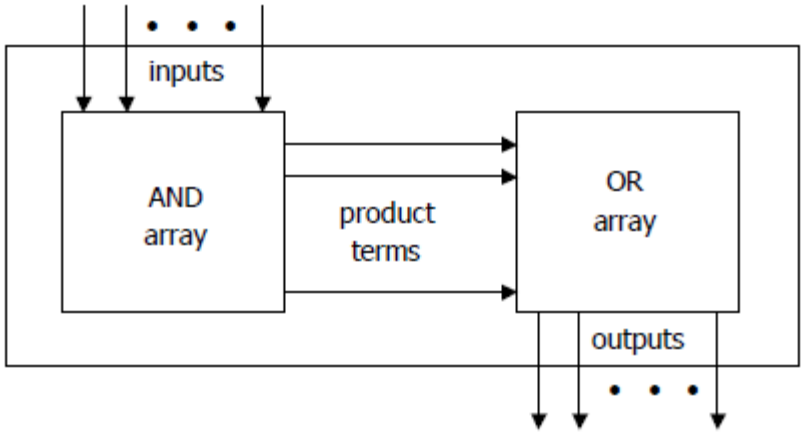
CSE 205: Digital Logic Design

Dr. Tanzima Hashem
Associate Professor
CSE, BUET

Programmable Logic Arrays (PLAs)

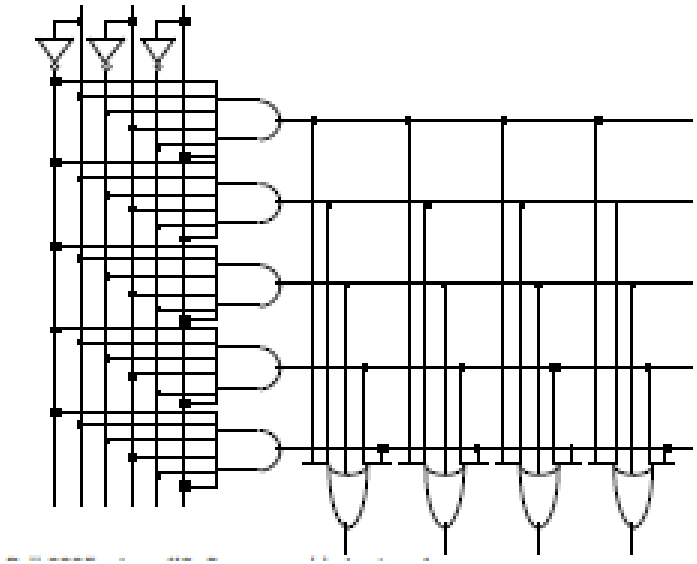
- Pre-fabricated building block of many AND/OR gates



- An array of programmable AND gates
  - can generate any product terms of the inputs
- An array of programmable OR gates
  - can generate the sums of the products

Programmable Logic Arrays (PLAs)

- Before programming, all possible connections available before "programming“
- ”Personalized" by making or breaking connections among gates



PLA Example

- Example: AND/OR/XOR
  - F1 = AB′ + AC + A′BC′
  - F2 = (AC + BC)′
- XOR gates can invert the outputs
  - invert: connected to 1
  - not change: connected to 0
- PLA programming table: 3 sections
  1. list the product terms
  2. specify the required paths between inputs and AND gates
  3. specify the paths between the AND and OR gates

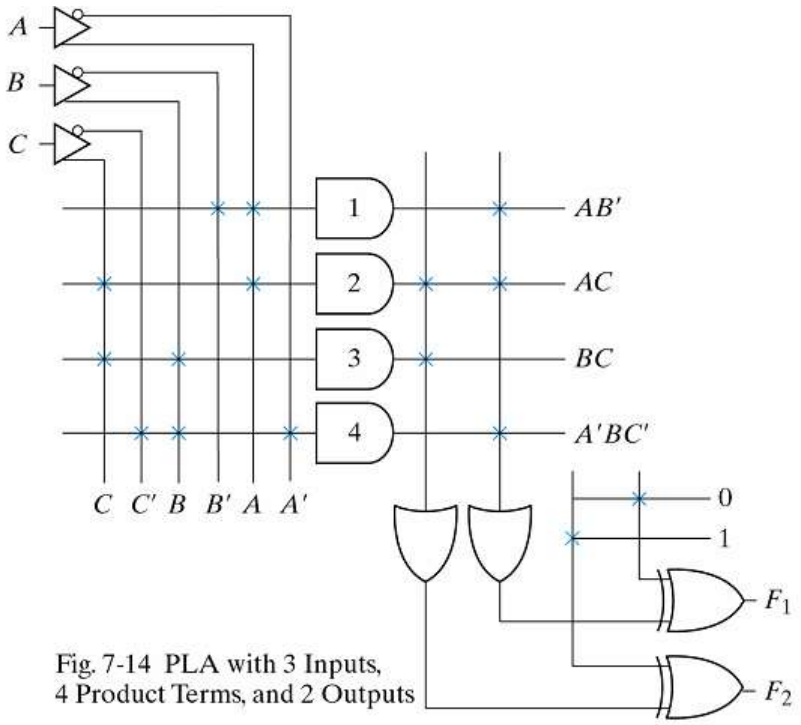


Fig. 7-14 PLA with 3 Inputs, 4 Product Terms, and 2 Outputs

Table 7-5 PLA Programming Table

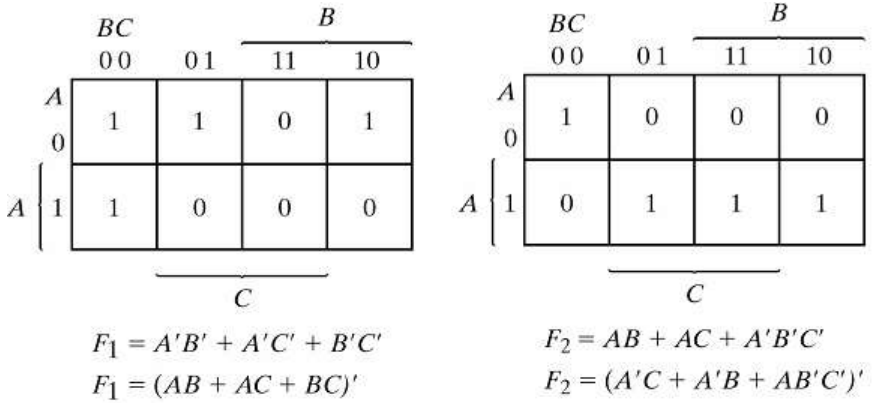
Product Term	Inputs A	Inputs B	Inputs C	Outputs (T) F_1	Outputs (C) F_2
AB′ 1	1	0	–	1	–
AC 2	1	–	1	1	1
BC 3	–	1	1	–	1
A′BC′ 4	0	1	0	1	–

Programmable Logic Arrays (PLAs)

- Size of PLA: specified by # of inputs, product terms and outputs
  - *n* inputs, *k* product terms and *m* outputs
  - *n* buffer-inverter gates, *k* AND gates, *m* OR gates, and *m* XOR gates
- Designing a digital system with a PLA
  - reduce the number of distinct product terms
  - the number of literals in a product is not important

Implement with PLAs

- F1(A, B, C) = Σ (0, 1, 2, 4);
  F2(A, B, C) = Σ (0, 5, 6, 7)
- Simply both the true and complement of the functions in sum of products
- Find the combination with minimum number of product terms
  - F1=(AB+AC+BC)'
  - F2=AB+AC+A'B'C'
- Obtain the PLA programming table



PLA programming table

	Product term	Inputs A	B	C	Outputs (C) F_1	(T) F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1

Fig. 7-15 Solution to Example 7-2

Syllabus

- Chapter 7:7.6