Read-Only Memory, Programmable Devices & Sequential Programmable Devices
Lecture 5

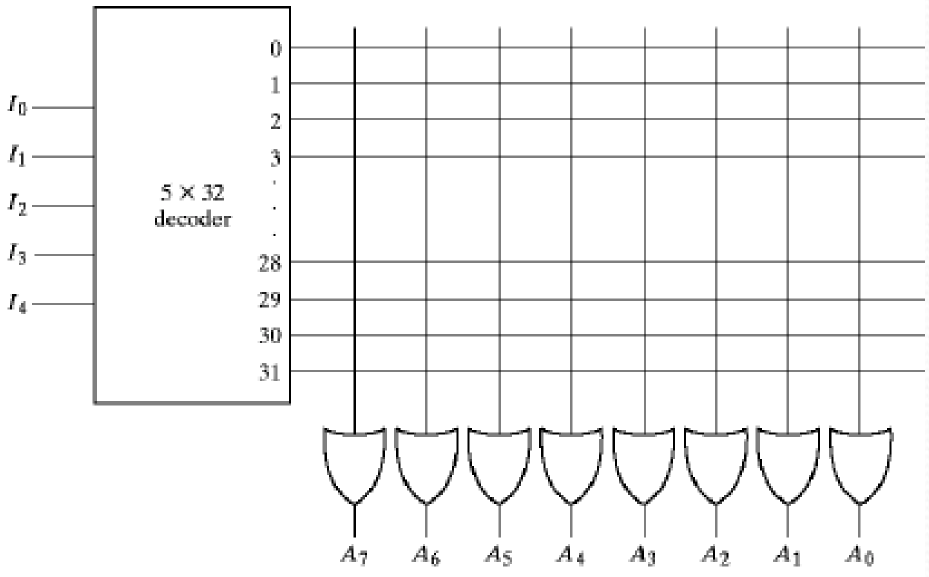
Dronacharya Group of Institutions

Outline

- Introduction
- Random-Access Memory
- Memory Decoding
- Error Detection and Correction
- Read-Only Memory
- Programmable Devices
- Sequential Programmable Devices

Read Only Memory

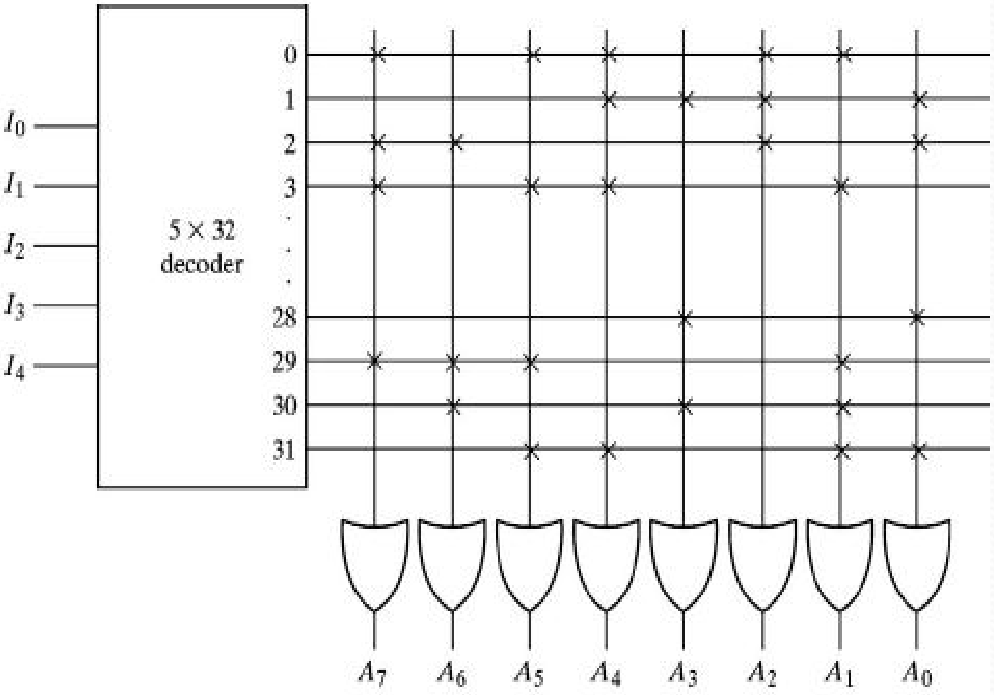
- A memory device that can permanently keep binary data.
- Even when power is turned off and on again.
- For a 2k x n ROM, it consists of
- k inputs (address line) and n outputs (data).
- 2k words of n-bit each
- A k x 2k decoder (generate all minterms).
- n OR gates with 2k inputs
- Initially, all inputs of OR gates and all outputs of the decoder are fully connected.



Internal Logic of a 32 × 8 ROM

Programming the ROM

- Each intersection (crosspoint) in the ROM is often implemented with a fuse
- Blow out unnecessary connections according to the truth table
- 1 means connected (marked as X)
- 0 means unconnected
- Cannot recovered after programmed



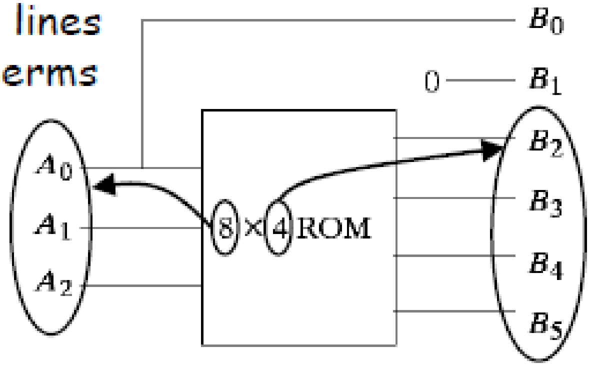
Programming the ROM According to Table 7-3

Design Comb. Circuit with ROM

- Derive the truth table of the circuit
- Determine minimum size of ROM
- Program the ROM

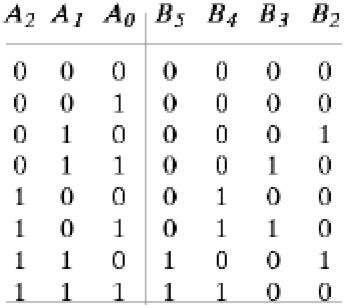
Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

3 select lines
= 8 minterms



word
length
= 4

(a) Block diagram



A_2	A_1	A_0	B_5	B_4	B_3	B_2
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.
- Program the ROM in the semiconductor factory.
- Economic for large quantity of the same ROM.
- Programmable ROM (PROM).
- Contain all fuses at the factory.
- Program the ROM by burning out the undesired fuses (irreversible process).
- Erasable PROM (EPROM).
- Can be restructured to the initial state under a special ultraviolet light for a given period of time.
- Electrically erasable PROM (EEPROM or E2PROM
- Like the EPROM except being erased with electrical signals.

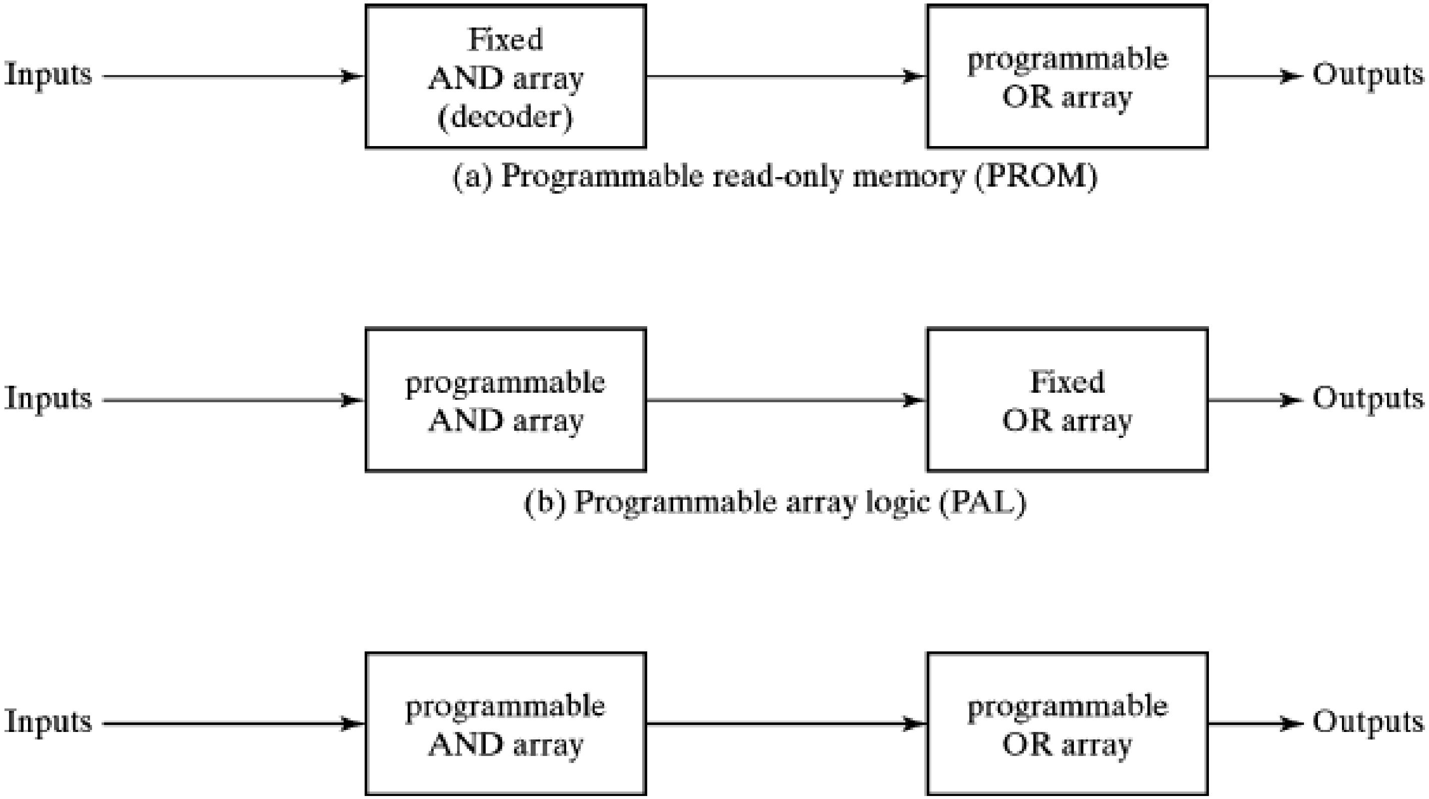
Programmable Logic Devices

- ROM provides full decoding of variables.
- Waste hardware if the functions are given.
- For known combinational functions, Programmable
- Logic Devices (PLD) are often used.
- Programmable read-only memory (PROM)
- Programmable array logic (PAL)
- Programmable logic array (PLA)
- For sequential functions, we can use Sequential (simple) programmable logic device (SPLD)
- Complex programmable logic device (CPLD)
- Field programmable gate array (FPGA)

Outline

- Introduction
- Random-Access Memory
- Memory Decoding
- Error Detection and Correction
- Read only Memory
- Programmable Devices
- Sequential Programmable Devices

Configurations of Three PLDs



(a) Programmable read-only memory (PROM)

(b) Programmable array logic (PAL)

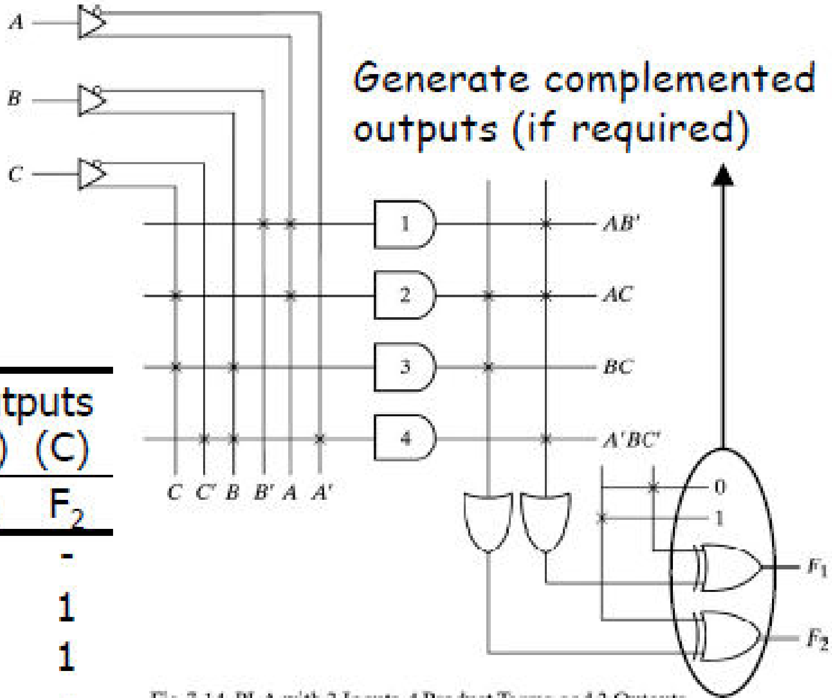
(c) Programmable logic array (PLA)

Basic Configuration of Three PLDs

Programmable Logic Array

- PLA does not provide full decoding of the variables
  - Only generate the terms you need
- The decoder is replaced by an array of AND gates that can be programmed

F1 = AB′ + AC + A′BC′
F2 = (AC +BC)′



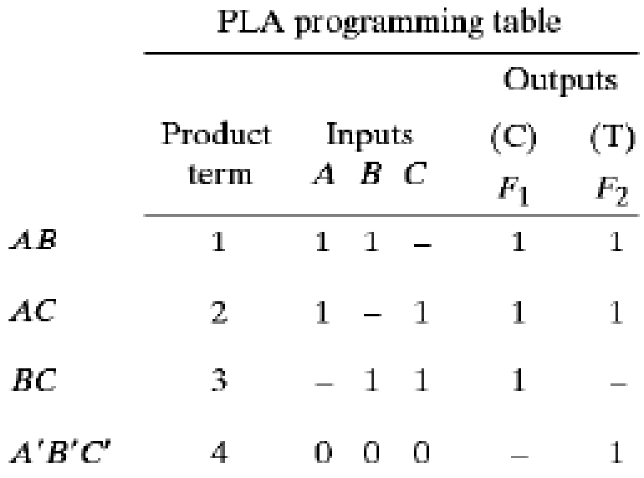
Generate complemented outputs (if required)

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

		Inputs			Outputs	
					(T)	(C)
	Product Term	A	B	C	F_1	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	-

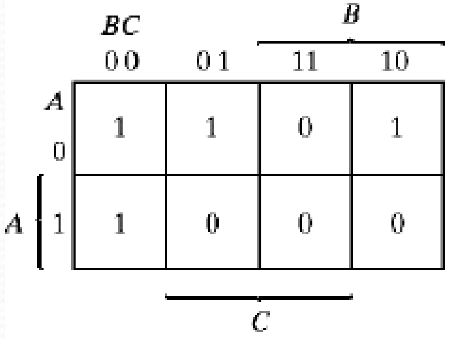
Implementation with PLA

- Implement the two functions with PLA
- F1(A, B, C) = Σ (0, 1, 2, 4)
- F2(A, B, C) = Σ (0, 5, 6, 7)
- Goal: minimize the number of
- **Distinct product terms between** two functions



PLA programming table

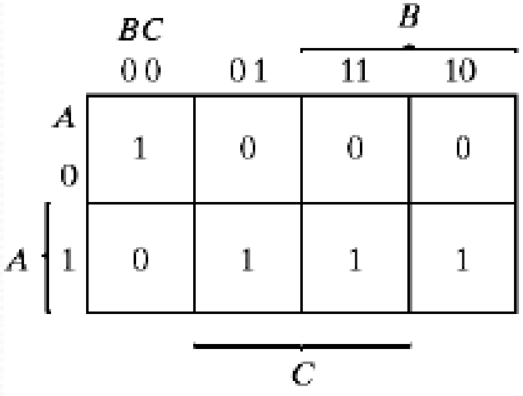
	Product term	Inputs			Outputs	
					(C)	(T)
		A	B	C	F_1	F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1



$A \backslash BC$	00	01	11	10
0	1	1	0	1
1	1	0	0	0

$F_1 = A'B' + A'C' + B'C'$

$F_1 = (AB + AC + BC)'$



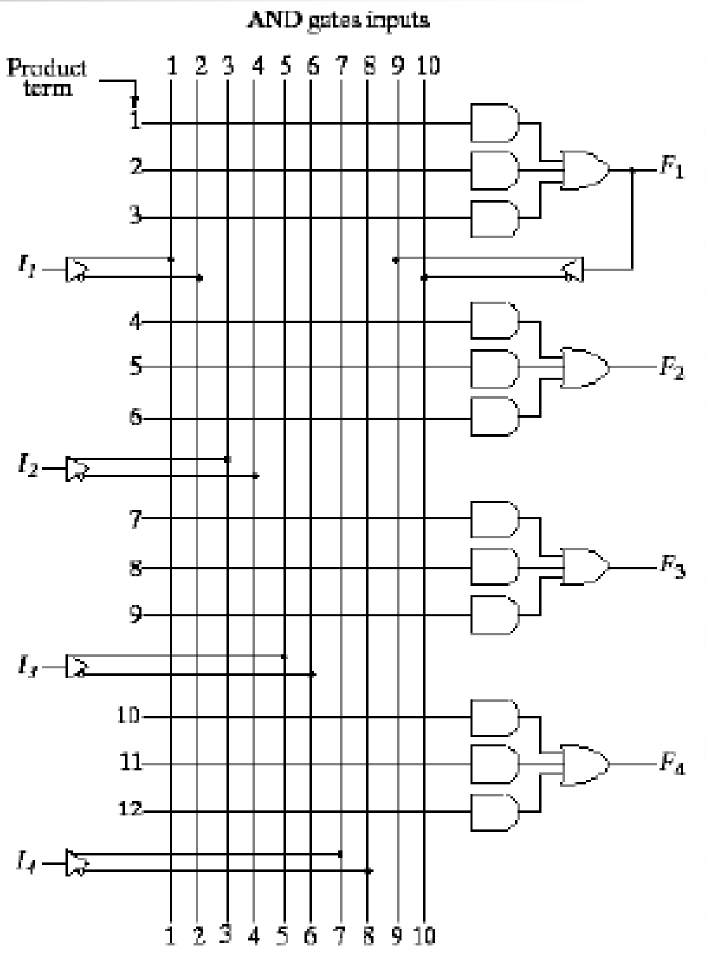
$A \backslash BC$	00	01	11	10
0	1	0	0	0
1	0	1	1	1

$F_2 = AB + AC + A'B'C'$

$F_2 = (A'C + A'B + AB'C')'$

Programmable Array Logic

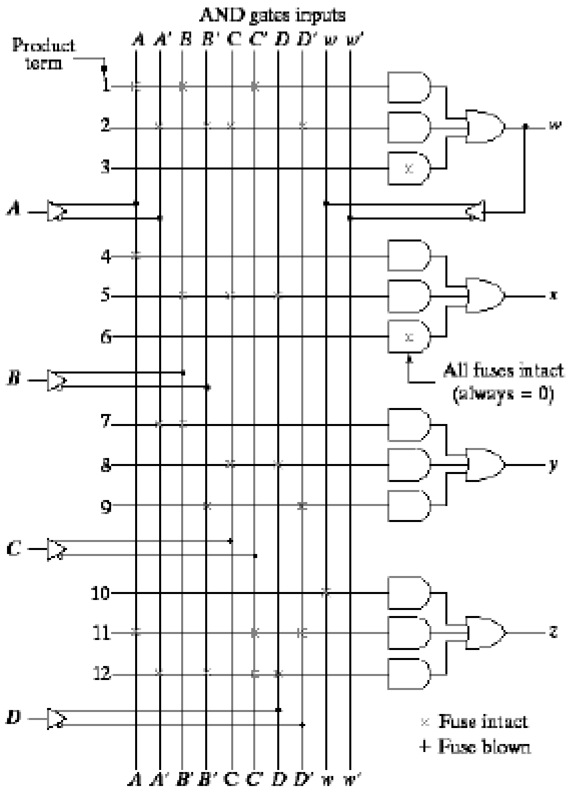
- PAL has a fixed OR array and a programmable AND array.
- Easier to program but not as flexible as PLA.
- Each input has a buffer inverter Gate.
- One of the outputs is fed back as two inputs of the AND gates.
- Unlike PLA, a product term cannot be shared among gates
- Each function can be simplified by itself without common terms.



Implementation with PAL

$w=\Sigma(2,12,13)$ $x=\Sigma(7,8,9,10,11,12,13,14,15)$
$y=\Sigma(0,2,3,4,5,6,7,8,10,11,15)$ $z=\Sigma(1,2,8,12,13)$

Product	AND Inputs					
Term	A	B	C	D	W	Outputs
1	1	1	0	-	-	w = ABC'
2	0	0	1	0	-	+ A'B'CD'
3	-	-	-	-	-	
4	1	-	-	-	-	x = A
5	-	1	1	1	-	+ BCD
6	-	-	-	-	-	
7	0	1	-	-	-	y = A'B
8	-	-	1	1	-	+ CD
9	-	0	-	0	-	+ B'D'
10	-	-	-	-	1	z = W
11	1	-	0	0	-	+ AC'D'
12	0	0	0	1	-	+ A'B'C'D

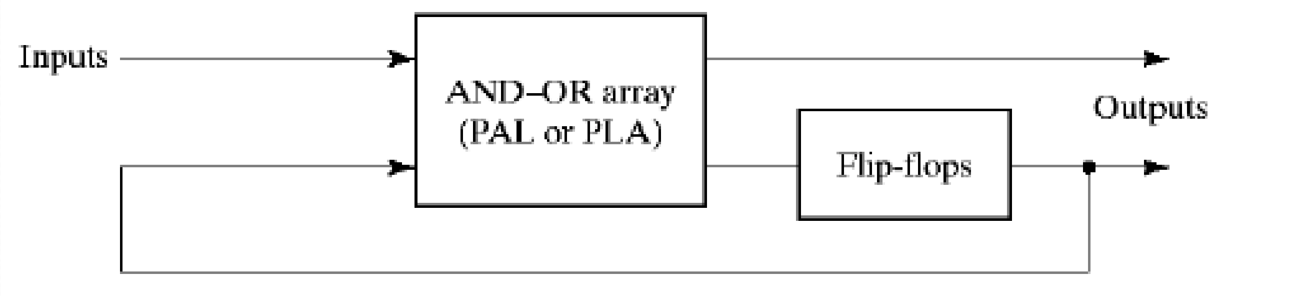


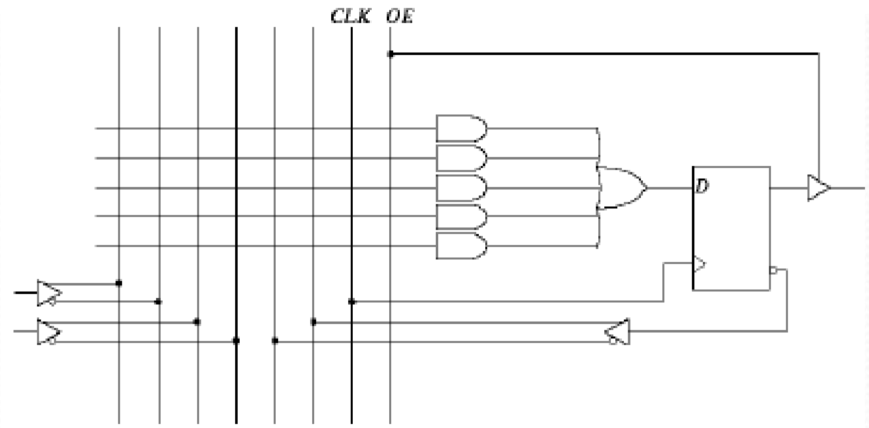
Outline

- Introduction
- Random-Access Memory
- Memory Decoding
- Error Detection and Correction
- Read only Memory
- Programmable Devices
- Sequential Programmable Devices

Sequential PLD

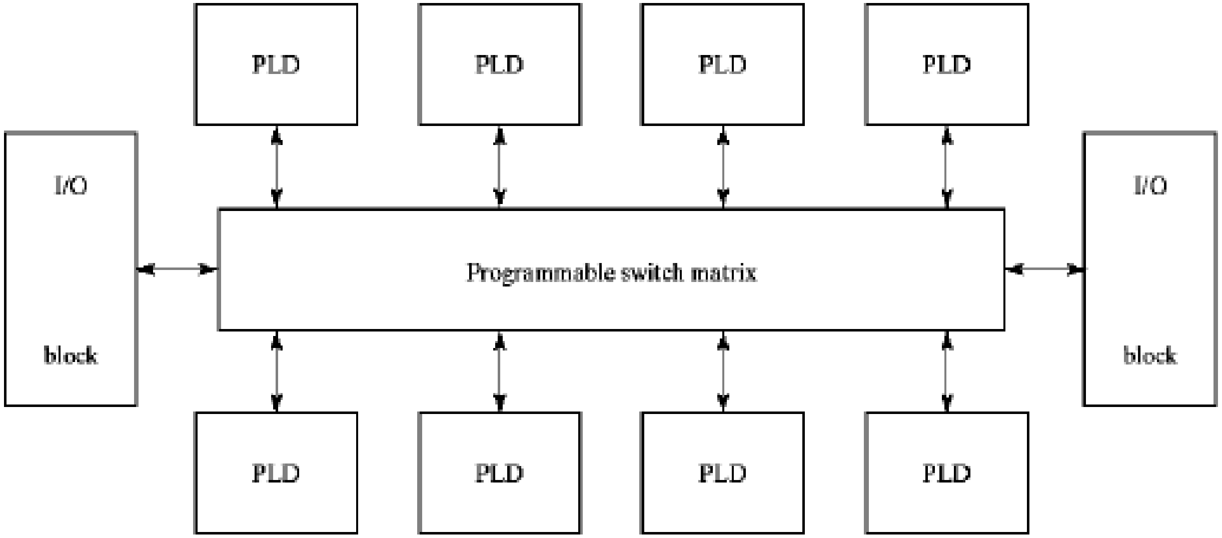
- The most simple sequential PLD = PLA (PAL) + Flip-Flops.
- The mostly used configuration for SPLD is constructed with 8 to 10 macrocells as shown right.





Complex PLD

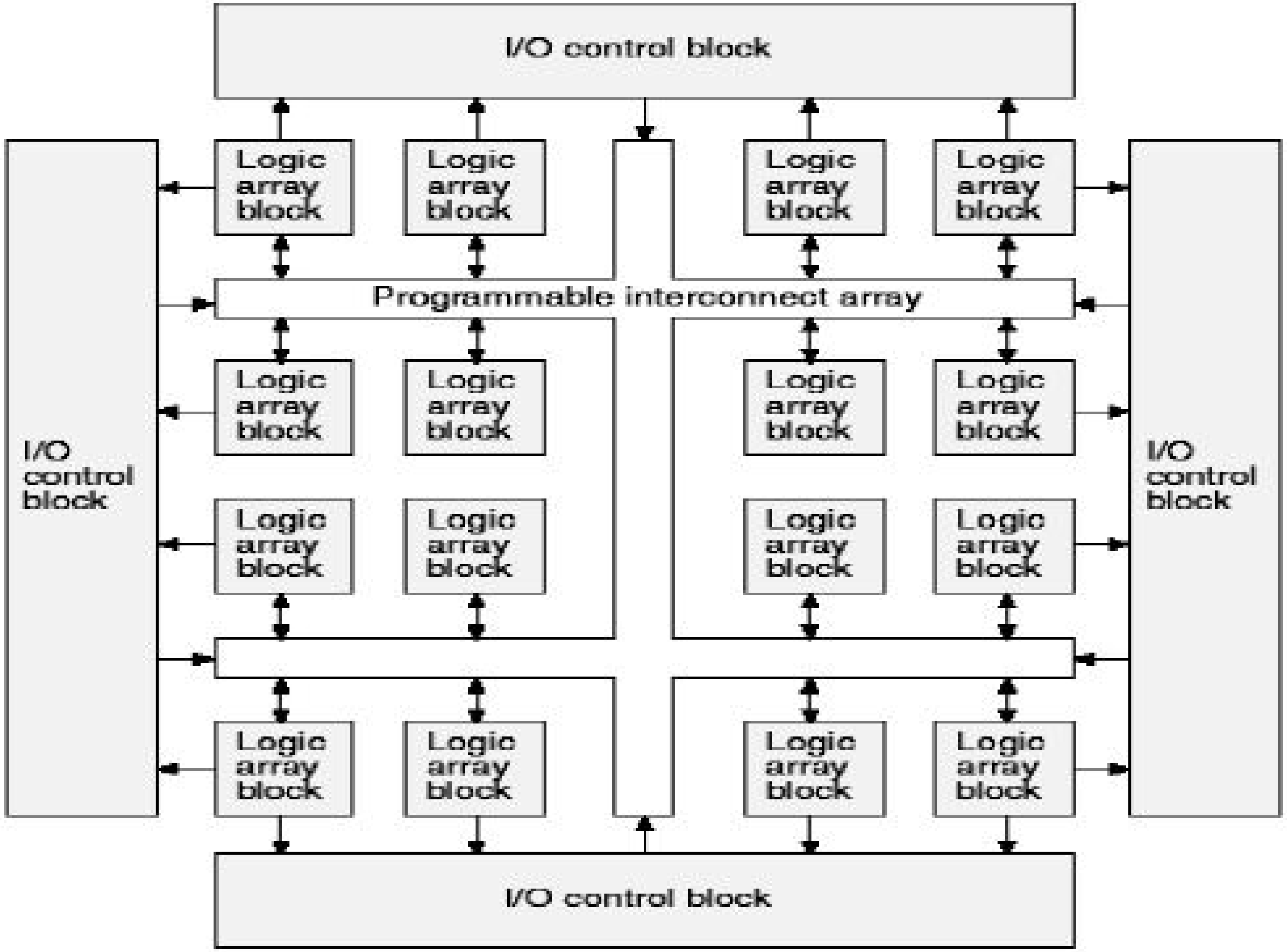
- Complex digital systems often require the connection of several devices to produce the complex specification.
- More economical to use a complex PLD (CPLD).
- CPLD is a collection of individual PLDs on a single IC with programmable interconnection structure.



Field Programmable Gate Array

- Gate array: a VLSI circuit with some pre-fabricated gates repeated thousands of times.
- Designers have to provide the desired interconnection patterns to the manufacturer (factory).
- A field programmable gate array (FPGA) is a VLSI circuit that can be programmed in the user’s location.
- Easier to use and modify
- Getting popular for fast and reusable prototyping
- There are various implementations for FPGA

FPGA Structure (Altera)


I/O control block
I/O control block
Logic array block
Logic array block
Logic array block
Logic array block
Programmable interconnect array
I/O control block
Logic array block
Logic array block
Logic array block
Logic array block
Logic array block
Logic array block
Logic array block
Logic array block
I/O control block
Logic array block
Logic array block
Logic array block
Logic array block
I/O control block