浙江大学
ZHEJIANG UNIVERSITY

Digital Logic Design

Chapter 6 – Digital Hardware Implementation

浙江大学计算机学院　王总辉
zhwang@zju.edu.cn
http://course.zju.edu.cn
2021年10月

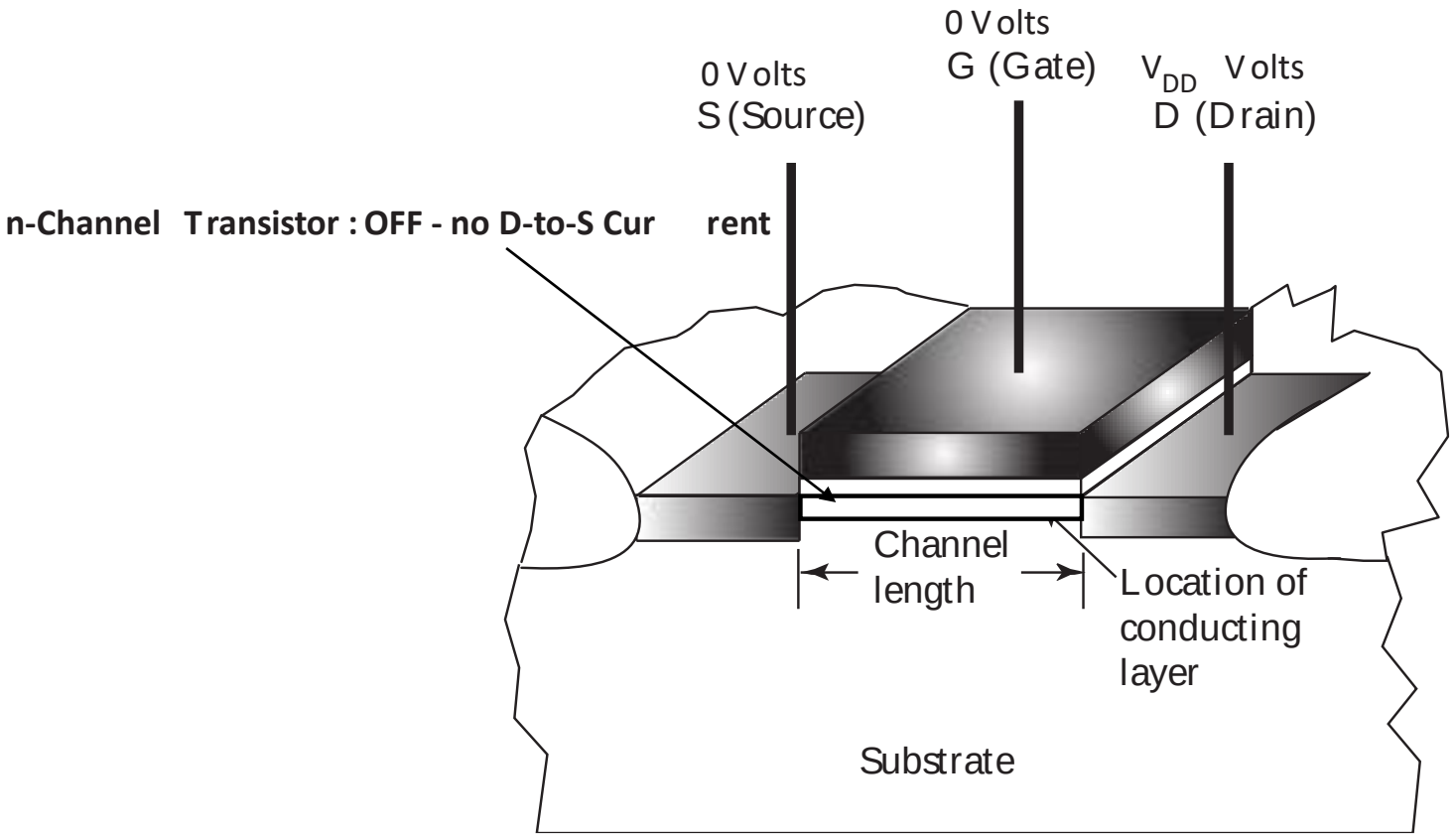
Overview

- **The Design Space**
  - Integrated Circuits
    - Levels of Integration
  - CMOS Circuit Technology
    - CMOS Transistor Models
    - Circuits of Switches
    - Fully Complementary CMOS Circuits
    - Technology Parameters
- **Programming Technologies**
  - ROMs
  - PLAs
  - PALs
  - Lookup Tables

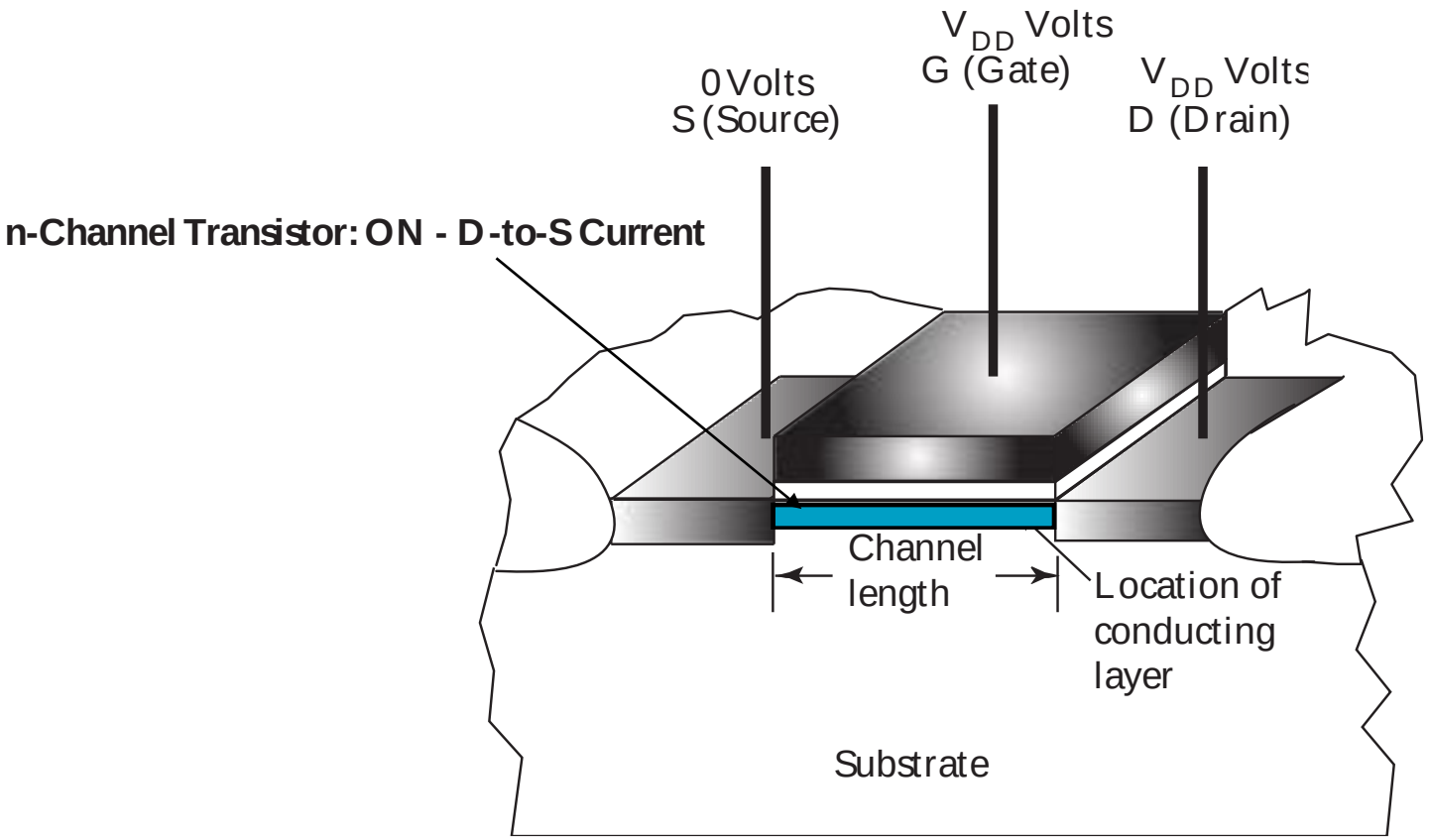
Integrated Circuits

- **Integrated circuit (informally, a "chip") is a semiconductor crystal (most often silicon) containing the electronic components for the digital gates and storage elements which are interconnected on the chip.**
- **Terminology - Levels of chip integration**
  - *SSI* (*small-scale integrated*) - fewer than 10 gates
  - *MSI* (*medium-scale integrated*) - 10 to 100 gates
  - *LSI* (*large-scale integrated*) - 100 to thousands of gates
  - *VLSI* (*very large-scale integrated*) - thousands to 100s of millions of gates

MOS Transistor

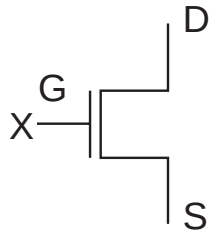


MOS Transistor



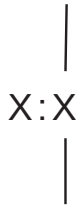
Switch Models for MOS Transistors

- **n-Channel – Normally Open (NO) Switch Contact**



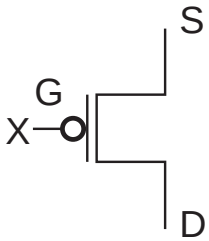
Symbol

Switch Model:



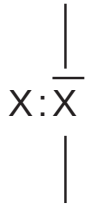
Simplifed
Switch Model

- **p-Channel – Normally Closed (NC) Switch Contact**



Symbol

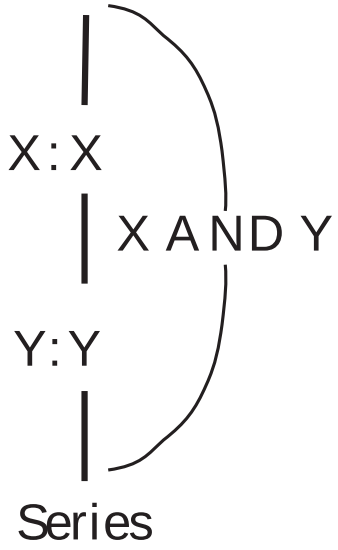
Switch Model

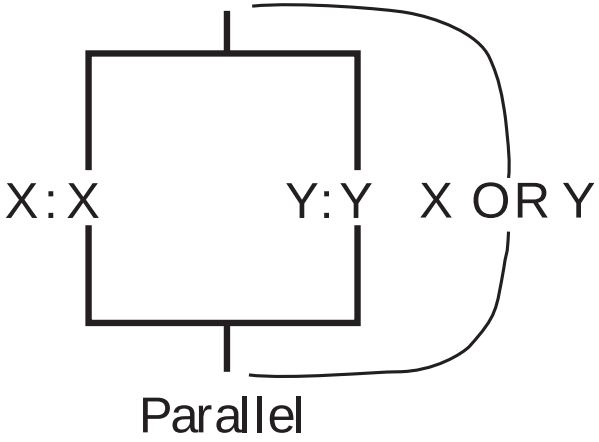


Simplified
Switch Model

Circuits of Switch Models

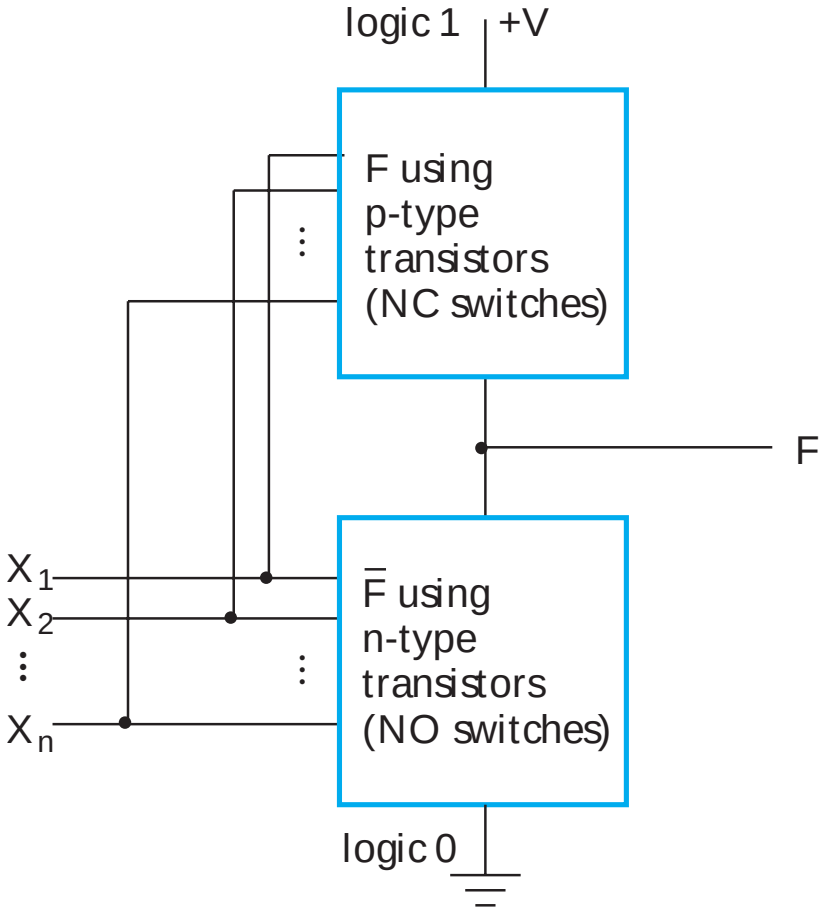
- **Series**
- **Parallel**





Fully-Complementary CMOS Circuit

- **Circuit structure for fully-complementary CMOS gate**



General Structure

CMOS Circuit Design Example

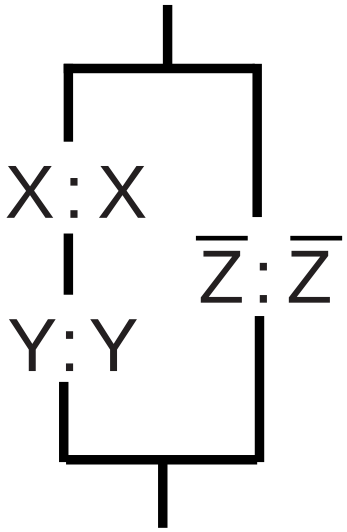
- **Find a CMOS gate with the following function:**

$$F = \overline{X}\,Z + \overline{Y}\,Z = (\overline{X} + \overline{Y})Z$$

- **Beginning with F0, and using $\overline{F}$**

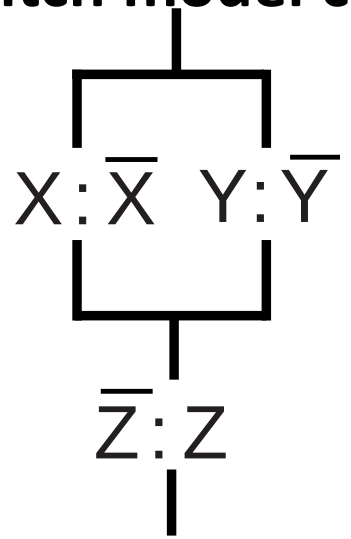
$$\text{F0 Circuit: } \overline{F} = X\,Y + \overline{Z}$$

- **The switch model circuit in terms of NO switches:**



CMOS Circuit Design Example

- **The switch model circuit for F1 in terms of NC contacts is the dual of the switch model circuit for F0:**



- **The function for this circuit is:**

$$F1\ \text{Circuit}: F = (\overline{X} + \overline{Y})\ Z$$

which is the correct F.

Technology Parameters

- **Specific gate implementation technologies are characterized by the following parameters:**
  - *Fan-in* – the number of inputs available on a gate
  - *Fan-out* – the number of standard loads driven by a gate output
  - *Noise Margin* – the maximum external noise voltage superimposed on a normal input value that will not cause an undesirable change in the circuit output
  - *Cost for a gate* - a measure of the contribution by the gate to the cost of the integrated circuit
  - *Propagation Delay* – The time required for a change in the value of a signal to propagate from an input to an output
  - *Power Dissipation* – the amount of power drawn from the power supply and consumed by the gate

Fan-out

- **Fan-out can be defined in terms of a standard load**
  - Example: 1 standard load equals the load contributed by the input of 1 inverter.
  - *Transition time* -the time required for the gate output to change from H to L, t_{HL}, or from L to H, t_{LH}
  - The *maximum fan-out* that can be driven by a gate is the number of standard loads the gate can drive without exceeding its specified *maximum transition time*

Cost

- **In an integrated circuit:**
  - The cost of a gate is proportional to the chip area occupied by the gate
  - The gate area is roughly proportional to the number and size of the transistors and the amount of wiring connecting them
  - Ignoring the wiring area, the gate area is roughly proportional to the gate input count
  - So gate input count is a rough measure of gate cost
- **If the actual chip layout area occupied by the gate is known, it is a far more accurate measure**

Programming Technologies

- **Programming technologies are used to:**
  - Control connections
  - Build lookup tables
  - Control transistor switching
- **The technologies**
  - Control connections
    - Mask programming
    - Fuse
    - Antifuse
    - Single-bit storage element

Programming Technologies

- **The technologies (continued)**
  - Build lookup tables
    - Storage elements (as in a memory)
  - Transistor Switching Control
    - Stored charge on a floating transistor gate
      - Erasable
      - Electrically erasable
      - Flash (as in Flash Memory)
    - Storage elements (as in a memory)

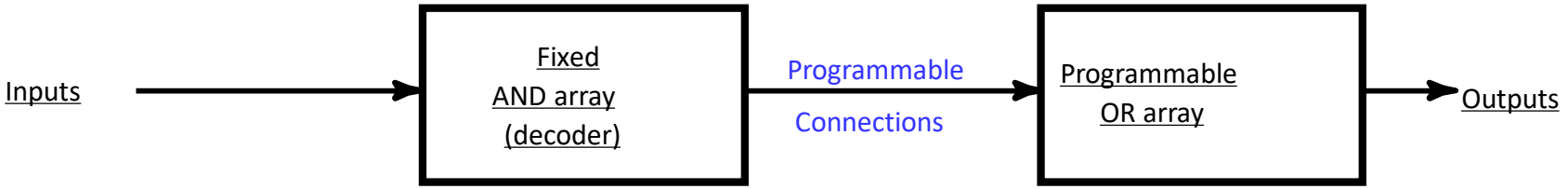
Technology Characteristics

- **Permanent - Cannot be erased and reprogrammed**
  - Mask programming
  - Fuse
  - Antifuse
- **Reprogrammable**
  - Volatile - Programming lost if chip power lost
    - Single-bit storage element
  - Non-Volatile
    - Erasable
    - Electrically erasable
    - Flash (as in Flash Memory)

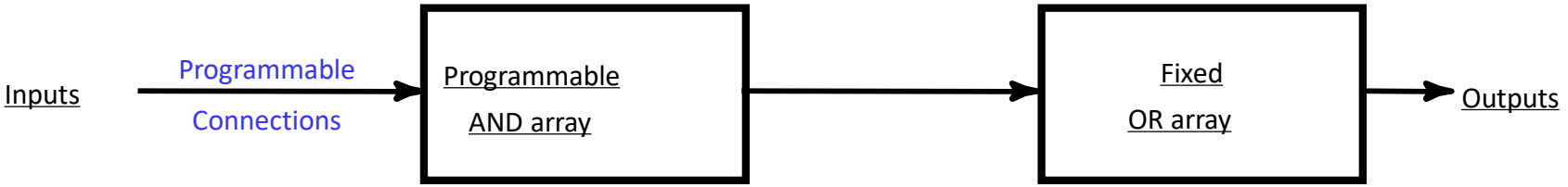
Programmable Configurations

- ***Read Only Memory*** **(*ROM*) - a fixed array of AND gates and a programmable array of OR gates**
- ***Programmable Array Logic*** **(*PAL*)Ò - a programmable array of AND gates feeding a fixed array of OR gates.**
- ***Programmable Logic Array*** **(*PLA*) - a programmable array of AND gates feeding a programmable array of OR gates.**
- ***Complex Programmable Logic Device*** **(*CPLD*) /*Field-Programmable Gate Array*** **(*FPGA*) - complex enough to be called "architectures" - See VLSI Programmable Logic Devices reading supplement**

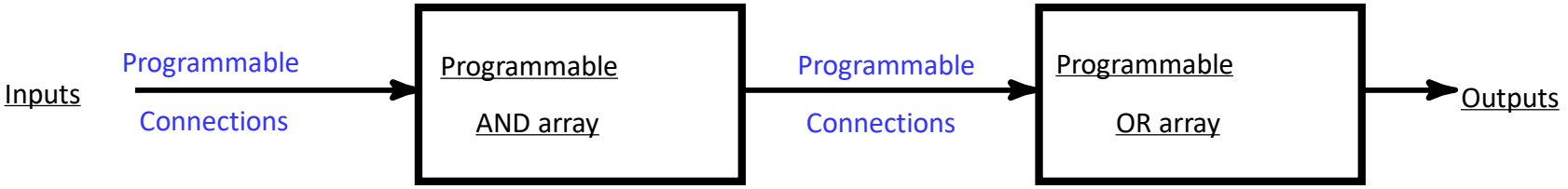
ROM, PAL and PLA Configurations



(a) Programmable read-only memory (PROM)

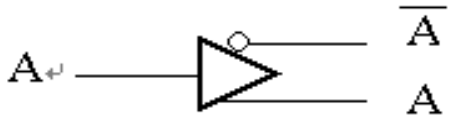


(b) Programmable array logic (PAL) device



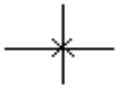
(c) Programmable logic array (PLA) device

Logical symbols



Buffer

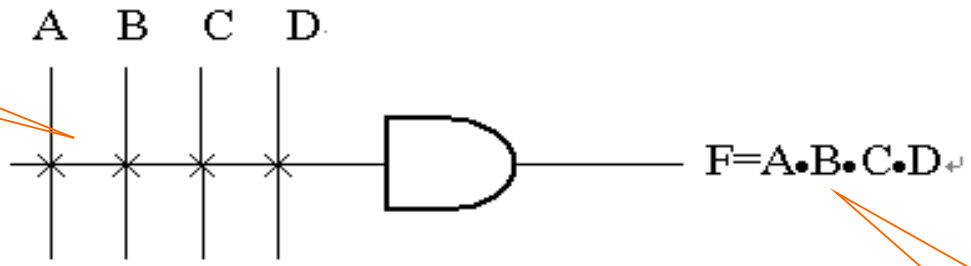
wire connecting
(not programmable)



wire connecting
(programmable)

Disconnection
(erasable)

AND Array



Array of AND **gates**
(programmable)

Product terms

Read Only Memory

- **Read Only Memories (ROM) or Programmable Read Only Memories (PROM) have:**
  - N input lines,
  - M output lines, and
  - 2^N decoded minterms.
- **Fixed AND array with 2^N outputs implementing all N-literal minterms.**
- **Programmable OR Array with M outputs lines to form up to M sum of minterm expressions.**

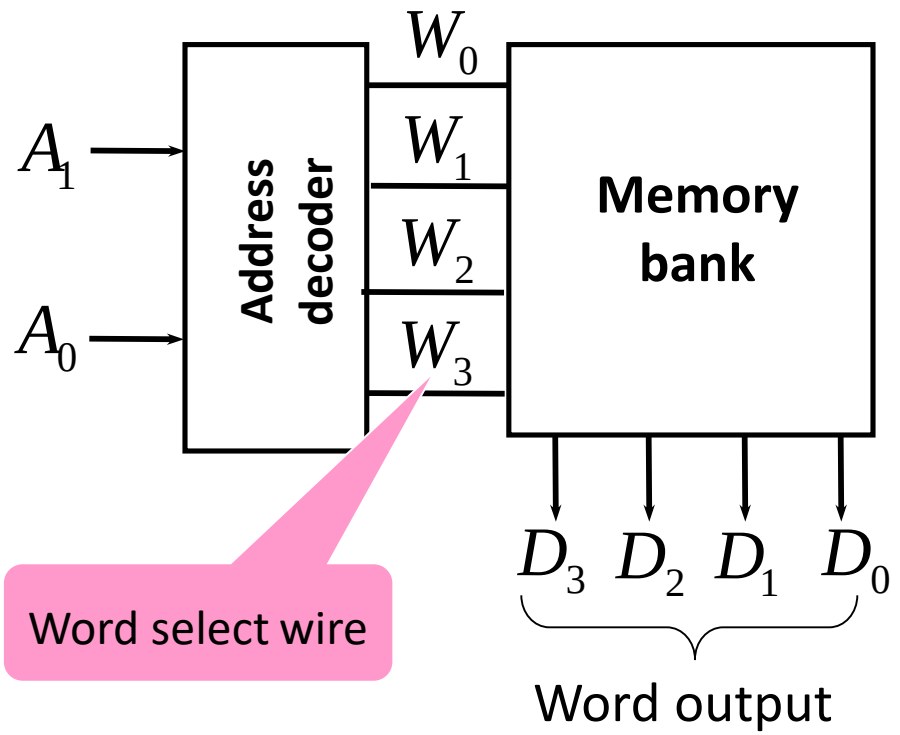
Read Only Memory-2

- **A program for a ROM or PROM is simply a multiple-output truth table**
  - If a 1 entry, a connection is made to the corresponding minterm for the corresponding output
  - If a 0, no connection is made
- **Can be viewed as a *memory* with the inputs as *addresses* of *data* (output values), hence ROM or PROM names!**

The general structure of the read-only memory

The general structure of the ROM

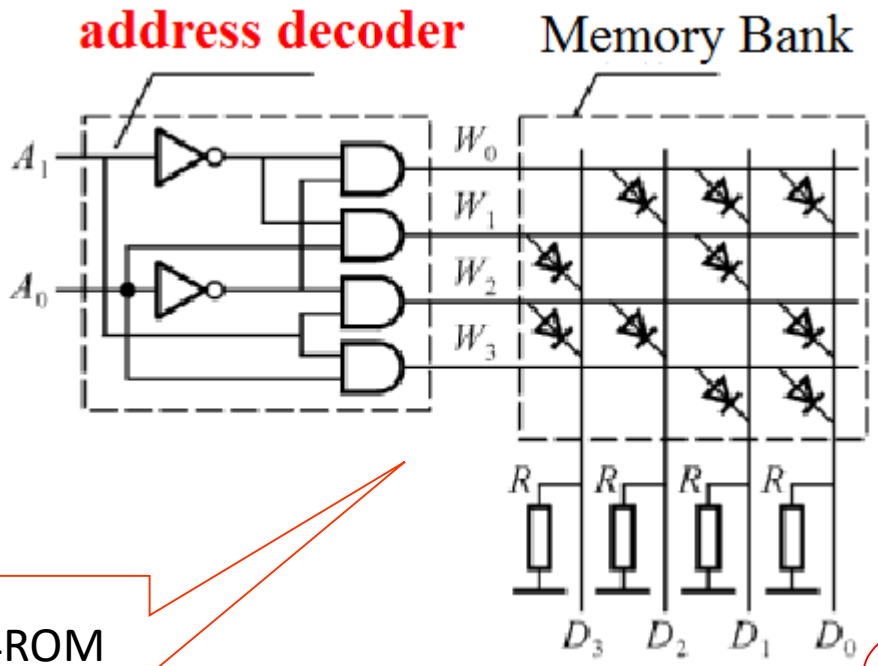
Size=2^2×4 bit=16 ROM structure

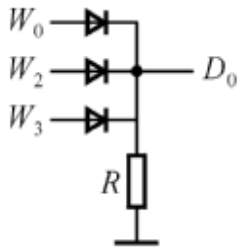


The logical structure of the read-only memory

The **address decoder** is a completely minterms (Full decoder) circuit, that is a non-programmable, “AND" array

The memory bank is a “OR" array

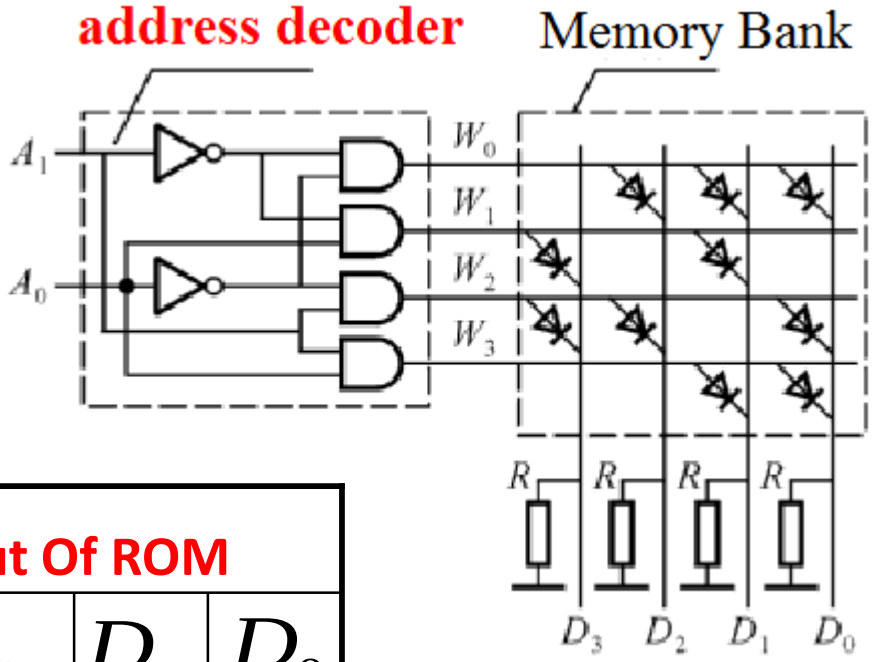




A 4 × 4ROM specific circuit

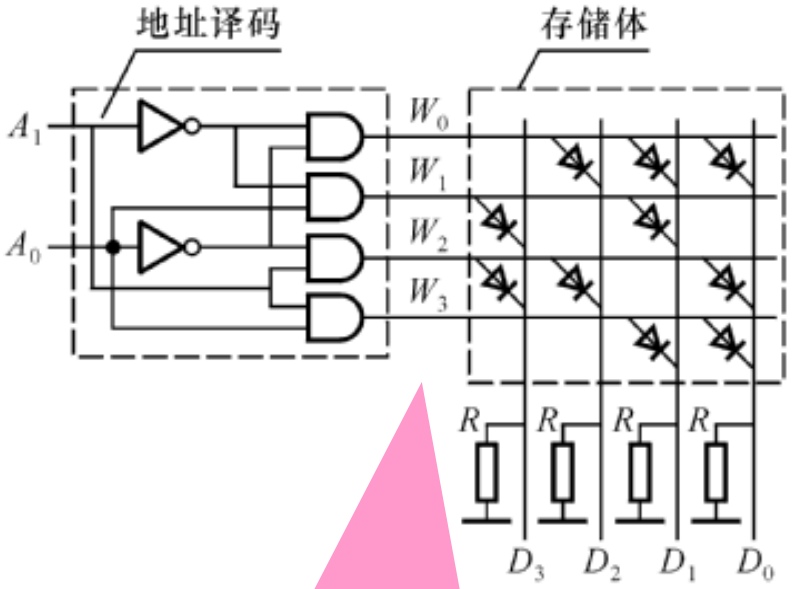
"D0" bit wire "OR" gate structure

Output information of ROM

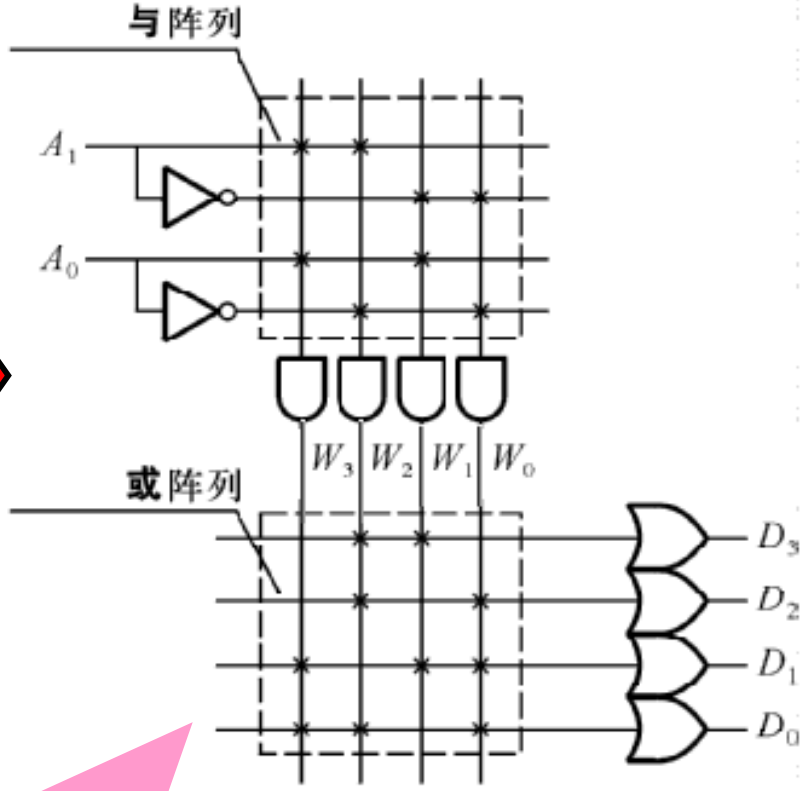


address		Word select wire	Output Of ROM			
A_1	A_0	W	D_3	D_2	D_1	D_0
0	0	W_0	0	1	1	1
0	1	W_1	1	0	1	0
1	0	W_2	1	1	0	1
1	0	W_3	0	0	1	1

ROM simplified logic symbol



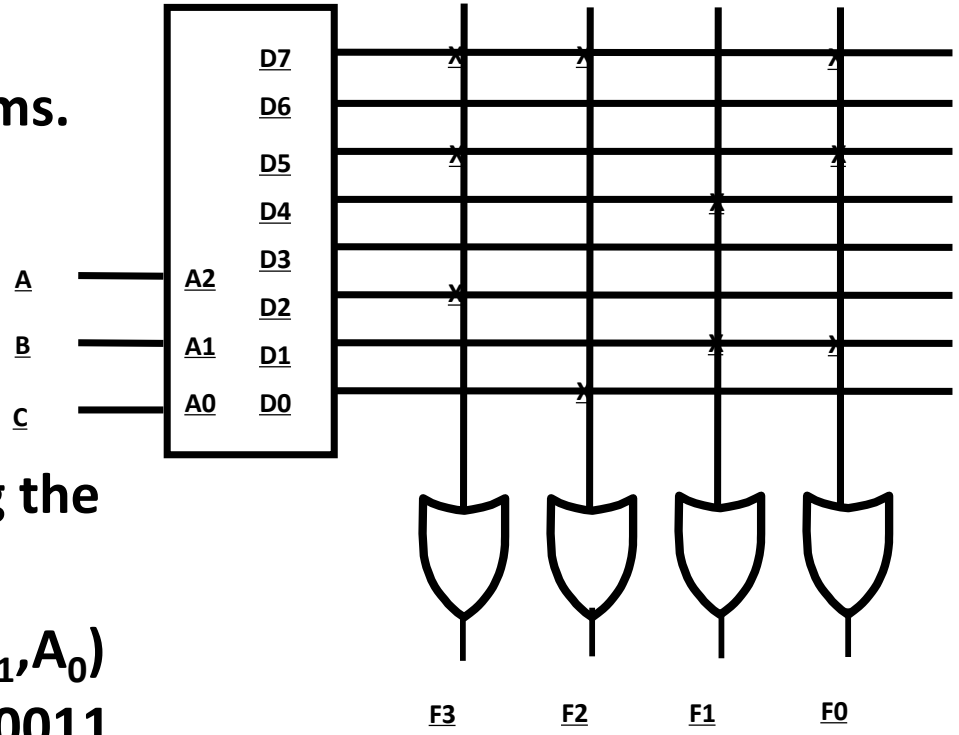
4×4ROM schematic



4×4ROM Simplified logic diagram

Read Only Memory Example

- Example: A 8 X 4 ROM (N = 3 input lines, M= 4 output lines)
- The fixed "AND" array is a "decoder" with 3 inputs and 8 outputs implementing minterms.
- The programmable "OR" array uses a single line to represent all inputs to an OR gate. An "X" in the array corresponds to attaching the minterm to the OR



- Read Example: For input (A_2,A_1,A_0) = 001, output is (F_3,F_2,F_1,F_0) = 0011.
- What are functions F_3, F_2 , F_1 and F_0 in terms of (A_2, A_1, A_0)?

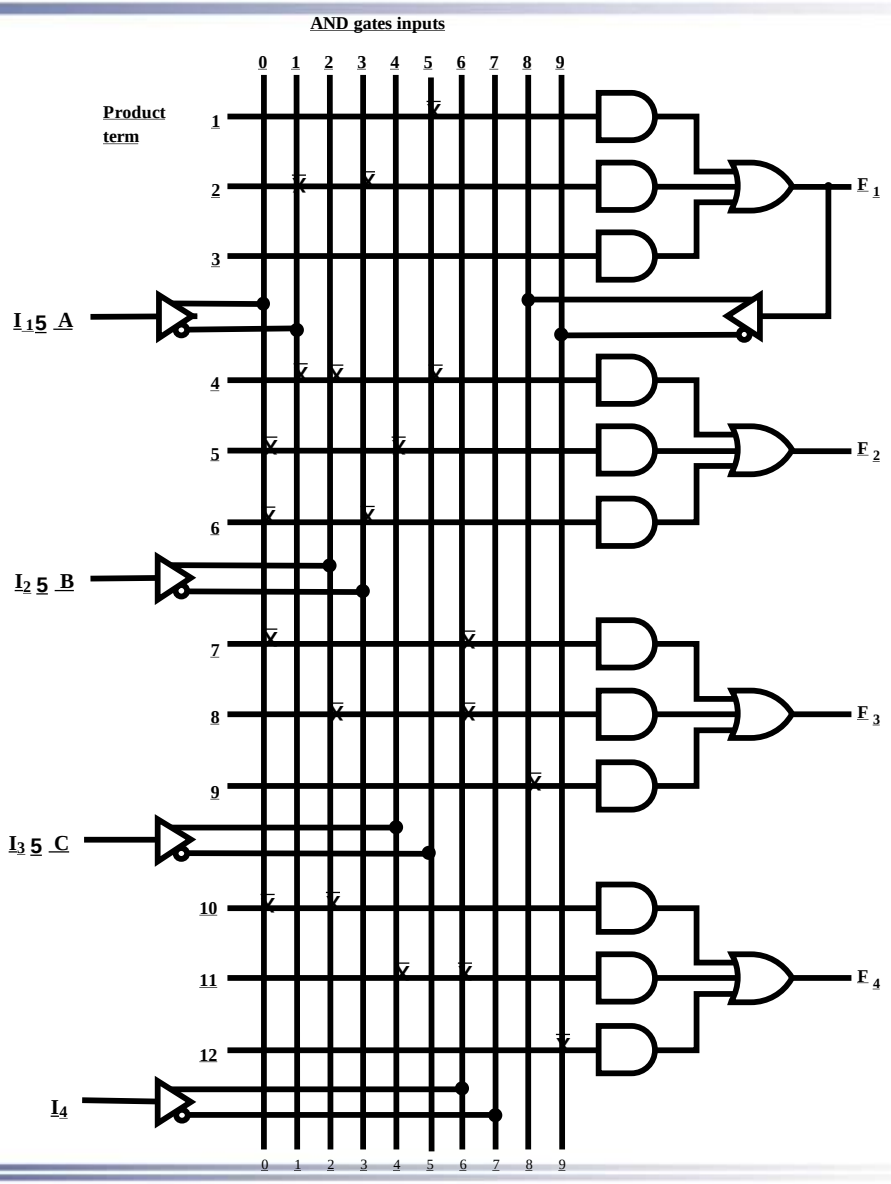
Programmable Array Logic (PAL)

- **The PAL is the opposite of the ROM, having a programmable set of ANDs combined with fixed ORs.**
- **Disadvantage**
  - ROM guaranteed to implement any M functions of N inputs. PAL may have too few inputs to the OR gates.
- **Advantages**
  - For given internal complexity, a PAL can have larger N and M
  - Some PALs have outputs that can be complemented, adding POS functions
  - No multilevel circuit implementations in ROM (without external connections from output to input). PAL has outputs from OR terms as internal inputs to all AND terms, making implementation of multi-level circuits easier.

Programmable Array Logic Example

- **4-input, 3-output PAL with fixed, 3-input OR terms**
- **What are the equations for F1 through F4?**

$$F1 = \overline{A}\,\overline{B} + \overline{C}$$
$$F2 = \overline{A}B\overline{C} + AC + A\overline{B}$$
$$F3 =$$
$$F4 =$$



Programmable Logic Array (PLA)

- **Compared to a ROM and a PAL, a PLA is the most flexible having a programmable set of ANDs combined with a programmable set of ORs.**
- **Advantages**
  - A PLA can have large N and M permitting implementation of equations that are impractical for a ROM (because of the number of inputs, N, required
  - A PLA has all of its product terms connectable to all outputs, overcoming the problem of the limited inputs to the PAL Ors
  - Some PLAs have outputs that can be complemented, adding POS functions

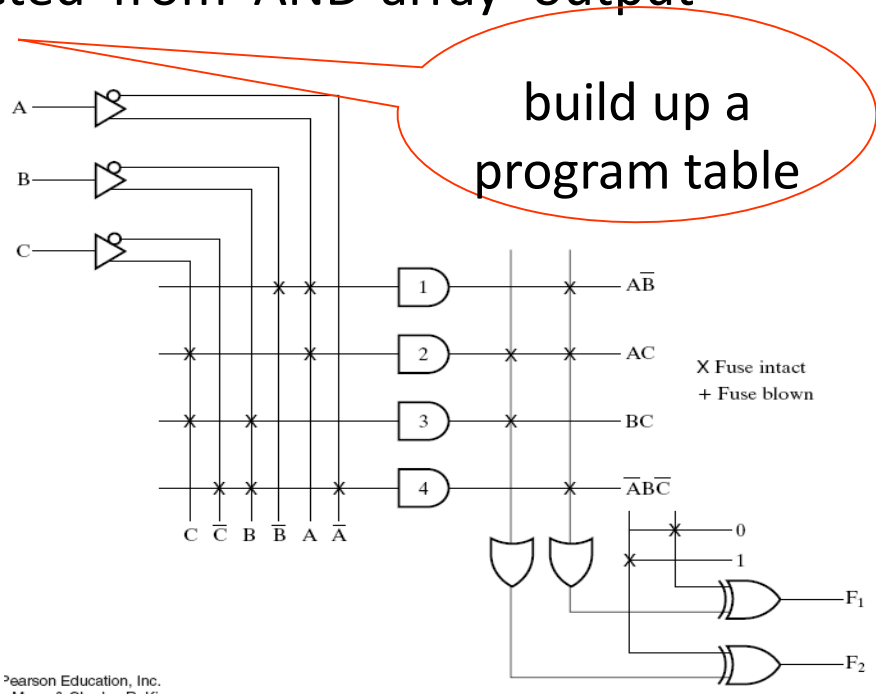
Programmable Logic Array (PLA)

- **Disadvantages**
  - Often, the product term count limits the application of a PLA.
  - Two-level multiple-output optimization is required to reduce the number of product terms in an implementation, helping to fit it into a PLA.
  - Multi-level circuit capability available in PAL not available in PLA. PLA requires external connections to do multi-level circuits.

Programmable Logic Array Example

- **Use of the PLA to implement a set of functions:**
  - First, must be listed AND-Items of function (As much as possible many public items)
  - Specifies whether need to be connected form the input to the AND gate array
  - Specifies whether need to be connected from AND-array- output to the OR-array
  - Whether the output is to be negated

	Product term	Inputs A	Inputs B	Inputs C	Outputs (T) F_1	Outputs (C) F_2
$A\bar{B}$	1	1	0	—	1	—
AC	2	1	—	1	1	1
BC	3	—	1	1	—	1
$\bar{A}B\bar{C}$	4	0	1	0	1	—



Generic array logic structure GAL

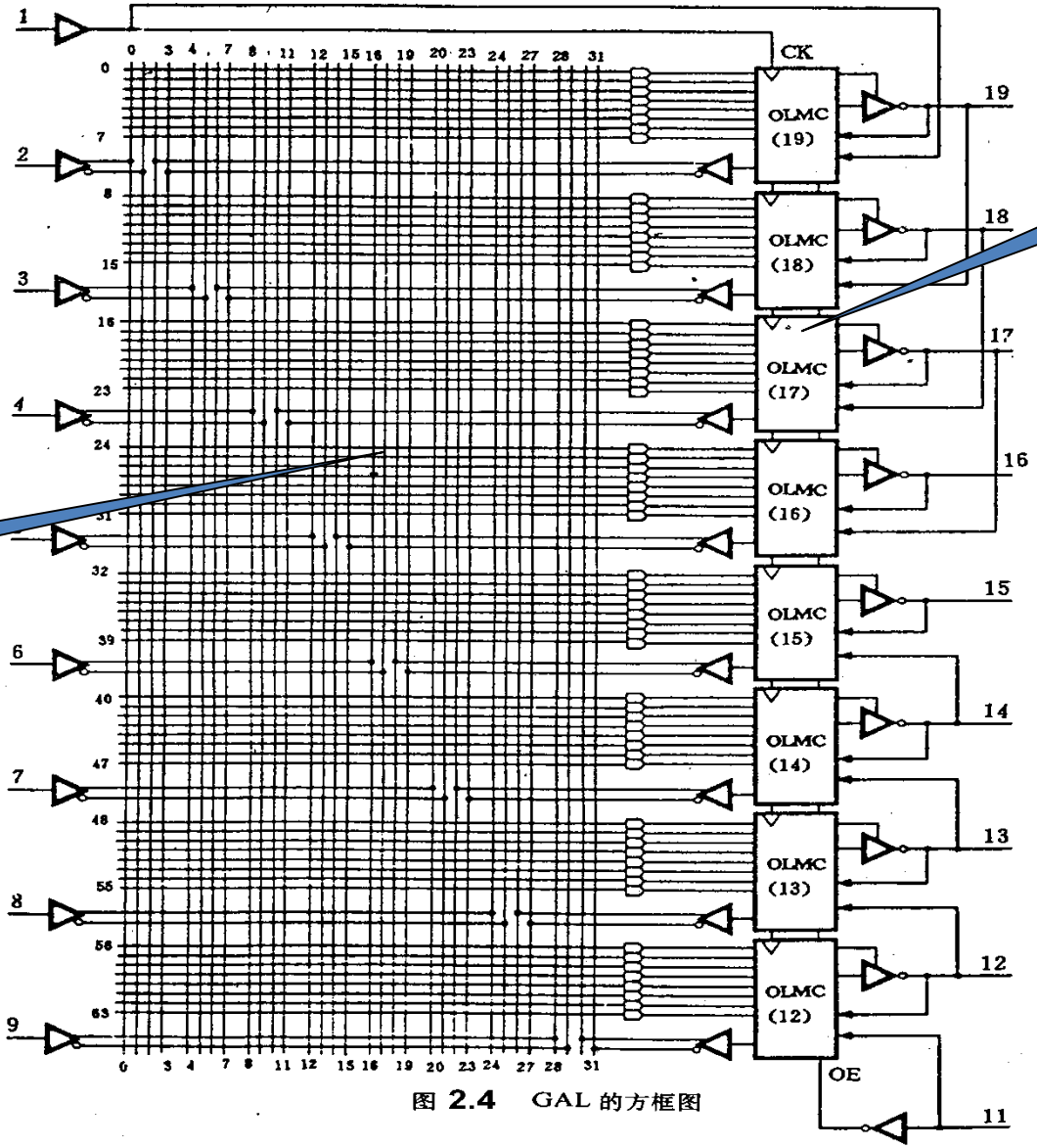


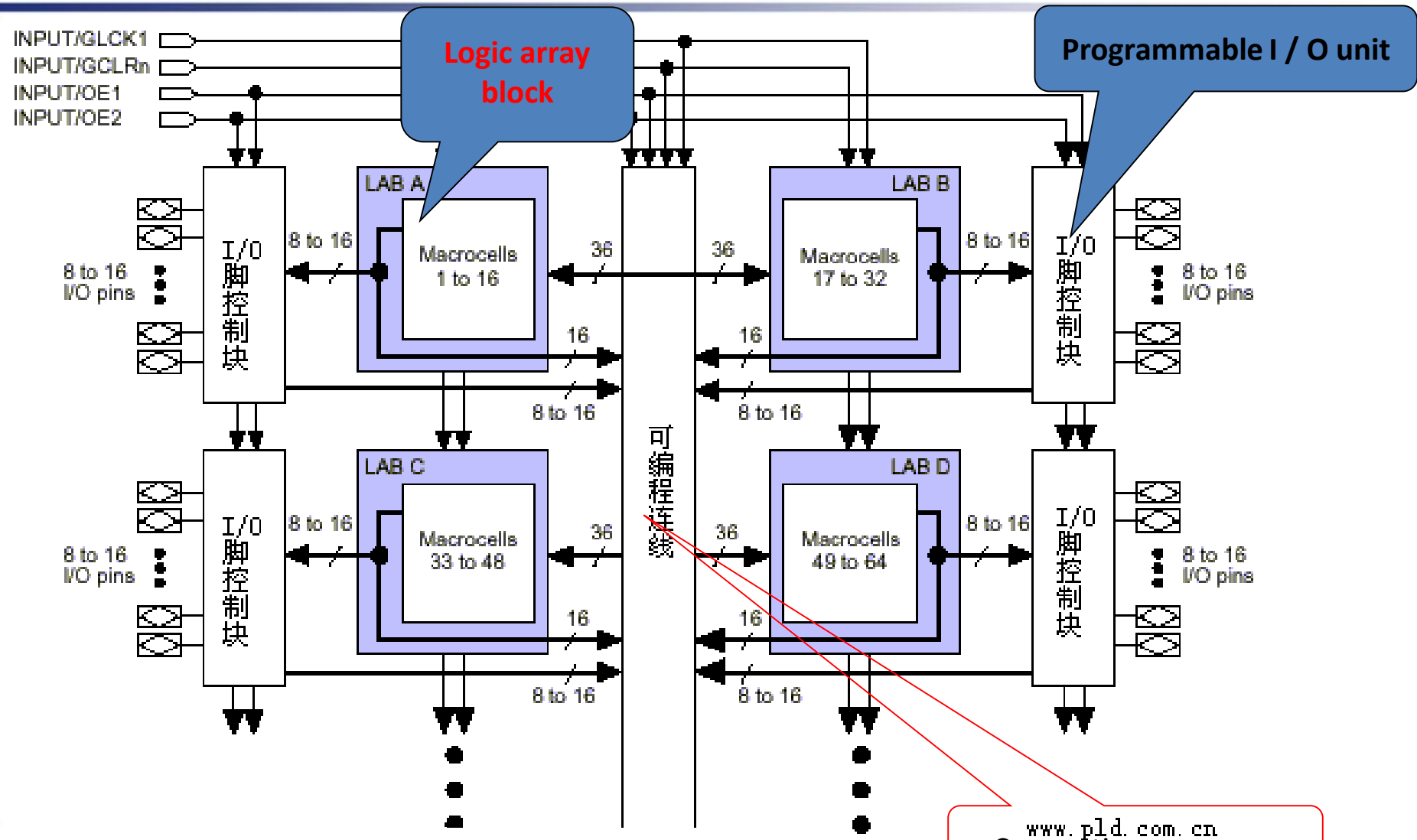
图 2.4 GAL 的方框图

OLMC

Logic macrocells

浙江大学 ZheJiang University

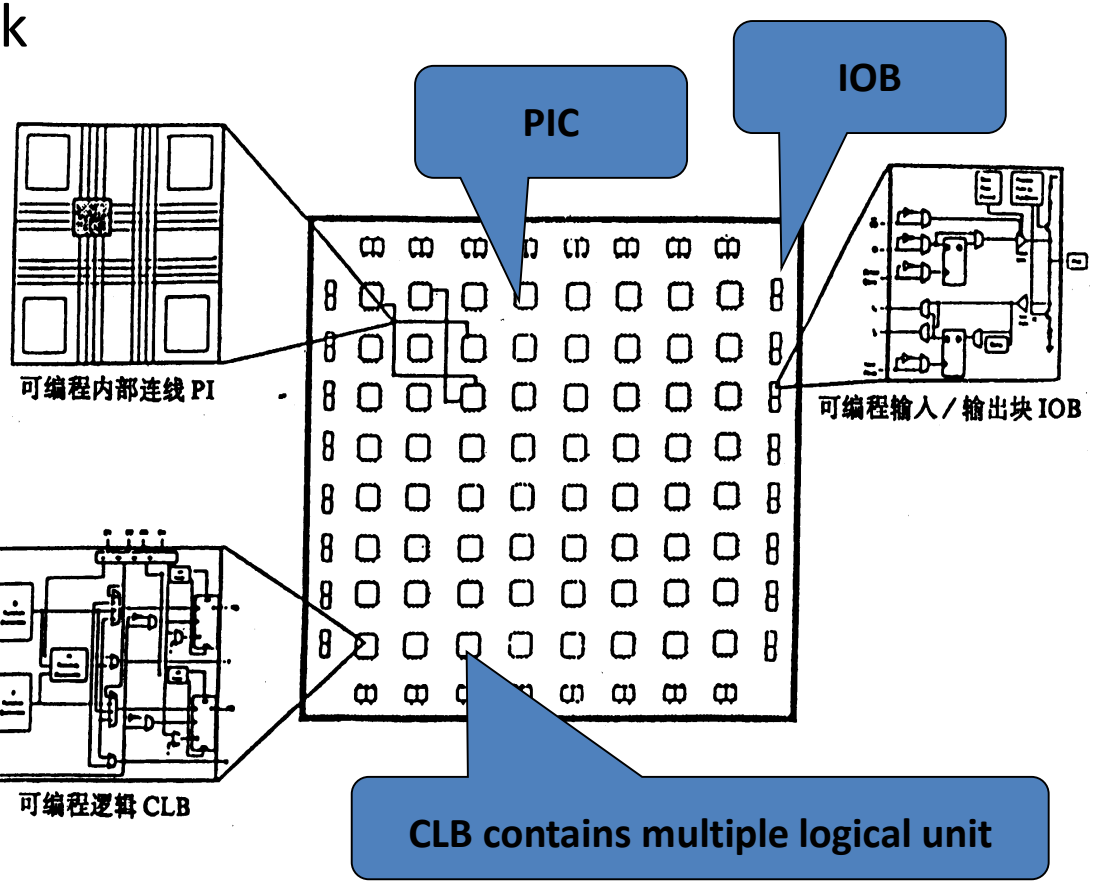
Complex Programmable Logic Devices CPLD
（Altera MAX7000S Series）



- Logic array block contains multiple macrocells

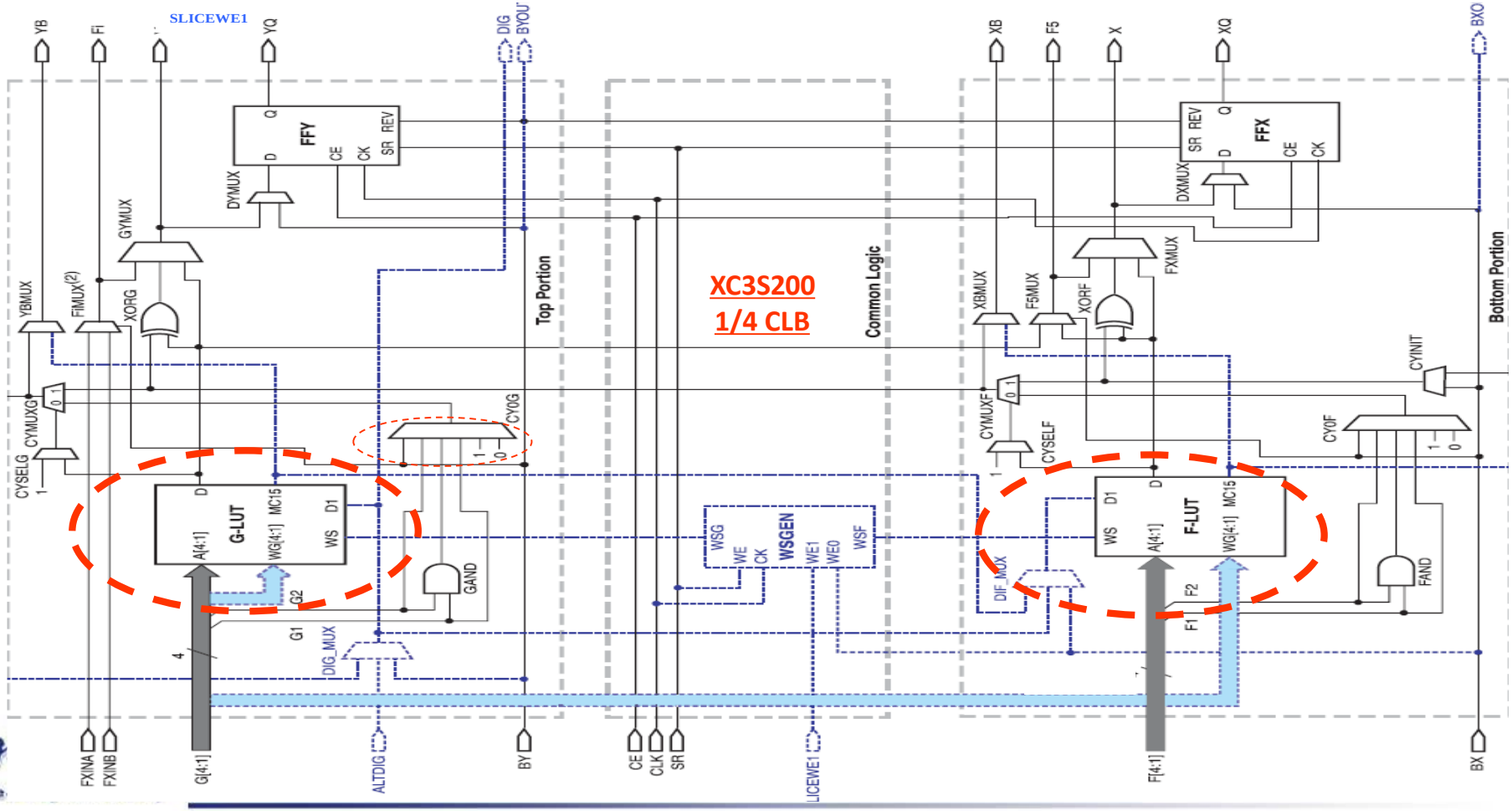
Programmable Gate Array: FPGA

- **The internal structure is known as LCA (Logic Cell Array) is composed of three parts:**
  - Programmable logic block （CLB）
  - Programmable in-put output module (IOB)
  - Programmable internal connection （PIC）

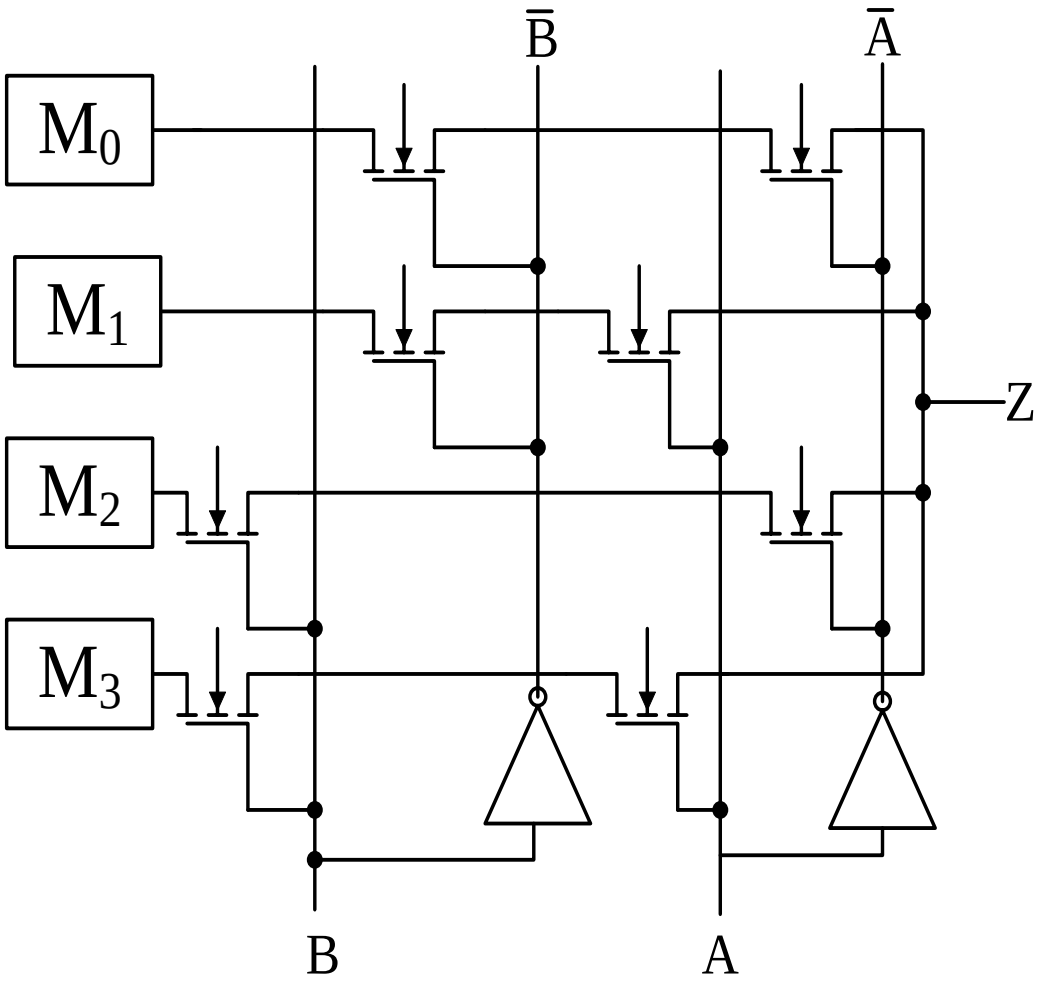


Configurable logic block

- **CLB（Configuration Logic Block）**
  - Is The Core of FPGA programmable universal logic function , composed of a RAM-based look-up table (LUT) may also be used as a memory.(480个CLB， 24×20）



Switch array look-up table Principle



Configuration Table

Store	B	A	Z
M_0	0	0	M_0
M_1	0	1	M_1
M_2	1	0	M_2
M_3	1	1	M_3

Combinational Logic Implementation -ROM

- **ROM is constituted by the variable decoder and OR gates**
- **function truth table is stored in the ROM**
- **Boolean function can be implemented in software**

Inputs			Outputs						Square for 3 bits
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

B0=A0

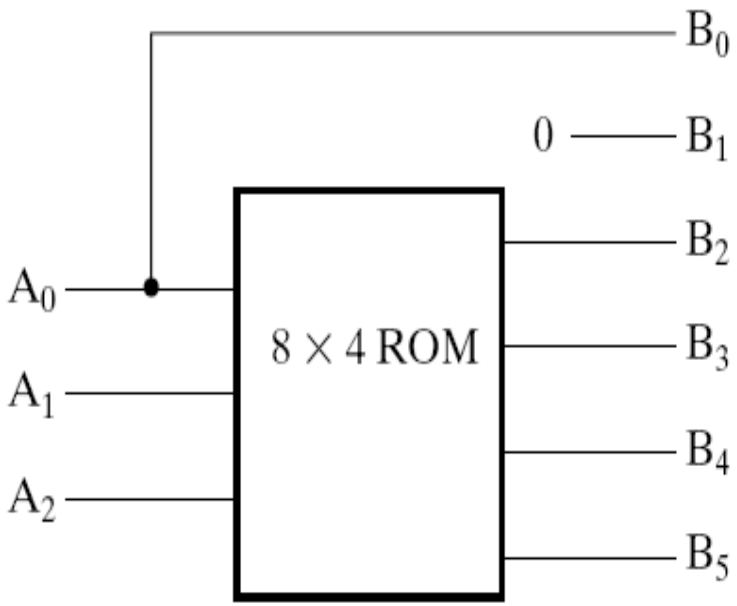
B1=”0”

ROM truth table

can select 8 × 4bit ROM

Function input corresponding to the ROM address

ROM output corresponding to the square value



ROM truth table

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

How to choose the ROM?

Combinational Logic Implementation-PLAs

- **Implement functions**

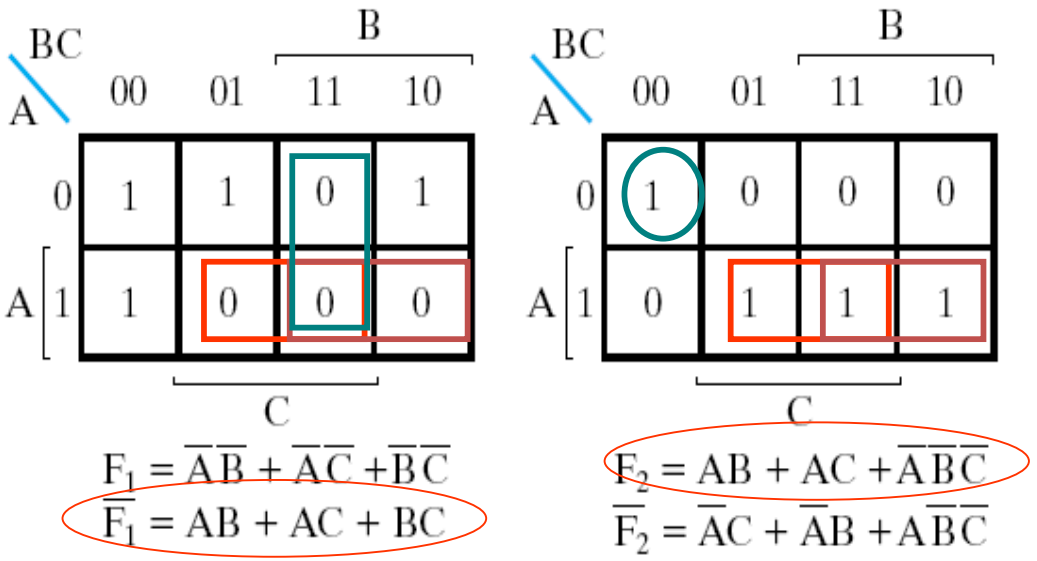
  F1(A,B,C)

  =Σm(0,1,2,4)

  F2(A,B,C)

  =Σm(0,5,6,7)

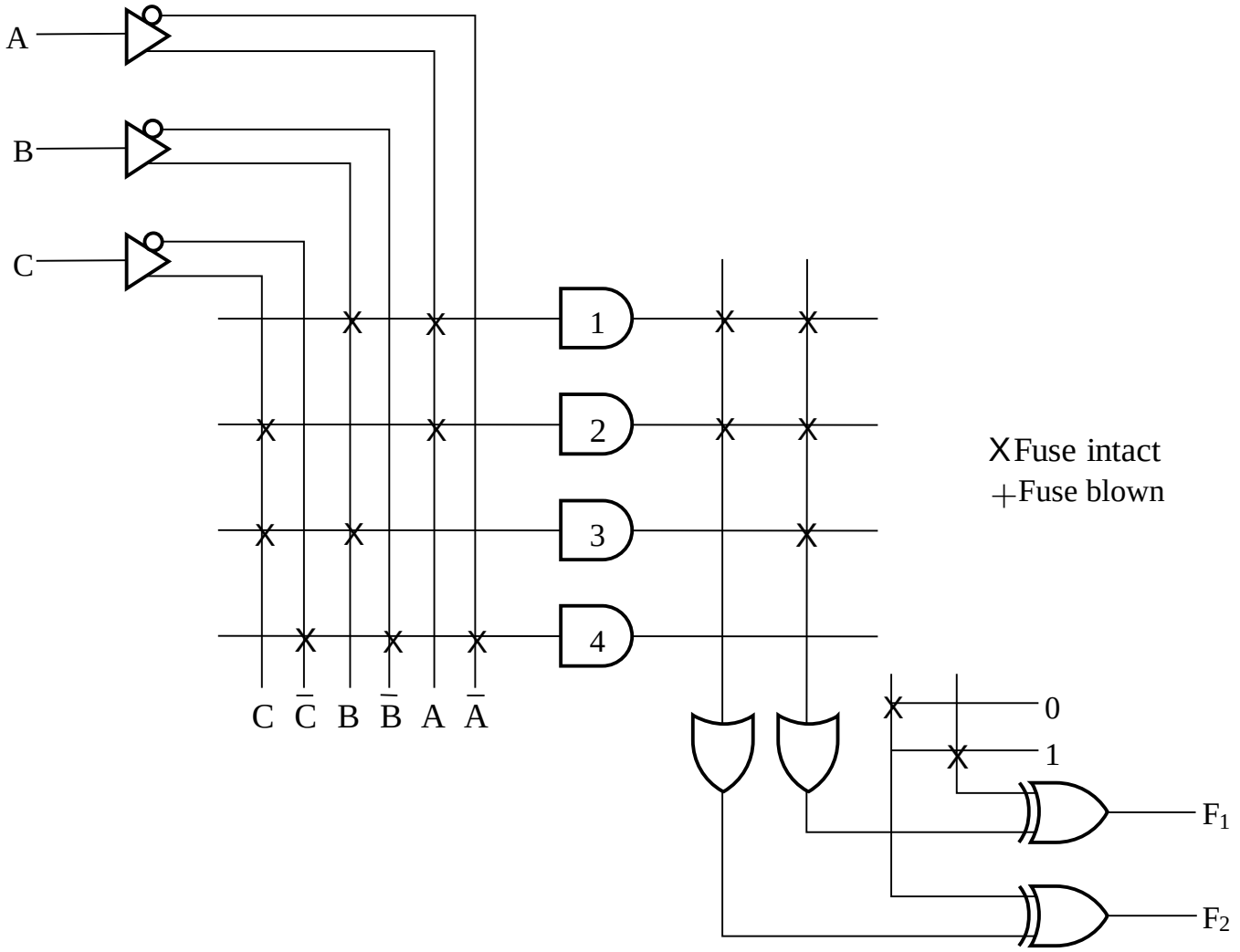
- **Use of K-map simplification**
- **How the entries at least (shared)?**
- **Completion of the programming table**



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	–
$\overline{A}\,\overline{B}\,\overline{C}$	4	0	0	0	–	1

浙江大學 ZheJiang University


A
B
C
1
2
3
4
C $\bar{C}$ B $\bar{B}$ A $\bar{A}$
X Fuse intact
+ Fuse blown
0
1
F_1
F_2

Combinational Logic Implementation-PALs

- **Implement functions :**

  W(A,B,C,D)=Σm(1,12,13)

  X(A,B,C,D)=Σm(7,8,9,10,11,12,13,14,15)

  Y(A,B,C,D)=Σm(0,2,3,4,5,6,7,8,9,10,11,15)

  Z(A,B,C,D)=Σm(1,2,8,12,13)

  simplification follows :

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

$$X=A+BCD$$

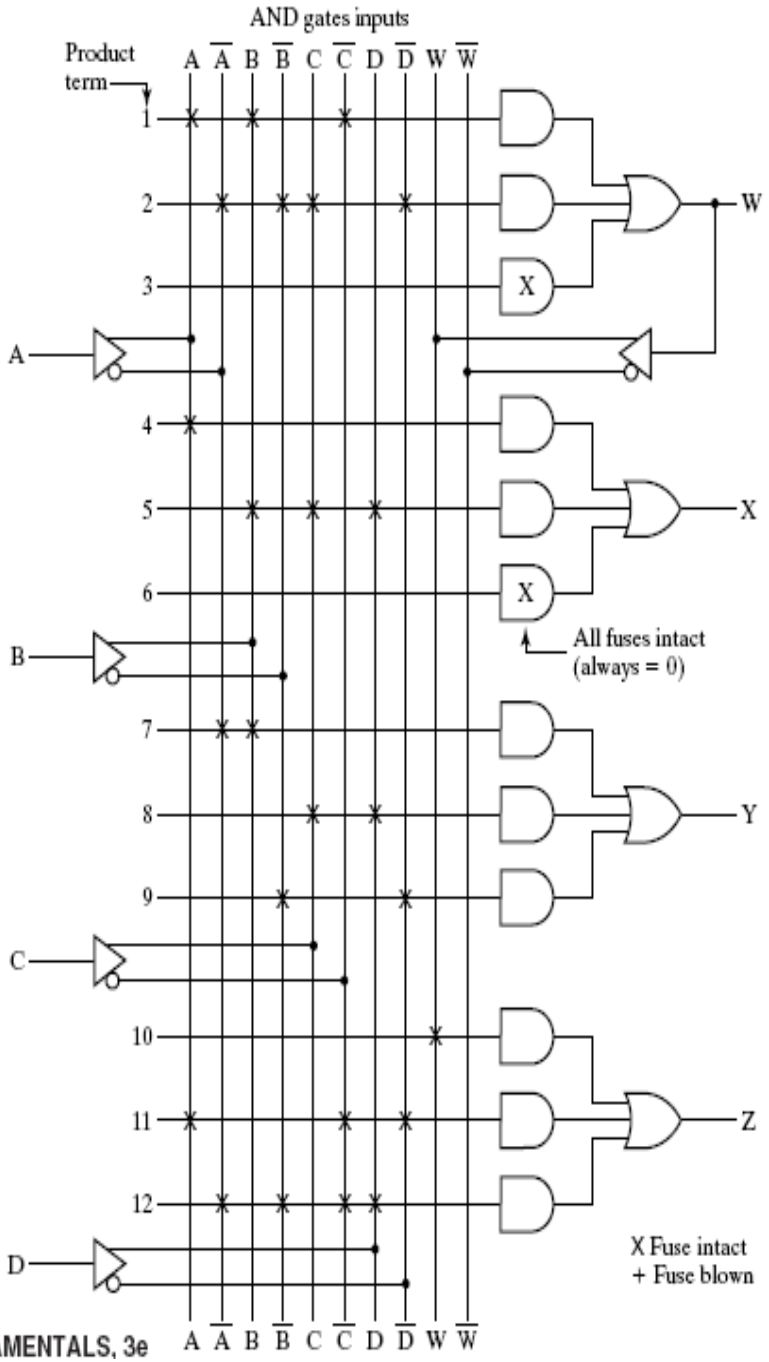
$$Y=\overline{A}B+CD+\overline{B}\overline{D}$$

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

$$=W+A\overline{C}\overline{D}+\overline{A}BC\overline{D}$$

PAL Implement connection

Product term	AND Inputs A	B	C	D	W	Outputs
1	1	1	0	—	—	$W = A\overline{B}\overline{C}$
2	0	0	1	0	—	$+\overline{A}\overline{B}C\overline{D}$
3	—	—	—	—	—	
4	1	—	—	—	—	$X = A$
5	—	1	1	1	—	$+BCD$
6	—	—	—	—	—	
7	0	1	—	—	—	$Y = \overline{A}B$
8	—	—	1	1	—	$+CD$
9	—	0	—	0	—	$+\overline{B}\overline{D}$
10	—	—	—	—	1	$Z = W$
11	1	—	0	0	—	$+A\overline{C}\overline{D}$
12	0	0	0	1	—	$+\overline{A}\overline{B}\overline{C}D$



Thank You !