SEE 3243
Digital System

Lecturers :
Muhammad Mun'im Ahmad Zabidi
Muhammad Nadzir Marsono
Kamal Khalil

Week 4:

Programmable Logic Devices (PLDs)

Problems by Using Basic Gates

- Many components on PCB
- As no. of components rise, nodes interconnection complexity grow exponentially
- Growth in interconnection will cause increase in interference, PCB size, PCB design cost, and manufacturing time

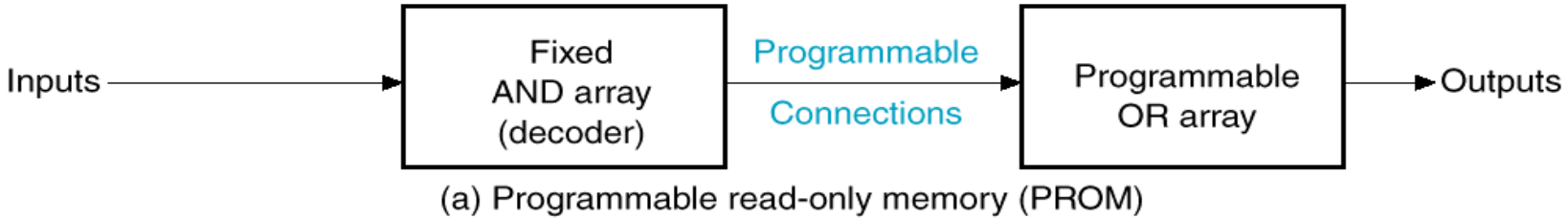
Solution: use PLDs

Types of Programmable Logic Devices

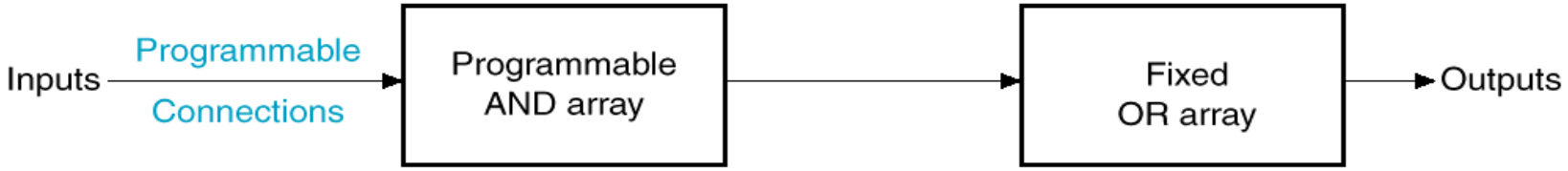
- SPLDs (Simple Programmable Logic Devices)
  - ROM (Read-Only Memory)
  - PLA (Programmable Logic Array)
  - PAL (Programmable Array Logic)
  - GAL (Generic Array Logic)

Device	AND-array	OR-array
PROM	Fixed	Programmable
PLA	Programmable	Programmable
PAL	Programmable	Fixed
GAL	Programmable	Fixed

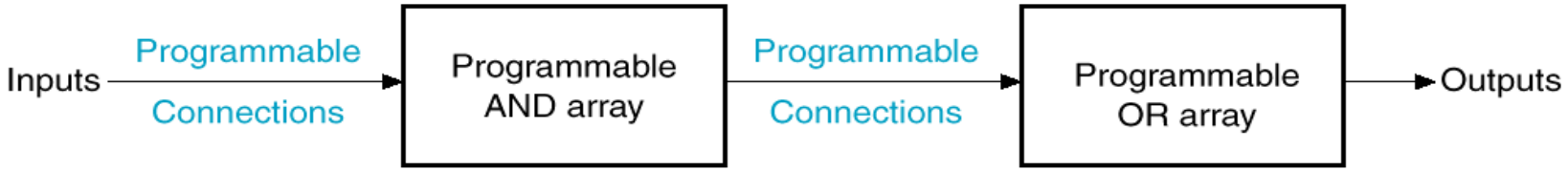
Basic Configuration of Three Simple PLDs



(a) Programmable read-only memory (PROM)



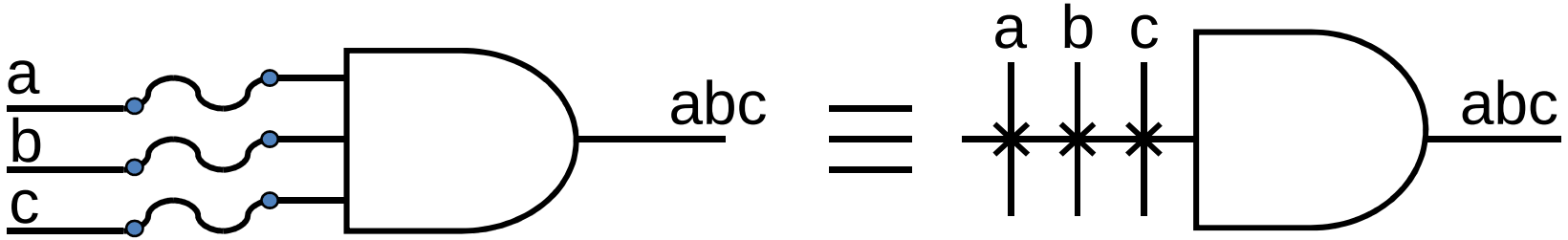
(b) Programmable array logic (PAL) device



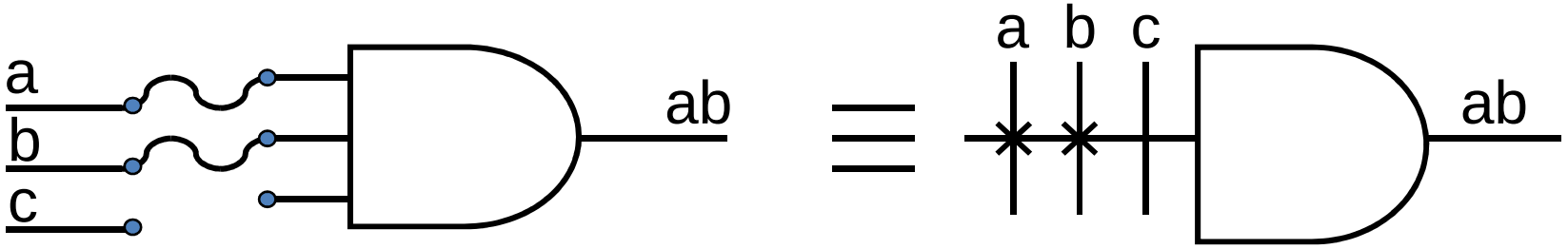
(c) Programmable logic array (PLA) device

AND PLD Notation

Programming is done by blowing the fuse



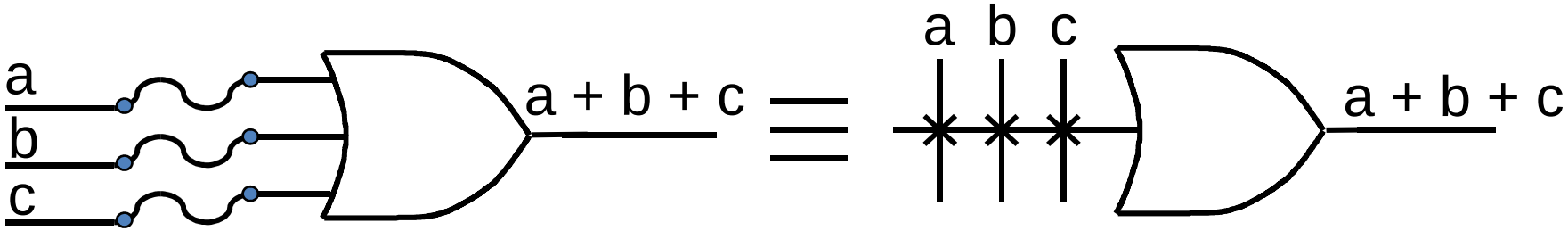
AND gate before programming



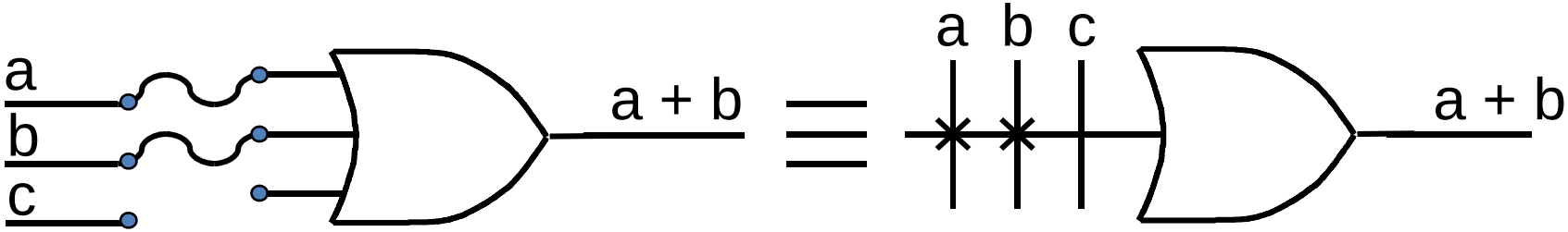
AND gate after programming

OR PLD Notation

Programming is done by blowing the fuse

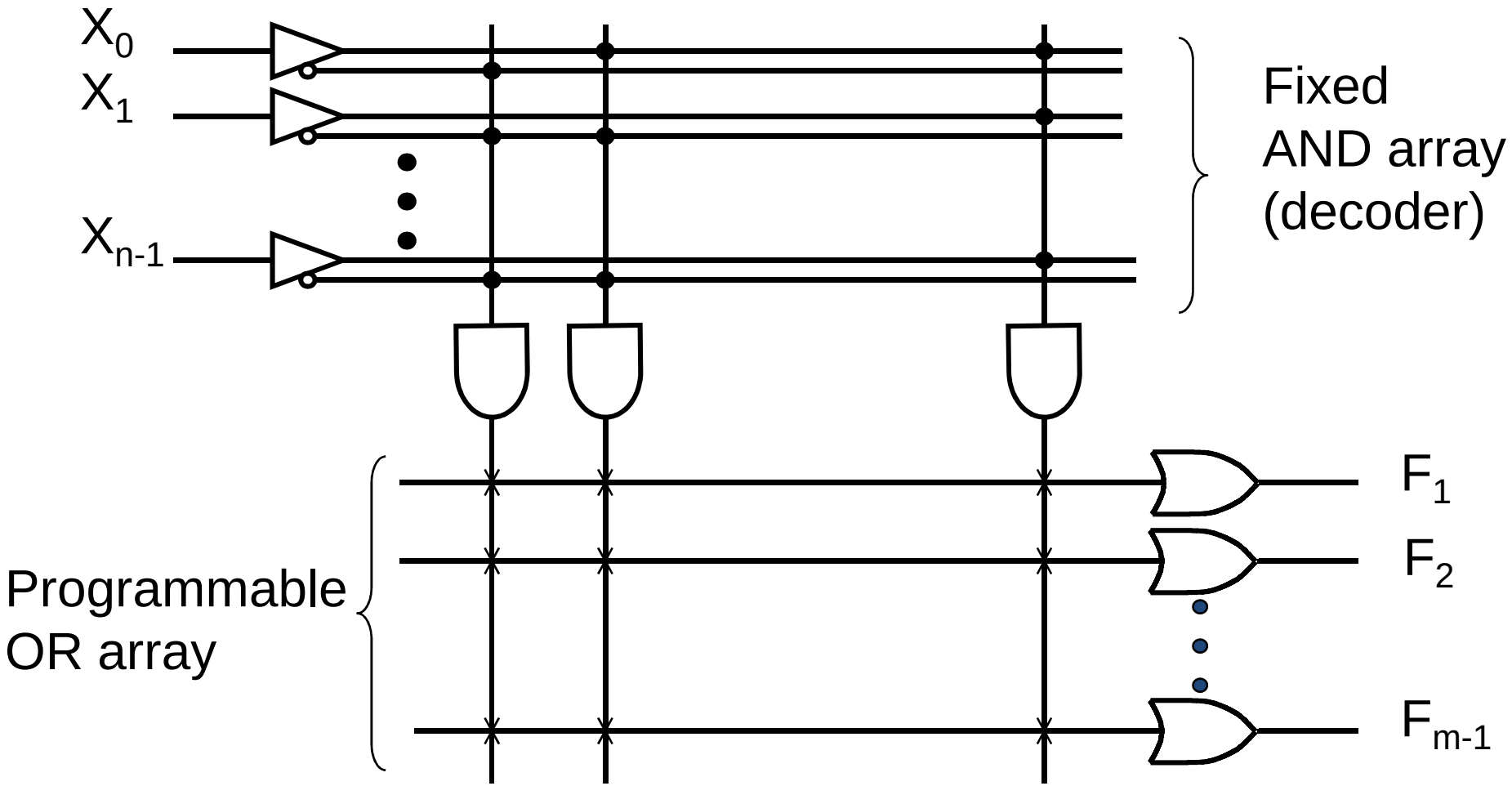


OR gate before programming

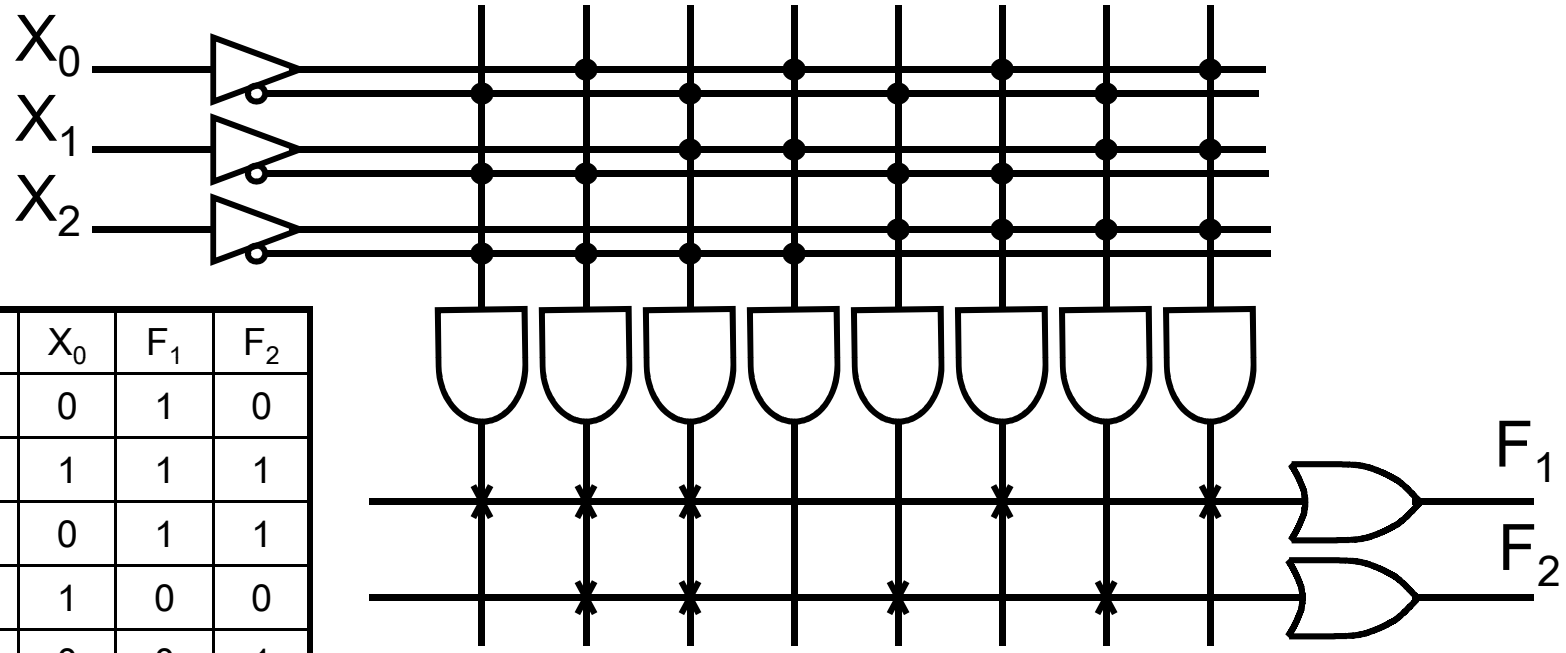


OR gate after programming

PROM Notation

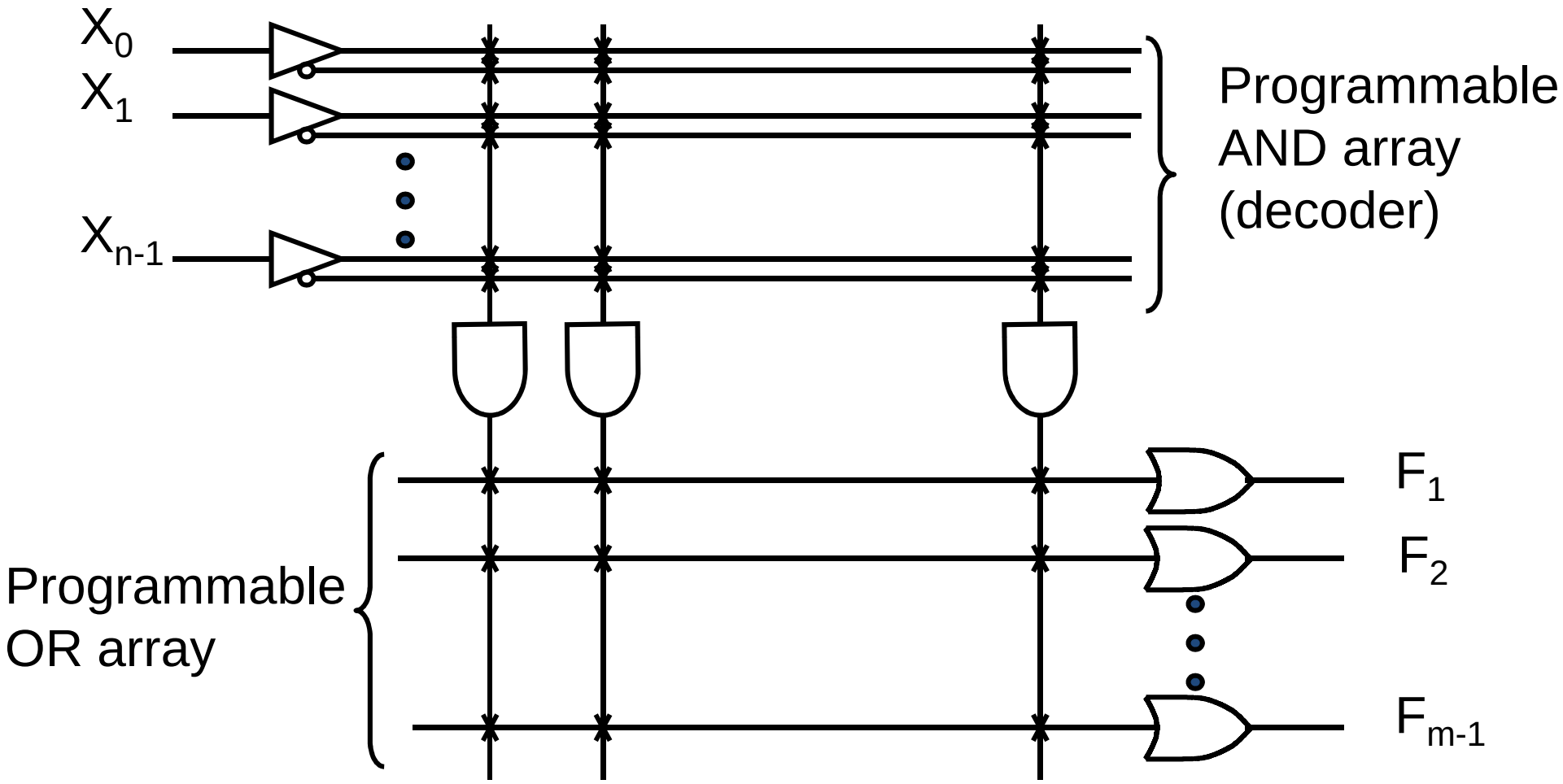


PROM Implementation



X_2	X_1	X_0	F_1	F_2
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

PLA Notation



A virgin PLA has X at all AND and OR junctions

PLA Implementation

X_2	X_1	X_0	F_1	F_2
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

x2 \ x1x0	00	01	11	10
0	1	1	0	1
1	0	1	1	0

$$F_1 = X_2 X_0 + X_2' X_0' + X_2' X_1' X_0$$

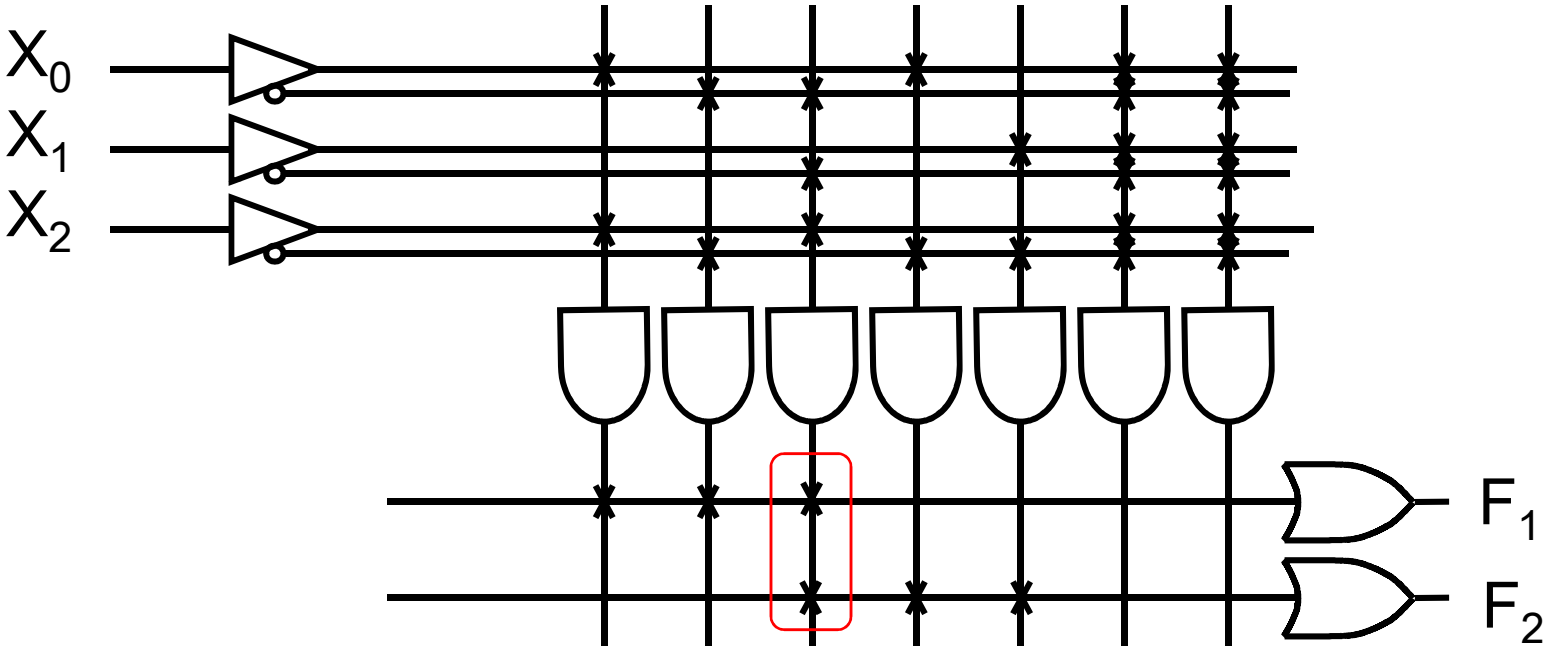
reused

x2 \ x1x0	00	01	11	10
0	0	1	0	1
1	1	0	0	1

$$F_2 = X_2 X_0' + X_1 X_0' + X_2' X_1' X_0$$

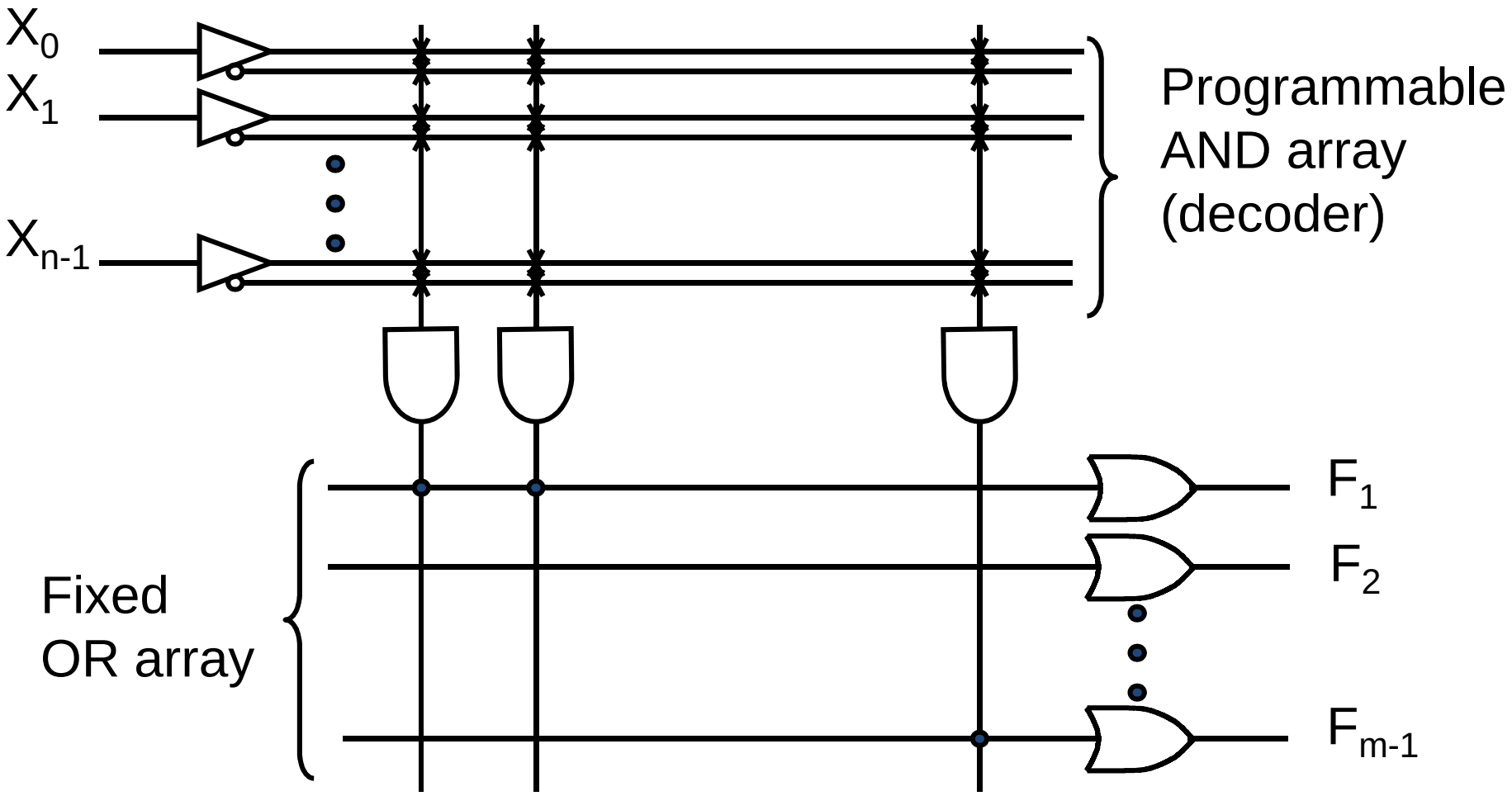
PLA Implementation

Try to implement both functions using minimum number of total terms (product sharing). Hence F_1 has 3-input term.



$F_1 = X_2 X_0 + X_2' X_0' + X_2' X_1' X_0$ $F_2 = X_2' X_1' X_0 + X_2 X_0' + X_1' X_0$

PAL Notation



Each AND gate is permanently connected to a certain OR gate

PAL Implementation

x2	x1	x0	F1	F2
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

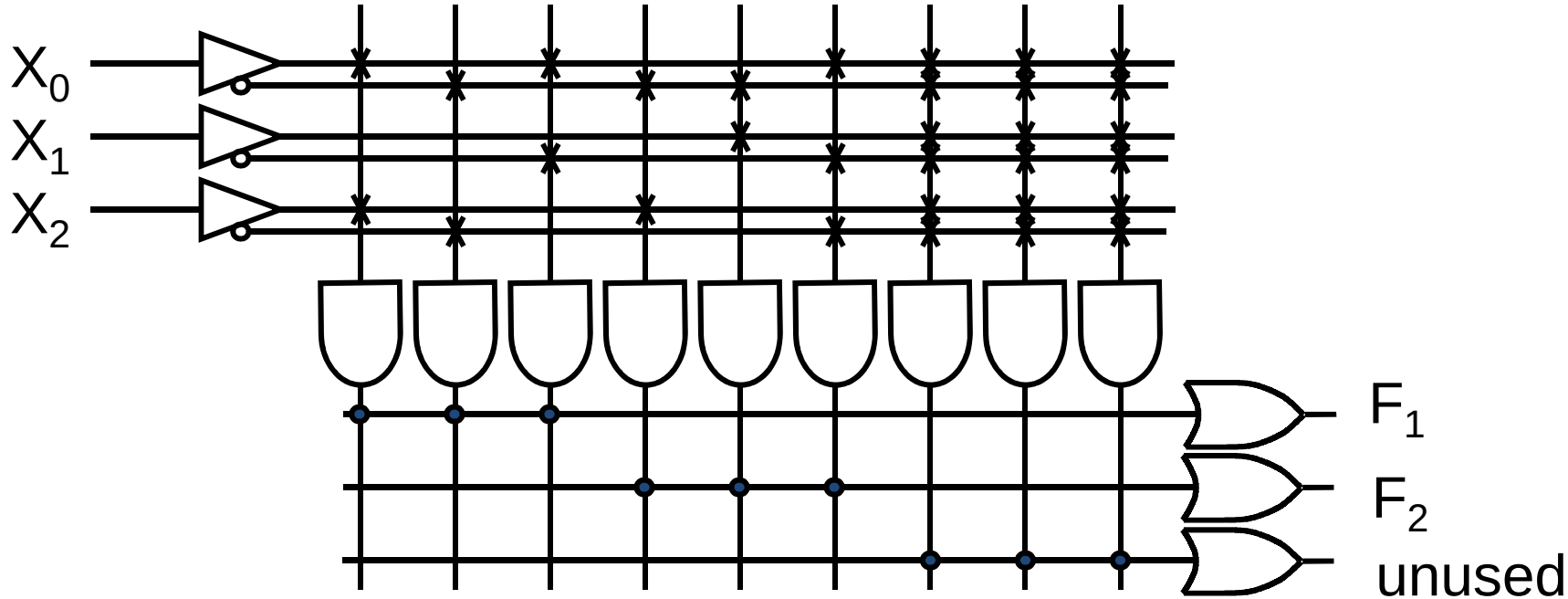
x2 \ x1x0	00	01	11	10
0	1	1	0	1
1	0	1	1	0

$$F_1 = X_2 X_0 + X_2' X_0' + X_1' X_0$$

x2 \ x1x0	00	01	11	10
0	0	1	0	1
1	1	0	0	1

$$F_2 = X_2' X_1' X_0 + X_2 X_0' + X_1' X_0$$

PAL Implementation

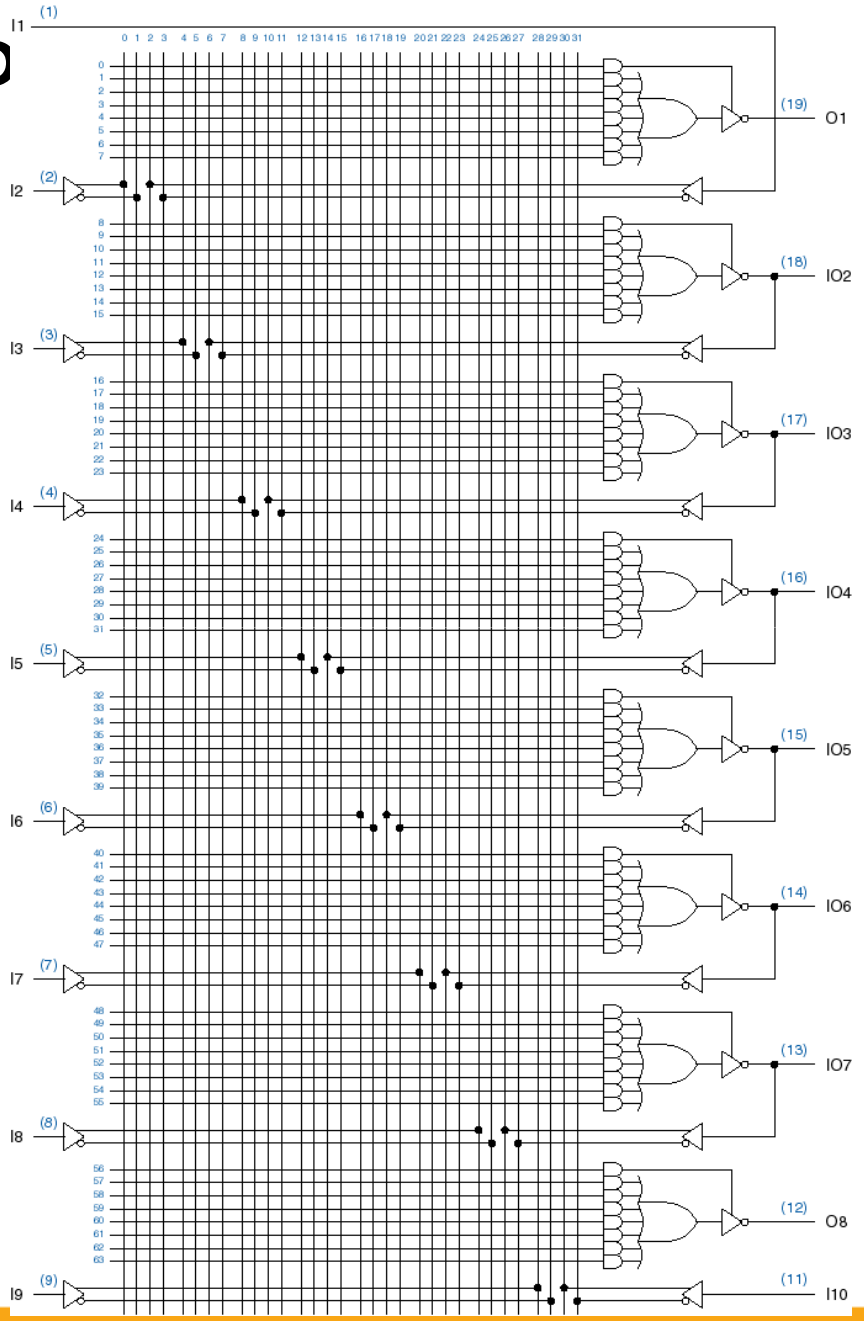


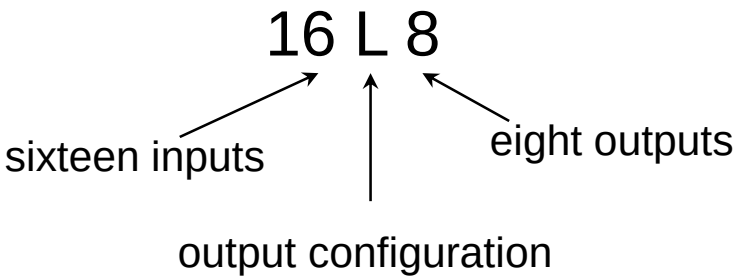
$F_1 = X_2 X_0 + X_2' X_0' + X_1' X_0$

$F_2 = X_2' X_1' X_0 + X_2 X_0' + X_1' X_0$

A Real P

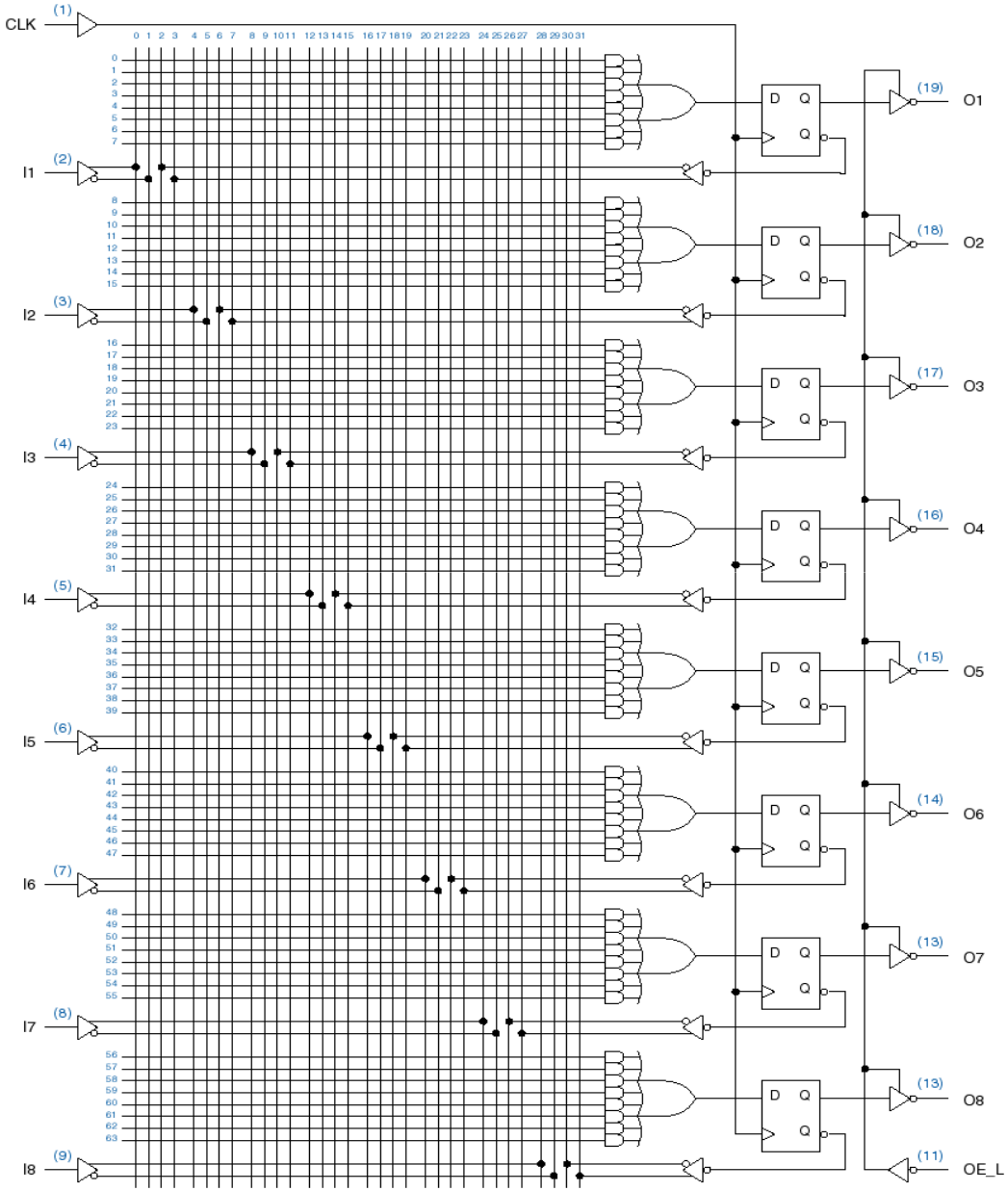
- There’re 64 AND gates
- Each AND gate can have 32 inputs (16 variables & complements)
- There’re 8 outputs
- Don’t count the I/O pins!
- Three common output configs:
  - L = pure combinational
  - R = registered
  - V = macrocell





Sequential PALs

- 16R8
- Notice 8 flip-flops at output



GAL (Generic Array Logic)

- GAL can emulate a number of PALs
- No need for PAL anymore!

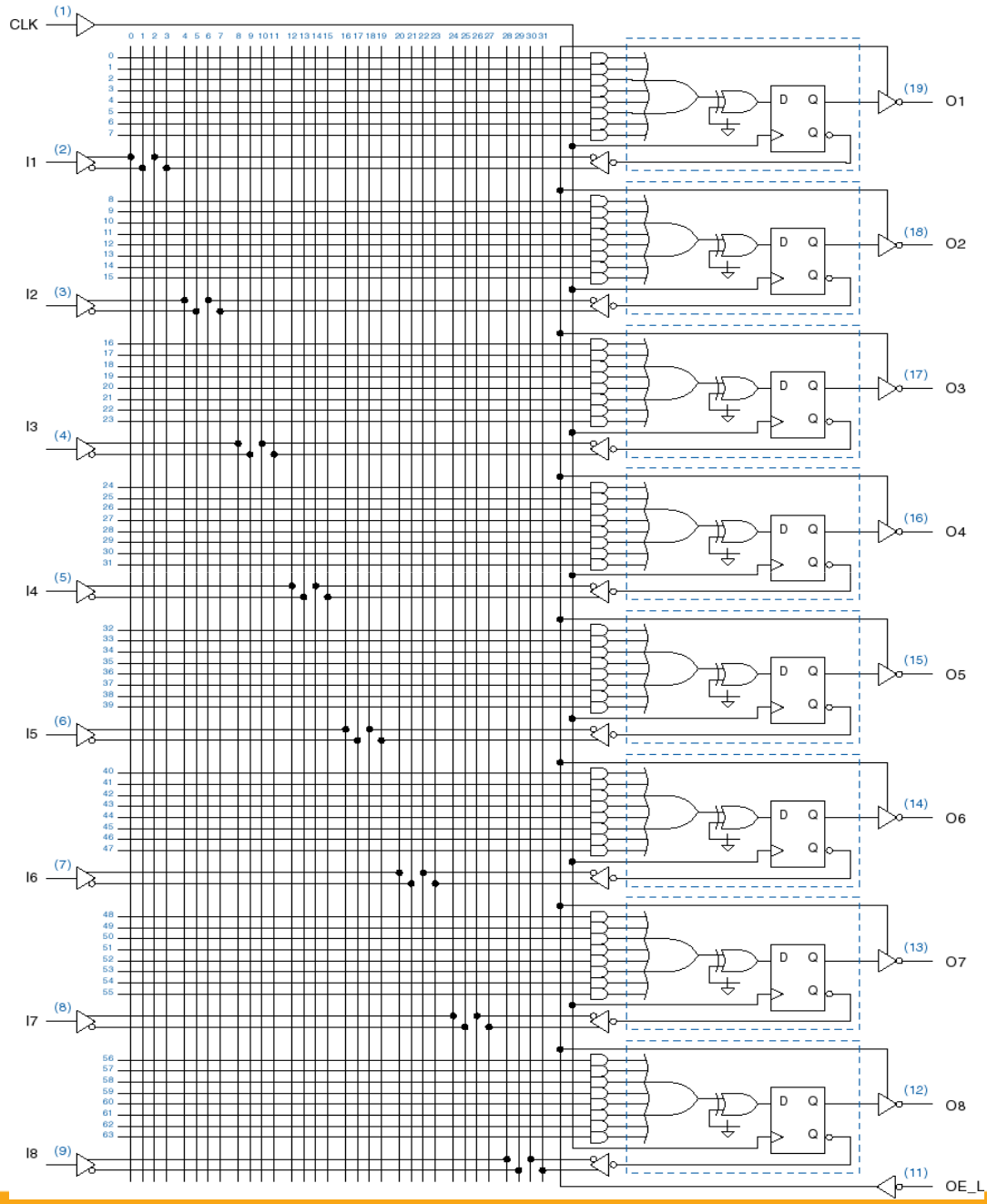
	PAL	GAL
Device technology	Fuse	Electrically erasable cell
Reconfigurability	One-time programmable	Erasable, reprogrammable
I/O	Fixed function	Selectable: input/output, combinational/registered

Oh, by the way…

GAL is trademark of Lattice Semiconductor: example GAL22V10
Universal PAL is the trademark of Vantis: example PALCE22V10 (but it's the same thing!)

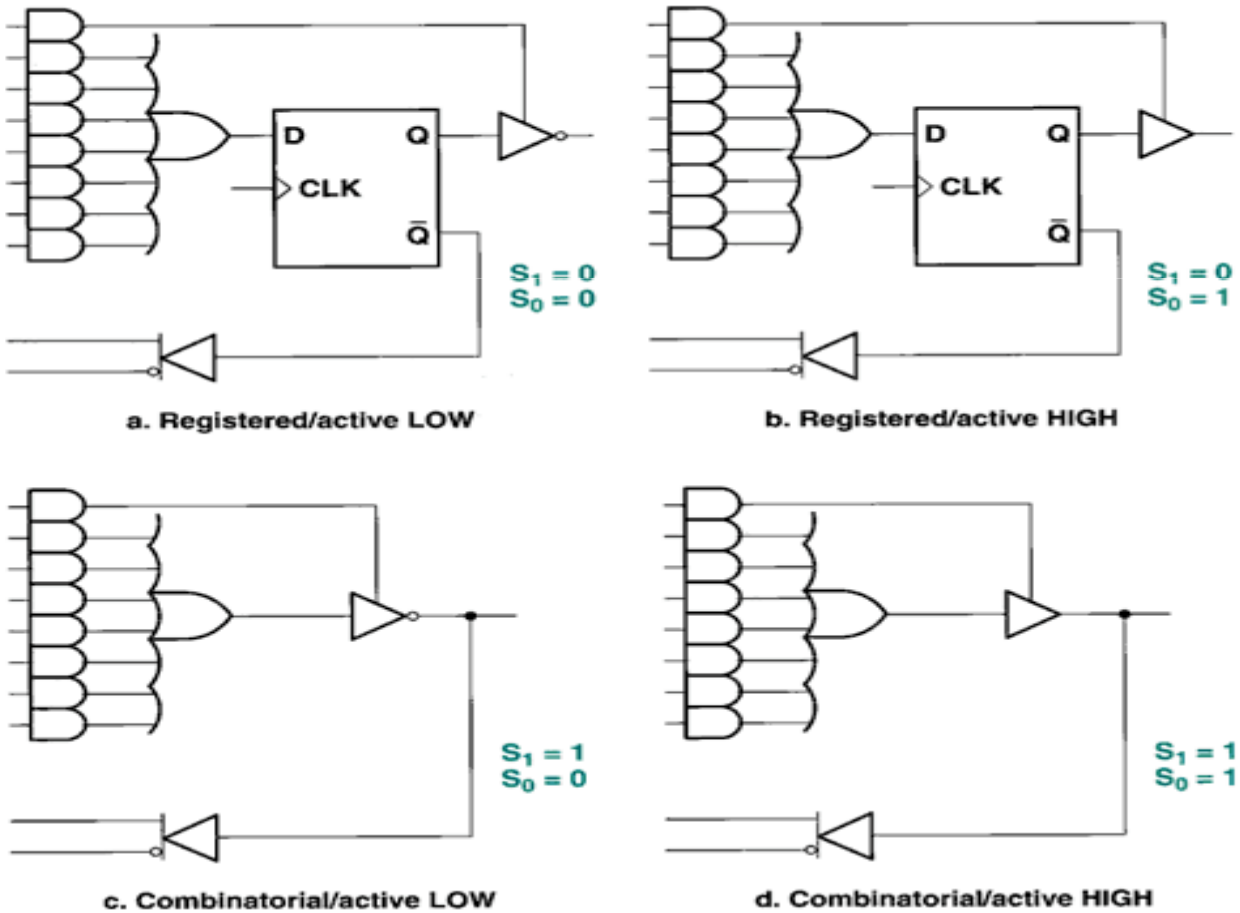
GAL16V8

- Each output is programmable as combinational or registered
- Also has programmable output polarity

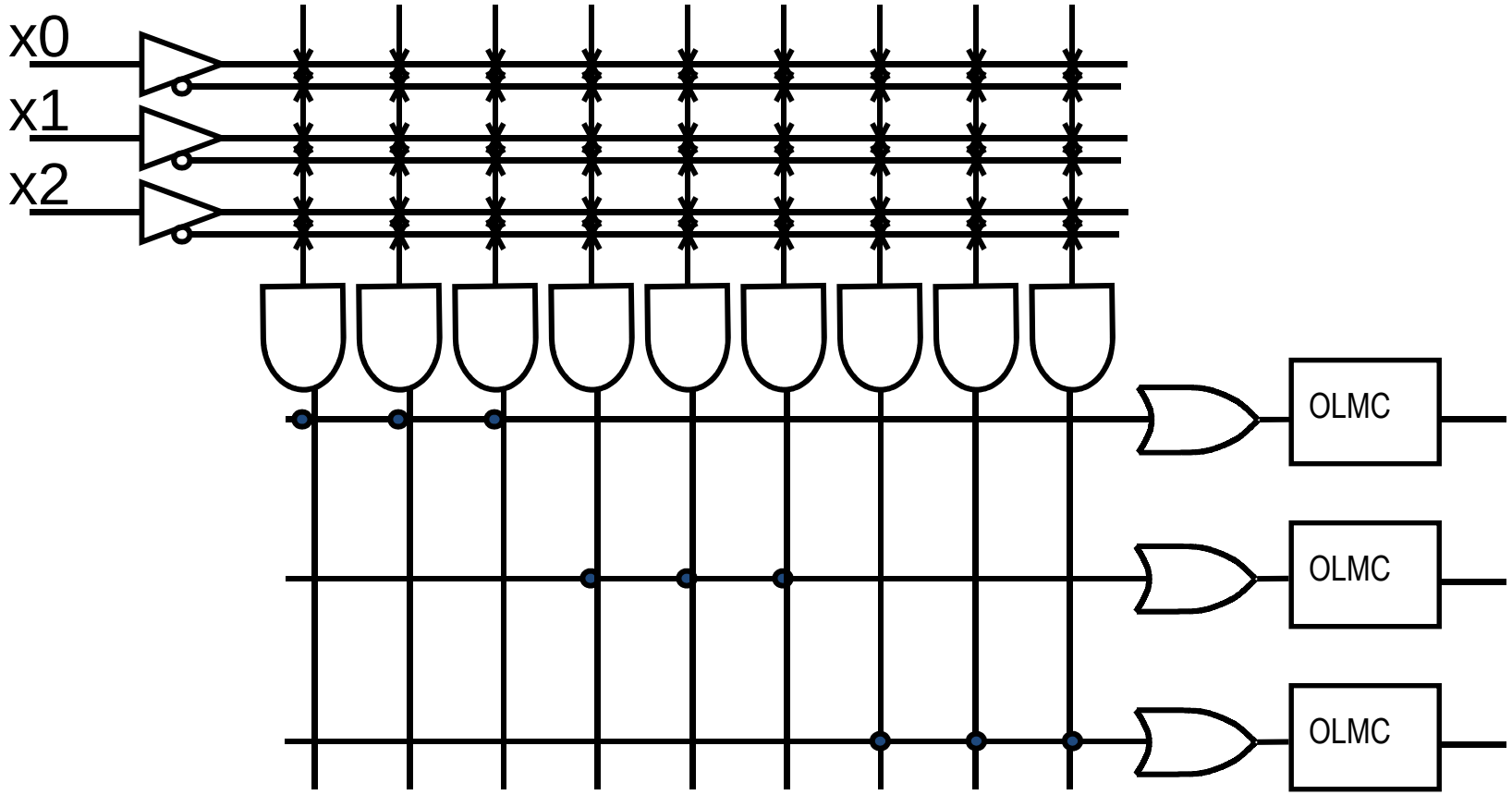


OLMC

- OLMC = Output Logic Macrocell
- Simplified, equivalent circuits:



Simplified GAL Representation



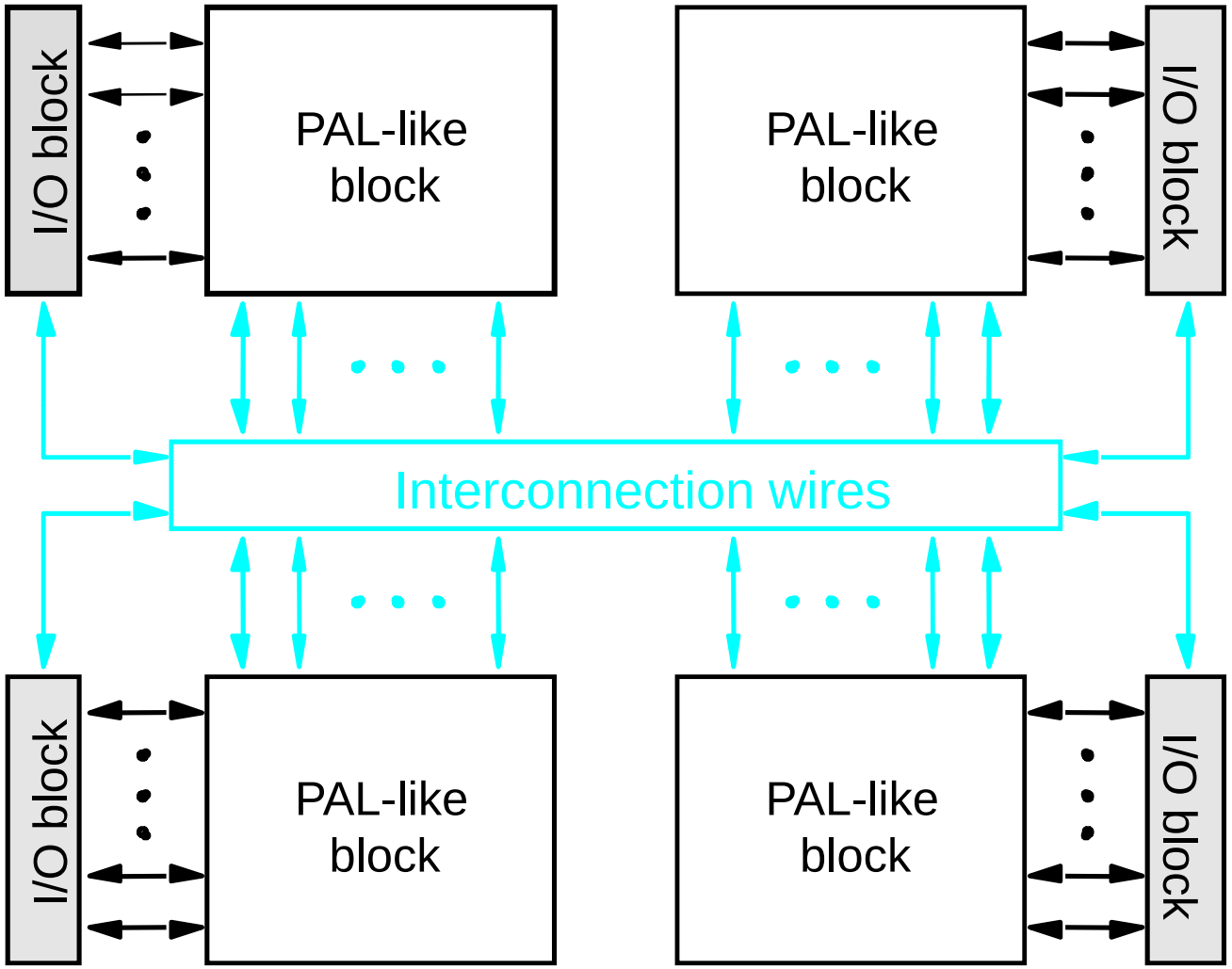
PLD programming unit (courtesy of Data IO Corp)

Problems Using SPLD

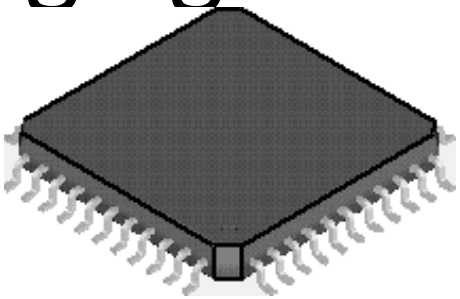
- Current trend is
  - Increasing gate count
  - Increase design complexity
  - Requirement for smaller size due to lower cost, lower power and higher reliability
  - Fast prototyping for quick design verification
  - PROM, PLA and PAL not used much except in small designs!

- Solution:
  - CPLD for intermediate complexity
  - FPGA for very complex designs (up to millions of gates)

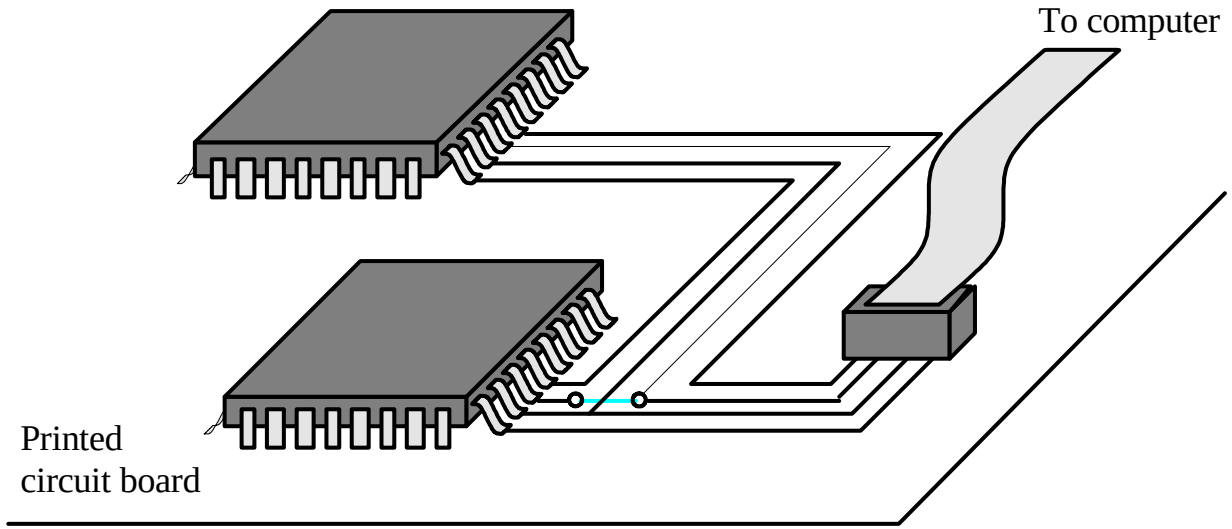
Structure of CPLD



CPLD Packaging and Programming



(a) CPLD in a Quad Flat Pack (QFP) package



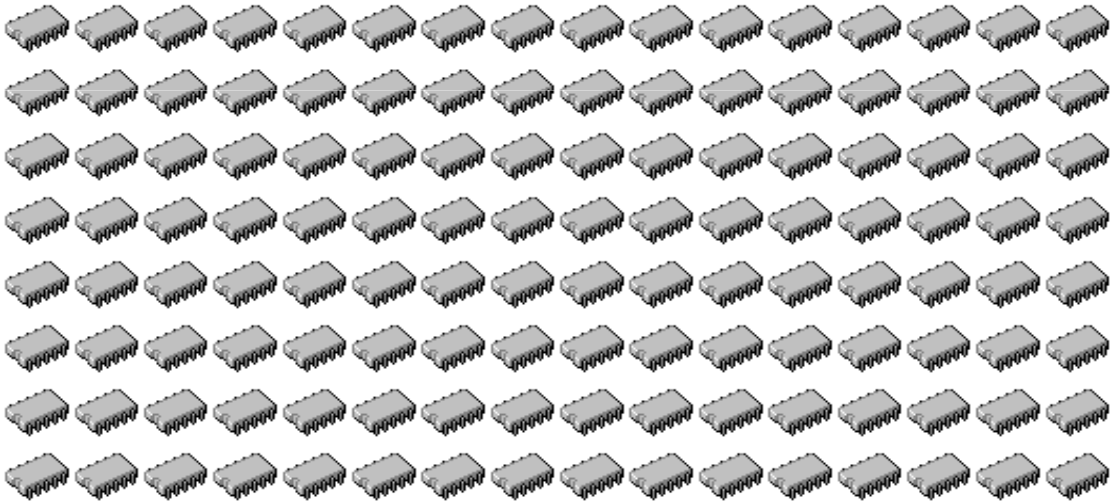
(b) JTAG programming

Advantages of CPLDs

- Reduced circuit board space utilization, and significant cost savings

A 44-pin CPLD is equivalent to 600 gates

=



Some CPLDs have gate equivalents in the millions and over 1000 pins

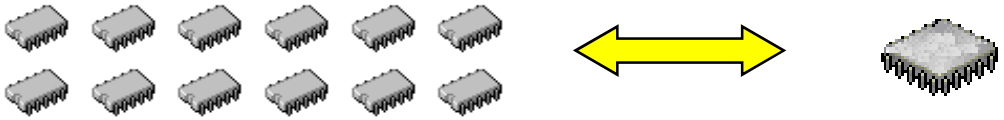
Advantages of CPLDs

- Easiest to modify
  - Use appropriate CAD tools

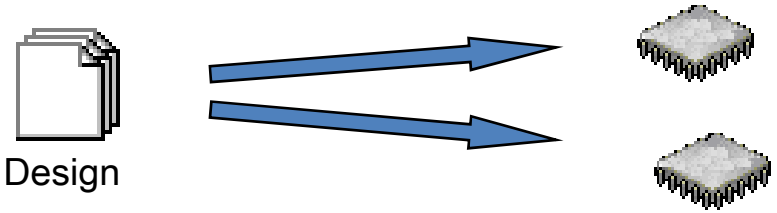
- Direct entry of conceptual design into functional circuit
  - Streamlined design to prototype process

Advantages of CPLDs

- Universal inventory, as one IC can be programmed for various applications

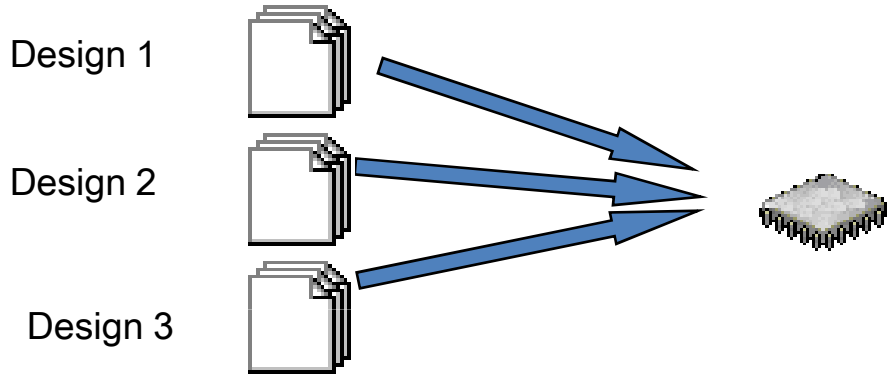


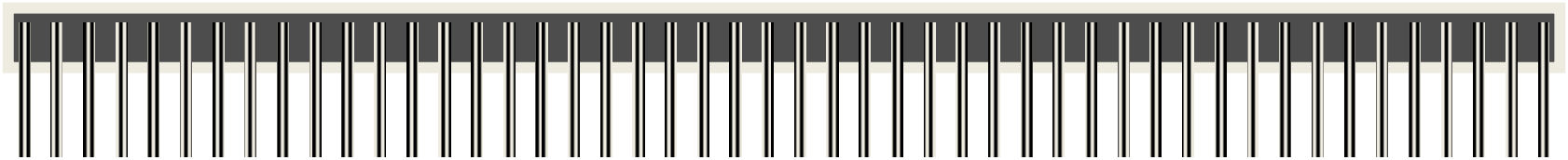
- Easy to duplicate



Advantages of CPLDs

- Reprogrammability
  - CPLD can be reprogrammed hundreds of times.



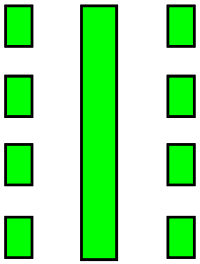
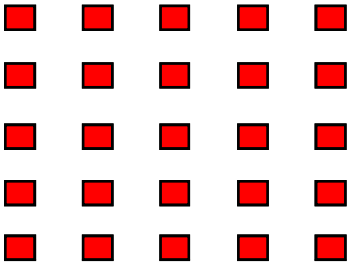


- Number of I/Os
  - CPLDs have large amounts of programmable input/output contacts.

Advantages of CPLDs

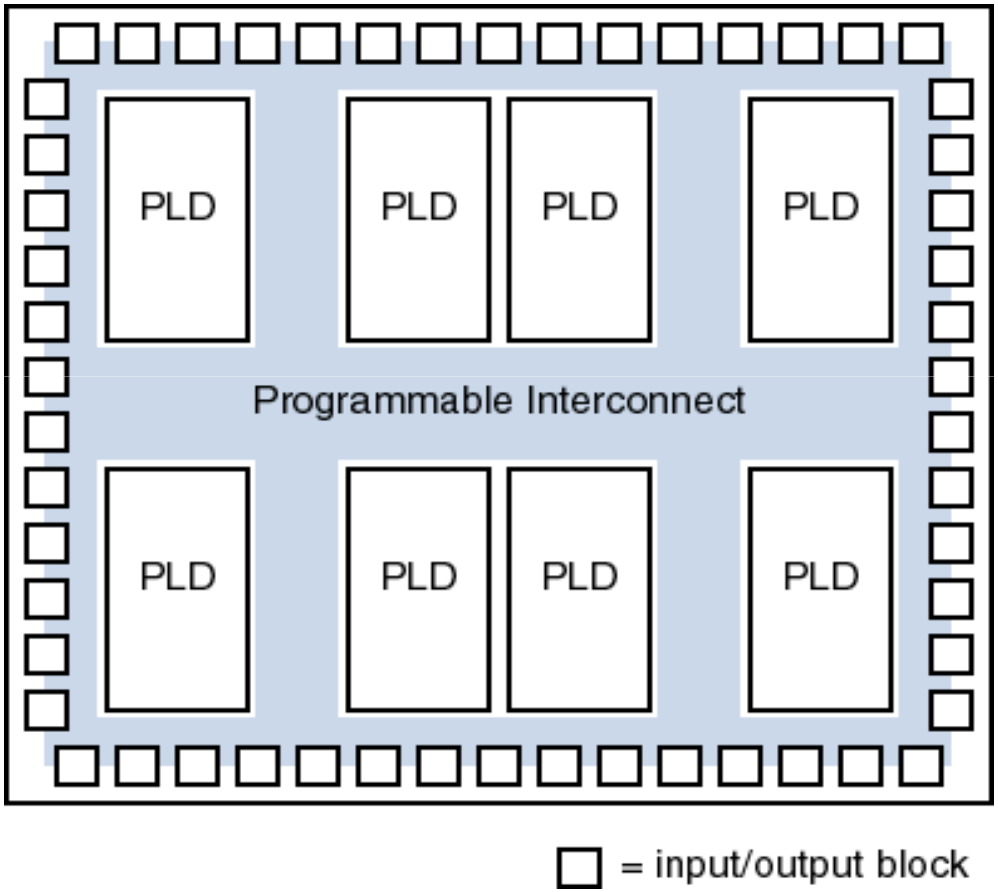
- **Software and Language**
  - **Manufacturers of CPLDs supply design software (software like Altera Quartus II is a free download)**
  - **VHDL is a standards-based language that most manufacturers conform to**
- **Simple Interface**
  - **Devices can be interfaced directly to a computer with a serial or USB connection**
- **In-circuit modifications**
  - **With the proper interface connections, the CPLD logic can be edited in-circuit**
- **Transportable design**
  - **As a designer you may easily exchange your designs and design modifications (e-mail, CD, web site, etc. …)**

CPLDs and FPGAs

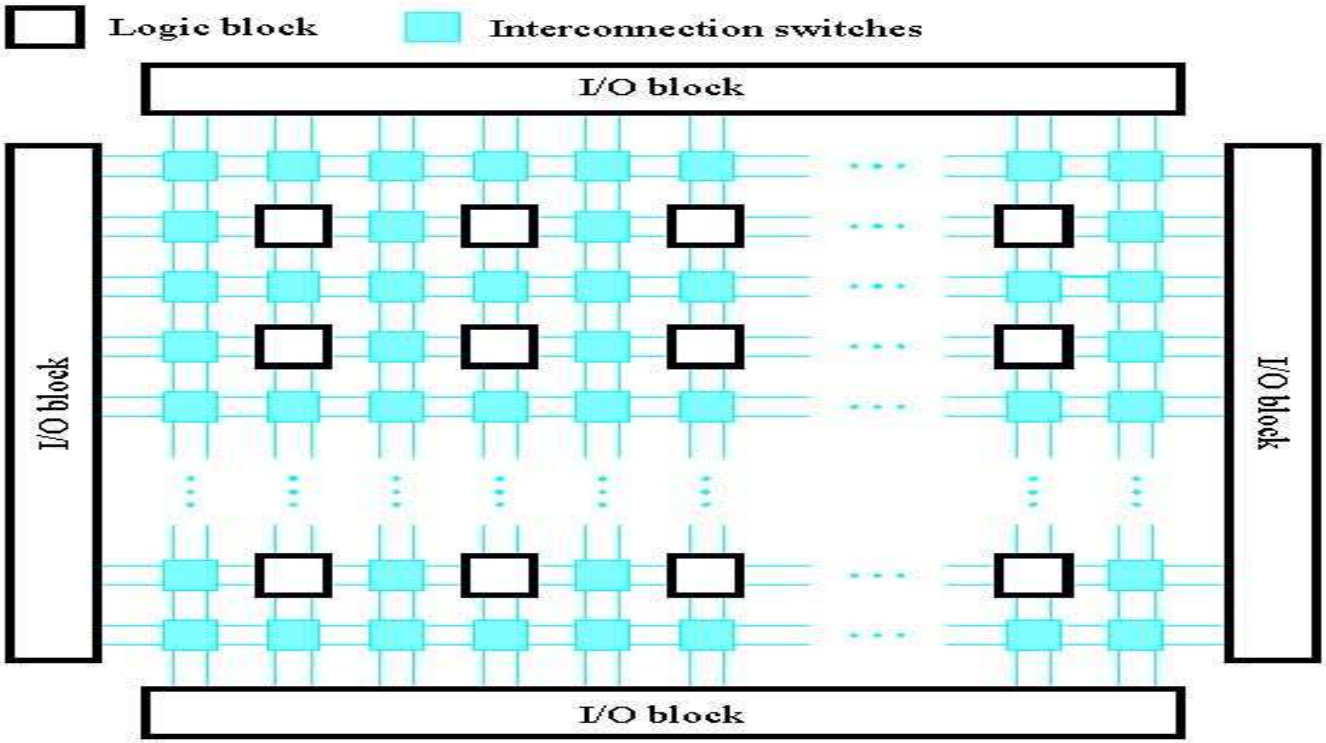
	CPLD	FPGA
	Complex Programmable Logic Device	Field-Programmable Gate Array
		
Architecture	PAL/22V10-like More Combinational	Gate array-like More Registers + RAM
Density	Low-to-medium 0.5-10K logic gates	Medium-to-high 1K to 1M system gates
Performance	Predictable timing Up to 250 MHz today	Application dependent Up to 150 MHz today
Interconnect	“Crossbar Switch”	Incremental

CPLDs vs. FPGAs

- CPLD architecture



FPGA architecture



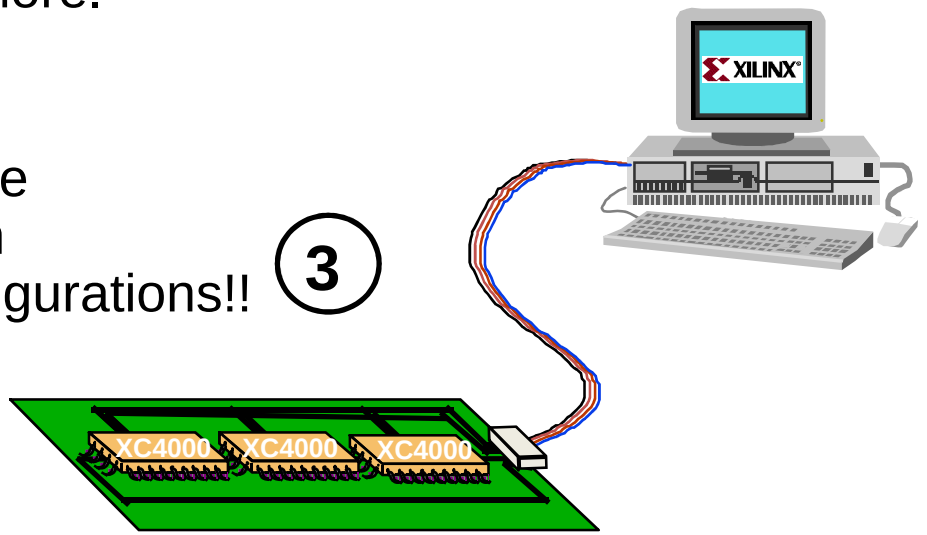
- Much larger number of smaller programmable logic blocks.
  - Xilinx calls them CLB (Configurable Logic Block)
  - Altera calls them LAB (Logic Array Block) and EAB (Embedded Array Block)
- Embedded in a sea of lots and lots of programmable interconnects.

Design Flow

1. **Design Entry** in schematic and/or HDL (ABEL, VHDL, Verilog). Vendors include Altera, Synopsys, Aldec (Xilinx Foundation), Mentor Graphics, Cadence, Viewlogic, etc.

2. **Implementation** includes Placement & Routing. Also, analyze timing, view layout, and more.

3. **Download** directly to the hardware device(s) with almost unlimited reconfigurations!!



Development Process