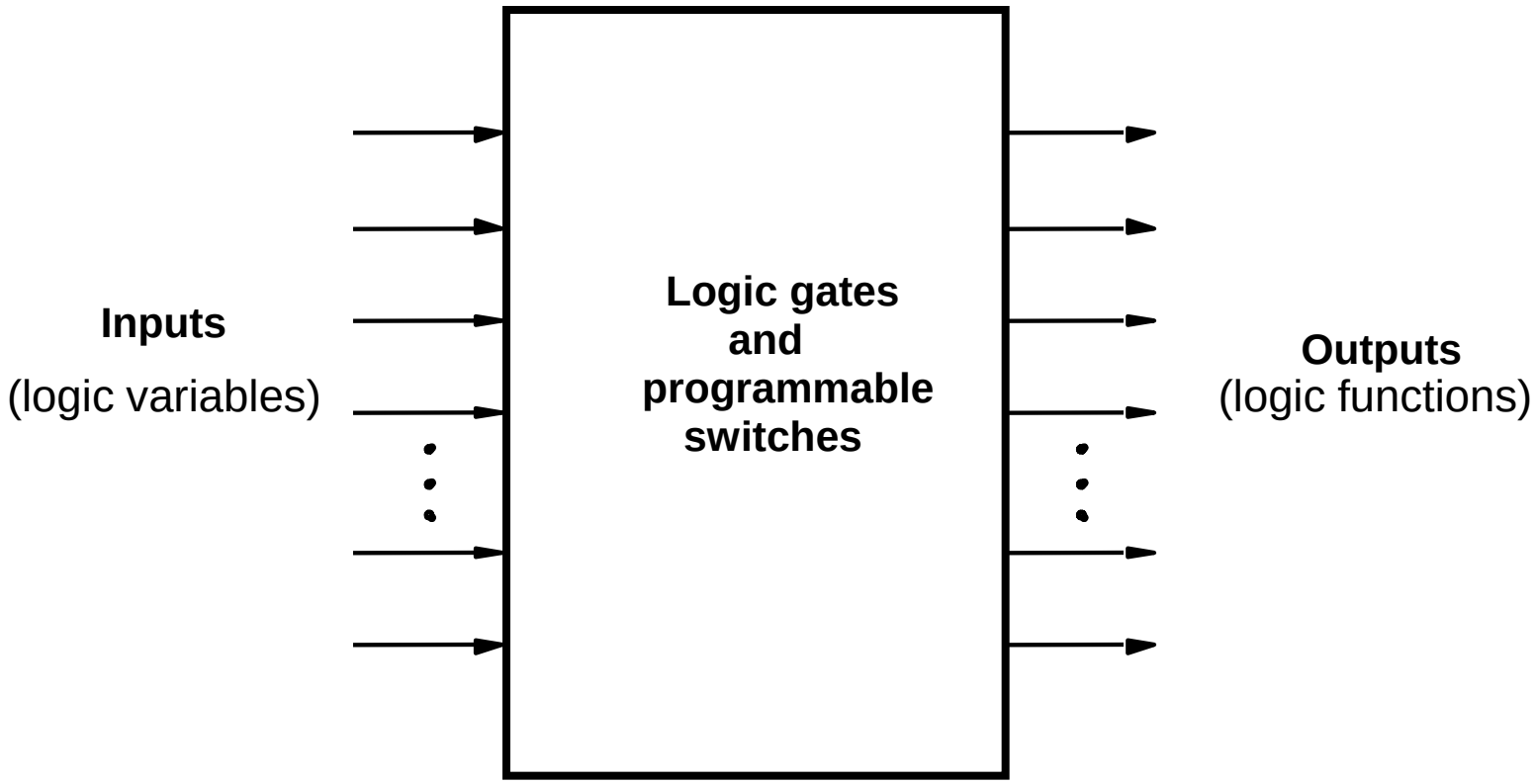
Programmable Logic Devices

PLDs

- Programmable Logic Devices (PLD)
  - General purpose chip for implementing circuits
  - Can be customized using programmable switches
- Programmable devices have their functionality programmed before they are first used.
- Range in complexity from 100's to 10,000's of logic gates.
- Main types of PLDs
  - ROM
  - PLA
  - PAL

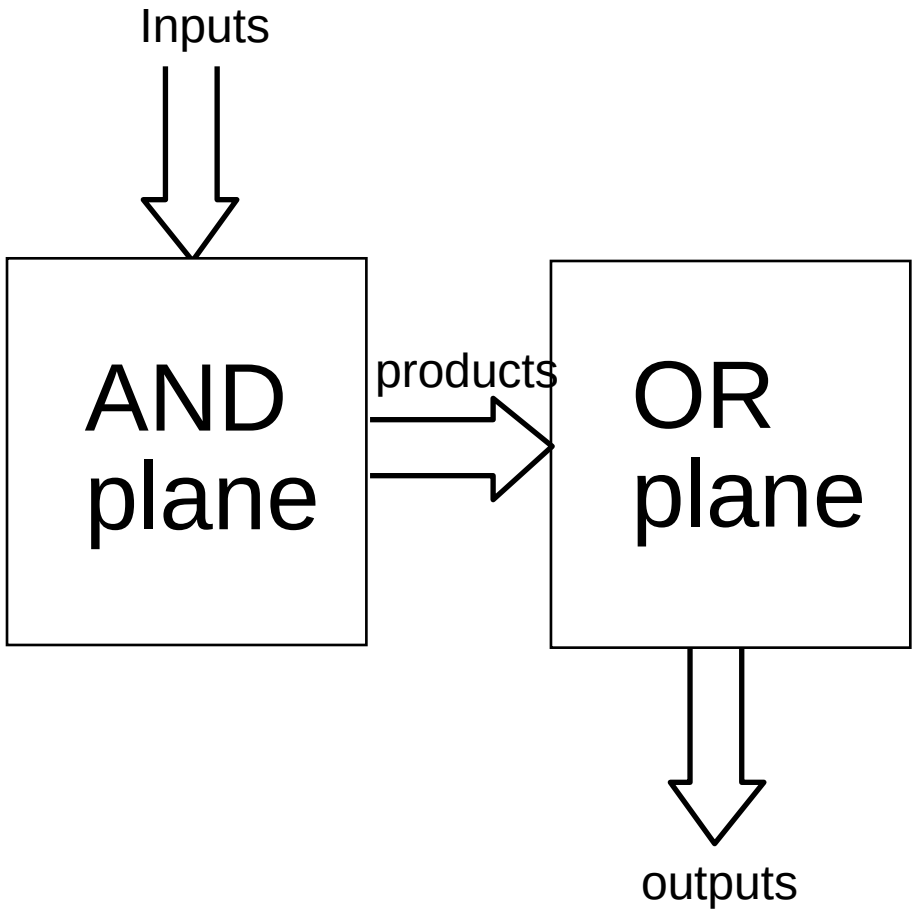
  → Simple/Sequential Prog Logic Devices
  - CPLD (Complex Prog. Logic Devices)
  - FPGA (Field Prog Gate Array)
- Custom chips: standard cells, sea of gates

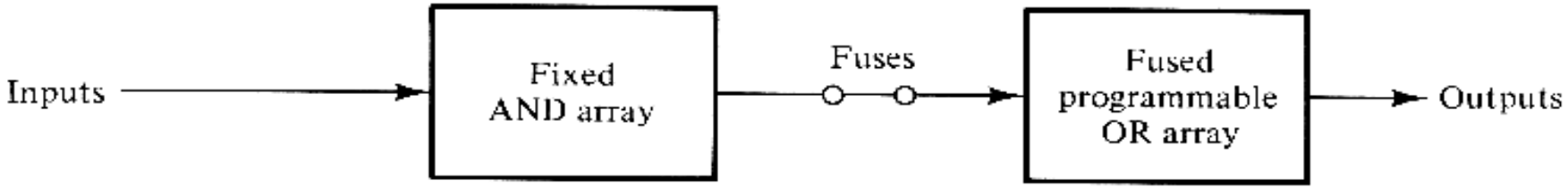
PLD as a Black Box



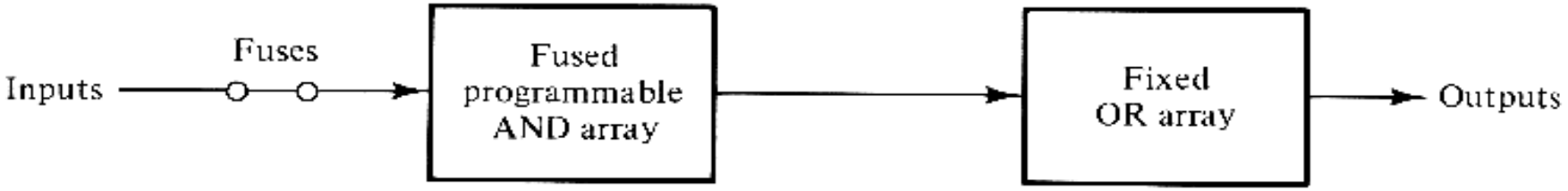
PLD's

Most of these devices are based on a two level structure (sum of products form).

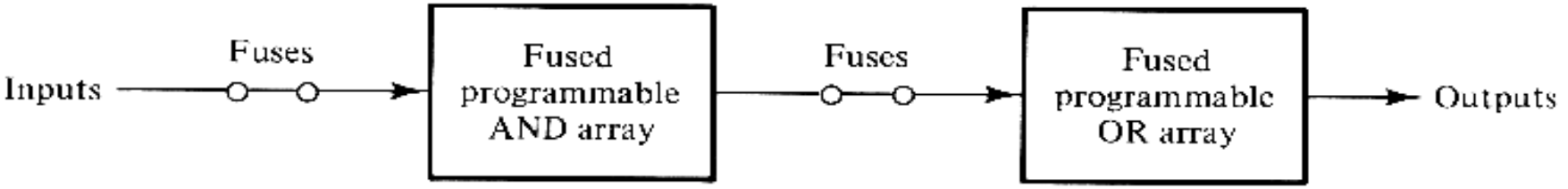




(a) Programmable read-only memory (PROM)



(b) Programmable array logic (PAL)



(c) Programmable logic array (PLA)

ROM

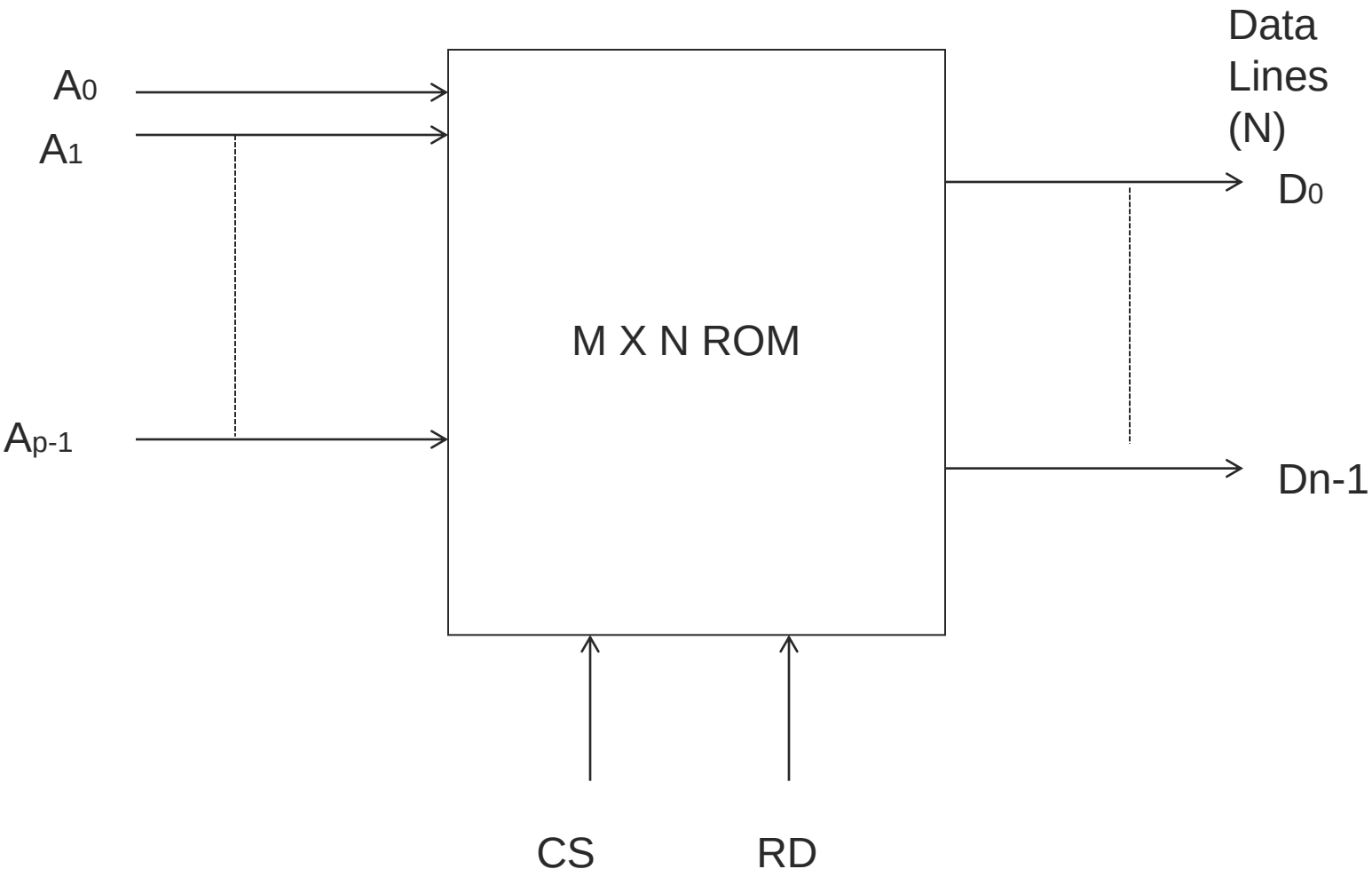
- A ROM (Read Only Memory) has a fixed AND plane and a programmable OR plane
- Size of AND plane is 2^n where n = number of input pins
  - Has an AND gate for every possible minterm so that all input combinations access a different AND gate
- OR plane dictates function mapped by the ROM

ROM

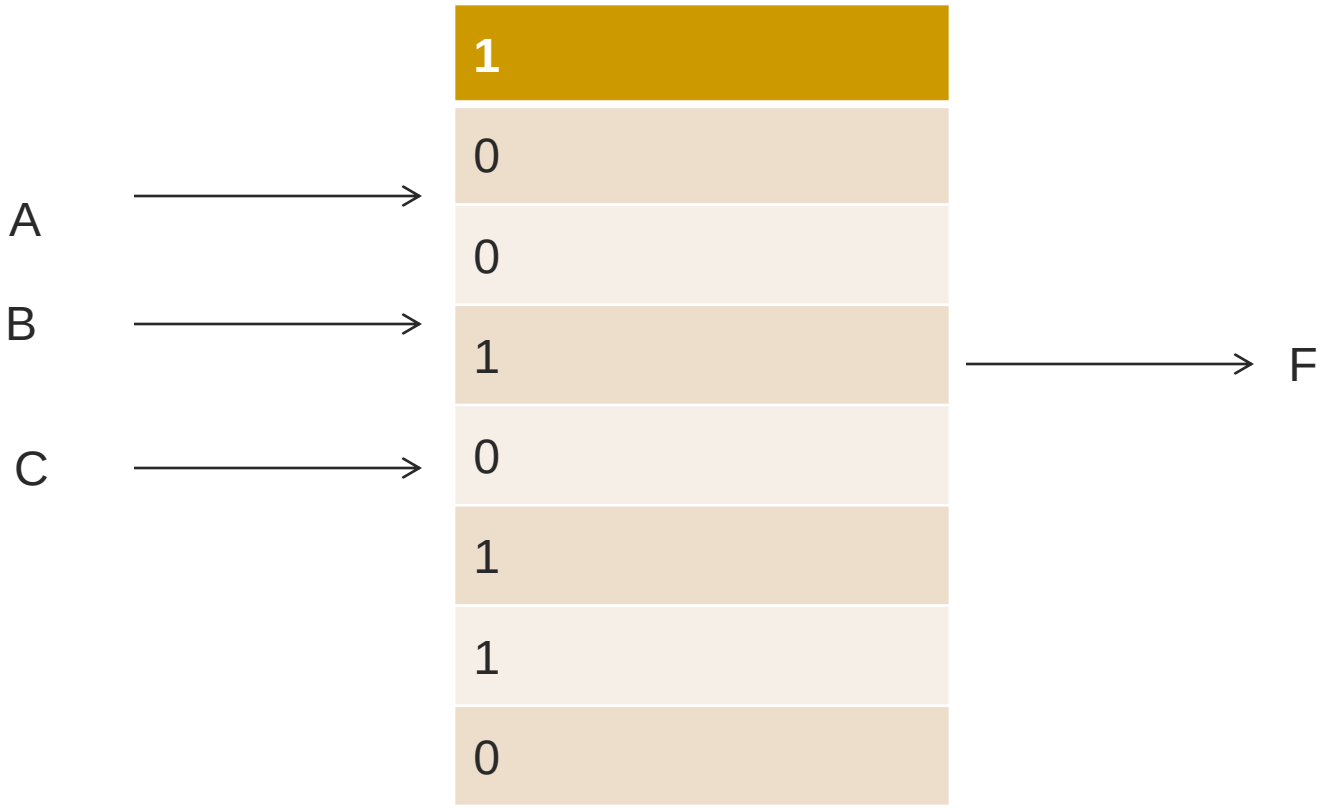
- a memory device of permanent binary information

- P input lines: address lines
- 2^p distinct addresses = M locations
- N of bits at each location = Data
- n output lines: word (no of bits stored data)
- CS Signal – Chip Select Signal
- RD Signal – Read Signal

Address Lines (P) such that $2^P >= M$ (M is no of locations)



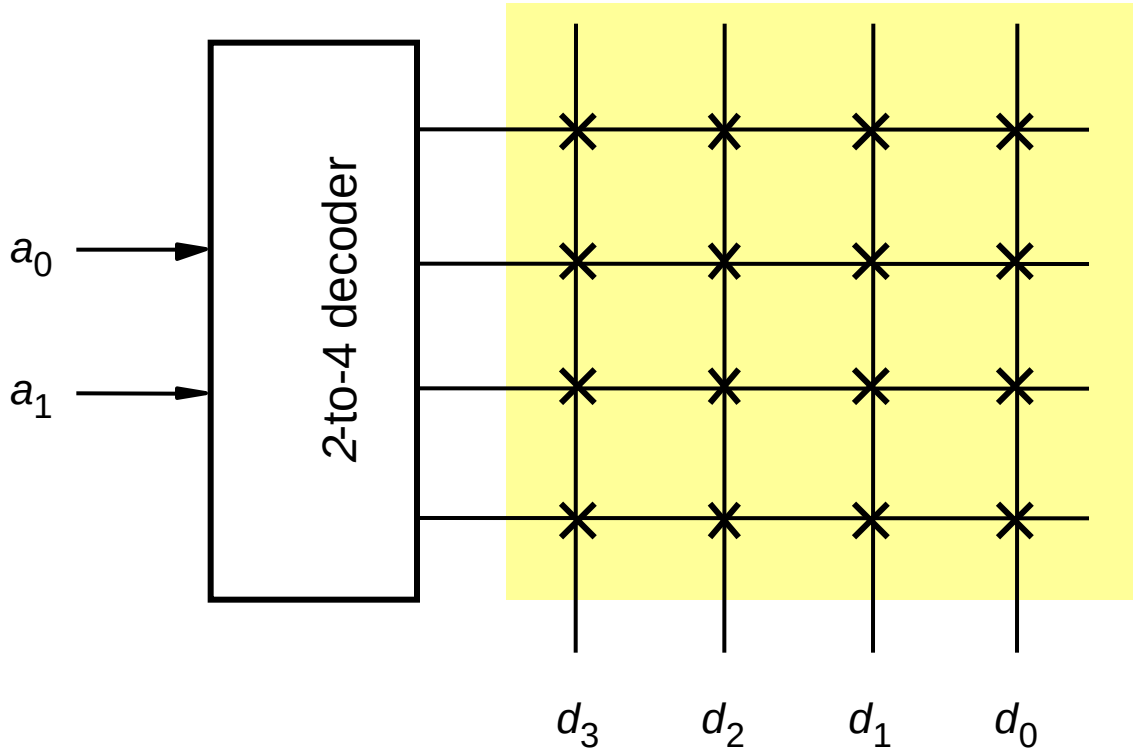
F (A,B,C) = Σ(0,3,5,6)



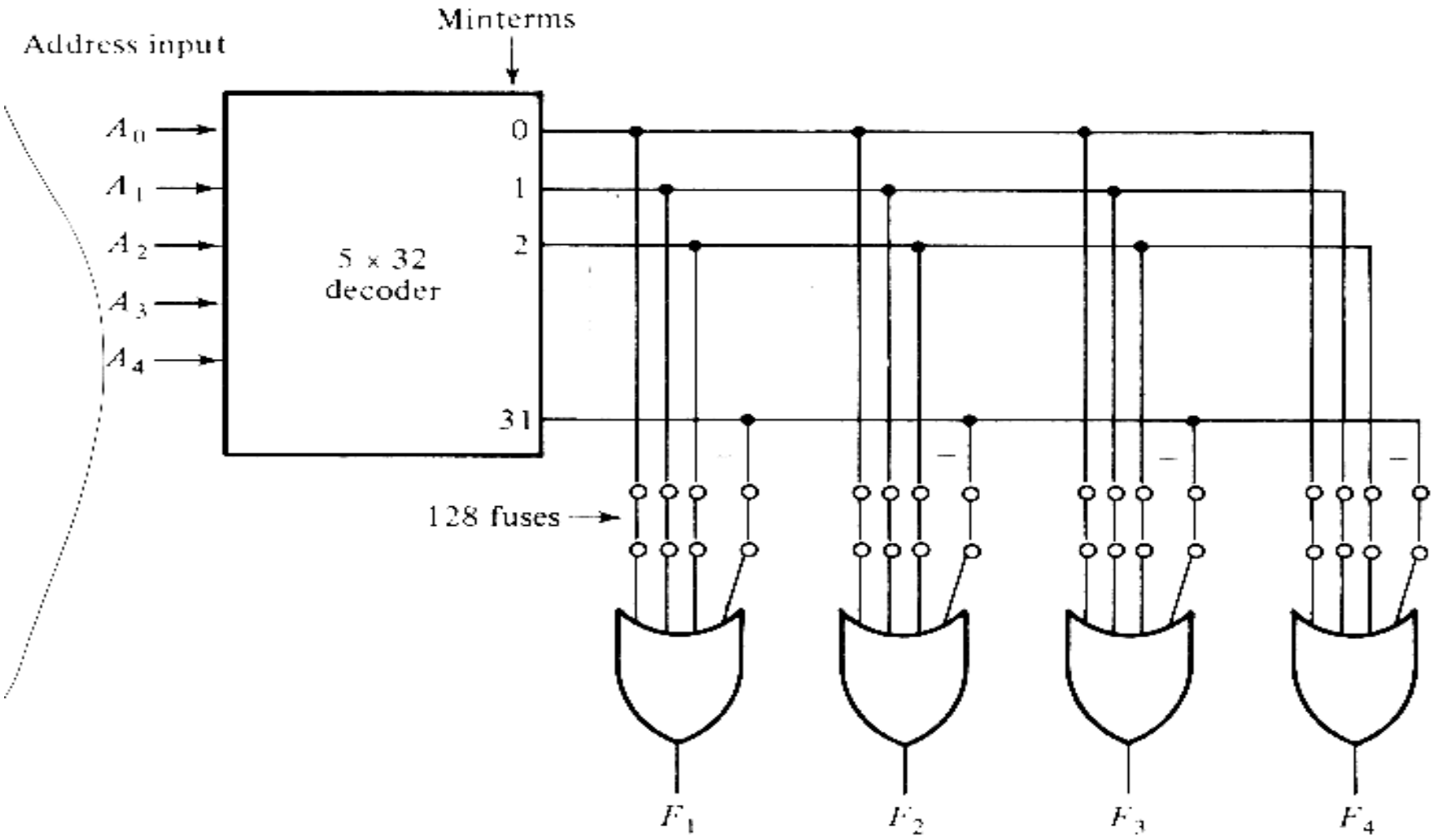
- a M x N decoder is used to select the location
- plus m OR gates
- can be used to implement any Boolean functions of n input variables
- a fixed AND array and a programmable OR array

4x4 ROM

- 4x4 ROM has 4 addresses that are decoded using 2 x 4 Decoder, 4 output lines and 4 data lines. Interconnections are called Crosslinks, are connected using fuses. X shows ON (1) status.



A 32x4 ROM

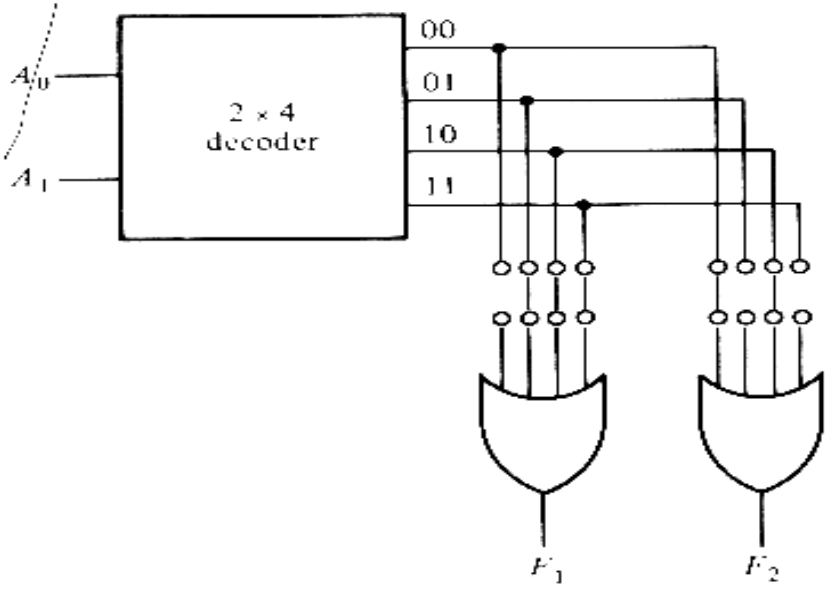

Minterms
Address input
A_0
A_1
A_2
A_3
A_4
5 × 32
decoder
0
1
2
31
128 fuses
F_1
F_2
F_3
F_4

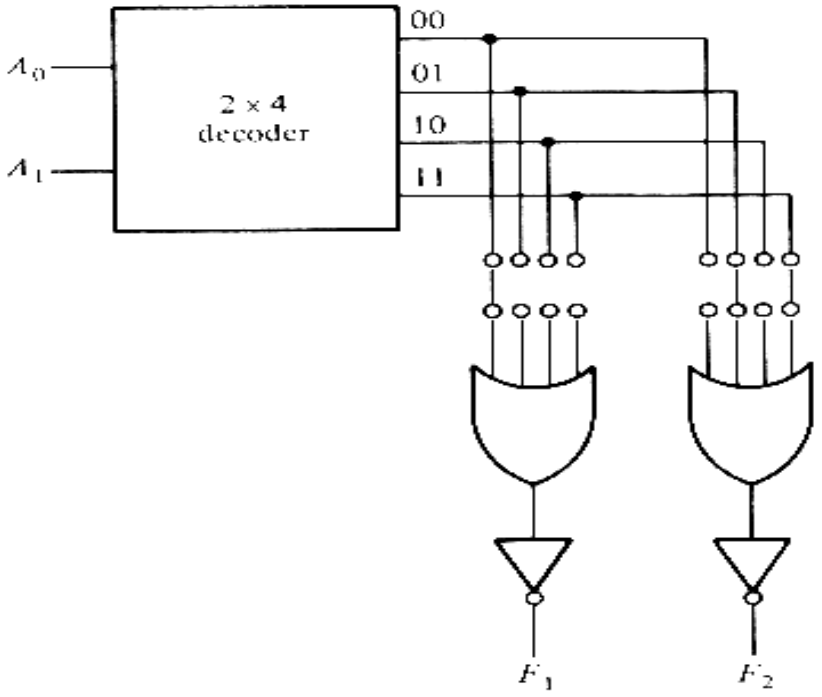
combinational logic Circuit implementation

- store the truth table in a ROM

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

(a) Truth table



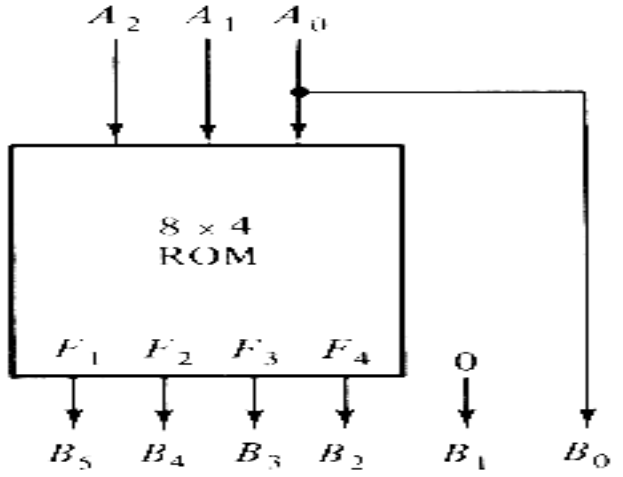


Examples 5-3

- generate the square of a 3-bit number

TABLE 5-5
Truth Table for Circuit of Example 5-3

Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49



(a) Block diagram

A_2	A_1	A_0	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

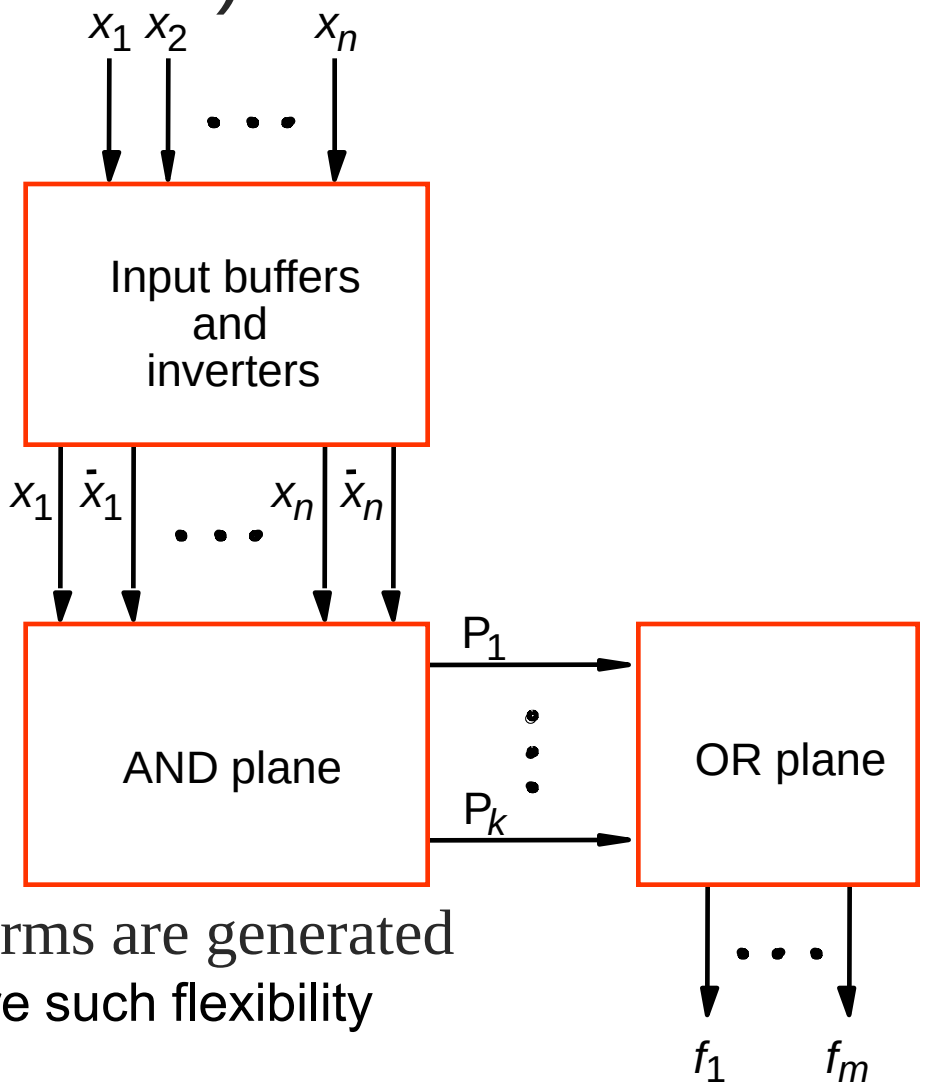
(b) ROM truth table

Types of ROMs

- mask programming ROM
  - IC manufacturers
  - is economical only if large quantities
- PROM: Programmable ROM
  - fuses
  - universal programmer
- EPROM: erasable PROM
  - floating gate
  - ultraviolet light erasable
- EEPROM: electrically erasable PROM
  - longer time is needed to write
  - flash ROM
  - limited times of write operations

Programmable Logic Array (PLA) – (Both Array are programmable)

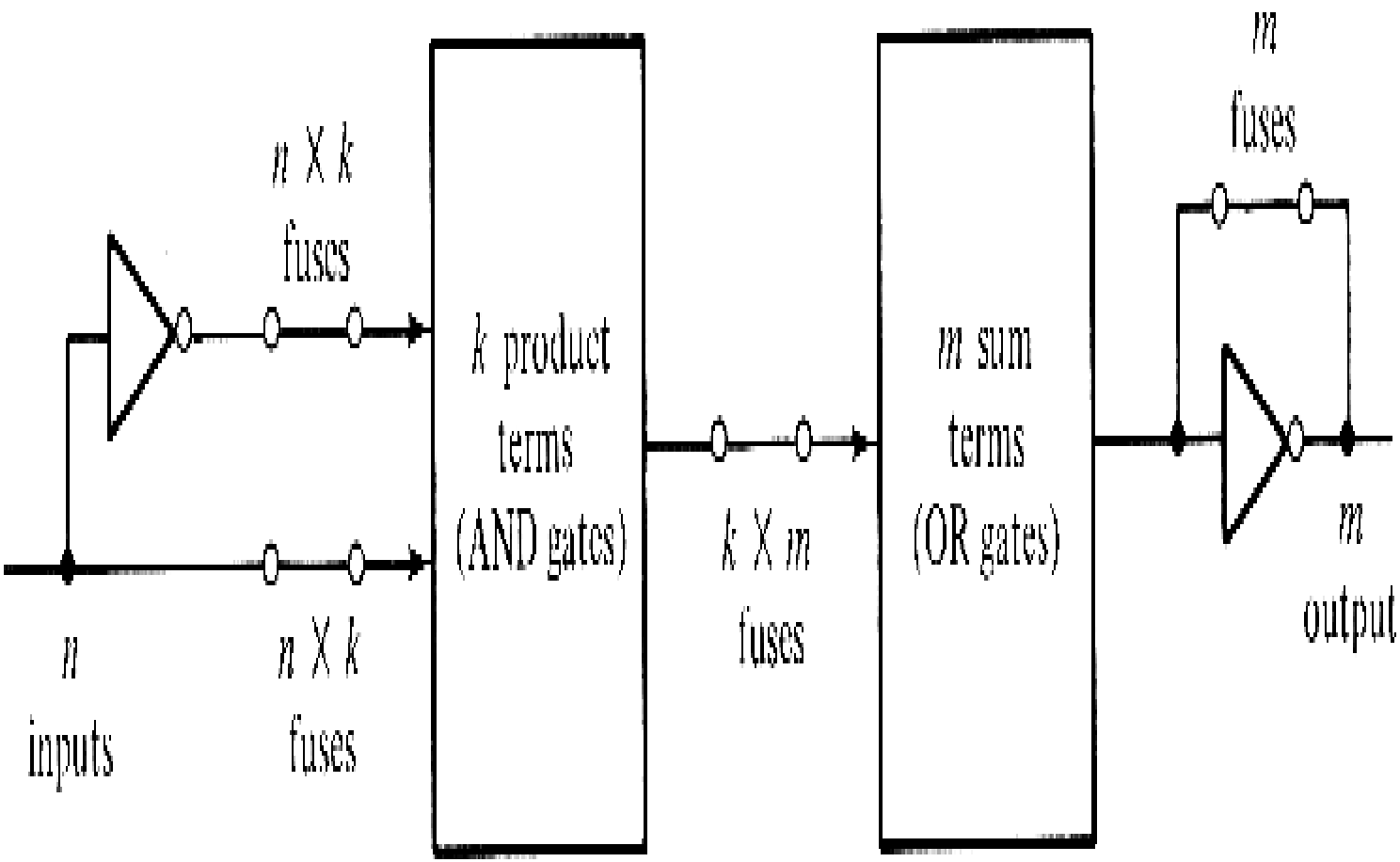
- Use to implement circuits in SOP form
- The connections in the AND plane are programmable
- The connections in the OR plane are programmable



only the needed product terms are generated

- Many applications don't require such flexibility

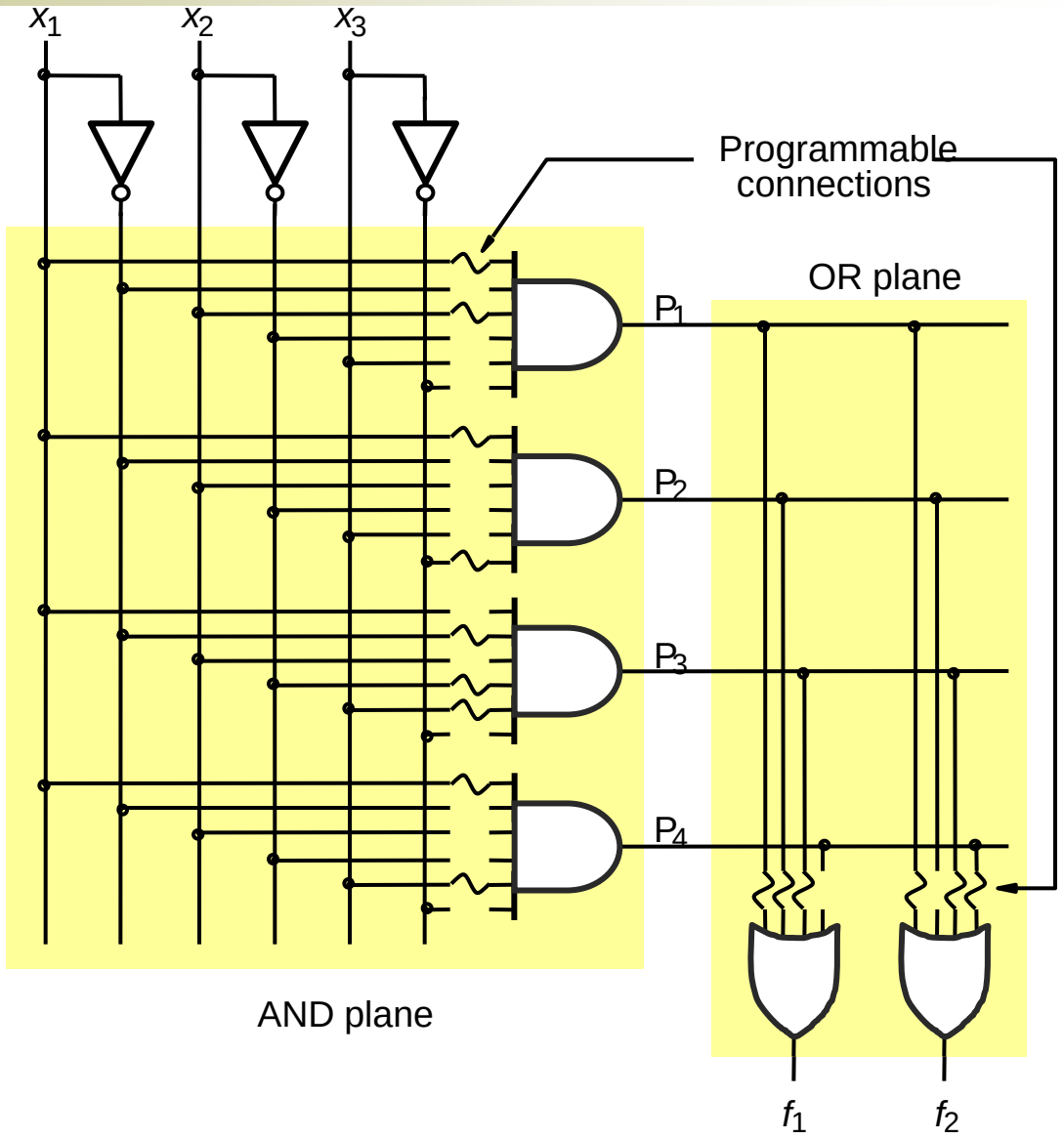
PLA


n X k
fuses
k product
terms
(AND gates)
k X m
fuses
m sum
terms
(OR gates)
m
fuses
m
output
n
inputs
n X k
fuses

Gate Level Version of PLA

$f_1 = x_1x_2+x_1x_3'+x_1'x_2'x_3$

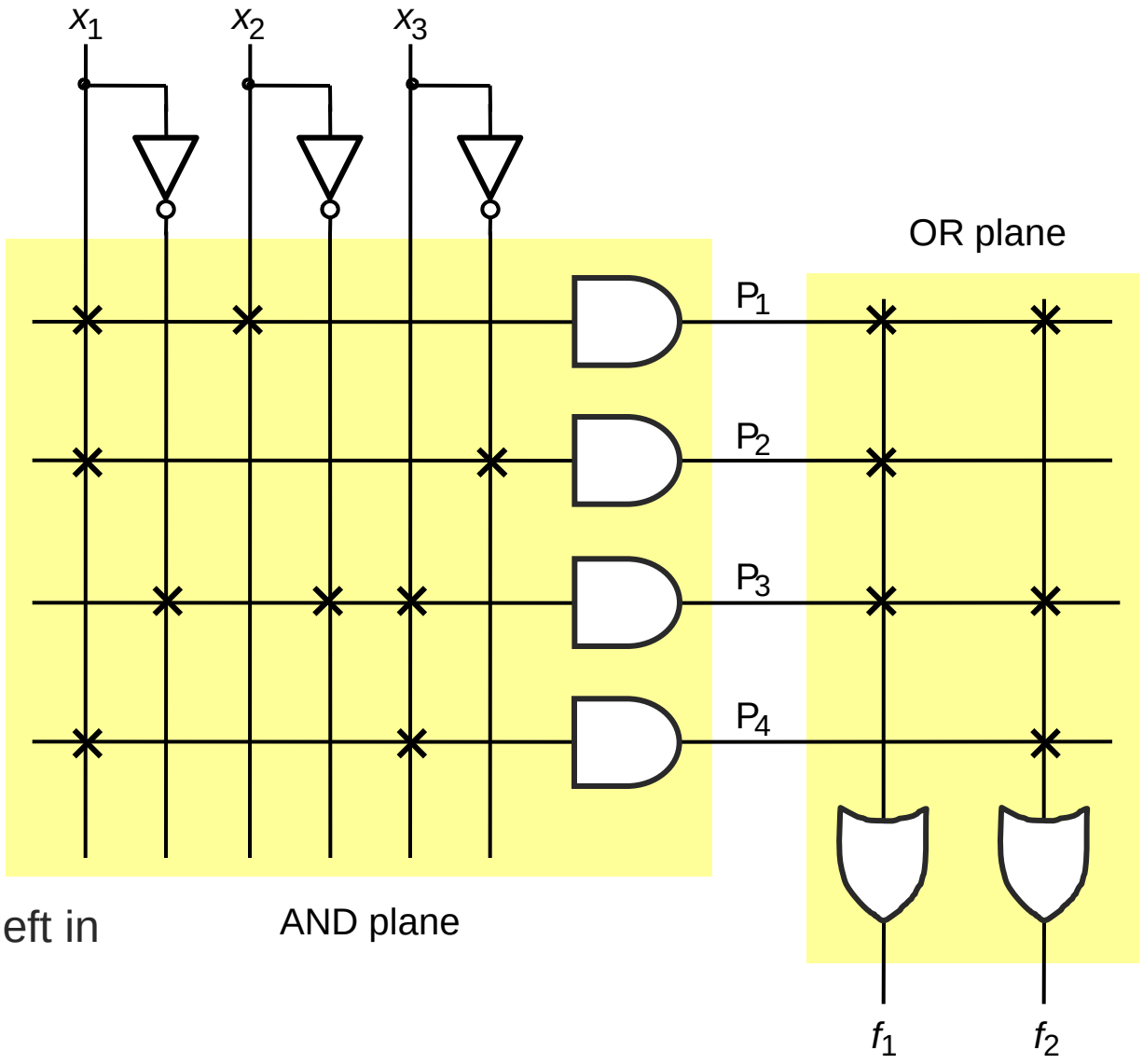
$f_2 = x_1x_2+x_1'x_2'x_3+x_1x_3$



Customary Schematic of a PLA

$f_1 = x_1x_2+x_1x_3'+x_1'x_2'x_3$

$f_2 = x_1x_2+x_1'x_2'x_3+x_1x_3$



x marks the connections left in place after programming

Advantages of PLA:

- Both AND and OR array are programmable,
- It gives flexibility for implementation of Logic design.
- Included as a part of larger chips such as microprocessors.
- Power requirement is less than ROM.
- Cost is also less.

Disadvantages:

Simplification of boolean expression is required.

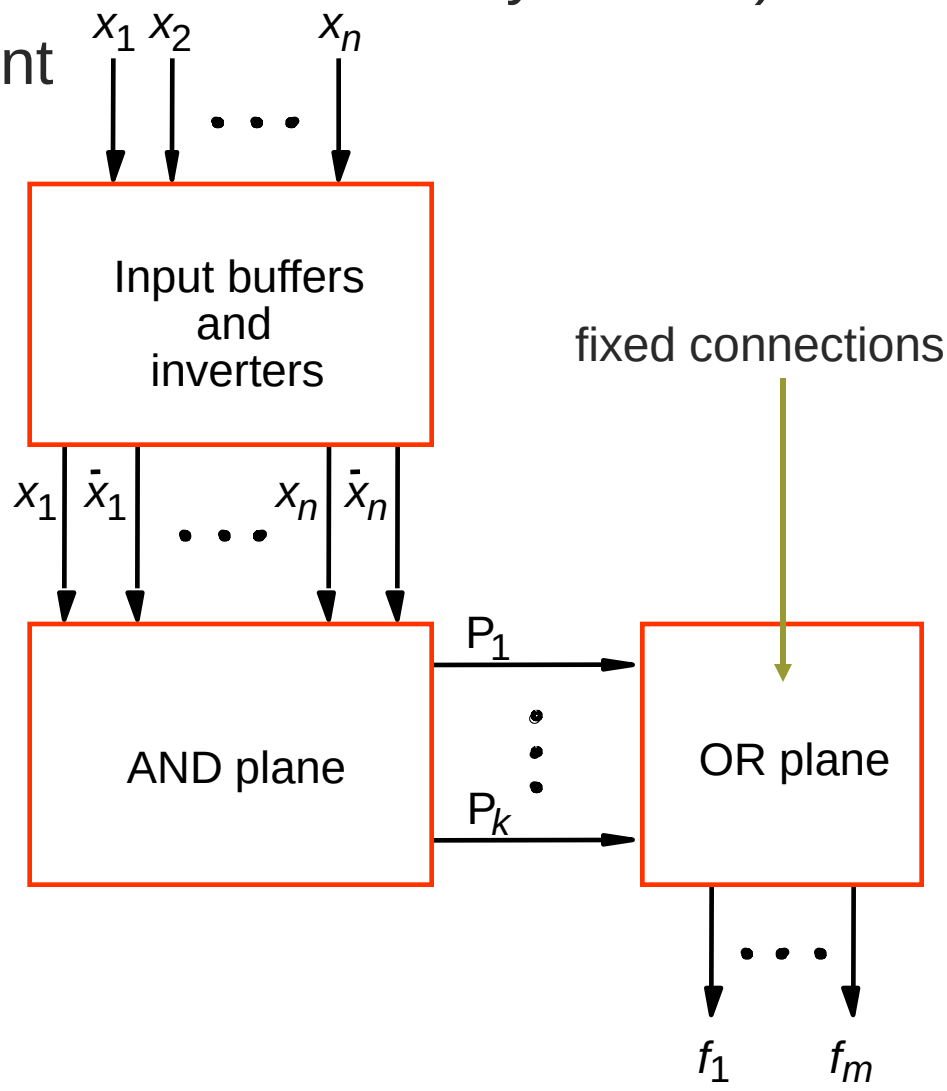
Limitations of PLAs

- PLAs come in various sizes
  - Typical size is 16 inputs, 32 product terms, 8 outputs
    - Each AND gate has large fan-in → this limits the number of inputs that can be provided in a PLA
    - 16 inputs → 3^{16} = possible input combinations; only 32 permitted (since 32 AND gates) in a typical PLA
    - 32 AND terms permitted → large fan-in for OR gates as well
      - This makes PLAs slower and slightly more expensive than some alternatives to be discussed shortly
    - 8 outputs → could have shared minterms, but not required

Programmable Array Logic (PAL)

(AND Array is programmable and OR array is fixed)

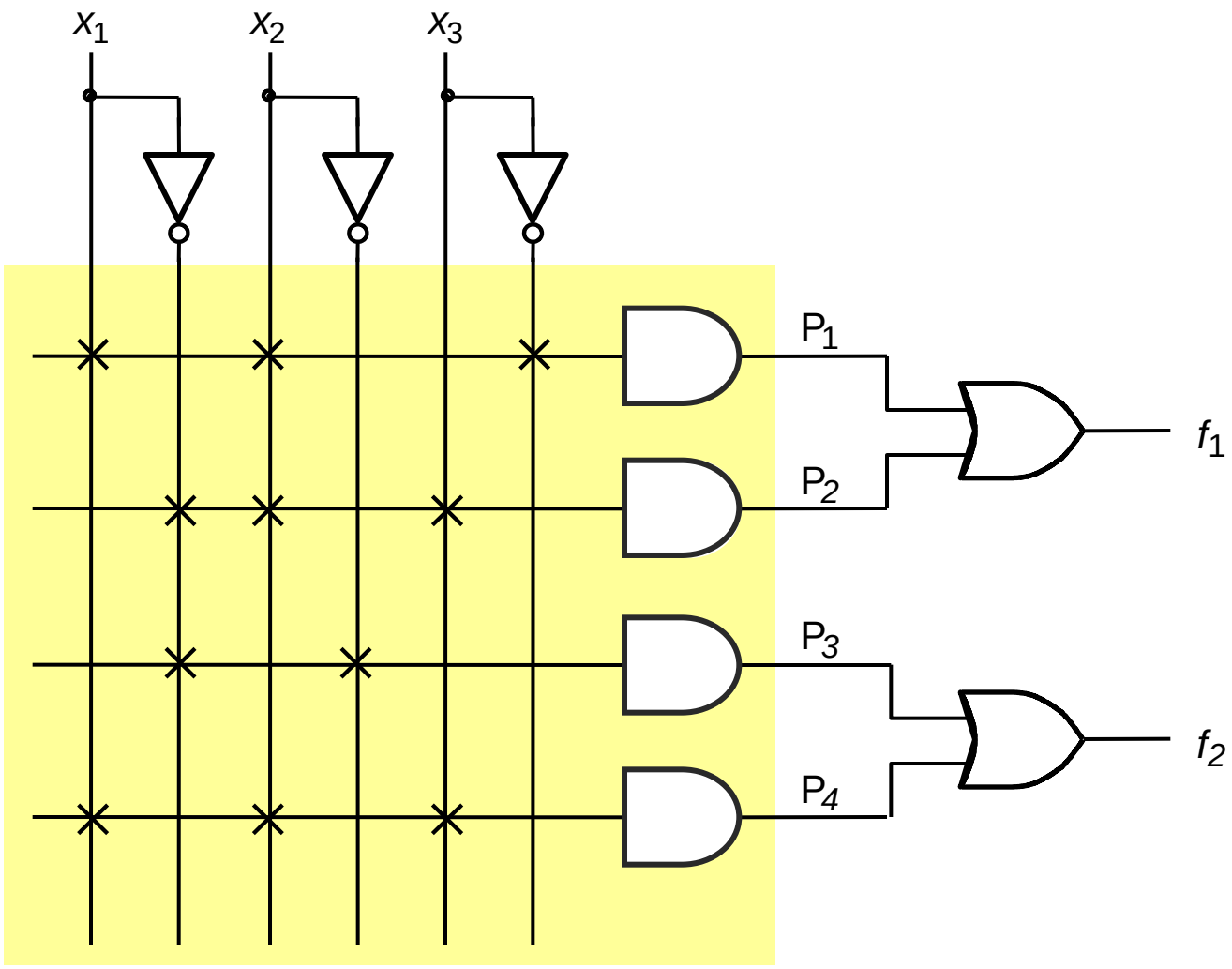
- Also used to implement circuits in SOP form
- The connections in the AND plane are programmable
- The connections in the OR plane are NOT programmable



Example Schematic of a PAL

$f_1 = x_1x_2x_3'+x_1'x_2x_3$

$f_2 = x_1'x_2'+x_1x_2x_3$



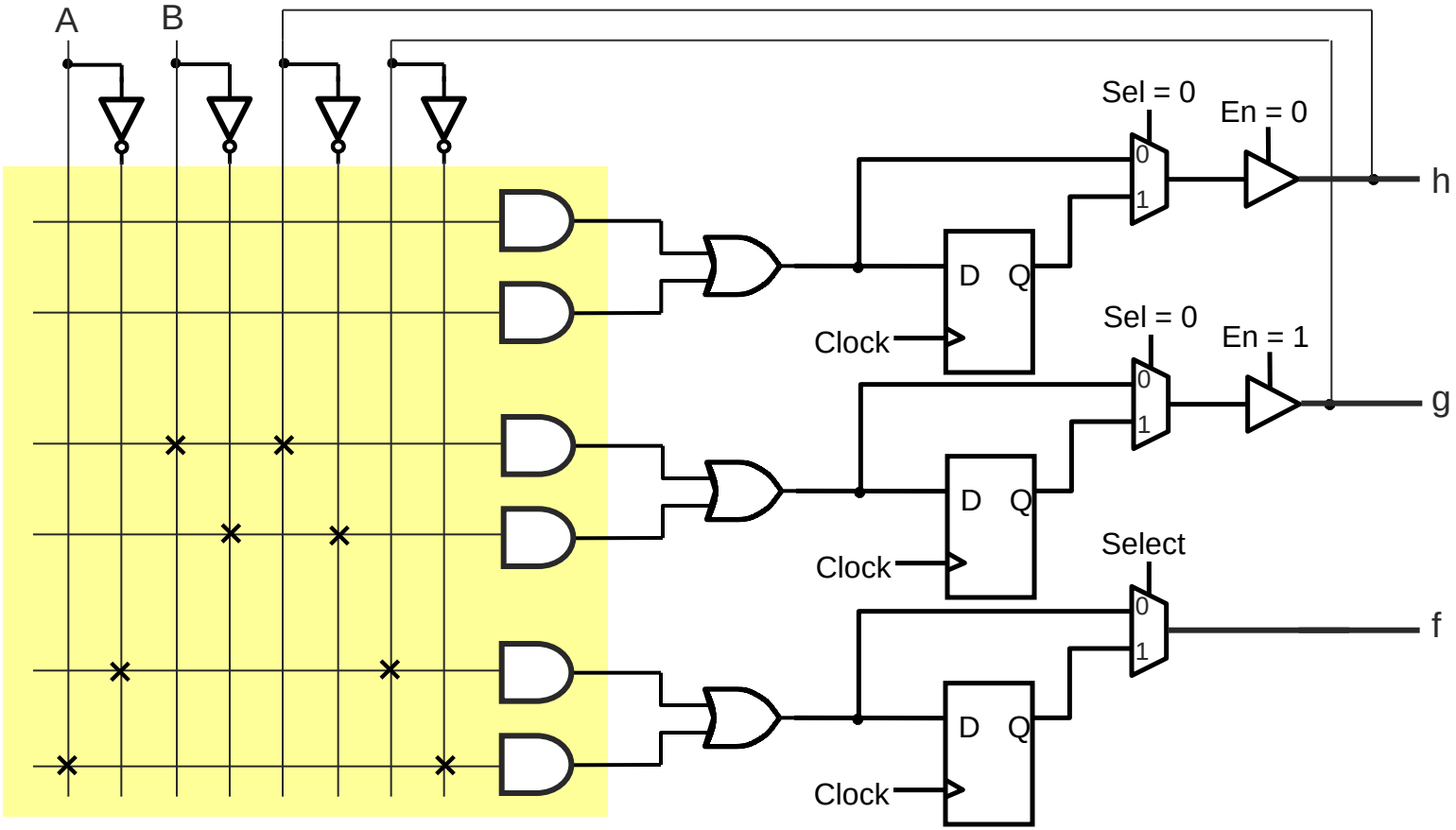
AND plane

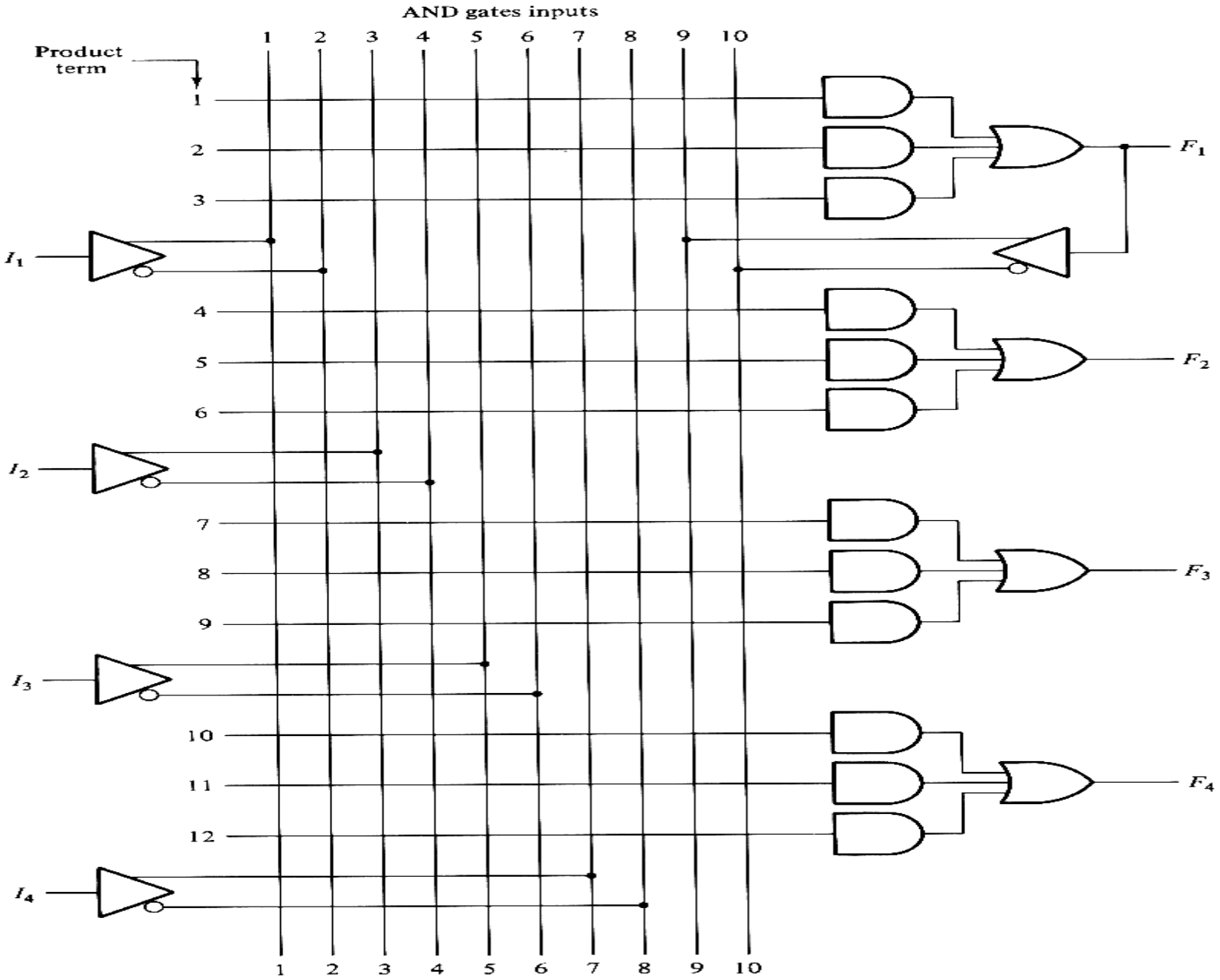
Comparing PALs and PLAs

- PALs have the same limitations as PLAs (small number of allowed AND terms) plus they have a fixed OR plane → less flexibility than PLAs
- PALs are simpler to manufacture, cheaper, and faster (better performance)
- PALs also often have extra circuitry connected to the output of each OR gate
  - The OR gate plus this circuitry is called a *macrocell*

Multi-Level Design with PALs

- f = A'BC + A'B'C' + ABC' + AB'C = A'g + Ag'
  - where g = BC + B'C' and C = h below



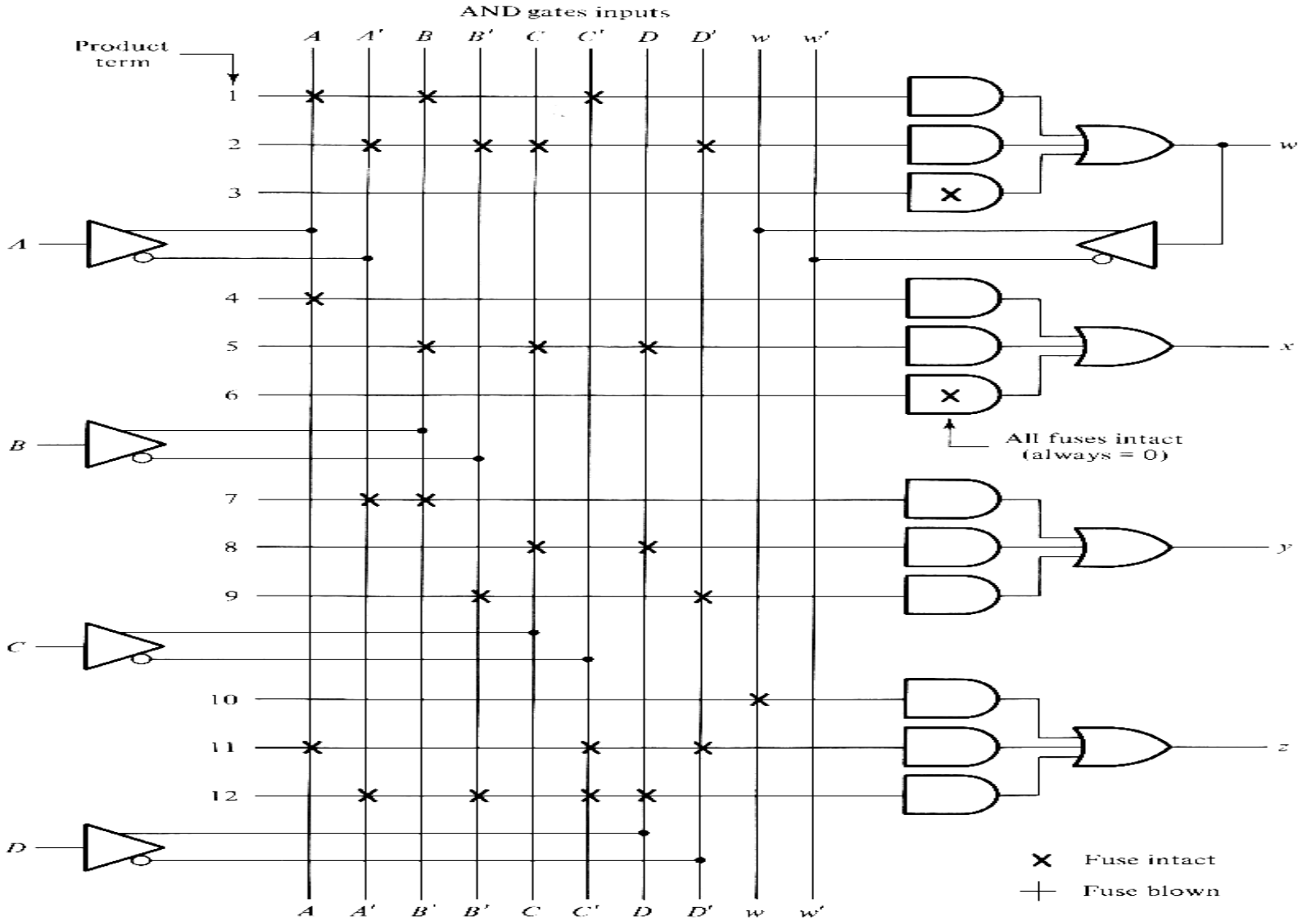

AND gates inputs
1 2 3 4 5 6 7 8 9 10
Product term
1
2
3
4
5
6
7
8
9
10
11
12
I_1
I_2
I_3
I_4
F_1
F_2
F_3
F_4
1 2 3 4 5 6 7 8 9 10

- can generate any product term
- each OR has only three inputs

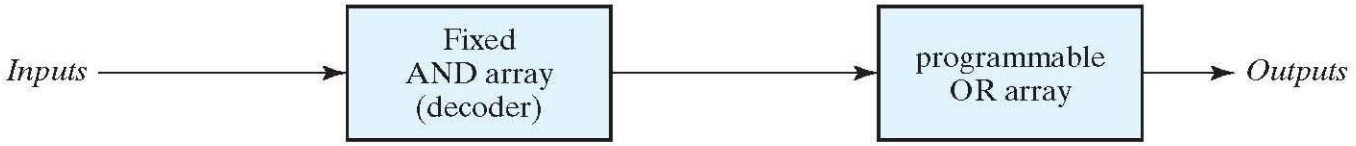
- Commercial PAL
  - more than 8 inputs
  - some of the output terminals are sometimes bidirectional
  - each OR gate may have 8 inputs
  - the fuse pattern may be unreadable
  - output terminals may be latched

- An example
  - w = Σ(2,12,13)
    x = Σ(7,8,9,10,11,12,13,14,15) y = Σ(0,2,3,4,5,6,7,8,10,11,15) z = Σ(1,2,8,12,13)
  - w = ABC'+A'B'CD'
    x = A+BCD y = A'B+CD+B'D' z = ABC'+A'B'CD'+AC'D'+A'B'C'D
    = w + AC'D'+A'B'C'D
  - using the output from w, the function z is reduced to three terms
  - if an input of the AND gates is not used, leave all the input fuses input intact

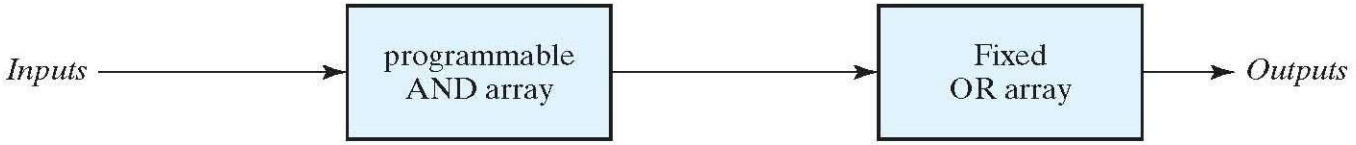
The fuse map


AND gates inputs
Product term
A A' B B' C C' D D' w w'
1
2
3
4
5
6
7
8
9
10
11
12
A
B
C
D
w
x
y
z
All fuses intact (always = 0)
Fuse intact
Fuse blown

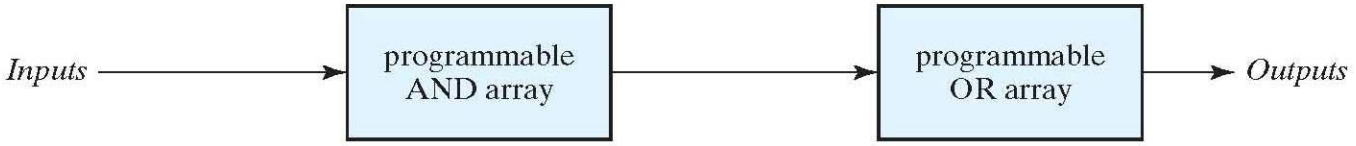
Combinational PLDs



(a) Programmable read-only memory (PROM)



(b) Programmable array logic (PAL)



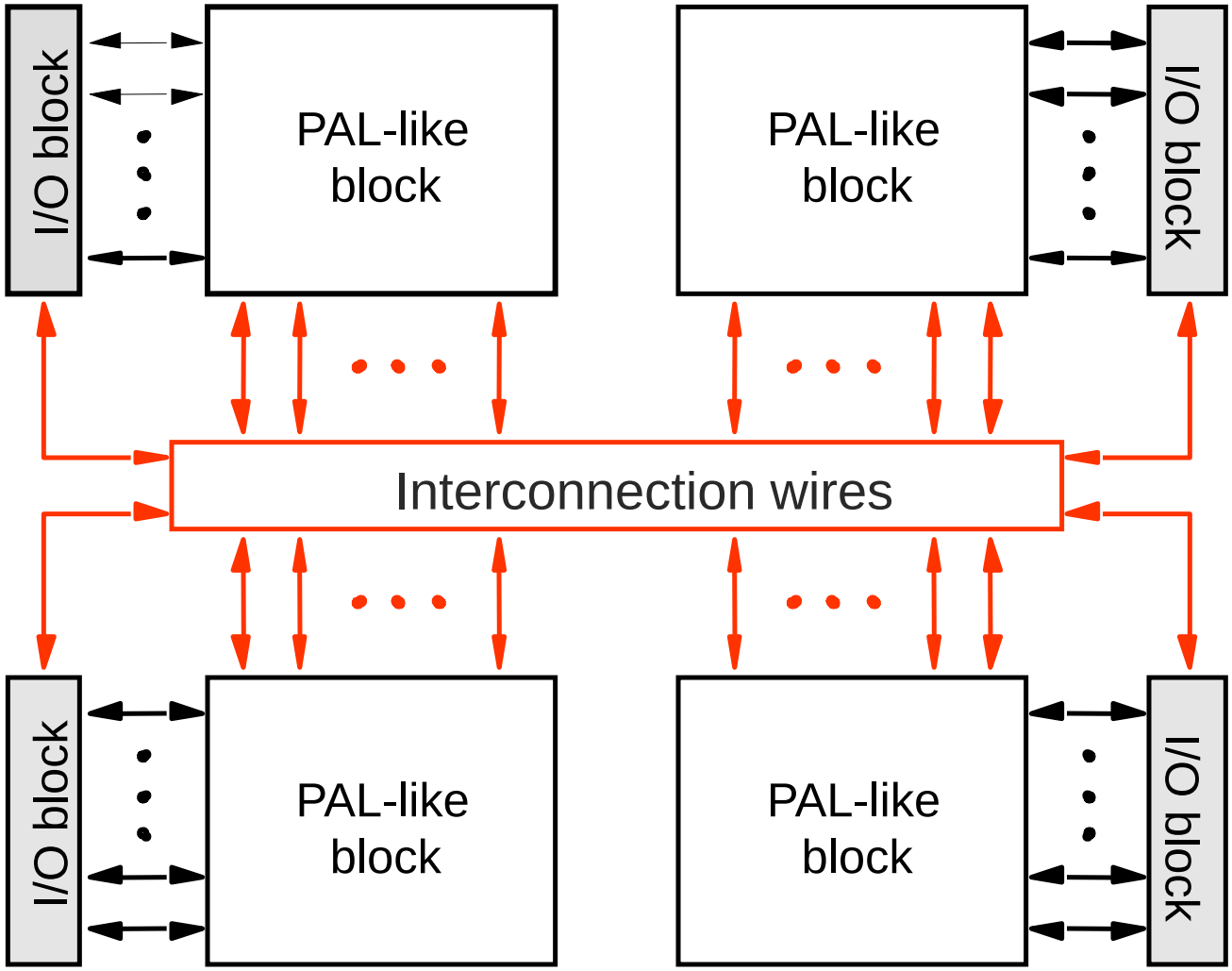
(c) Programmable logic array (PLA)

Simple Programmable Logic Devices
and
Complex Programmable Logic Devices

CPLD

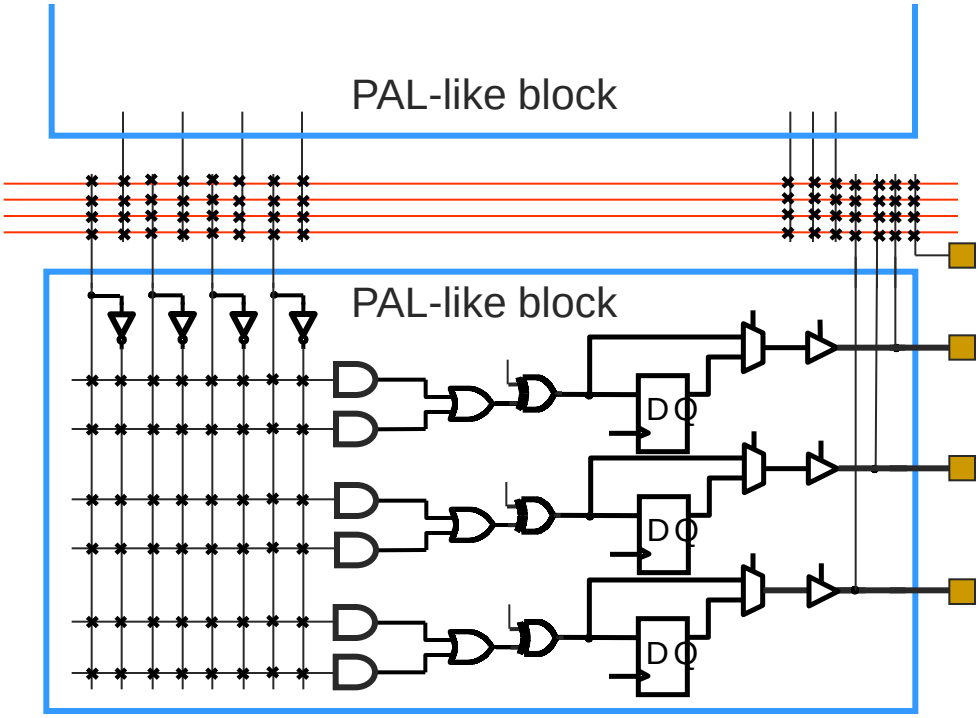
- Complex Programmable Logic Devices (CPLD)
- SPLDs (PLA, PAL) are limited in size due to the small number of input and output pins and the limited number of product terms
  - Combined number of inputs + outputs < 32 or so
- CPLDs contain multiple circuit blocks on a single chip
  - Each block is like a PAL: PAL-like block
  - Connections are provided between PAL-like blocks via an interconnection network that is programmable
  - Each block is connected to an I/O block as well

Structure of a CPLD



Internal Structure of a PAL-like Block

- Includes macrocells
  - Usually about 16 each
- Fixed OR planes
  - OR gates have fan-in between 5-20
- XOR gates provide negation ability
  - XOR has a control input

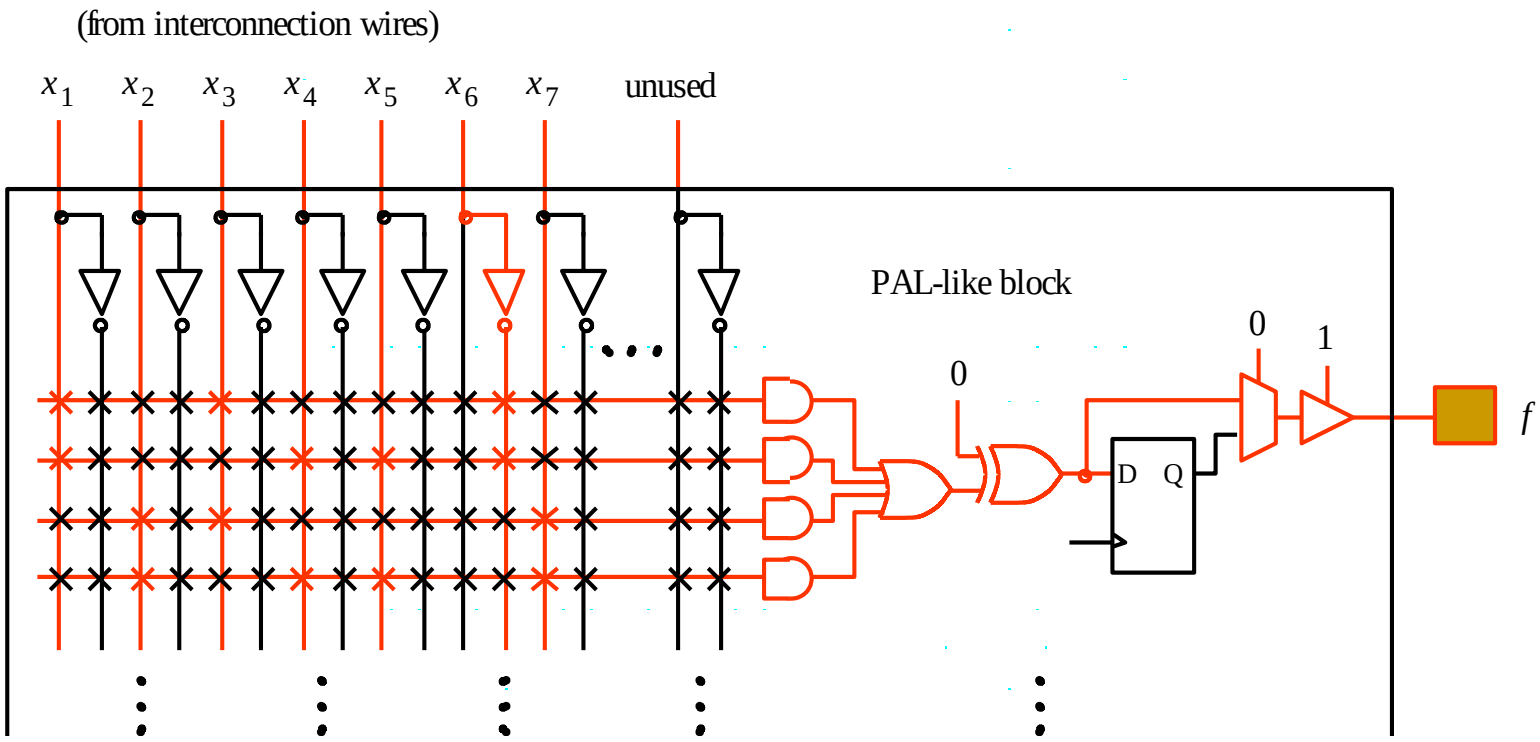


More on PAL-like Blocks

- CPLD pins are provided to control XOR, MUX, and tri-state gates
- When tri-state gate is disabled, the corresponding output pin can be used as an input pin
  - The associated PAL-like block is then useless
- The AND plane and interconnection network are programmable
- Commercial CPLDs have between 2-100 PAL-like blocks

Example CPLD

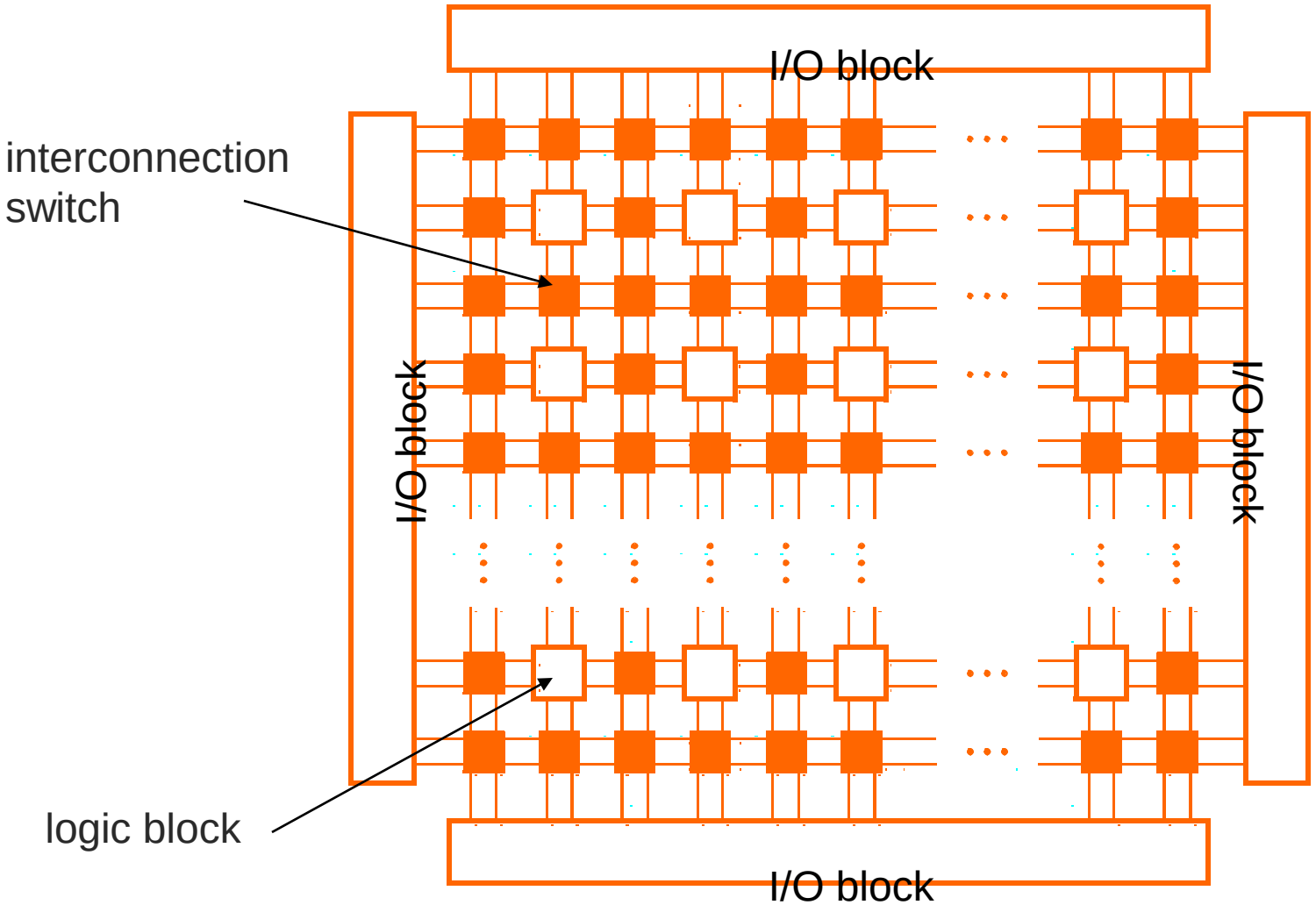
- Use a CPLD to implement the function
  - $f = x_1x_3x_6' + x_1x_4x_5x_6' + x_2x_3x_7 + x_2x_4x_5x_7$



FPGA

- SPLDs and CPLDs are relatively small and useful for simple logic devices
  - Up to about 20000 gates
- Field Programmable Gate Arrays (FPGA) can handle larger circuits
  - No AND/OR planes
  - Provide logic blocks, I/O blocks, and interconnection wires and switches
  - Logic blocks provide functionality
  - Interconnection switches allow logic blocks to be connected to each other and to the I/O pins

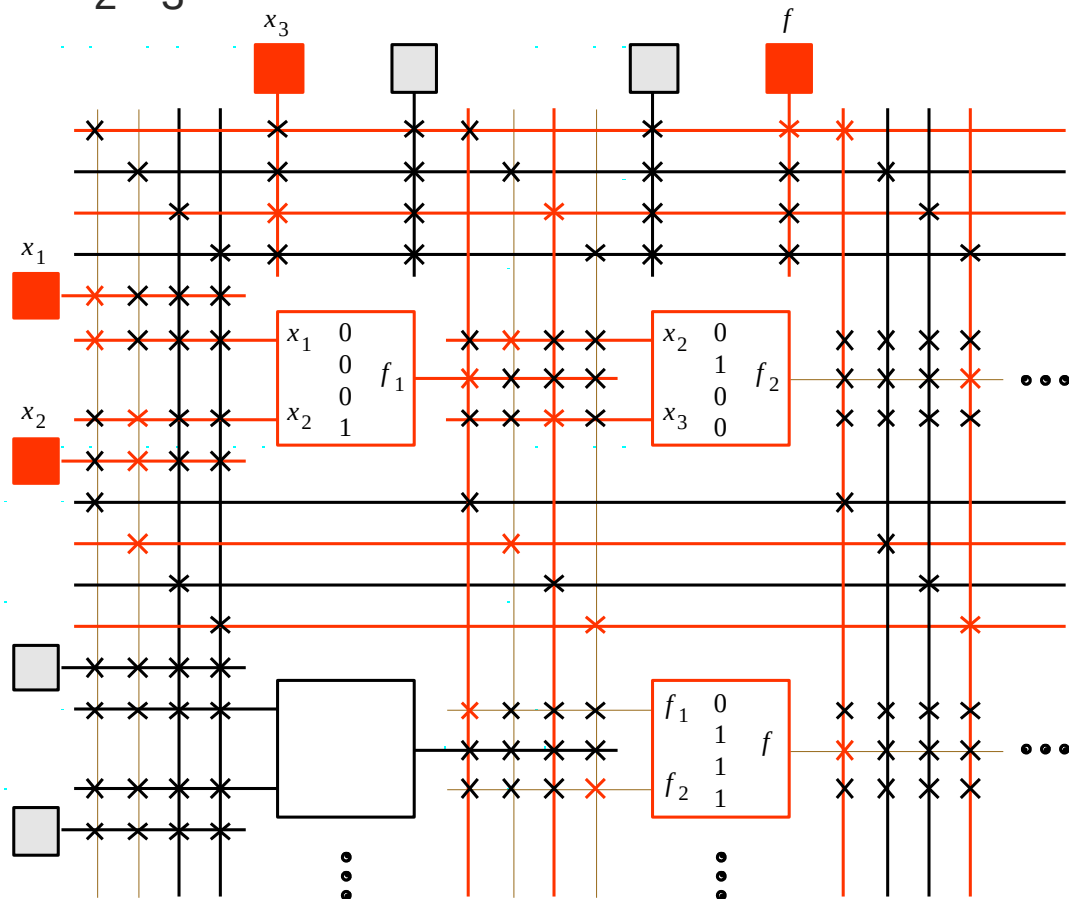
Structure of an FPGA



Example FPGA

- Use an FPGA with 2 input LUTS to implement the function $f = x_1x_2 + x_2'x_3$

  - $f_1 = x_1x_2$
  - $f_2 = x_2'x_3$
  - $f = f_1 + f_2$



Another Example FPGA

- Use an FPGA with 2 input LUTS to implement the function $f = x_1x_3x_6' + x_1x_4x_5x_6' + x_2x_3x_7 + x_2x_4x_5x_7$
  - Fan-in of expression is too large for FPGA (this was simple to do in a CPLD)
  - Factor f to get sub-expressions with max fan-in = 2
    - $f = x_1x_6'(x_3 + x_4x_5) + x_2x_7(x_3 + x_4x_5)$
      $= (x_1x_6' + x_2x_7)(x_3 + x_4x_5)$
  - Could use Shannon's expansion instead
    - Goal is to build expressions out of 2-input LUTs

Custom Chips

- PLDs are limited by number of programmable switches
  - Consume space
  - Reduce speed

- Custom chips are created from scratch
  - Expensive → used when high speed is required, volume sales are expected, and chip size is small but with high density of gates
  - ASICs (Application Specific Integrated Circuits) are custom chips that use a standard cell layout to reduce design costs