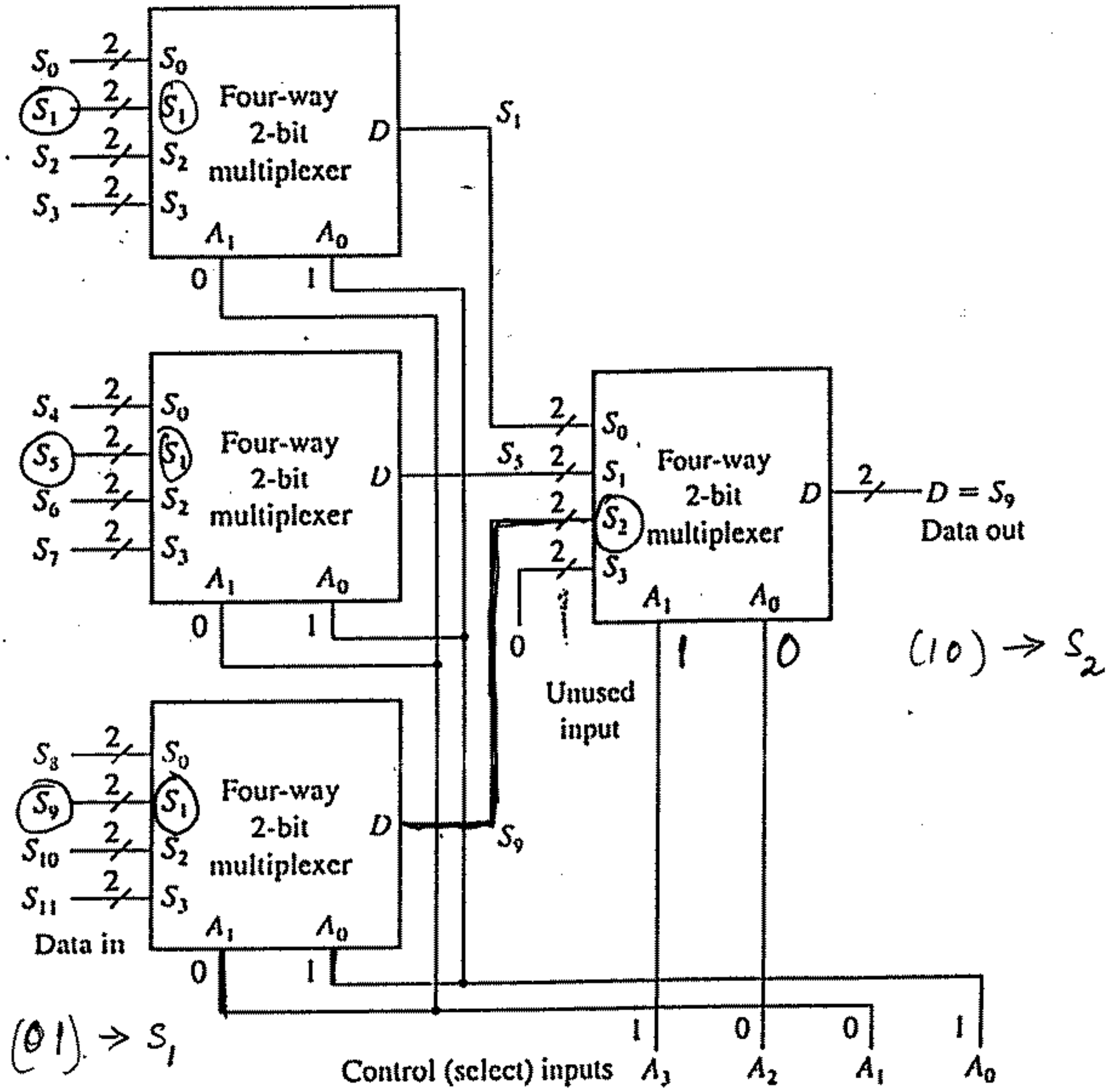


A four-way, 2-bit multiplexer.

4x1

1 0

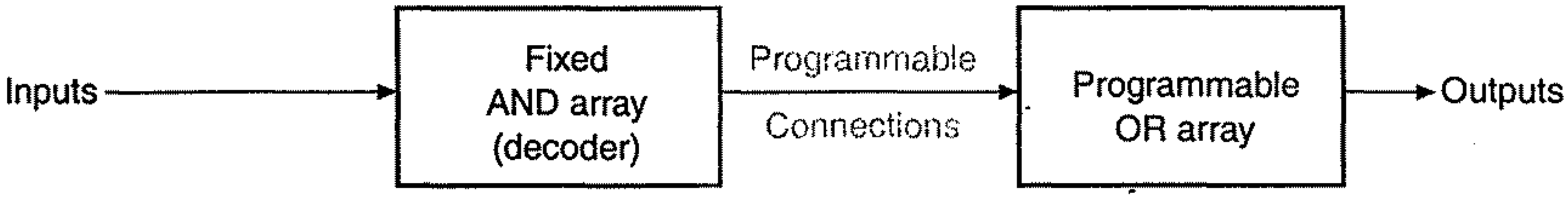
Example: 1 out of 12 is selected by an address of 4 bits. 64



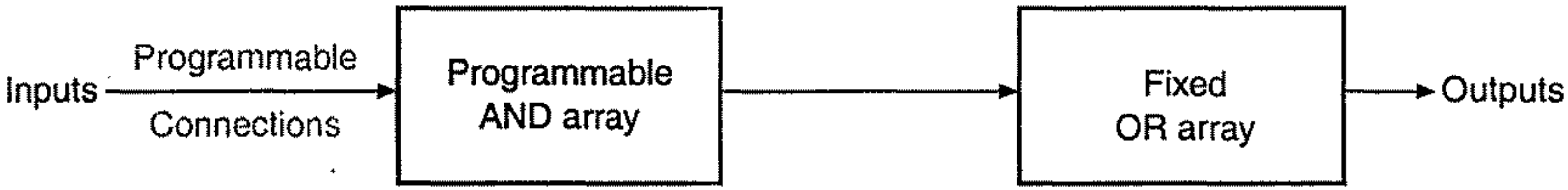
A 12-way multiplexer tree constructed from four-way multiplexers.

Basic Configuration of Three PLDs

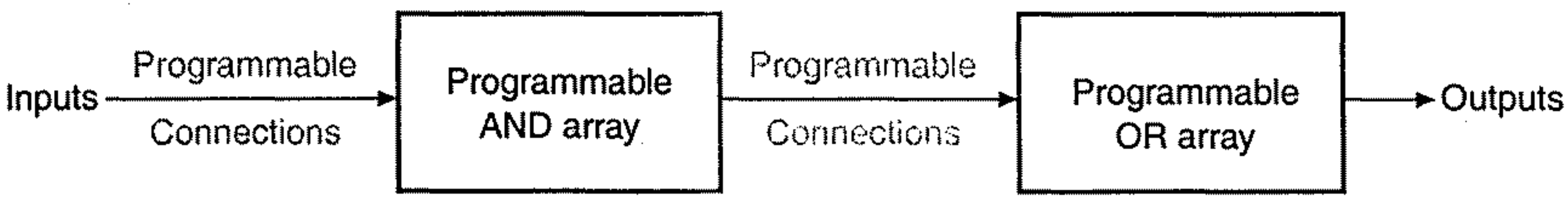
Programmable Logic Devices



(a) Programmable read-only memory (PROM)

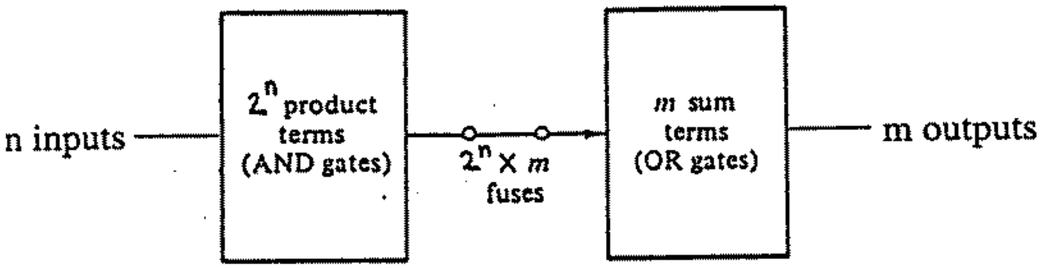


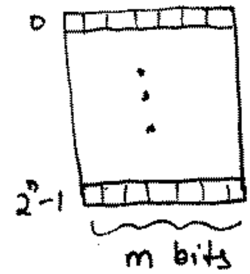
(b) Programmable array logic (PAL) device



(c) Programmable logic array (PLA) device

ROM (READ-ONLY MEMORY)





2^n x m ROM

IMPLEMENTING AN n-INPUT, m-OUTPUT COMBINATIONAL CIRCUIT

=

A 2^n x m ROM and one level of PROGRAMMING

e.g., Given the following two functions in CSOP form $n = 2$, $m = 2$

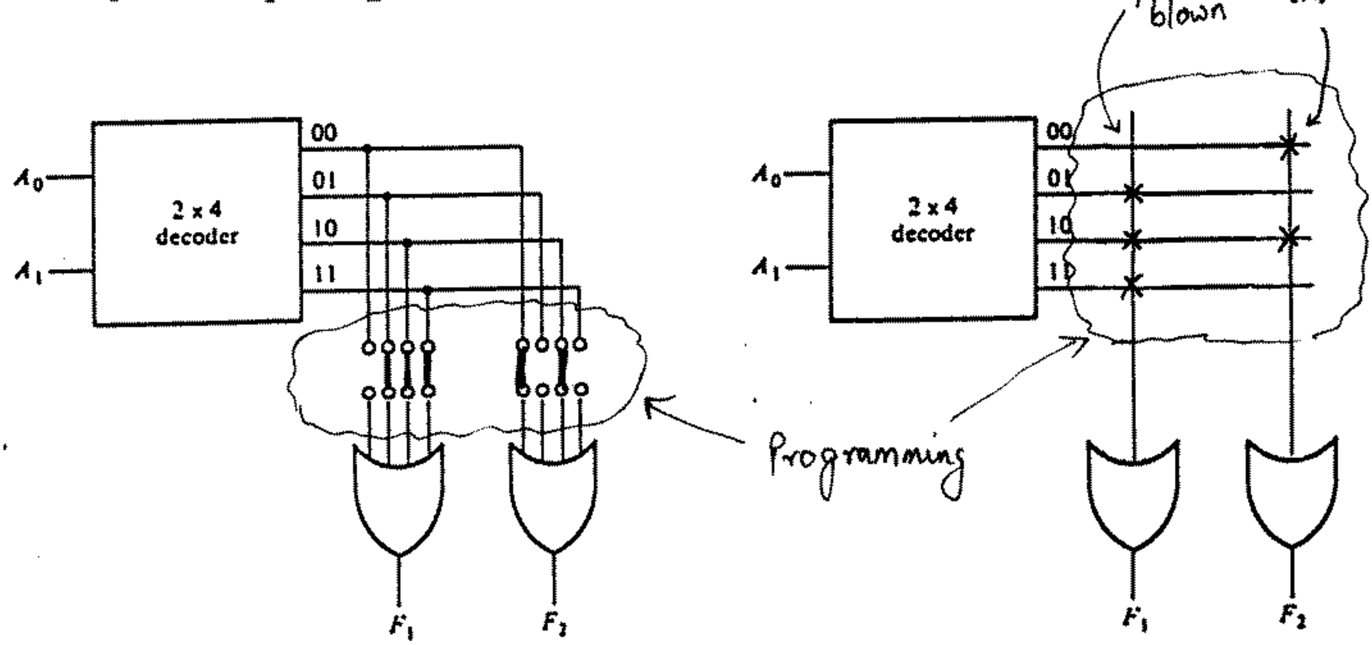
$f_1(A_1, A_0) = \sum m(1, 2, 3)$

$f_2(A_1, A_0) = \sum m(0, 2)$

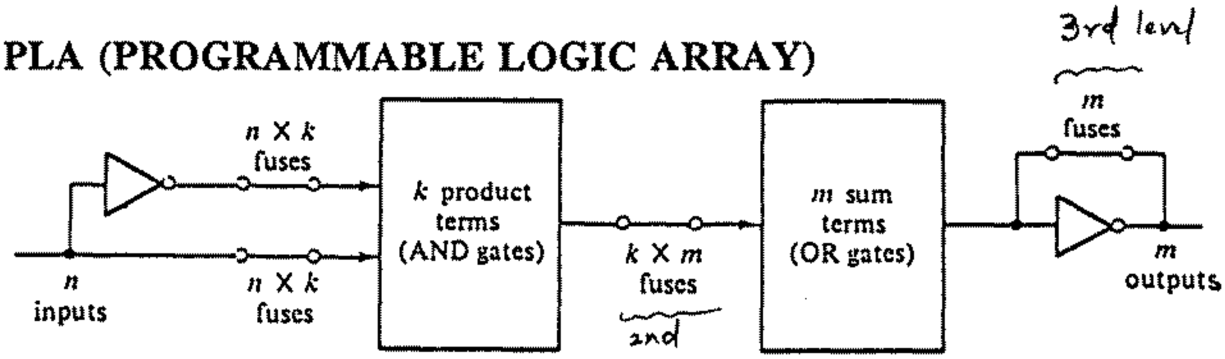
a) Obtain the 2^2 x 2 ROM Truth Table for f_1 and f_2

A_1	A_0	F_1	F_2
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	0

b) Implement f_1 and f_2 by using a 2^2 x 2 ROM and programming



PLA (PROGRAMMABLE LOGIC ARRAY)



n x k x m PLA

IMPLEMENTING AN n-INPUT, m-OUTPUT COMBINATIONAL CIRCUIT
=
An n x k x m PLA and three levels of PROGRAMMING

e.g., Given the following two functions in CSOP form

$f_1(A, B, C) = \Sigma m(4, 5, 7)$
$f_2(A, B, C) = \Sigma m(0, 1, 2, 4, 6)$

a) Obtain the minimal SOP form of f_1, f_1', f_2, and f_2'

B (over columns 3–4); A (beside row 2); C (under columns 2–3)

1	1	1	

$f_1 = AB' + AC$

1	1		1
1			1

$f_2 = C' + A'B'$

0	0	0	0
			0

$f_1' = A' + BC'$

		0	
	0	0	

$f_2' = AC + BC$

b) Determine the minimal number of product terms that will be sufficient to implement (f_1 or f_1') and (f_2 or f_2')

f_1 & f_2' ⇒ 3 product terms

c) Construct the PLA Program Table for f_1 and f_2

		A	B	C	f1	f2
1	AB'	1	0	–	1	–
2	AC	1	–	1	1	1
3	BC	–	1	1	–	1
				T/C	T	C

T: function itself
C: complement of function

– : don't need this term

		A	B	C	f1	f2
1	AB′	1	0	-	1	-
2	AC	1	-	1	1	1
3	BC	-	1	1	-	1
					T	C
					T/C	

d) Implement f_1 and f_2 by using a 3 x 3 x 2 PLA and programming