CMPE 212
Principles of Digital Design

Lecture 17

Programmable Logic Devices

March 28, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

Lecture's Overview

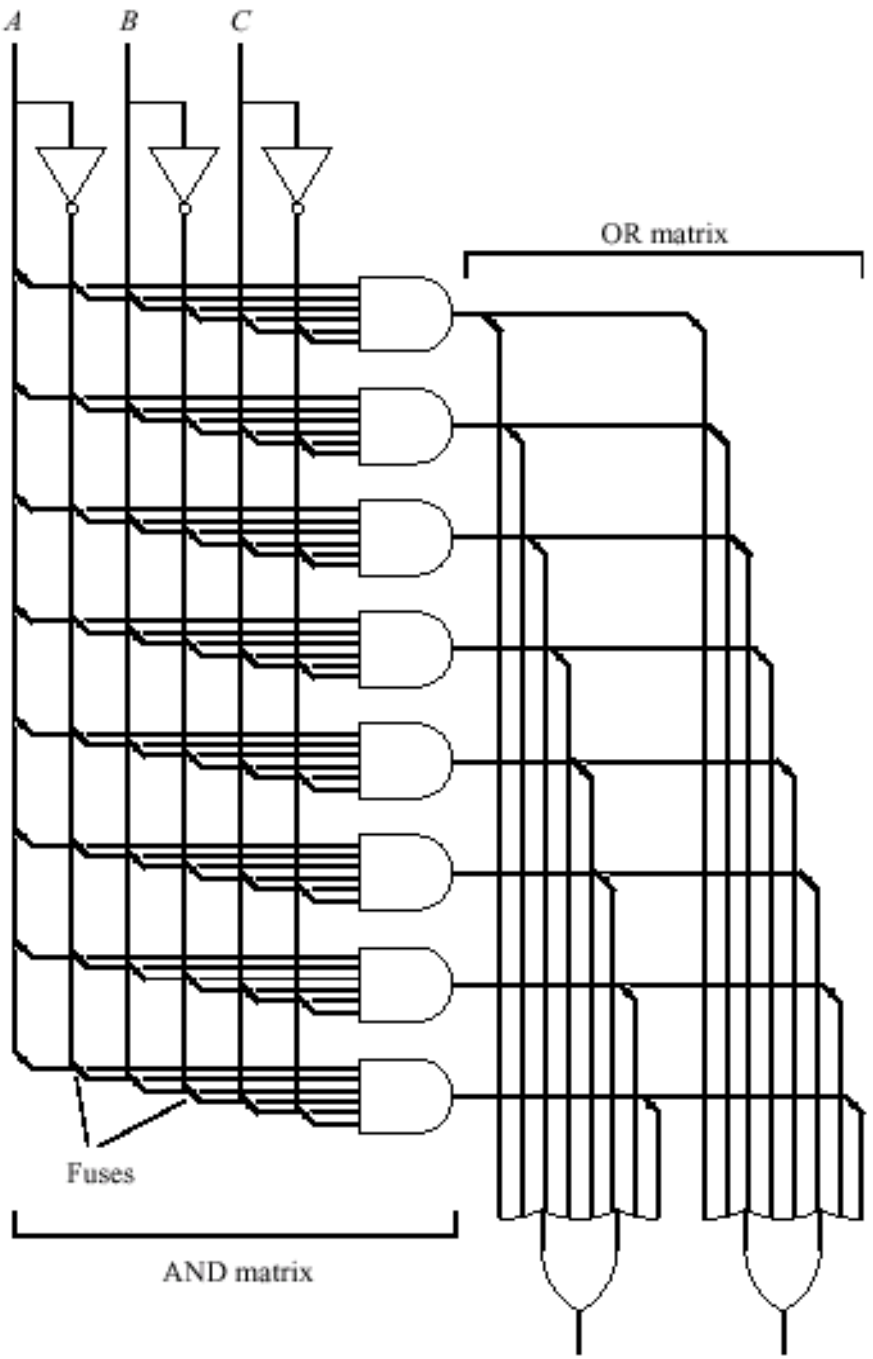
- ***Previous Lecture:***
  - ➔ Modular Combinational Logic
  - ➔ Examples of medium scale integration components (Multiplexers, Decoders, Encoders, etc.)
  - ➔ Designing with digital components (Tree-type arrangements, implementing switching functions)
  - ➔ Binary ripple-carry adders
- ***This Lecture***
  - ➔ Programmable logic devices
  - ➔ Logic using Diode Biasing
  - ➔ Programmable devices and how to configure them
  - ➔ PLA, PAL, ROM and FPGA devices

Design Options

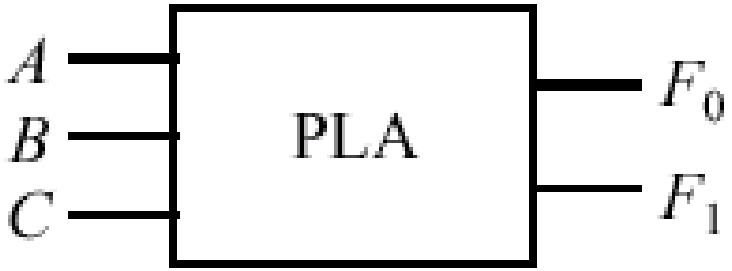
- Custom Integrated Circuits: Design and realize using a library modules on a single integrated circuit
  - Complete control over all circuit parameters (increases speed, reduces size and power consumption)
  - Most expensive and slowest to design and manufacture
- Semi-custom ICs: Use some pre-designed components, e.g.,
  - *Standard Cell libraries*
    - Use pre-designed units, just place and interconnect
    - All processing steps need to be done (same as fully costumed design)
  - *Generic gate arrays*
    - Integrated circuits with many unconnected gates
    - Flexible interconnections to enable implementation
    - Large volume enables commodity pricing and off-the-shelf availability
    - Examples are programmable logic arrays (PLA), programmable logic devices (PLD), and field programmable gate arrays (FPGAs)

Programmable Logic Array

- A PLA is a customizable AND matrix followed by a customizable OR matrix.
- Every intersection has a switch that controls the connection

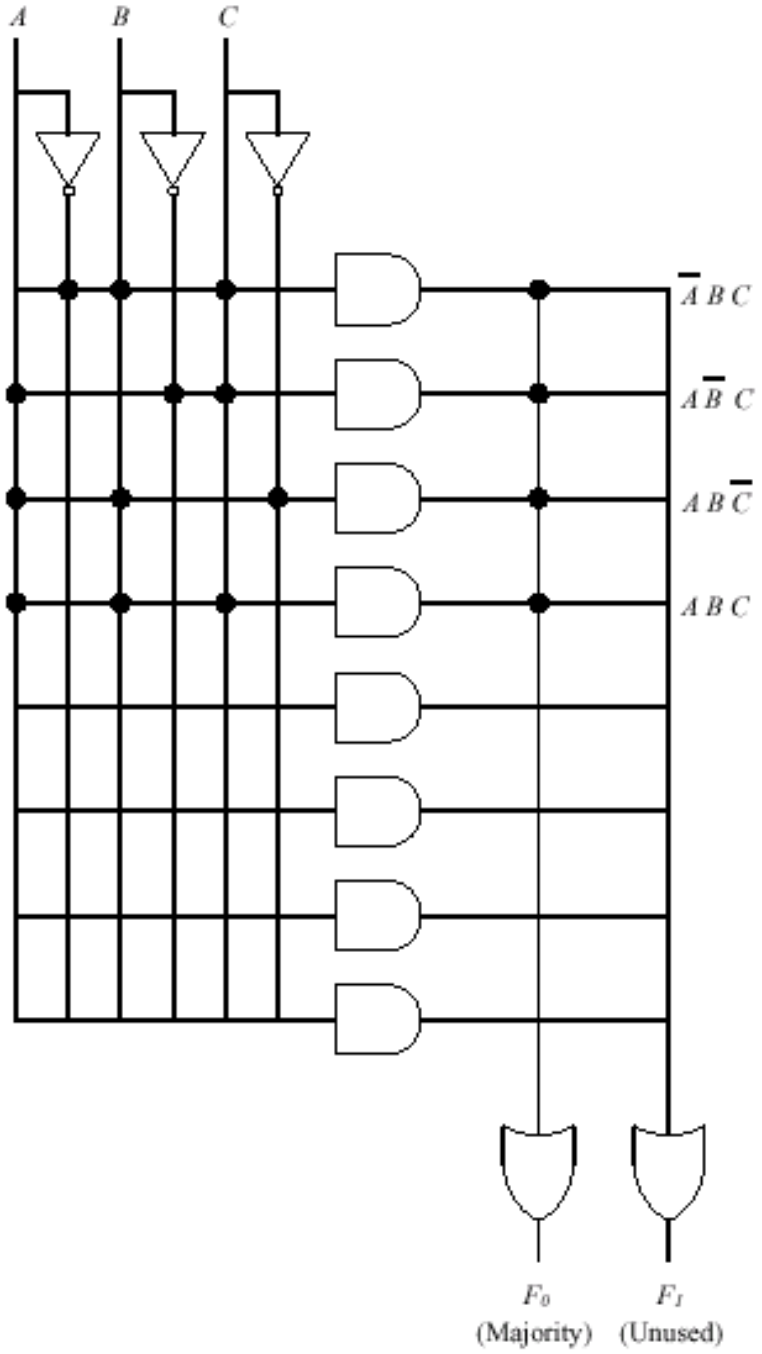


Black box view of PLA



* Slide is courtesy of M. Murdocca and V. Heuring

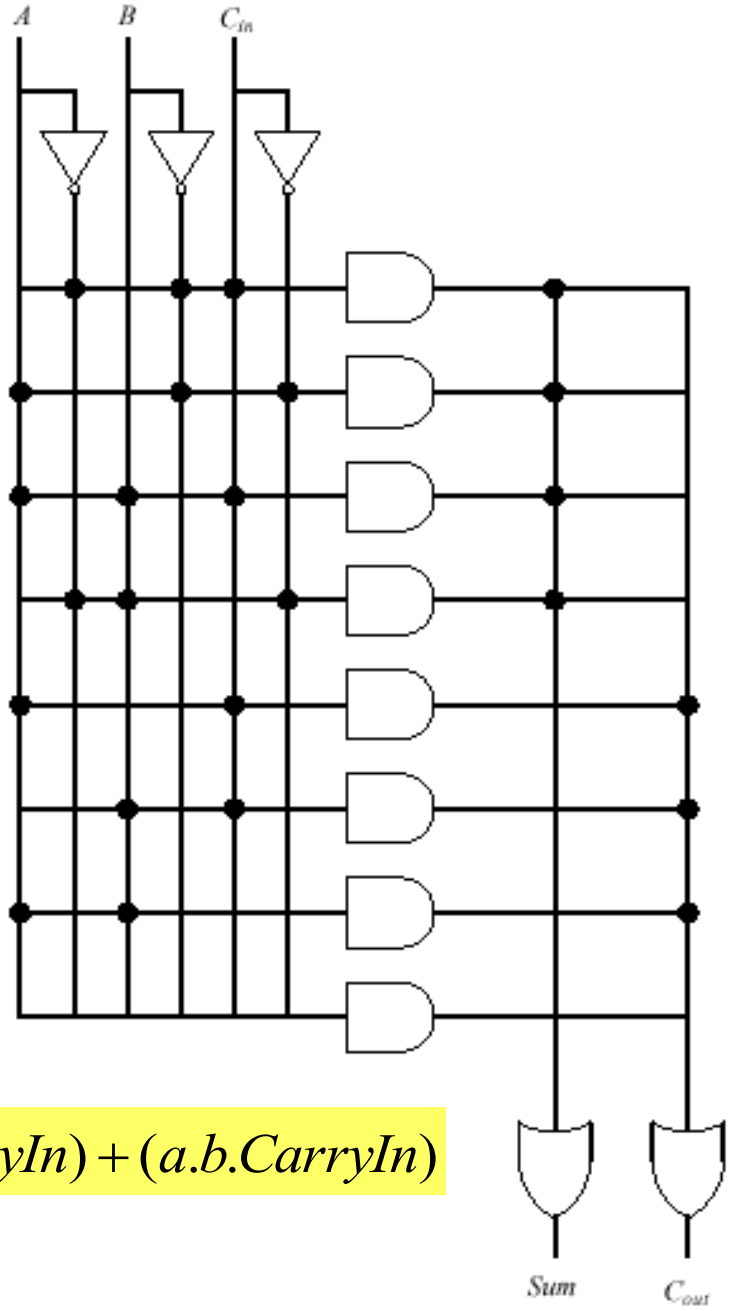
Implementation of Majority Function Using PLA



Dots reflects that the vertical line is an input to the AND gate or the horizontal line is an input to the OR gate

* Slide is courtesy of M. Murdocca and V. Heuring

PLA Realization of Full Adder



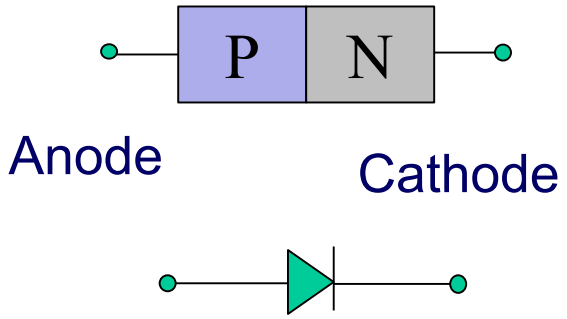
$$CarryOut = (b.CarryIn) + (a.CarryIn) + (a.b)$$

$$Sum = (a.\bar{b}.\overline{CarryIn}) + (\bar{a}.b.\overline{CarryIn}) + (\bar{a}.\bar{b}.CarryIn) + (a.b.CarryIn)$$

* Slide is courtesy of M. Murdocca and V. Heuring

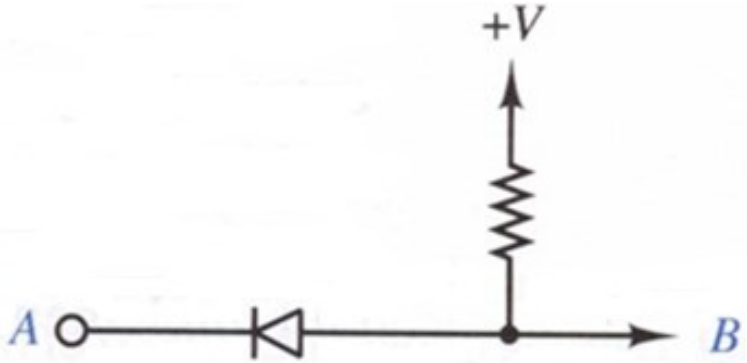
Programmable Logic Devices (PLD)

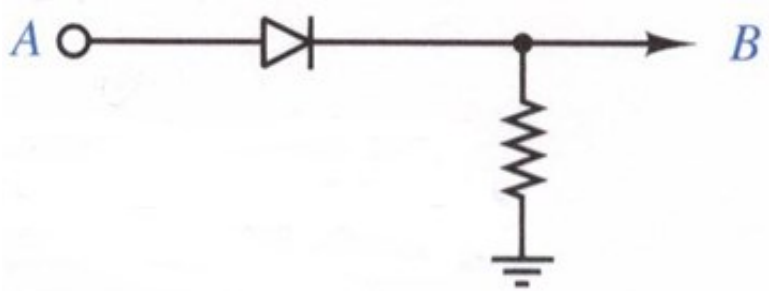
- PLD are built to enable different functional configurations using conductor diodes, transistor switches or similar elements
- Diodes can be used for implementing gates:
  - When voltage source is connected to the anode of a diode, it becomes forward-biased and acts as a short circuits
  - When the cathode is attached to the voltage source the diode becomes a large resistive element (reverse-biased) and acts as an open circuit 0

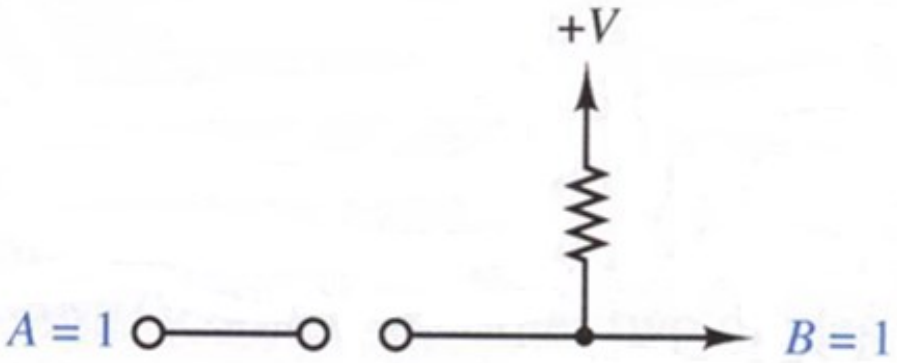


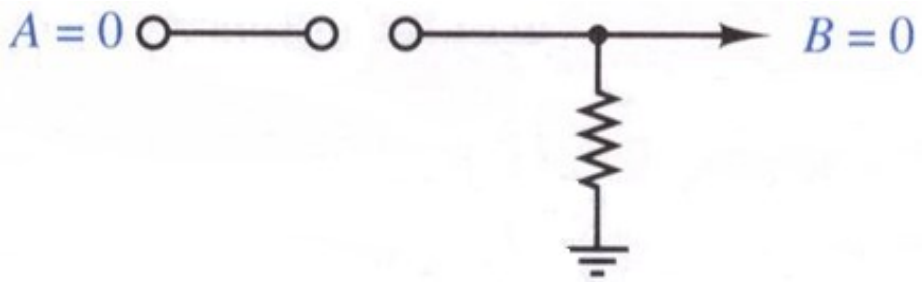
- When the PLD are programmed in the field, rather than at manufacturing facilities, they called field programmable
- Fuses are used to enable the programmability in the field (to connect internal blocks)
- Focus on AND-OR devices (NAND-NOR devices are popular)

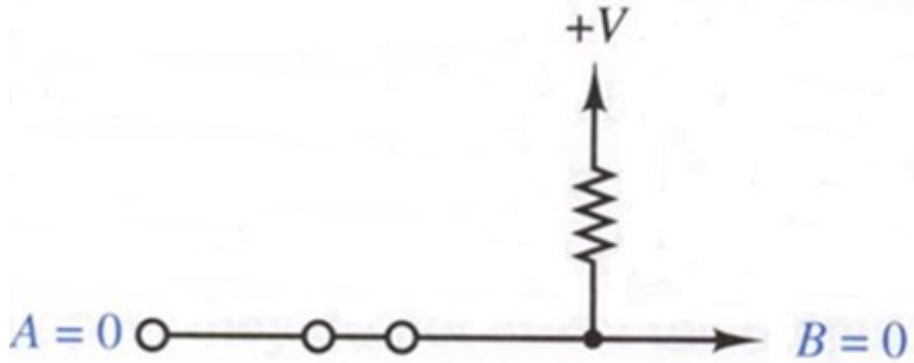
Logic using Diode Biasing

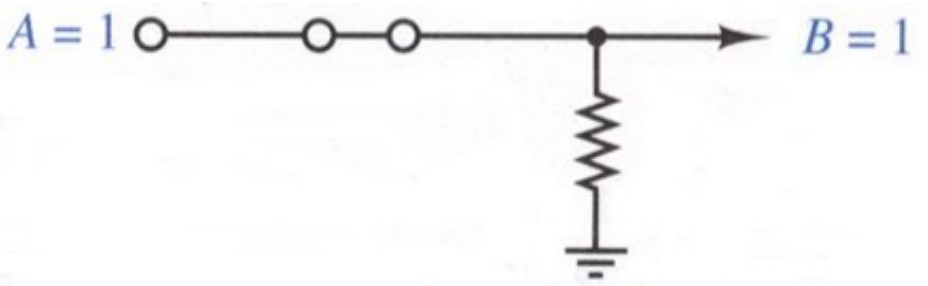




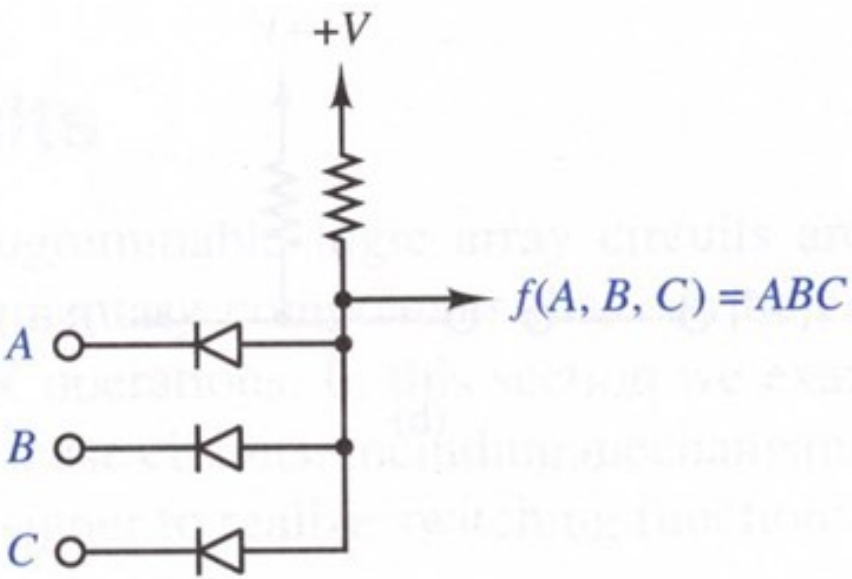


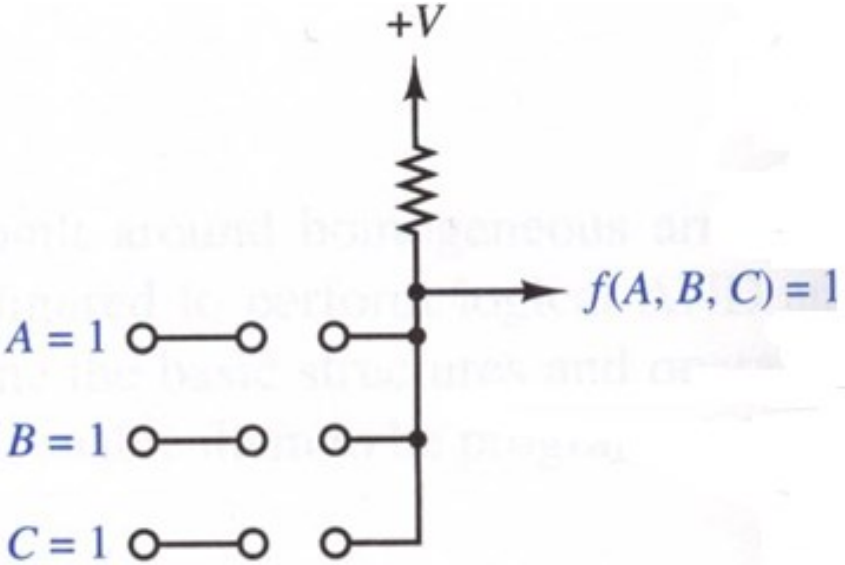


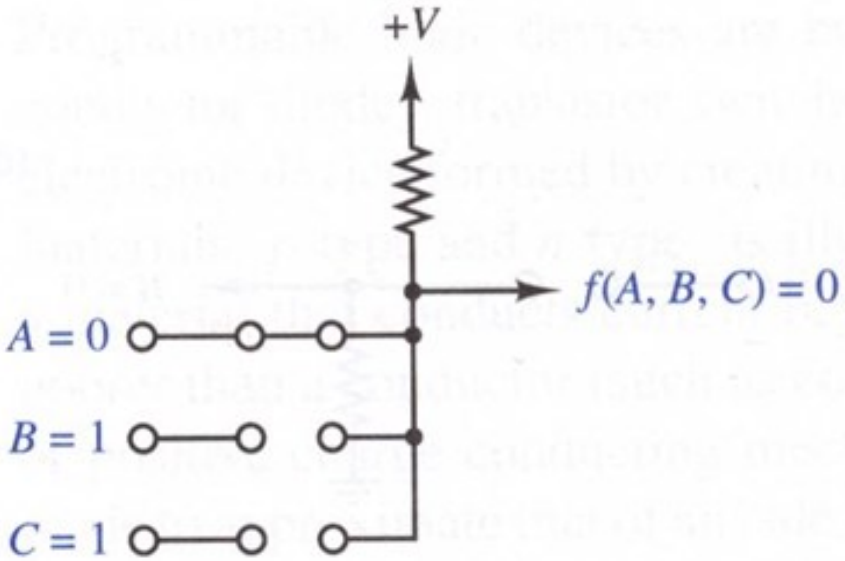


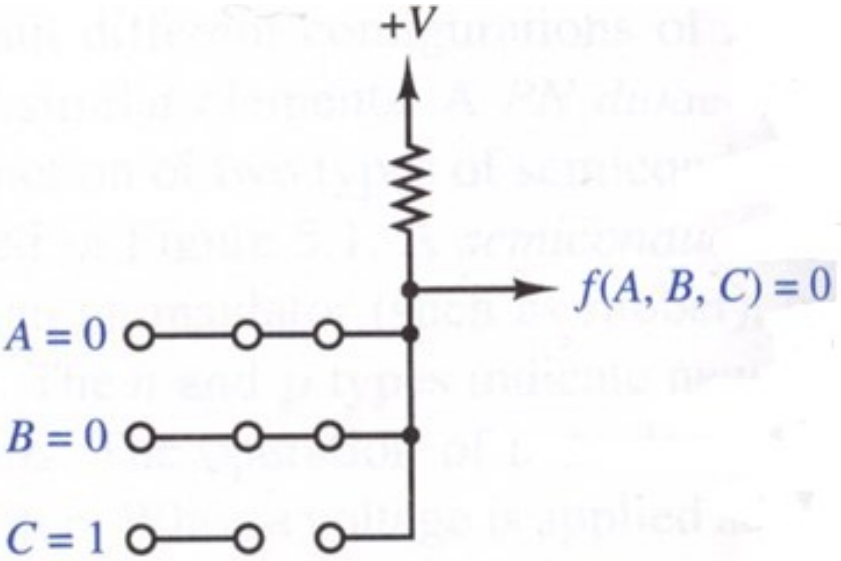


AND using Diodes



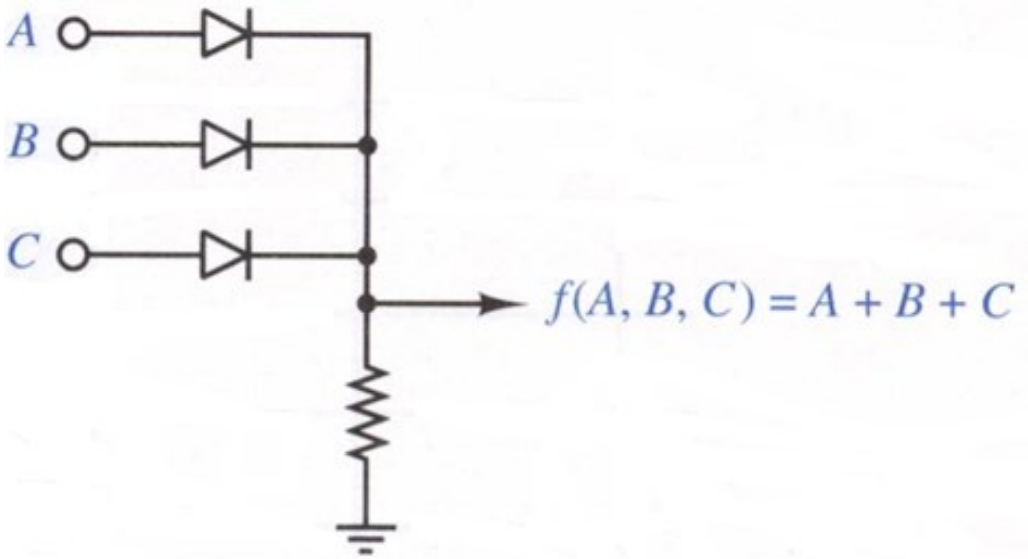


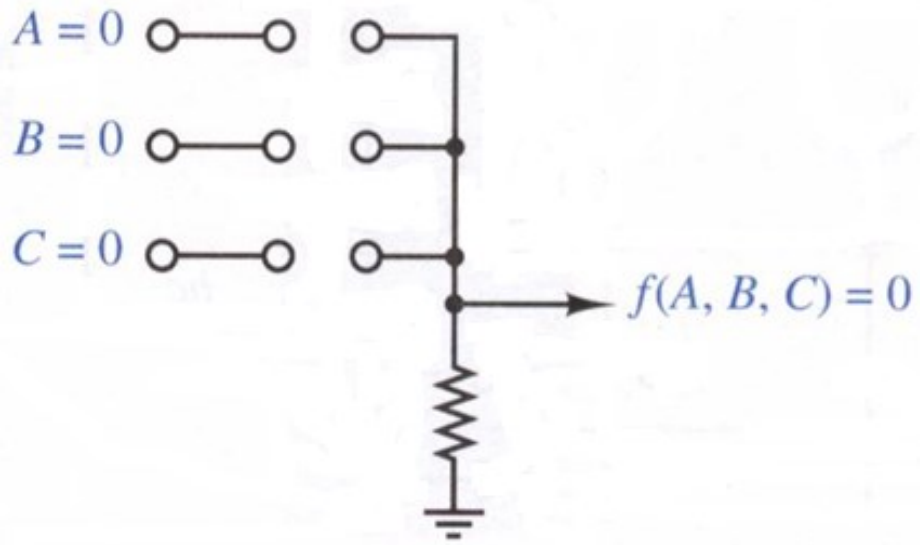


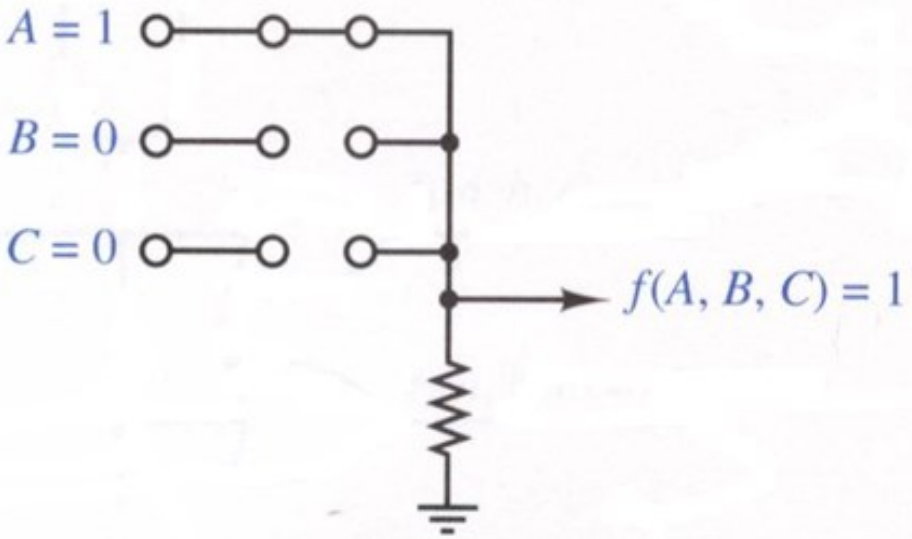


OR using Diodes

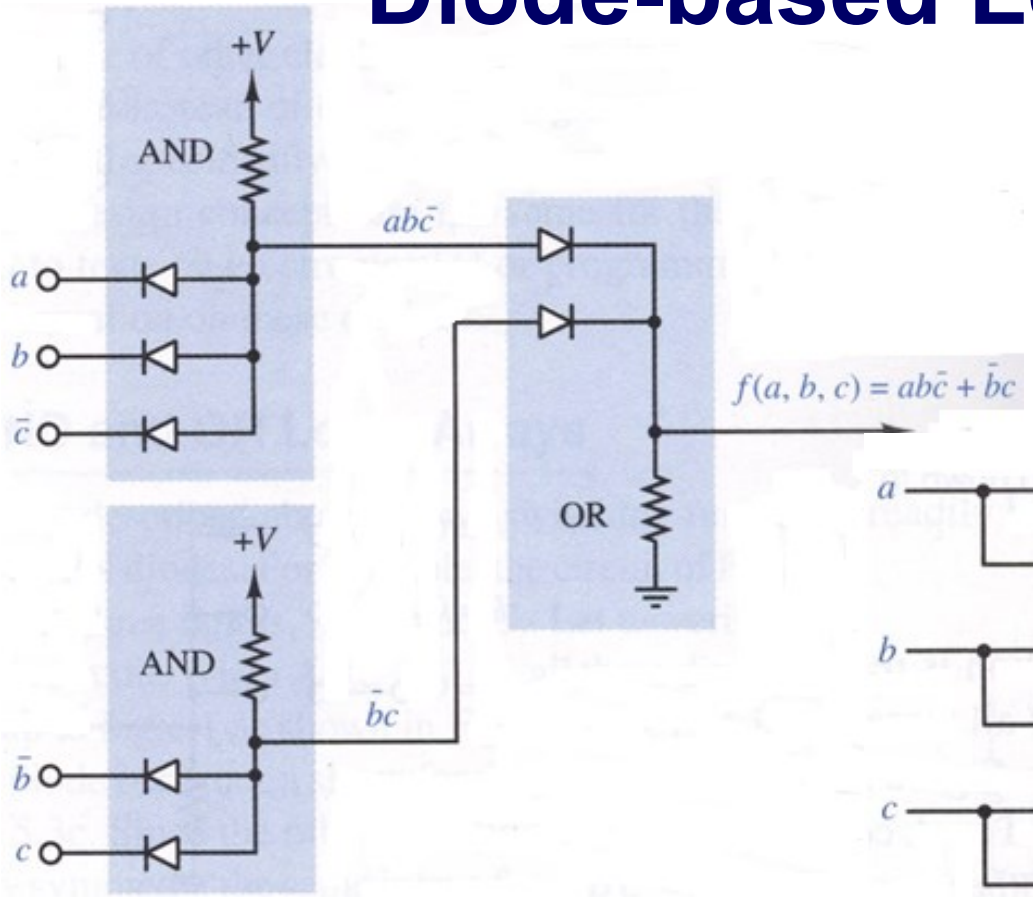
Key point:
Biasing and arranging the diodes enable implementation of gates, interconnecting them and connecting them to inputs lines



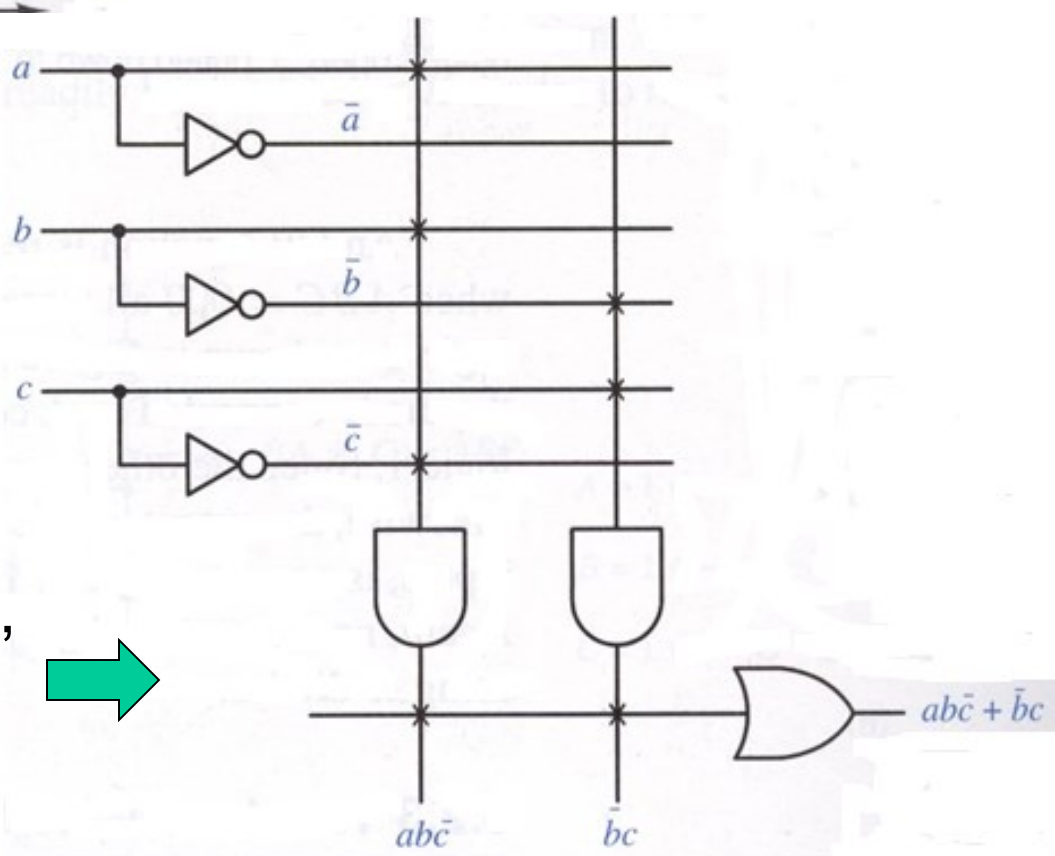




Diode-based Logic Circuit



Sum of product implementation realized with AND-OR arrays

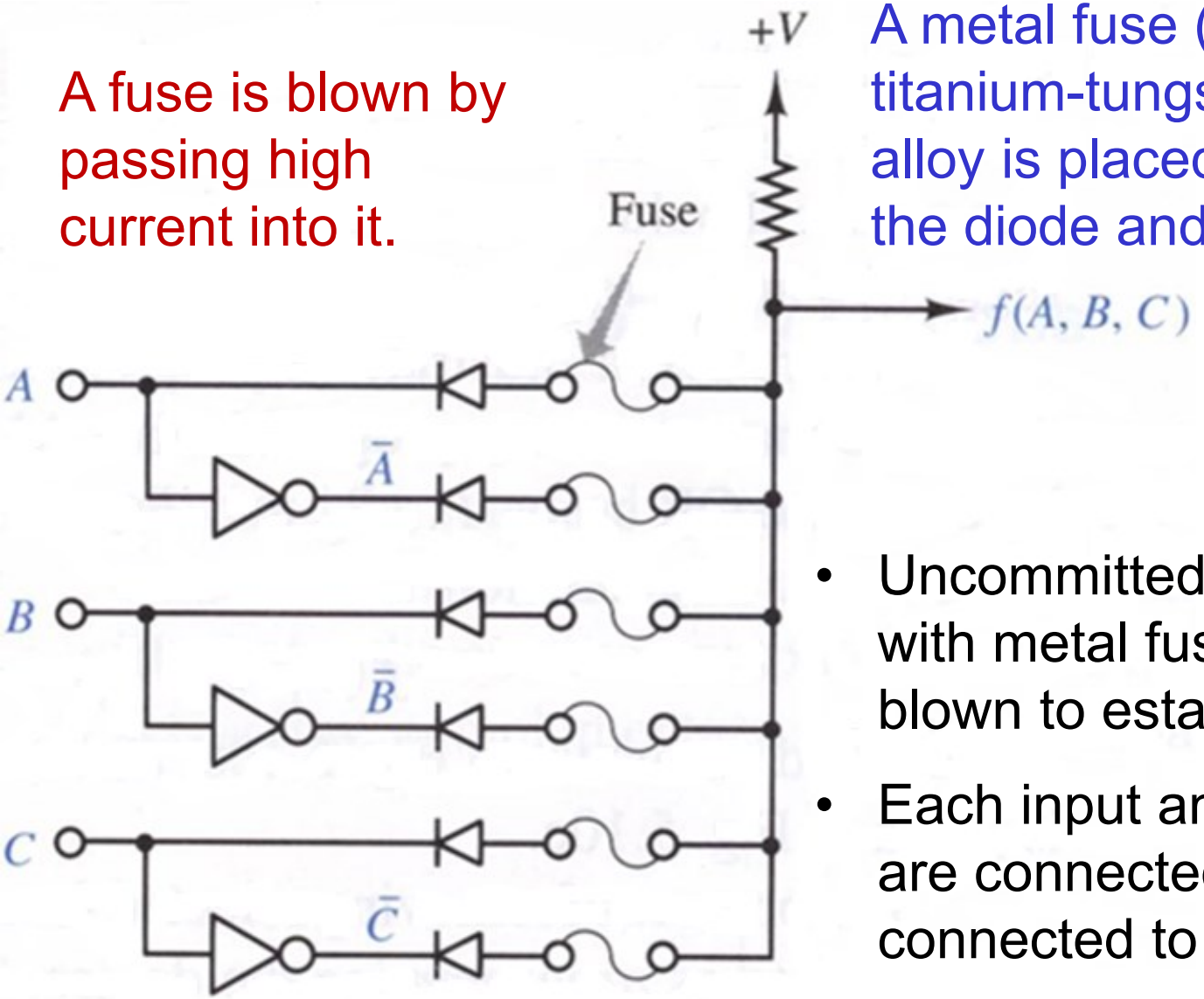


Compact form representation, where the mark X indicates presence of a diode

Field Programmable Logic Circuit

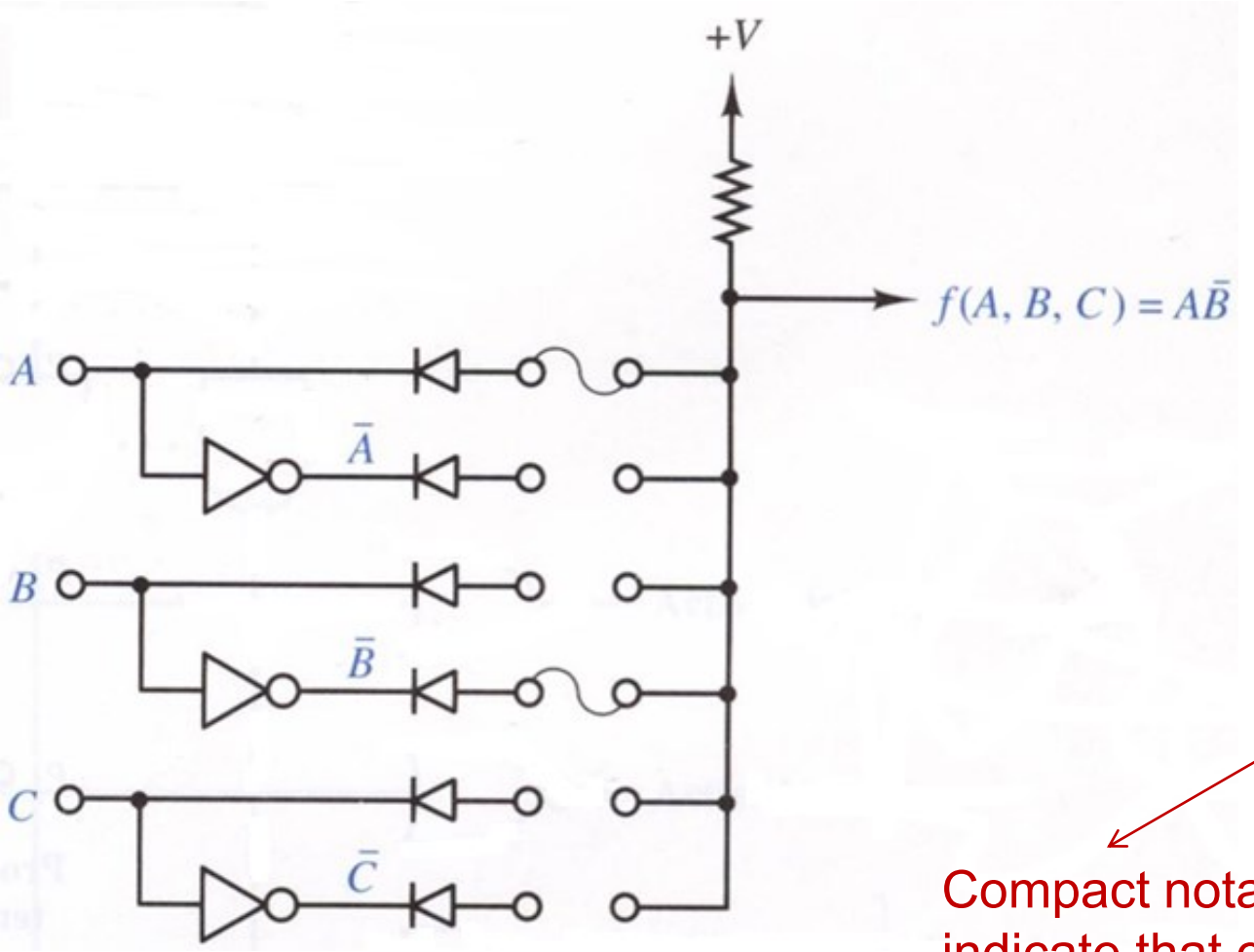
A fuse is blown by passing high current into it.

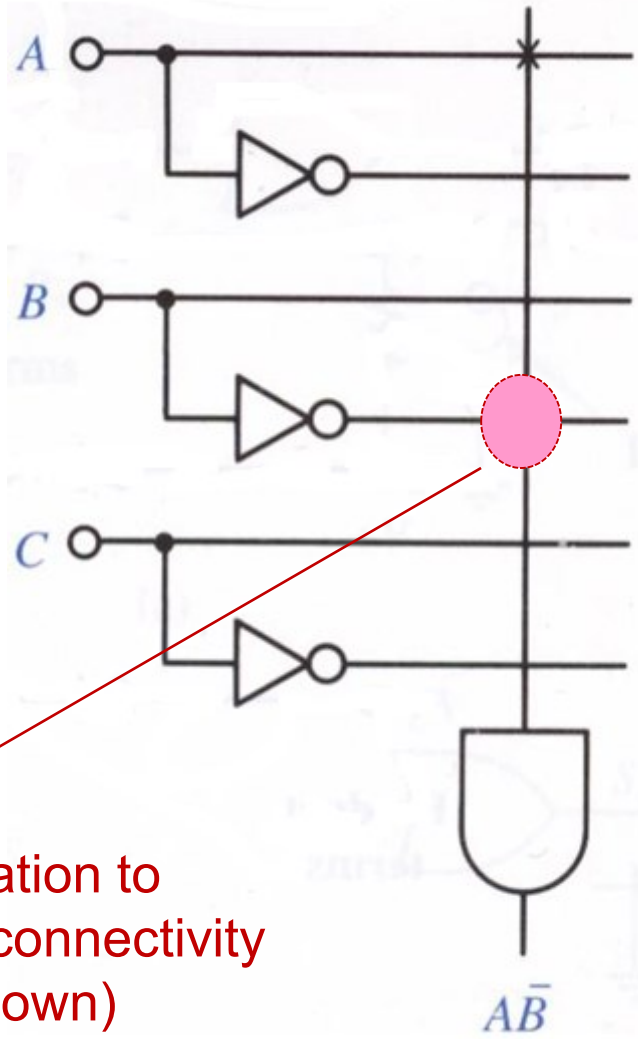
A metal fuse (nickel-chromium, titanium-tungsten, or similar alloy is placed in series between the diode and the output line



- Uncommitted AND/OR arrays with metal fuses that can be blown to establish connectivity
- Each input and its complement are connected and fuses are connected to the output

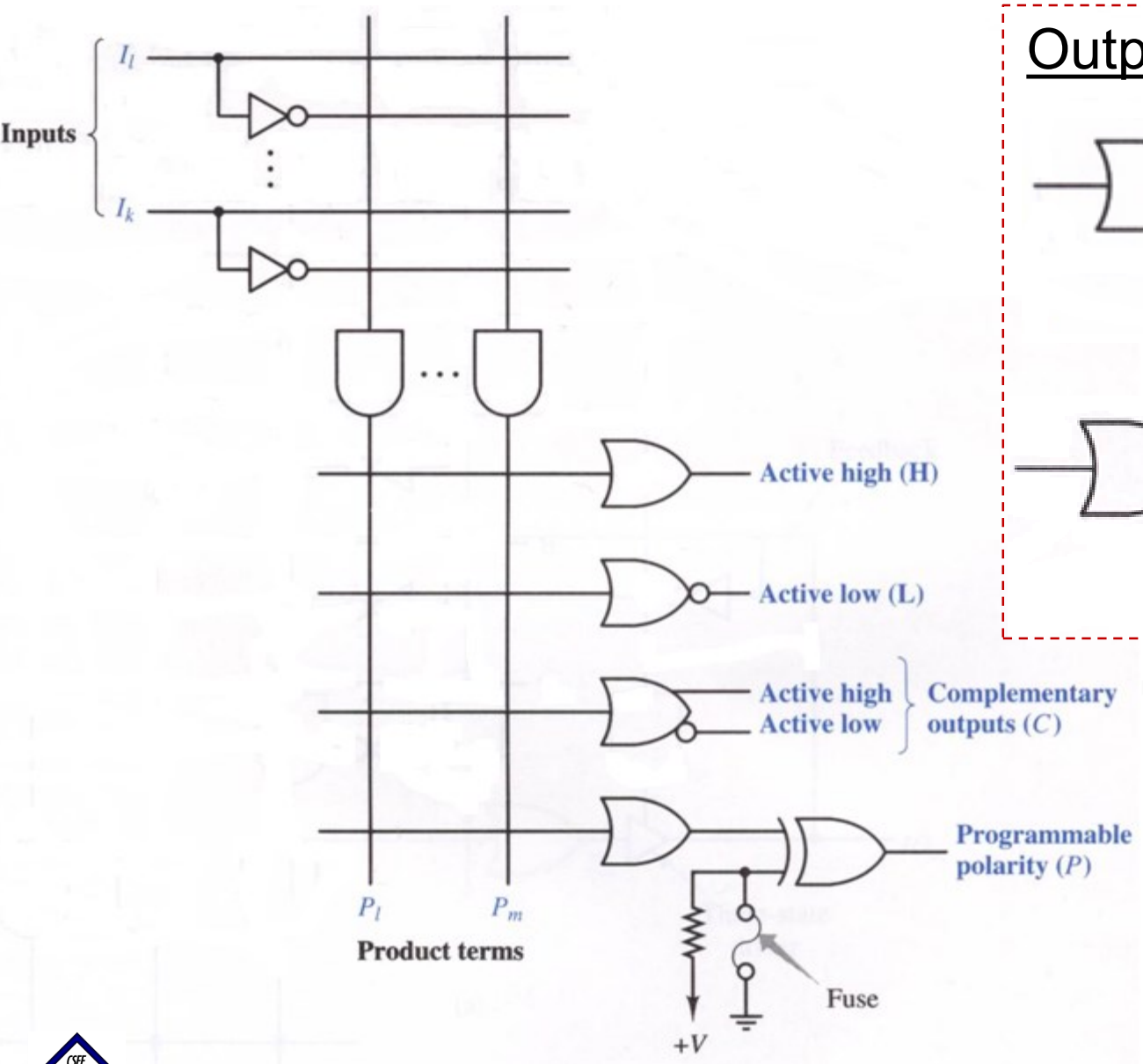
Example: Field Programmable Circuit



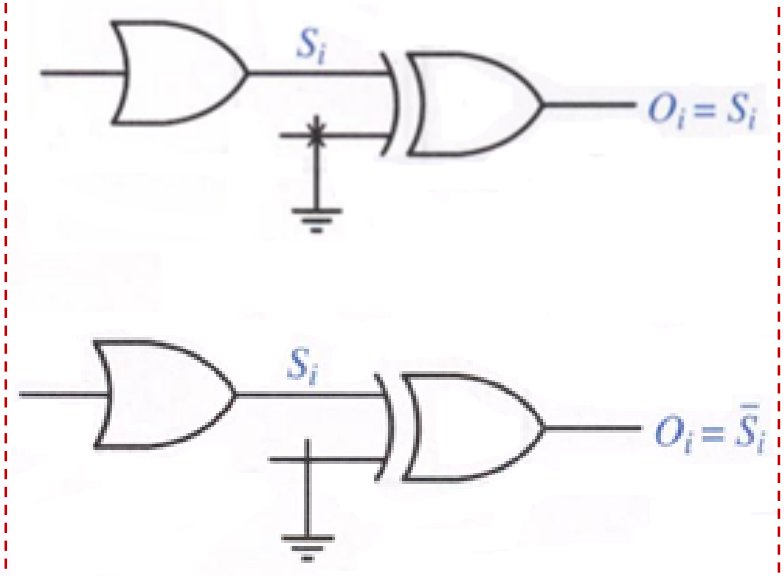


Compact notation to indicate that connectivity (fuse is not blown)

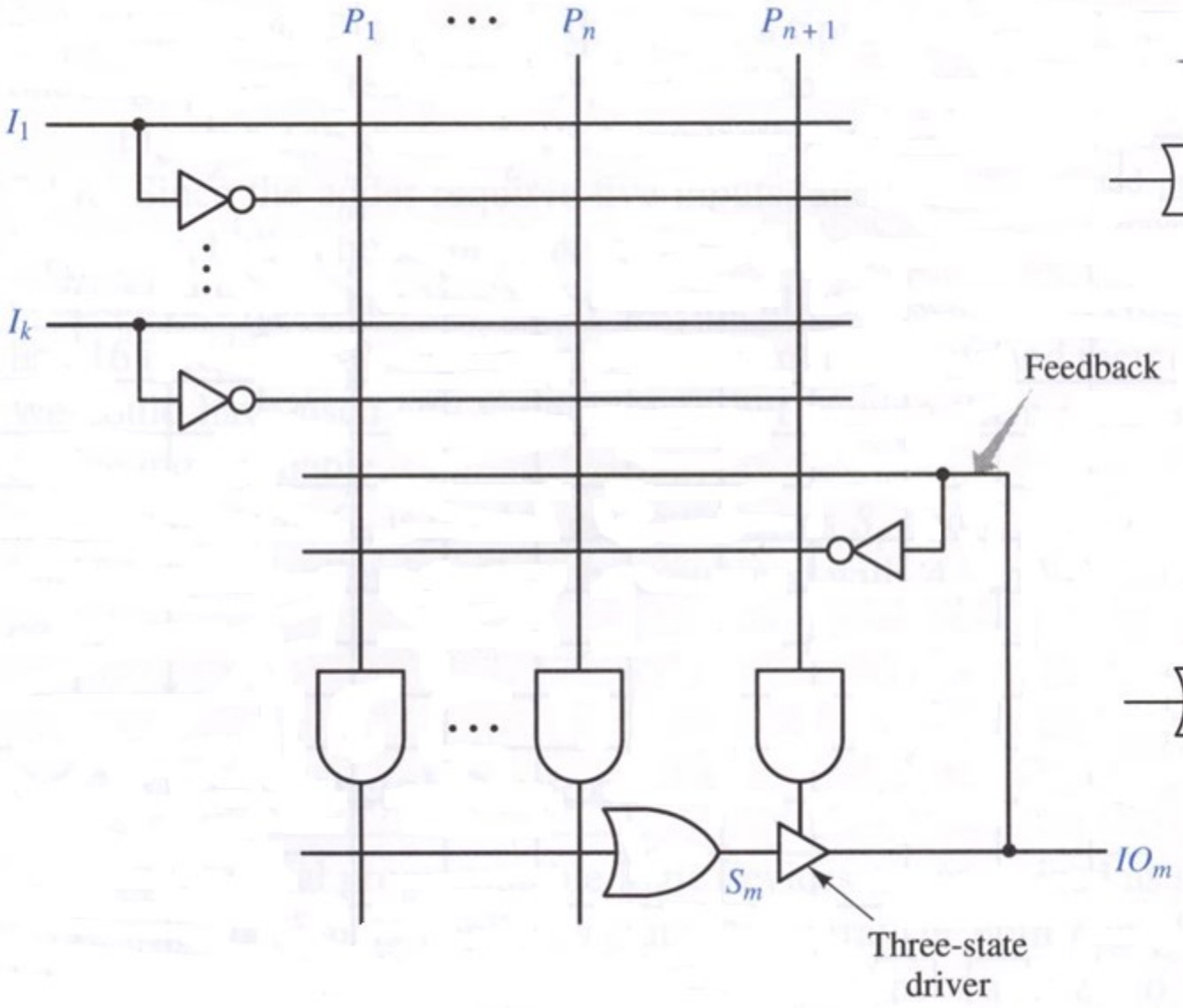
Input-Output Port Options

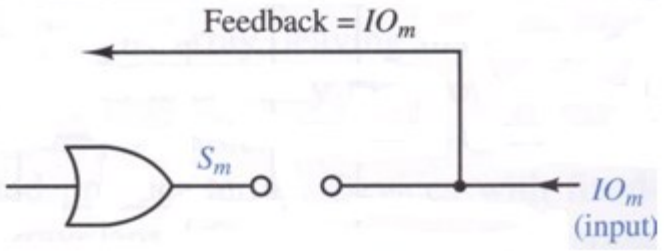


Output polarity

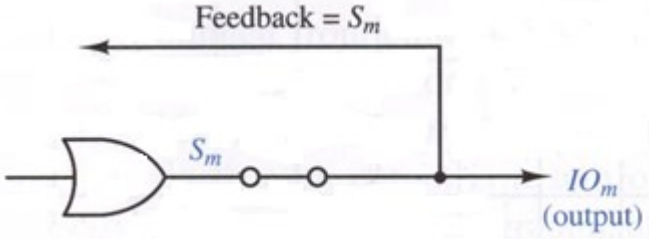


Bidirectional Port Option





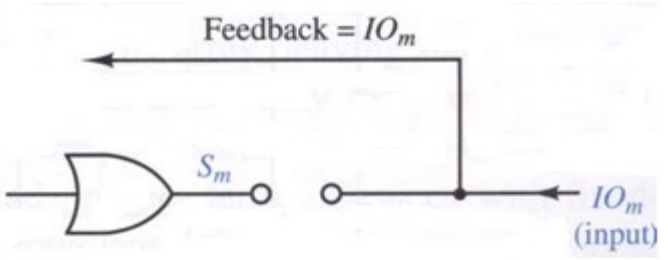
Tristate driver disabled ($P_{n+1} = 0$)



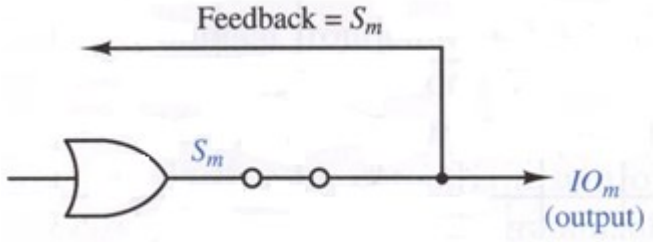
Tristate driver enabled ($P_{n+1} = 1$)

Bidirectional Pins

- Tristate driver configuration:
  - Tristate driver enabled: S_m is available as output and is also fed back to be used as an intermediate input to the gate array
  - Tristate driver disabled: IO_m becomes an external input port to the array

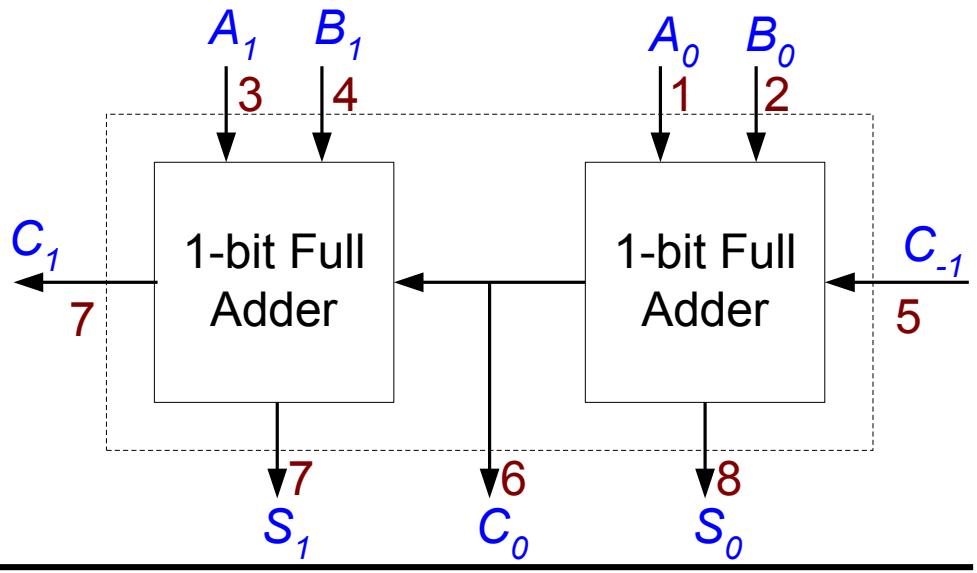


Tristate driver disabled

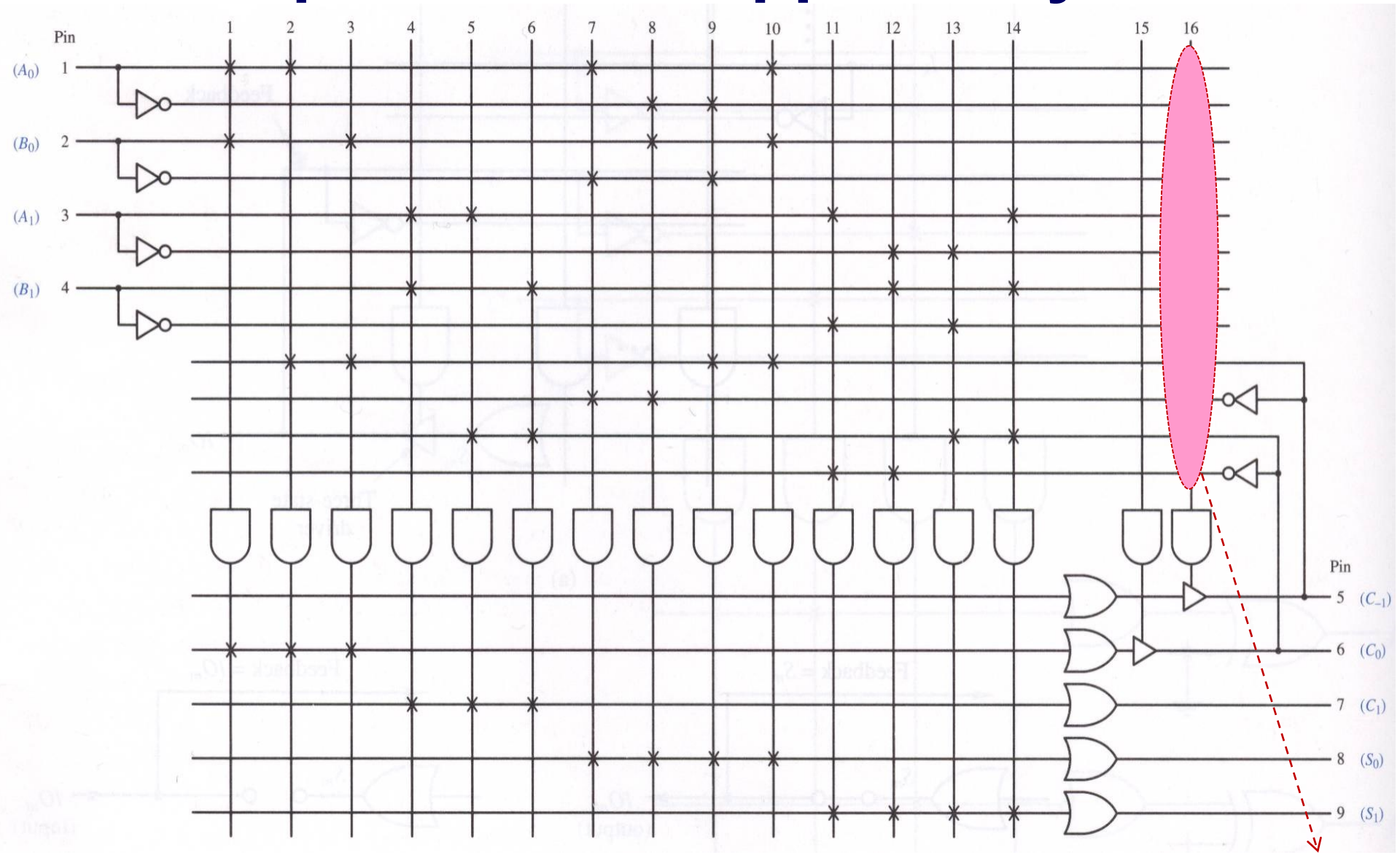


Tristate driver enabled

- Example: Implementing of a 2-bit ripple-carry adder using PLA
  - C_{-1} is considered an external input to the array (control line for subtract/add)
  - C_1 is the carry-out of the ripple adder, i.e., an output from the array.
  - C_0 is an intermediate carry (to be fed back to array)

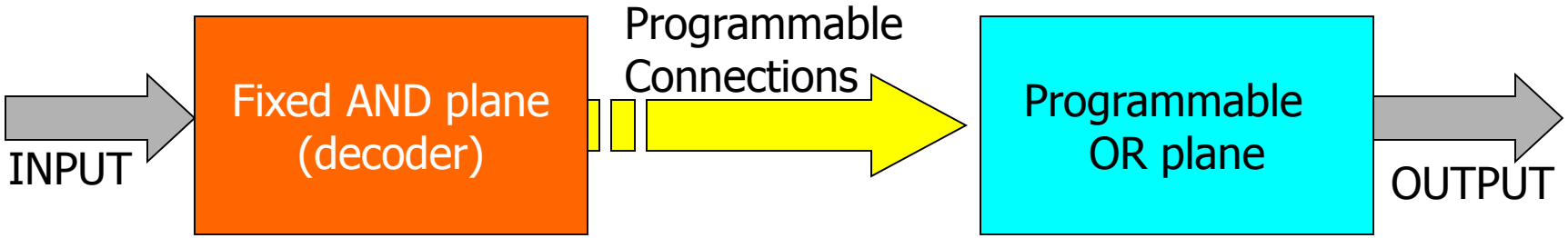


Example: Two-bit ripple carry adder

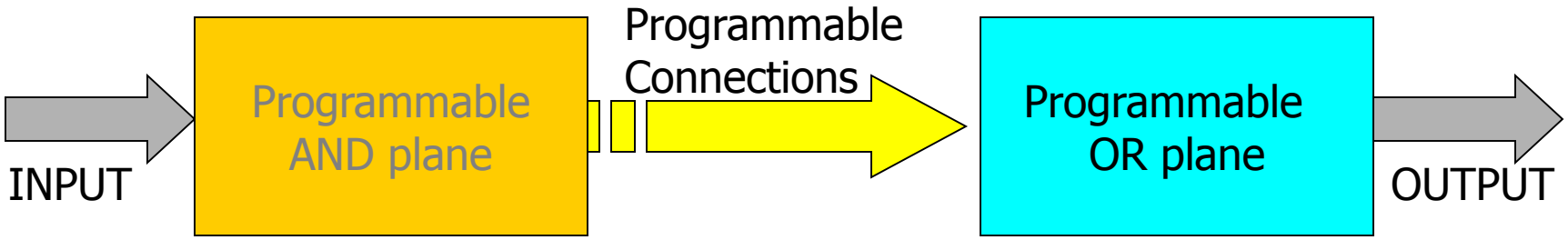


Pin_{16} connected to inputs & their complements to prevent feeding back C_{-1}

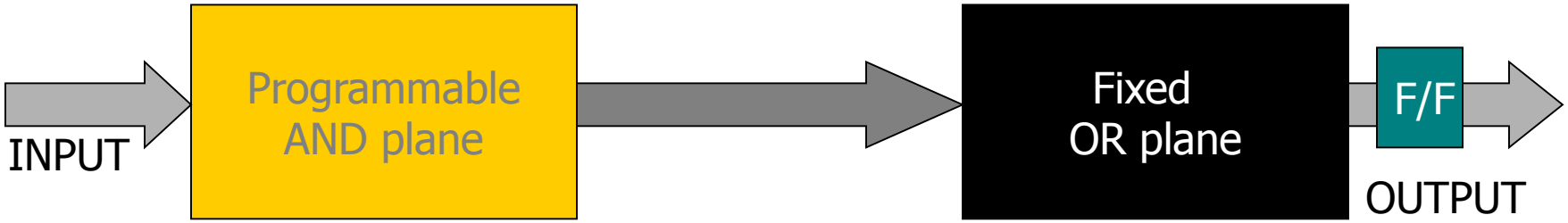
Classifying Three Basic PLDs



(Programmable) Read-Only Memory (ROM)



Programmable Logic Array (PLA)



Programmable Array Logic (PAL) Devices

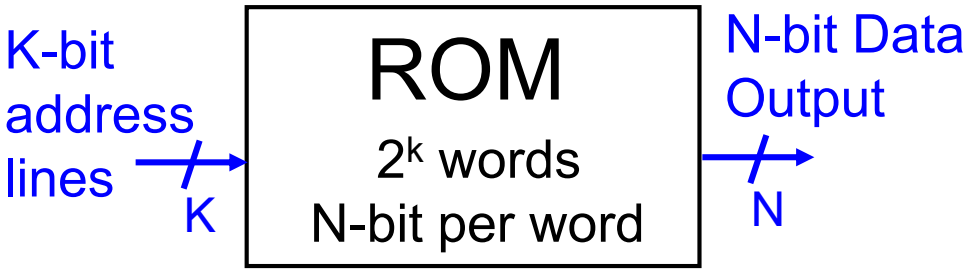
* Slide is courtesy of Hsien-Hsin Sean Lee

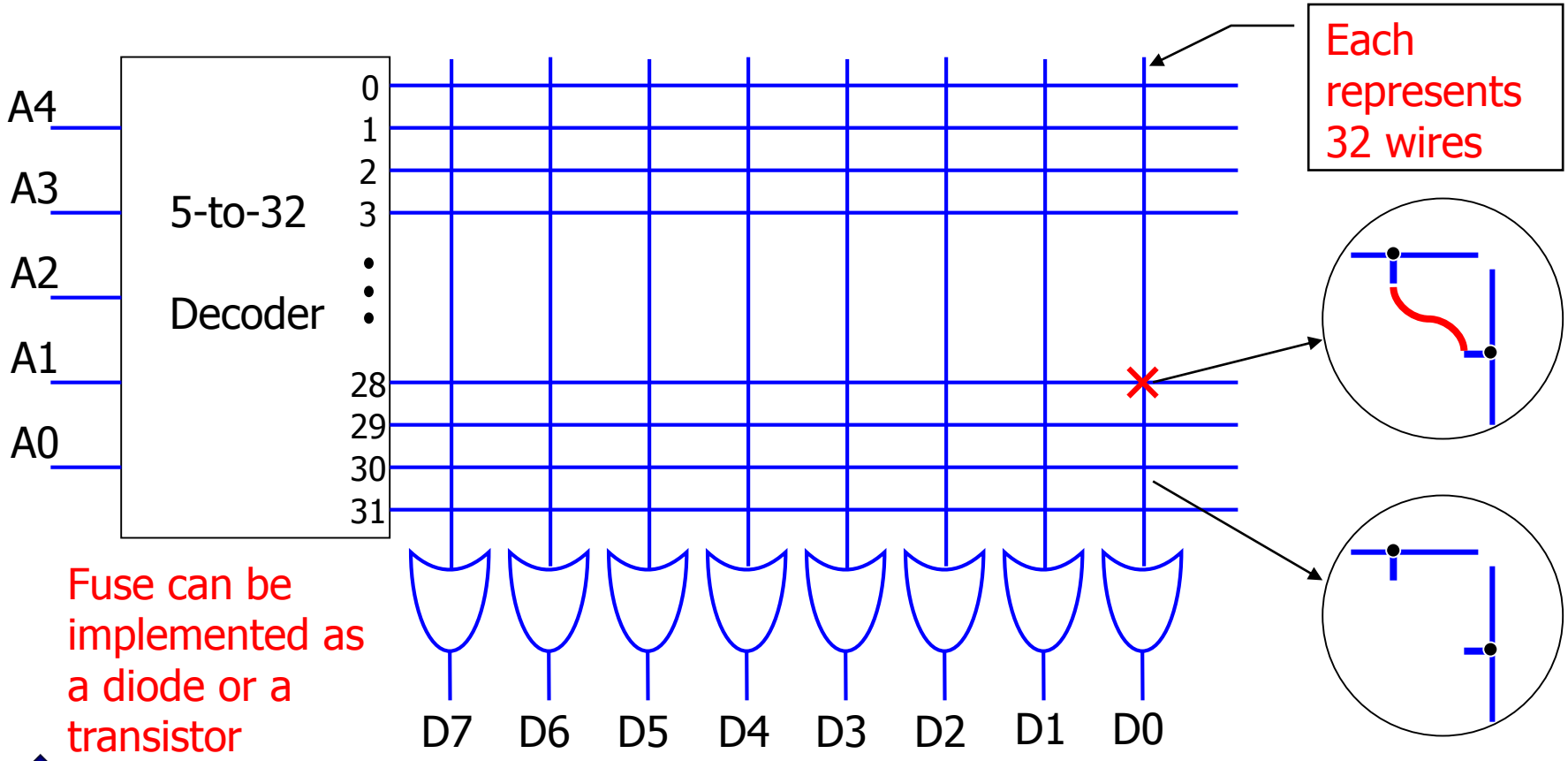
Example Commercial PLDs

Device	Description	Inputs	Product Terms	Outputs	Output Polarity
PLS100	PLA	16	48	8	Programmable
PLS153	PLA	16	42	10	Programmable
82S123	PROM	5	32	8	Active high
82S129	PROM	8	256	4	Active high
82S131	PROM	9	512	4	Active high
82S135	PROM	8	256	8	Active high
82S137	PROM	10	1024	4	Active high
82S147	PROM	9	512	8	Active high
82S181	PROM	10	1024	8	Active high
82S185	PROM	11	2048	4	Active high
82S191	PROM	11	2048	8	Active high
82S321	PROM	12	4096	8	Active high
PAL16L8	PAL	16	8	8	Active low
PAL14H4	PAL	14	4	4	Active high
PAL16C1	PAL	16	16	1	Complementary
PAL18P8	PAL	18	8	8	Programmable

Read Only Memory (ROM)

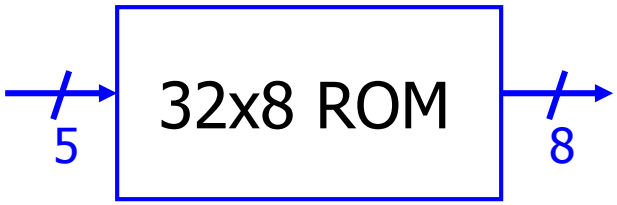
- Configure for "Permanent" storage of binary information
- Non-volatile: Power off does not erase stored information





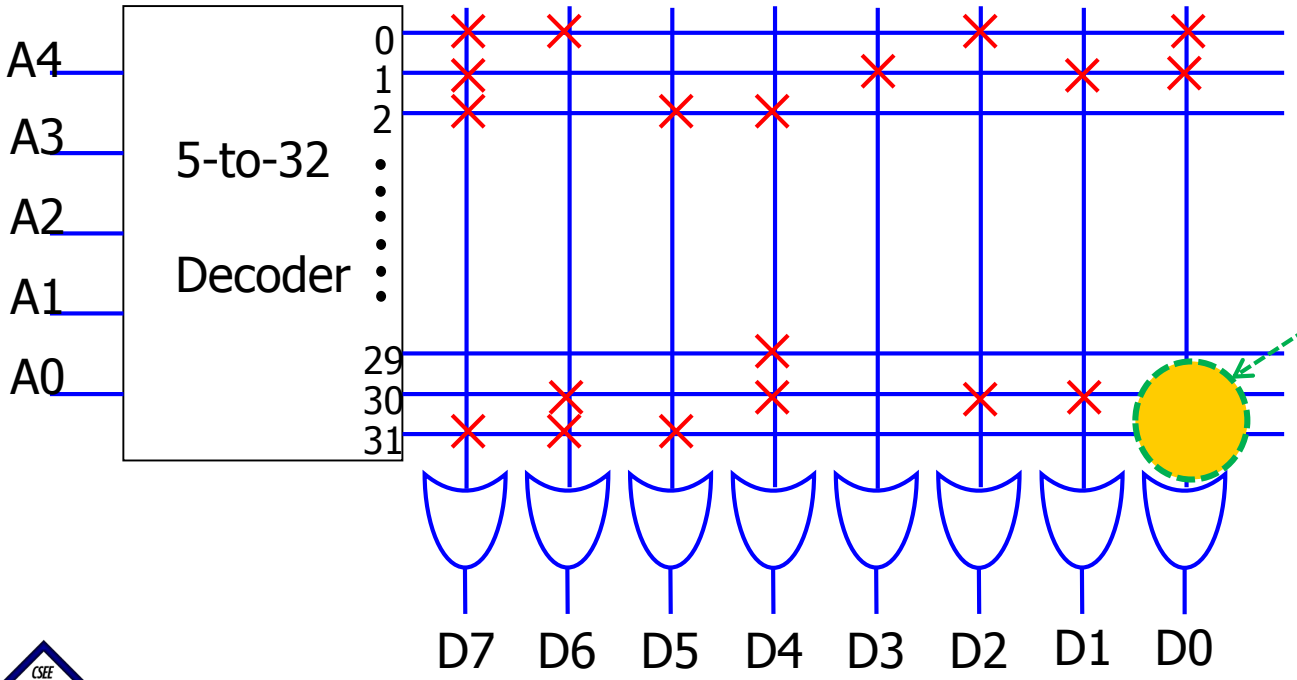
* Contents is courtesy of Hsien-Hsin Sean Lee

Example: Programming 32x8 ROM



Start with truth table

A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	1	0	0	0	1	0	1
0	0	0	0	1	1	0	0	0	1	0	1	1
0	0	0	1	0	1	0	1	1	0	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…
1	1	1	0	1	0	0	0	1	0	0	0	0
1	1	1	1	0	0	1	0	1	0	1	1	0
1	1	1	1	1	1	1	1	0	0	0	0	



For every minterm make a connection for all corresponding output OR gates

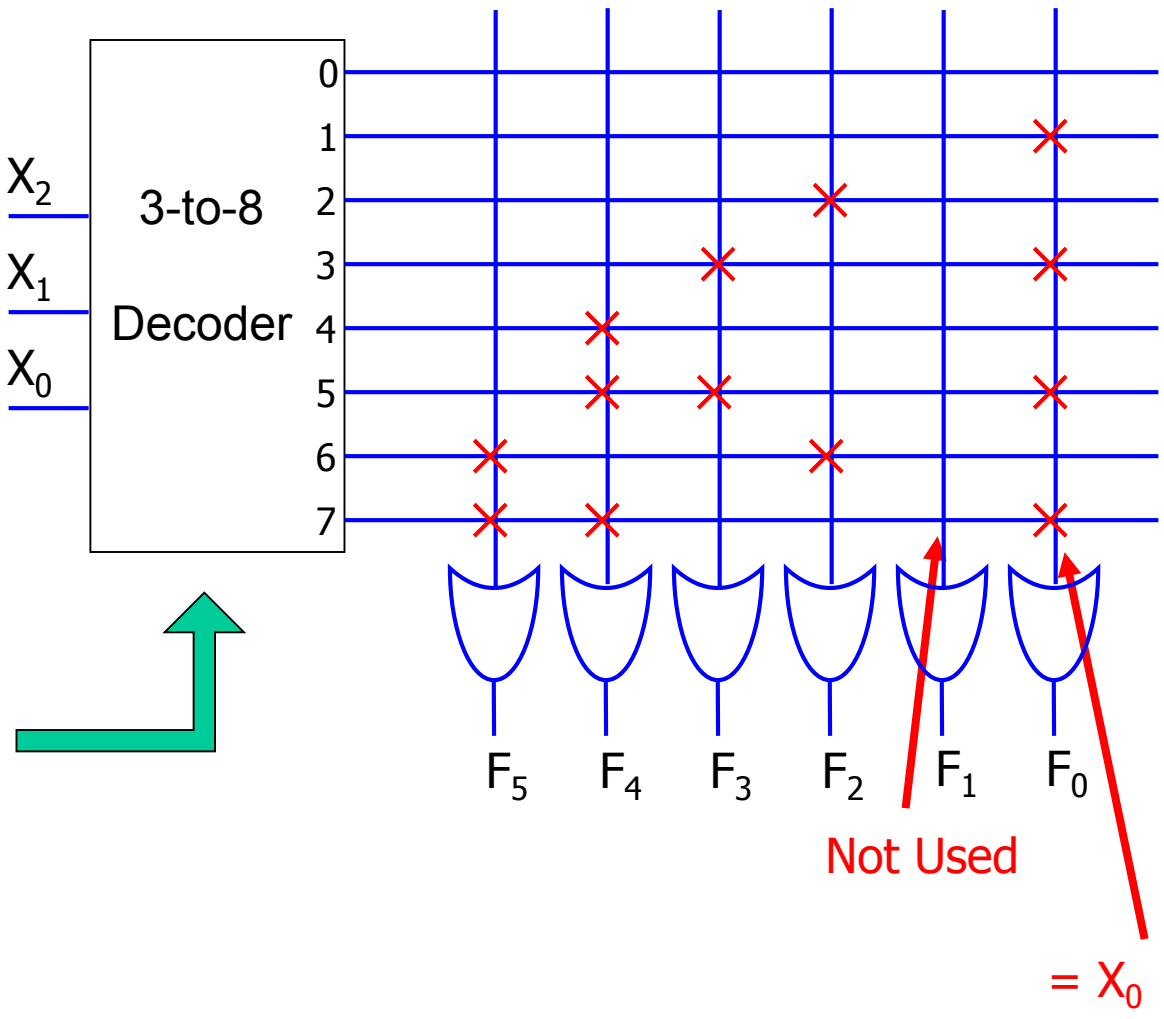
* Contents is courtesy of Hsien-Hsin Sean Lee

Example: Lookup Table using ROM

X	$F(X)=X^2$
0	0
1	1
2	4
3	9
4	16
5	25
6	36
7	49

- A square lookup table for $F(X) = X^2$ using ROM

X	$F(X)=X^2$
000	000000
001	000001
010	000100
011	001001
100	010000
101	011001
110	100100
111	110001



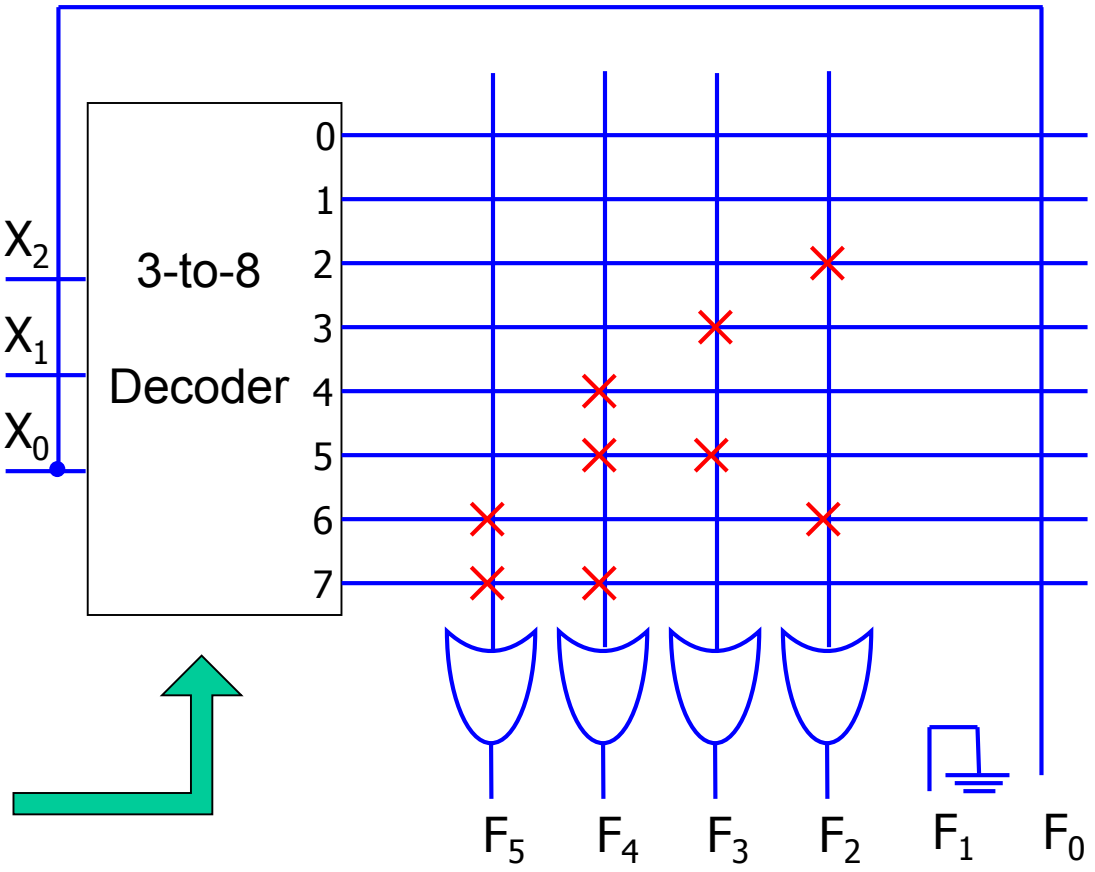
* Contents is courtesy of Hsien-Hsin Sean Lee

Example: Lookup Table using ROM

X	$F(X)=X^2$
0	0
1	1
2	4
3	9
4	16
5	25
6	36
7	49

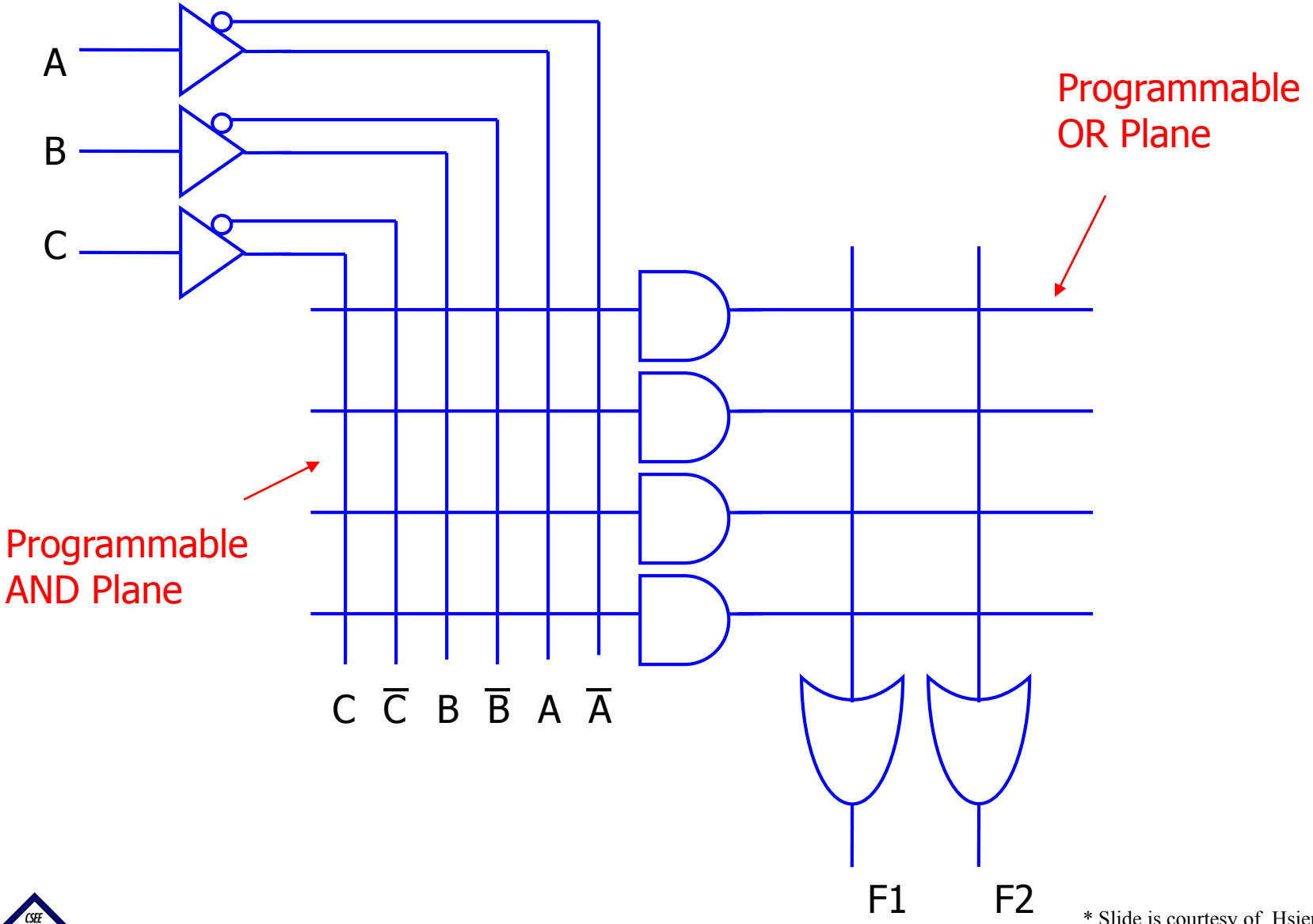
- A square lookup table for $F(X) = X^2$ using ROM

X	$F(X)=X^2$
000	000000
001	000001
010	000100
011	001001
100	010000
101	011001
110	100100
111	110001



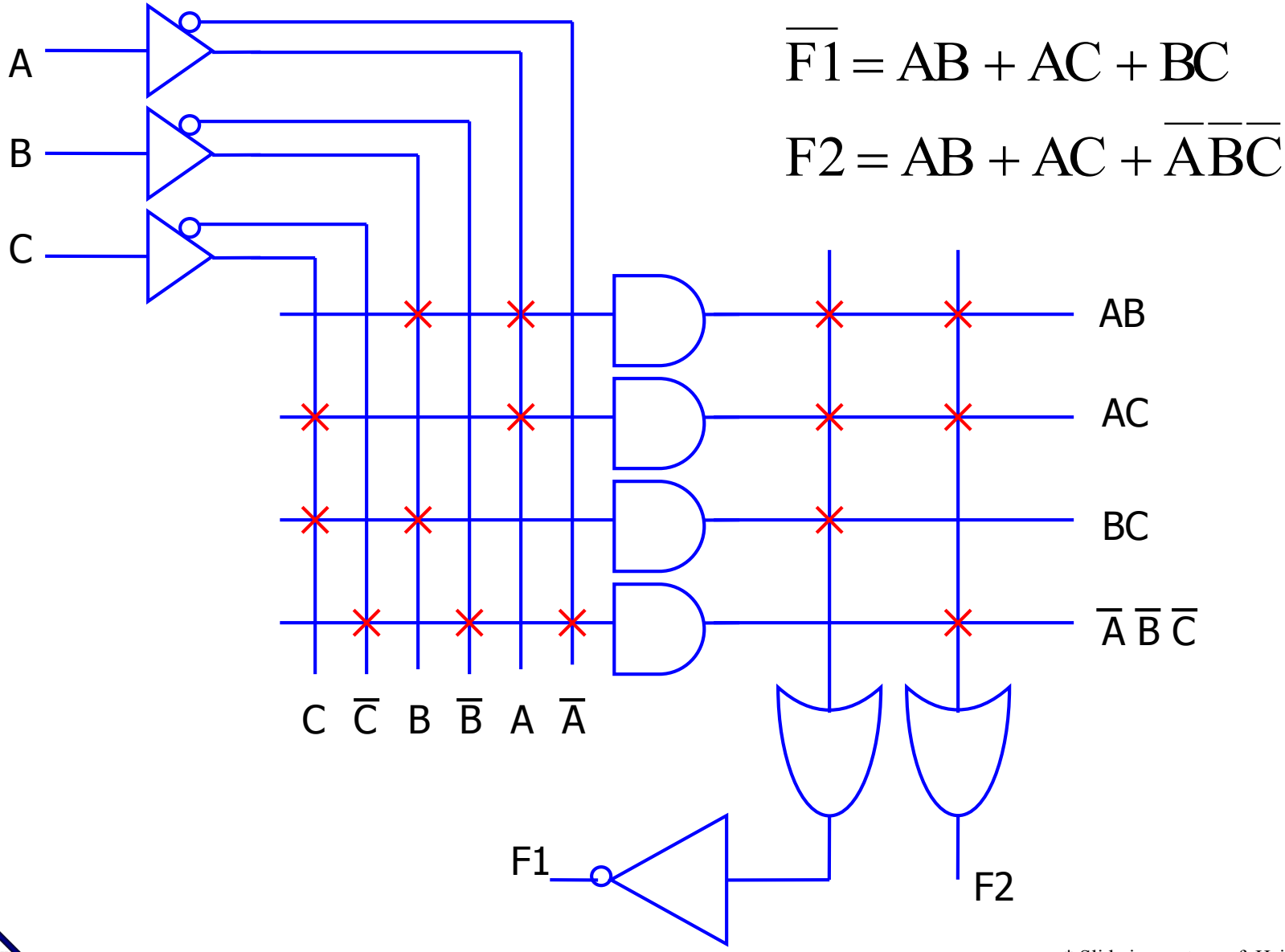
* Contents is courtesy of Hsien-Hsin Sean Lee

Programmable Logic Array (PLA)



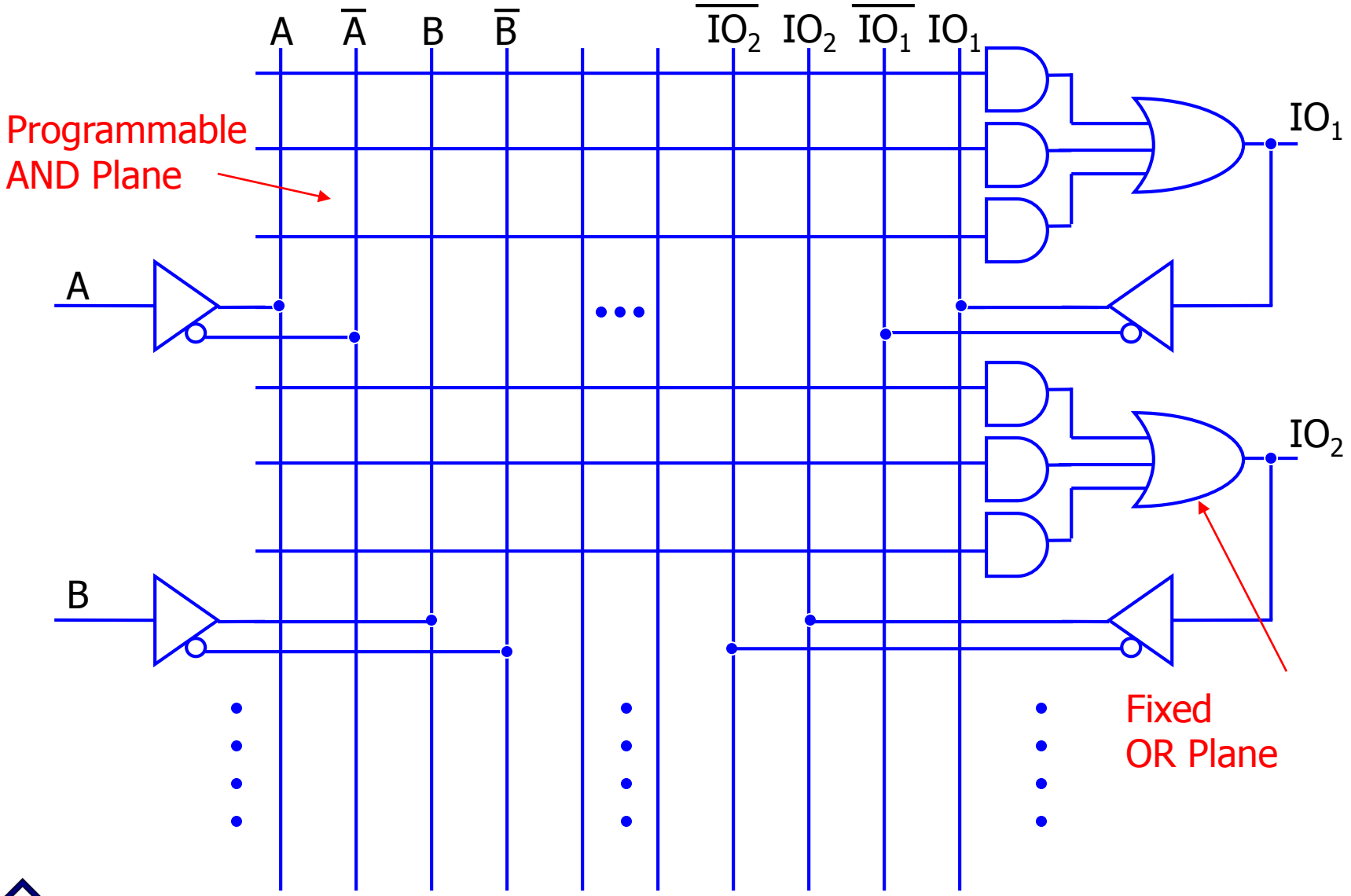
* Slide is courtesy of Hsien-Hsin Sean Lee

Example of Using PLA



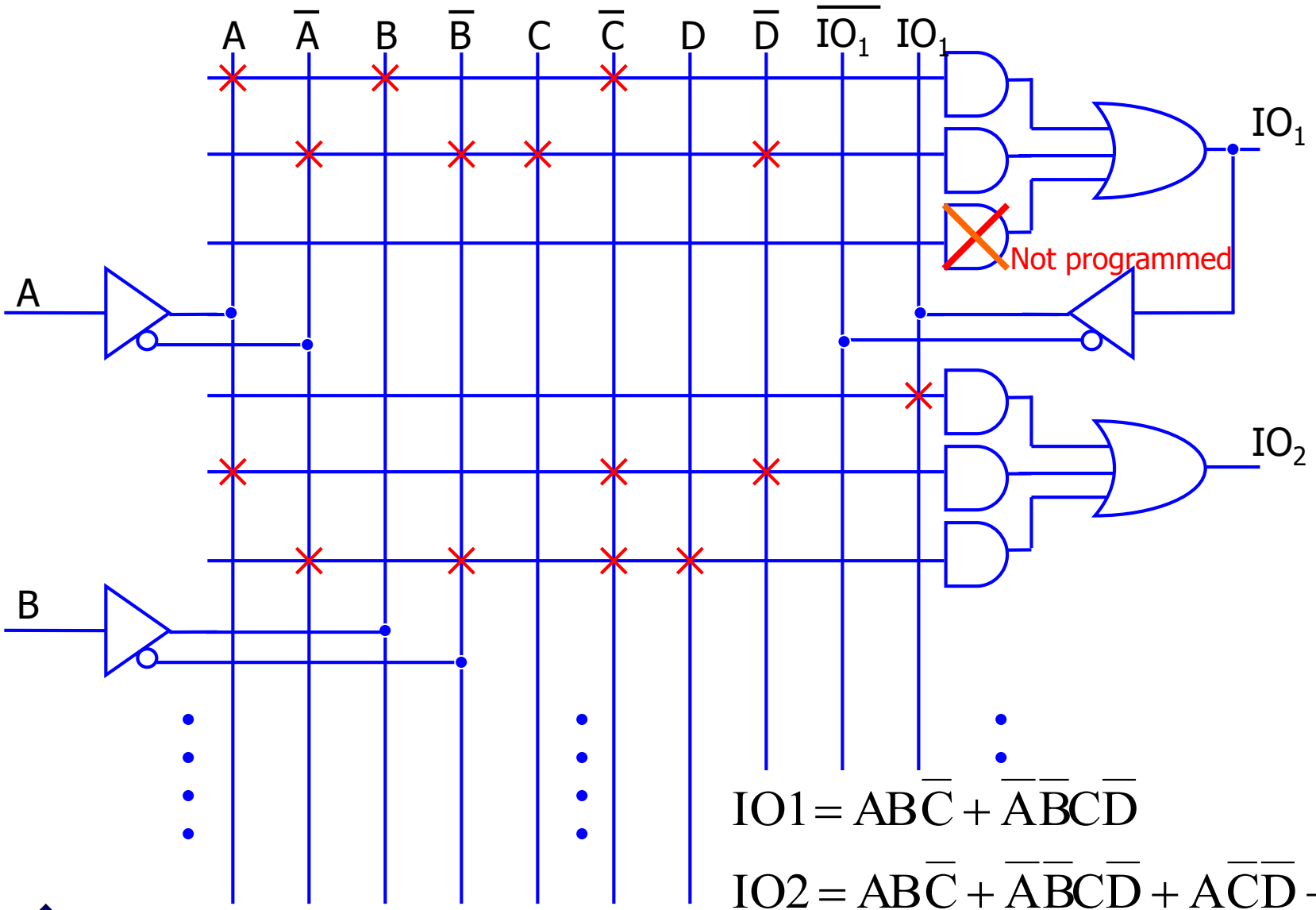
* Slide is courtesy of Hsien-Hsin Sean Lee

PAL Devices



* Slide is courtesy of Hsien-Hsin Sean Lee

Example: PAL Device Design

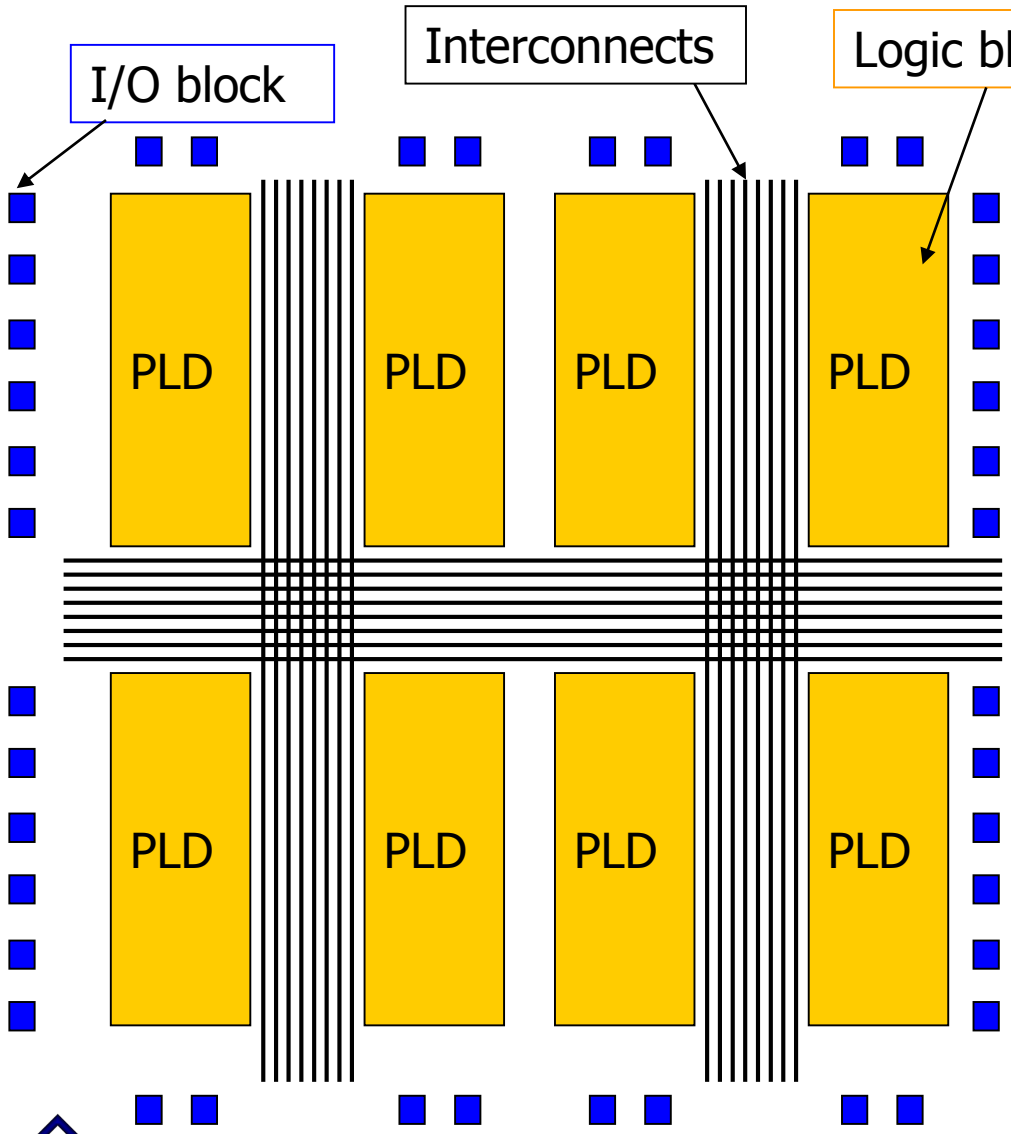


$$IO1 = AB\overline{C} + \overline{A}\overline{B}C\overline{D}$$

$$IO2 = AB\overline{C} + \overline{A}\overline{B}C\overline{D} + A\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D}$$

* Slide is courtesy of Hsien-Hsin Sean Lee

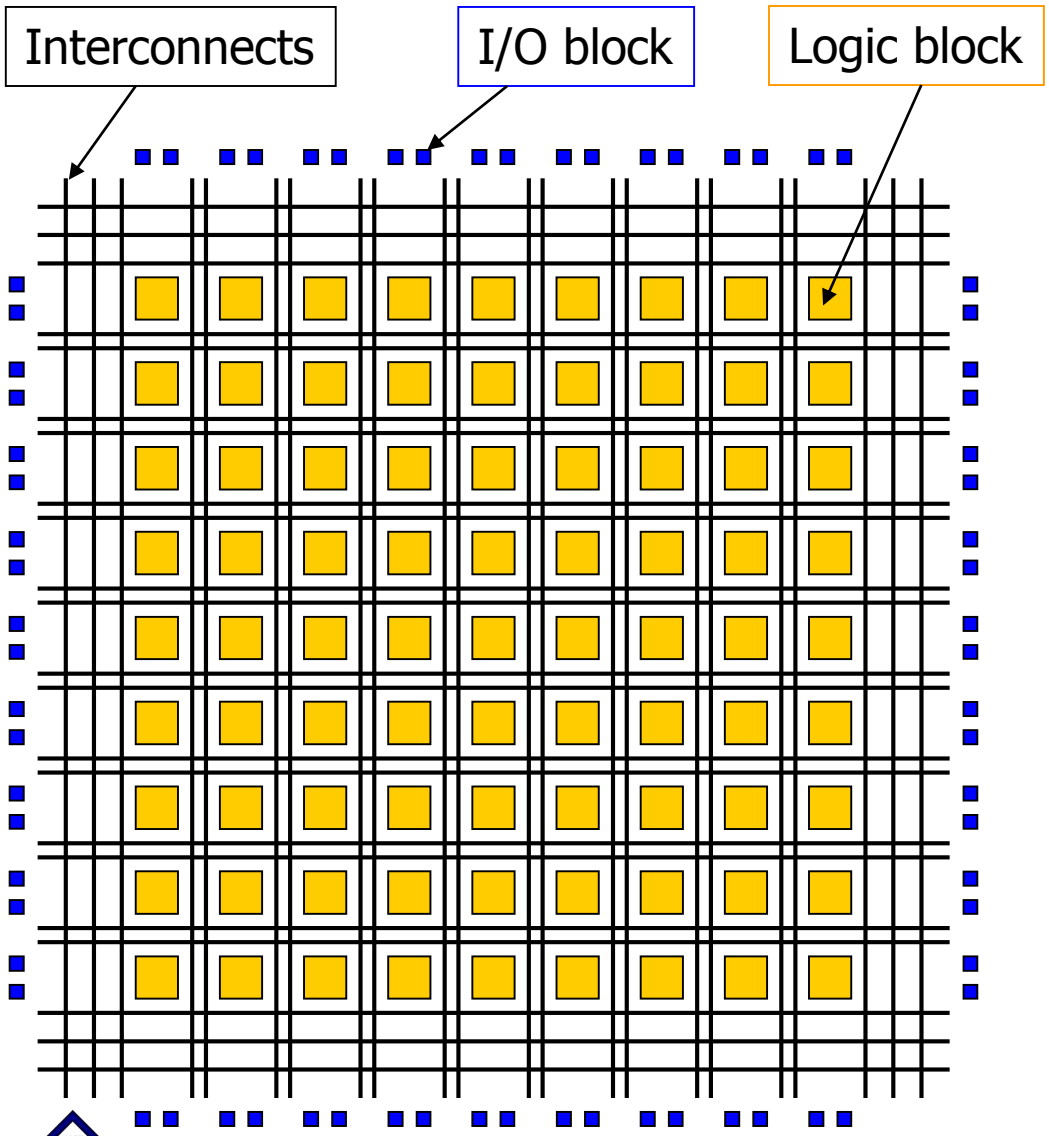
Complex Programmable Logic Devices



- Multiple PLDs (e.g. PALs, PLAs) with programmable interconnection structure
- Complex Programmable Logic Devices (CPLD) are very popular nowadays
- Pioneered by Altera (a major player in the market)
- Consists of logic, IO, and interconnects blocks (all programmable)
- CPLD offers logic resources with a wide number of inputs (AND planes)

* Slide is courtesy of Hsien-Hsin Sean Lee

Field-Programmable Gate Array (FPGA)



- High logic capacity with large distributed interconnection structure
- Logic capacity ≈ number of 2-input NAND gates
- Offers more narrow logic resources
- Offer a higher ratio of Flip-flops to logic resources than CPLD
- SRAM-controlled switches (e.g., Pass transistors)
- HCPLD (High Capacity PLD) is often used to refer to both CPLD and FPGA

* Slide is courtesy of Hsien-Hsin Sean Lee

Conclusion

- ***Summary***
  - ➔Programmable logic devices
    (Concept of logic matrix, example applications)
  - ➔ Logic using Diode Biasing
    (AND-OR circuits using diodes settings)
  - ➔ Programmable devices and how to configure them
    (fuse management, port configurations and applications)
  - ➔ PLA, PAL, ROM and FPGA devices
    (architectures, key features, examples)
- ***Next Lecture***
  - ➔ Introduction to sequential circuits
  - ➔ Memory devices

Reading assignment: Sections 5.1 – 5.5 in the textbook