EE2000 Logic Circuit Design

## Lecture 6 – Programmable Logic Devices

mouser.com

What will you learn?

6.1 What are Programmable Logic Devices

6.2 Implement a Boolean function using a Programmable Read-only Memory (PROM)

6.3 Implement a Boolean function using a Programming Array Logic (PAL)

6.4 Implement a Boolean function using a Programming Logic Array (PLA)

6.5 What are the basic components of a Field Programmable Gate Array Logic (FPGA) and their functions

6.1 Programmable Logic Devices

- Logic circuits constructed using discrete logic gates for a fixed function are not reconfigurable/ programmable

- Programmable Logic Devices (PLDs) are integrated circuit chips that have undefined function at the time of manufacture

- It has to be programmed by user to implement the desired function
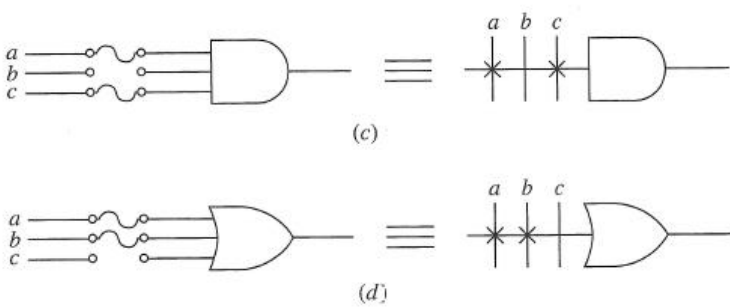
- For design and prototyping

General Structure of PLD

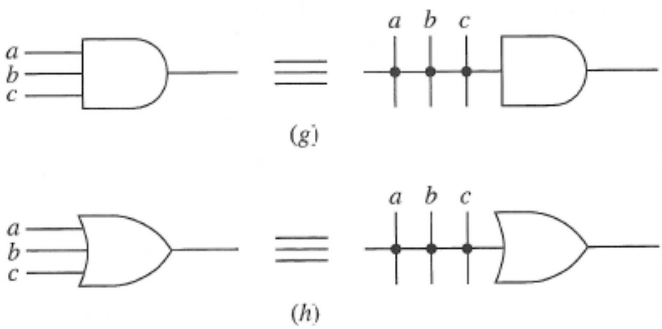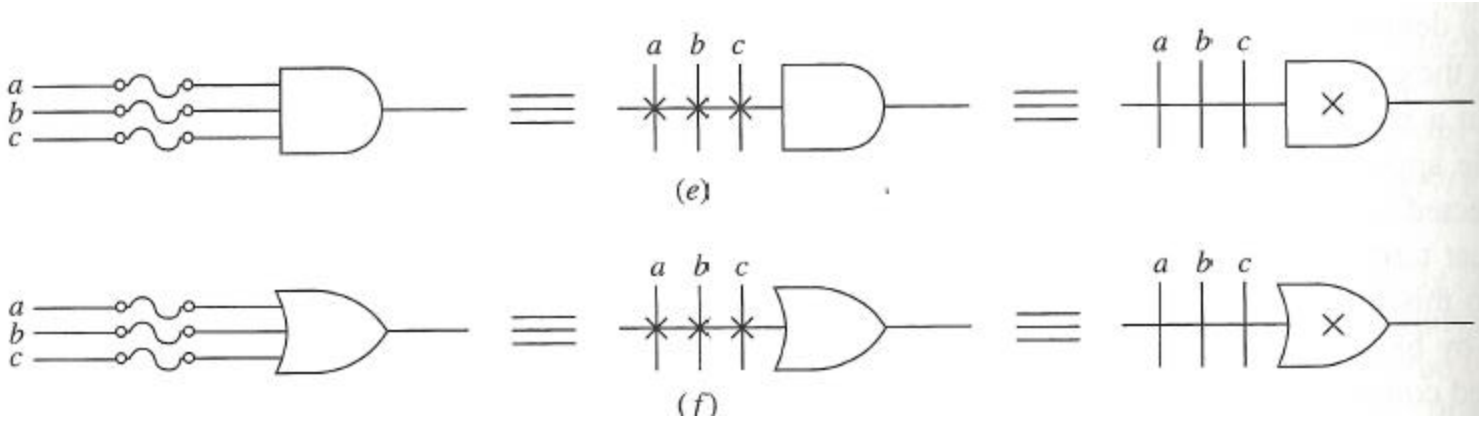To implement two-level combinational logic in Sum-of-Products Form
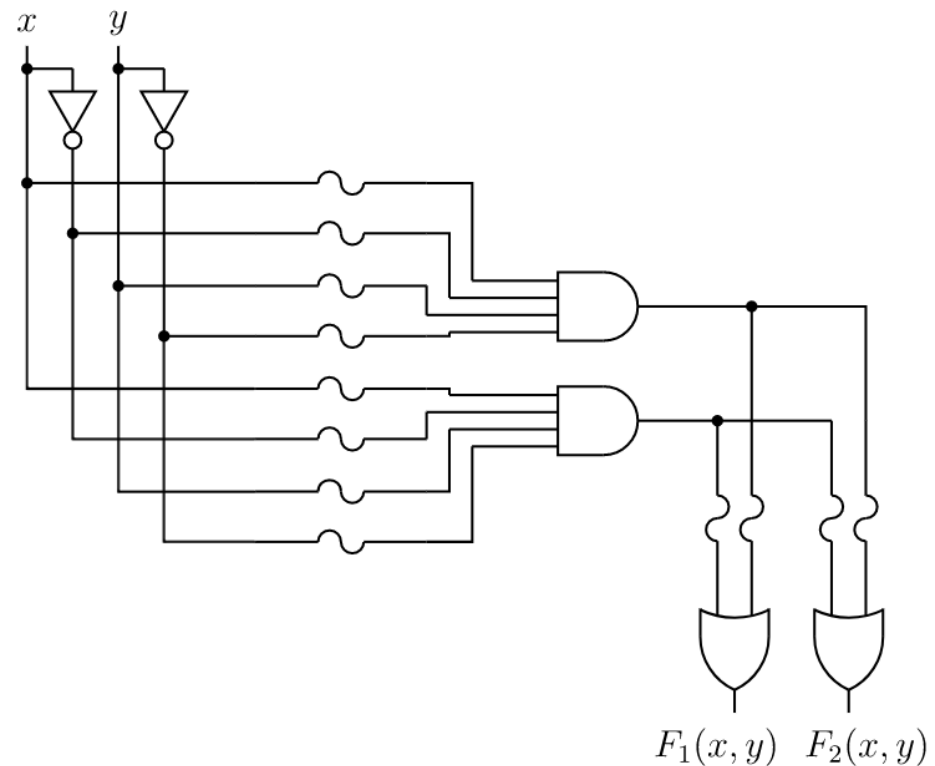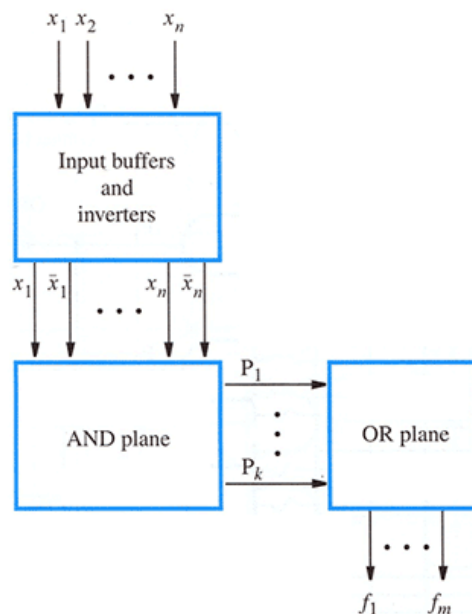
PLD Notation

Fusible link

(c)

(d)

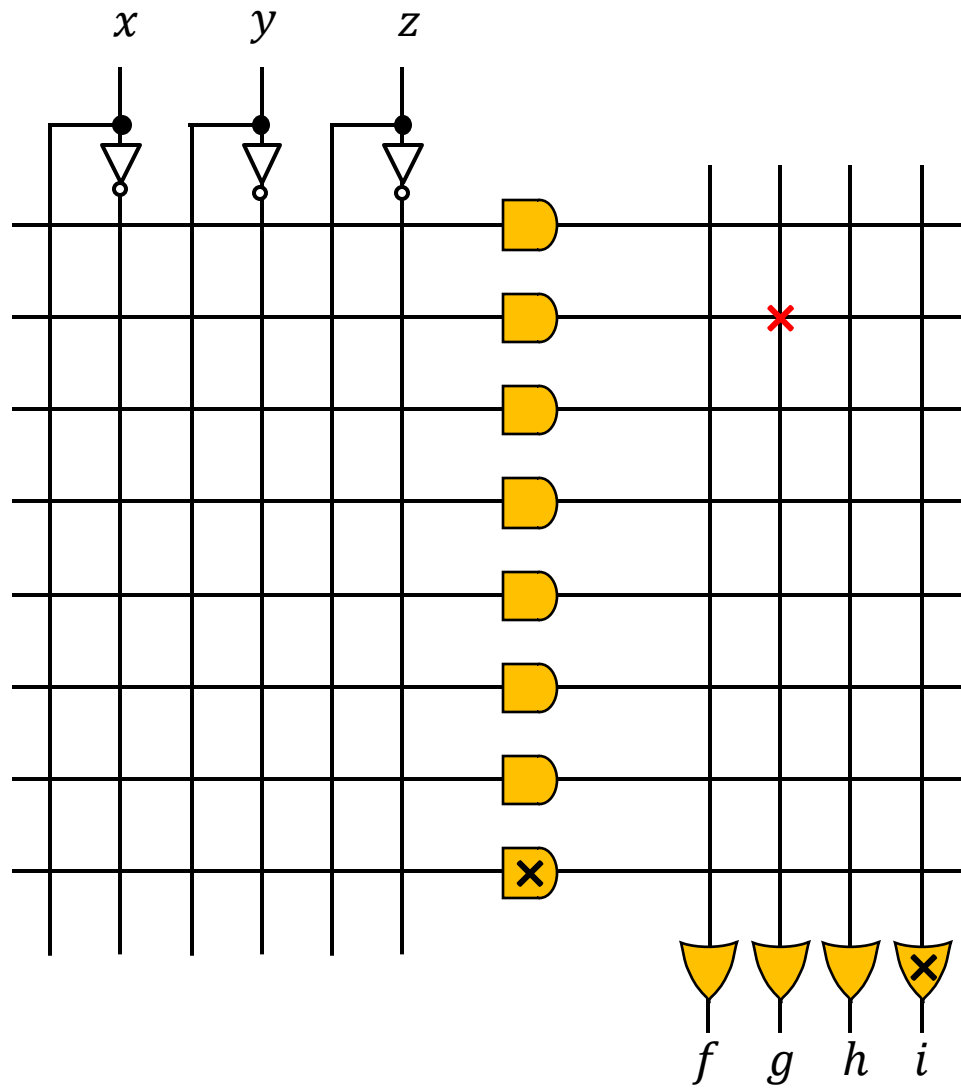Hard-wired connection
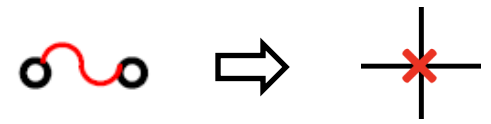

(g)

(h)

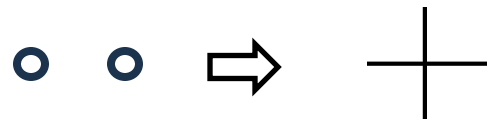All fuses are intact

(e)

(f)

PLD Notation

Closed connection
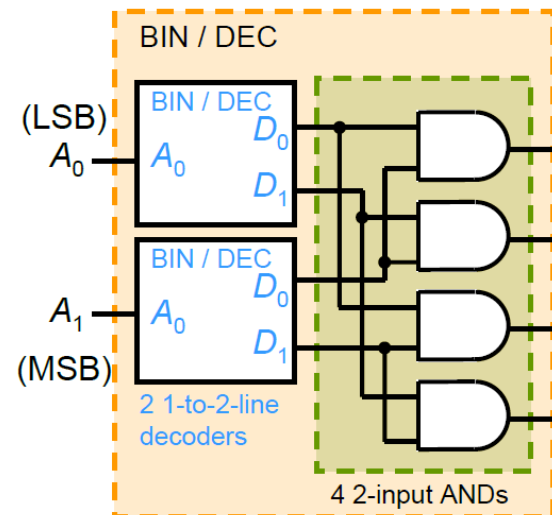
Open connection

Fixed connection at factory

All connected

6

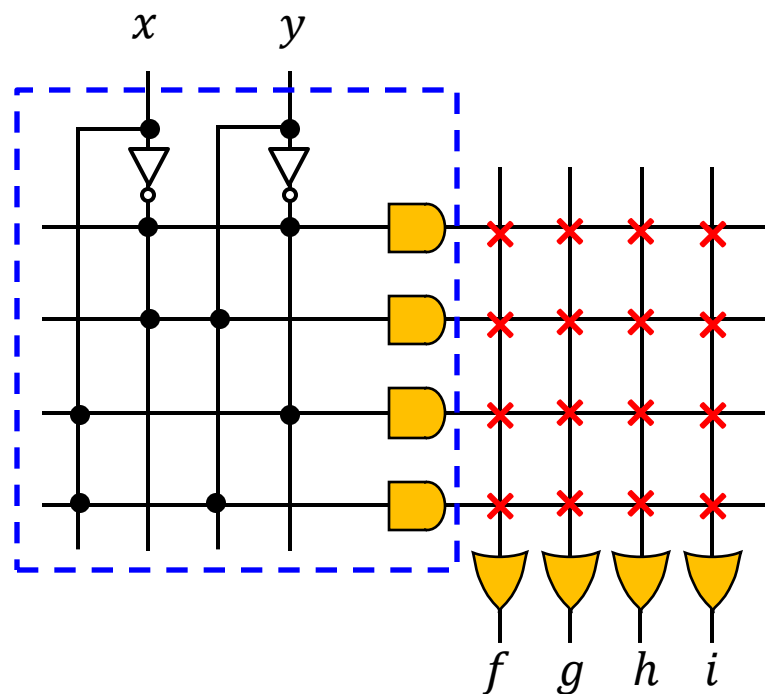# 6.2 Programmable Read-Only Memory

- Has fixed AND array and programmable OR array
- n inputs require 2^n AND gates and can be implemented using $n \times 2^n$ decoder
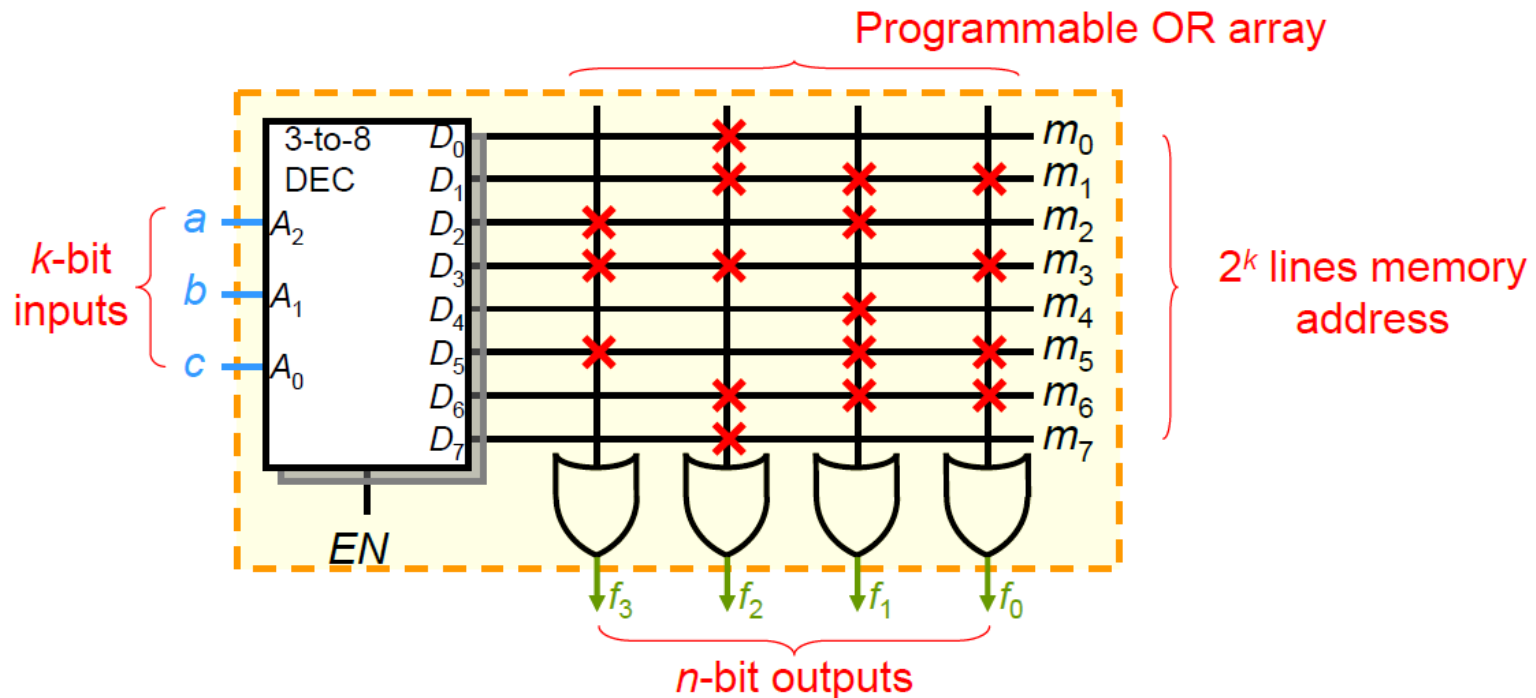
Example (8 × 4 PROM)

Realize the following functions with an 8 × 4 PROM

$$f_3(a, b, c) = \sum m(2, 3, 5) \qquad f_1(a, b, c) = \sum m(1, 2, 4, 5, 6)$$

$$f_2(a, b, c) = \sum m(0, 1, 3, 6, 7) \qquad f_0(a, b, c) = \sum m(1, 3, 5, 6)$$

Why is it called as memory?
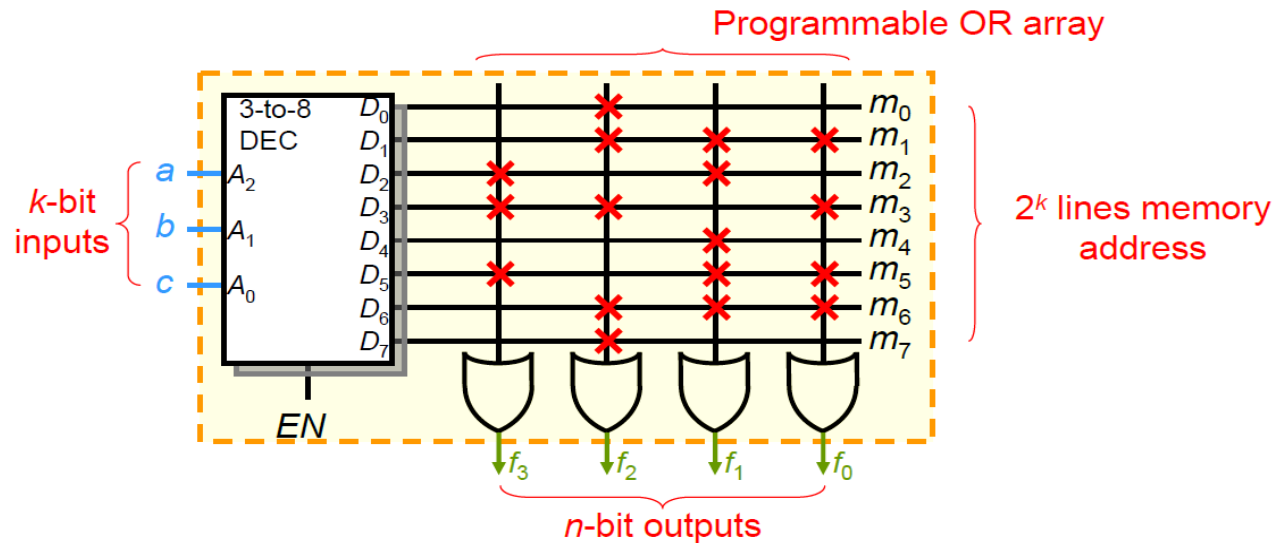
- An essential memory device in which **permanent** binary information is stored (even when the power is turned off, the information is still there).
- k-bit address inputs and n-bit data outputs
- Inputs: address for the memory
- Outputs: the word (n data bits) stored in ROM selected by the k-bit input address

Types of ROM

- PROM: Programmed by the user only once; non-reprogrammable

- EPROM (Erasable & Programmable Read-Only Memory): Possible to erase it by exposure to UV light; Charge is stored during programming and can dissipate upon UV exposure

- EEPROM (Electrically EPROM): Reprogrammable using voltage (10K times)

6.3 Programmable Array Logic (PAL)

- PAL: The OR array is fixed

- PLA is a general design for implementing sum-of-product terms, whereas the PAL has fixed sum-of-product terms

- Design for PAL device is easier, but not as flexible as that for PLA

PAL Diagram I
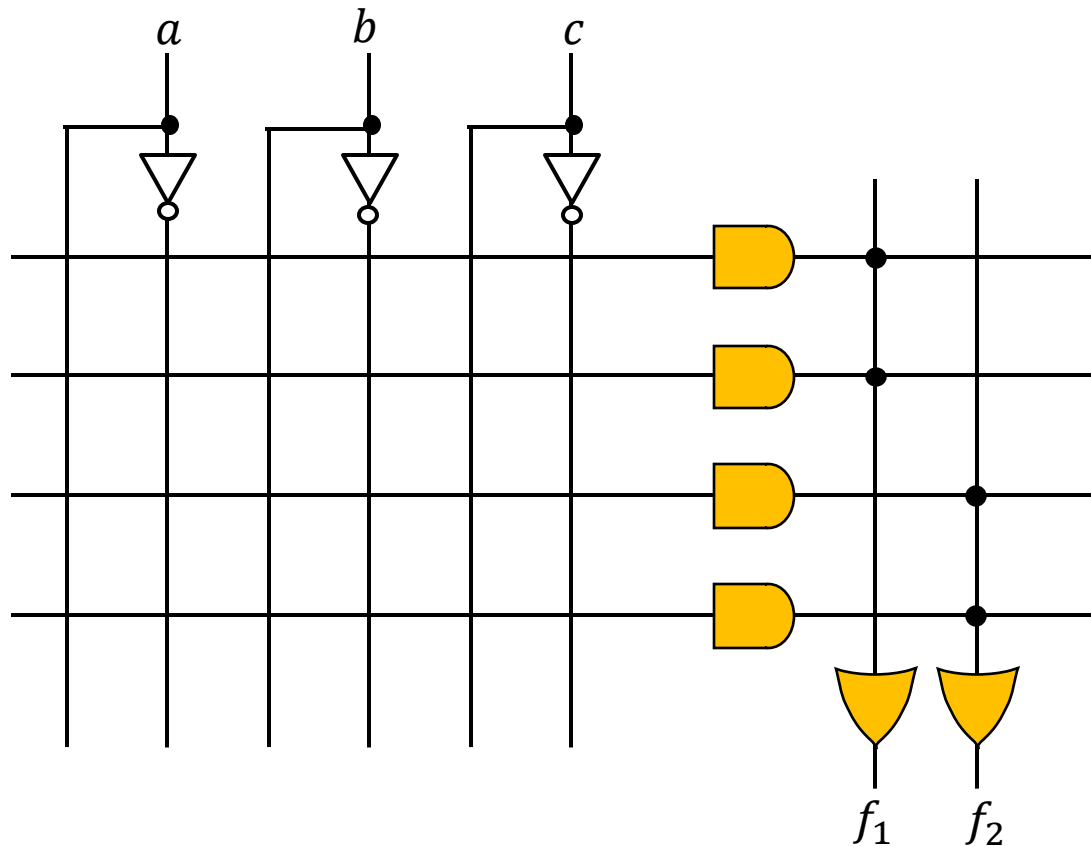
- 3 inputs
- 2 outputs
- Each output can have the most 2 product terms

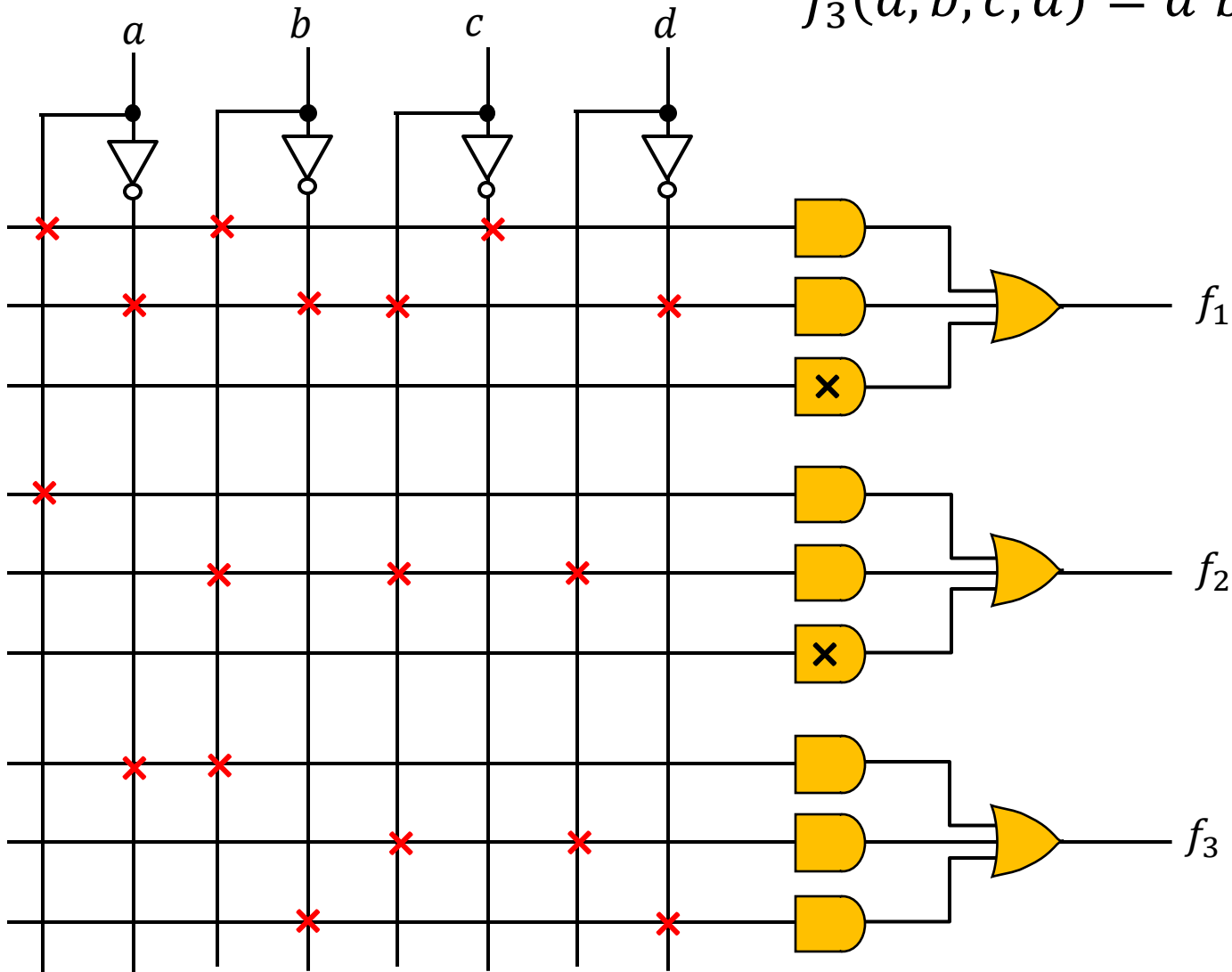# PAL Diagram II
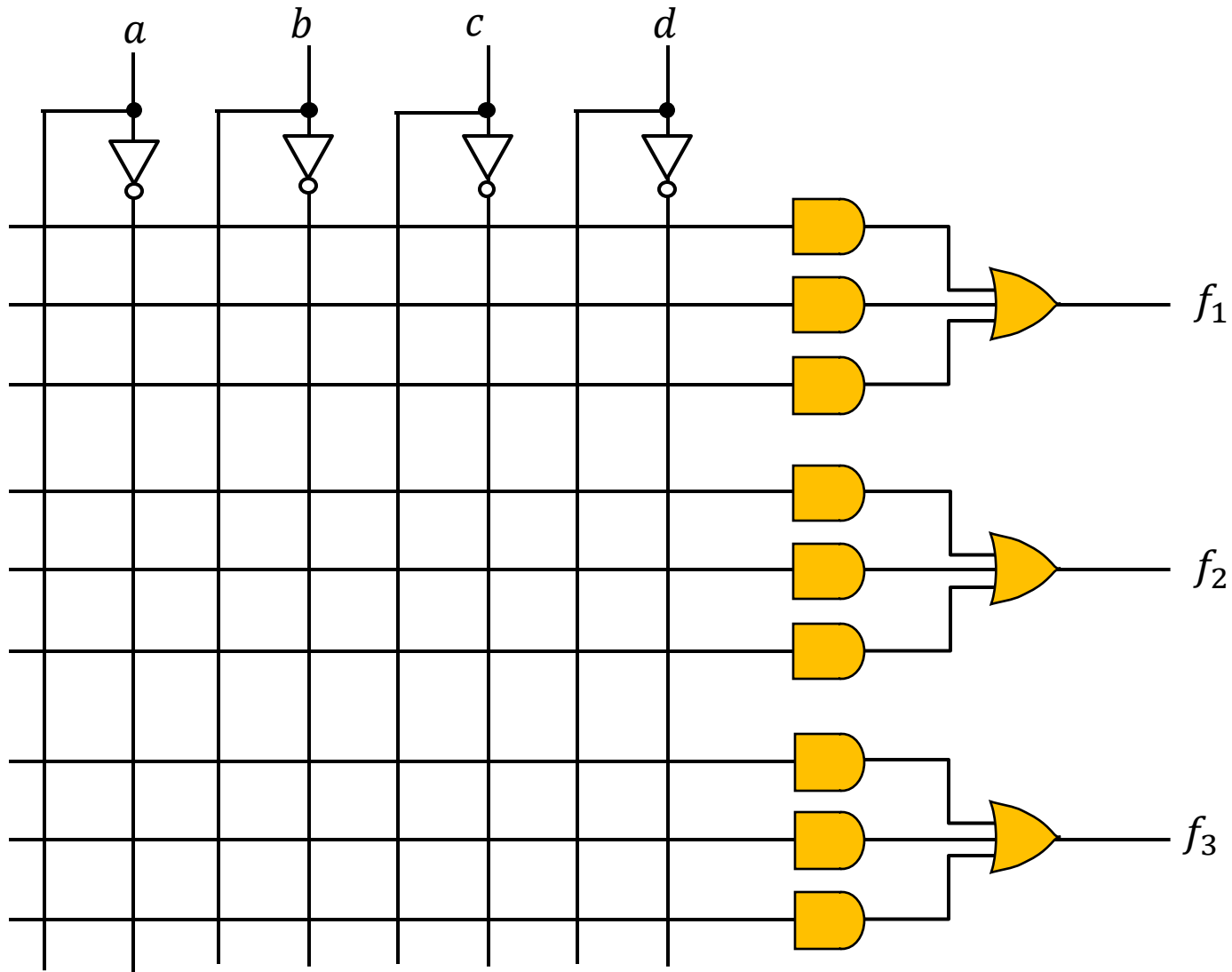
Example

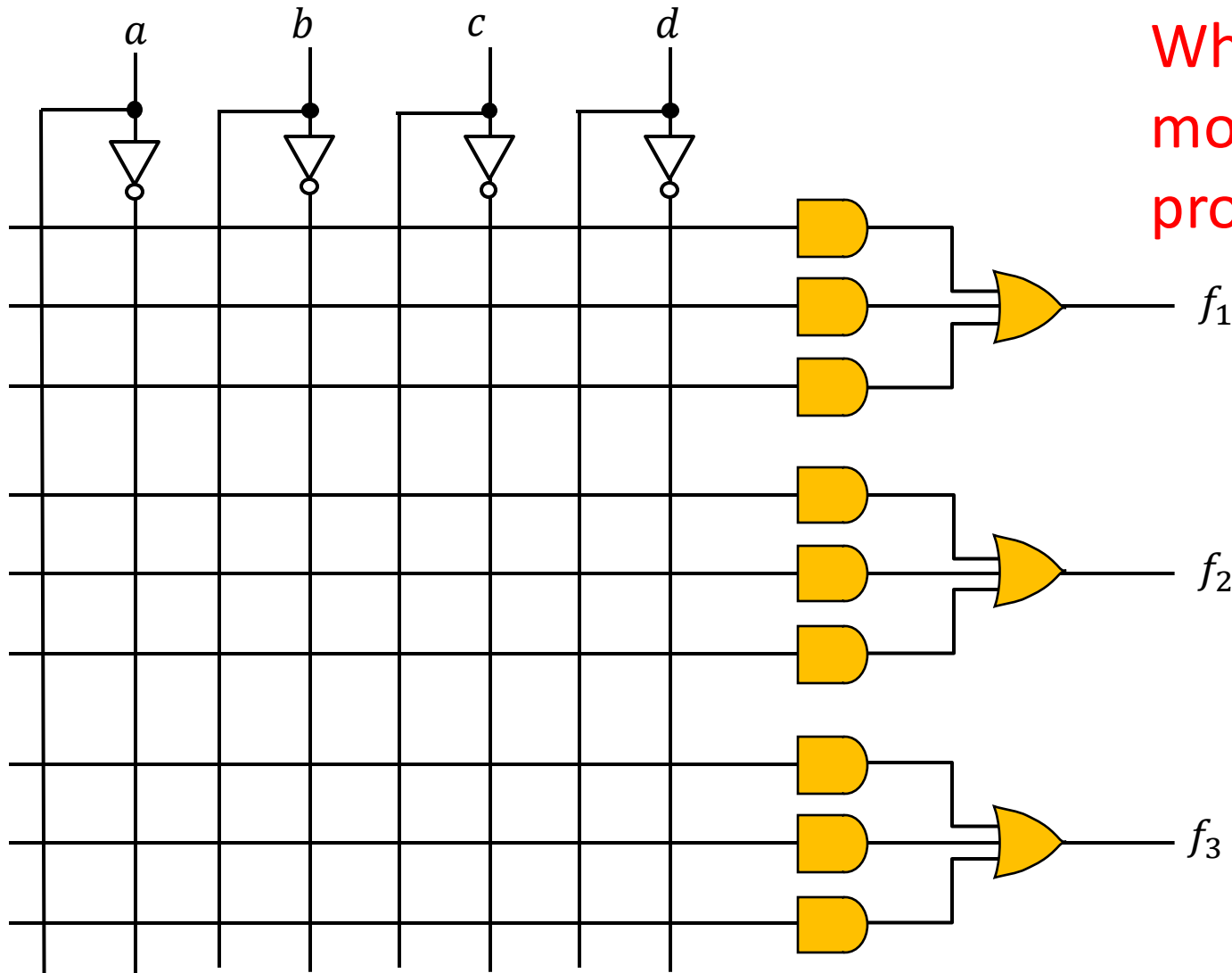$$f_1(a, b, c, d) = abc' + a'b'cd'$$
$$f_2(a, b, c, d) = a + bcd$$
$$f_3(a, b, c, d) = a'b + cd + b'd'$$

Limitation

$$f_1(a, b, c, d) = abc' + a'b'cd' + ad + ab'$$
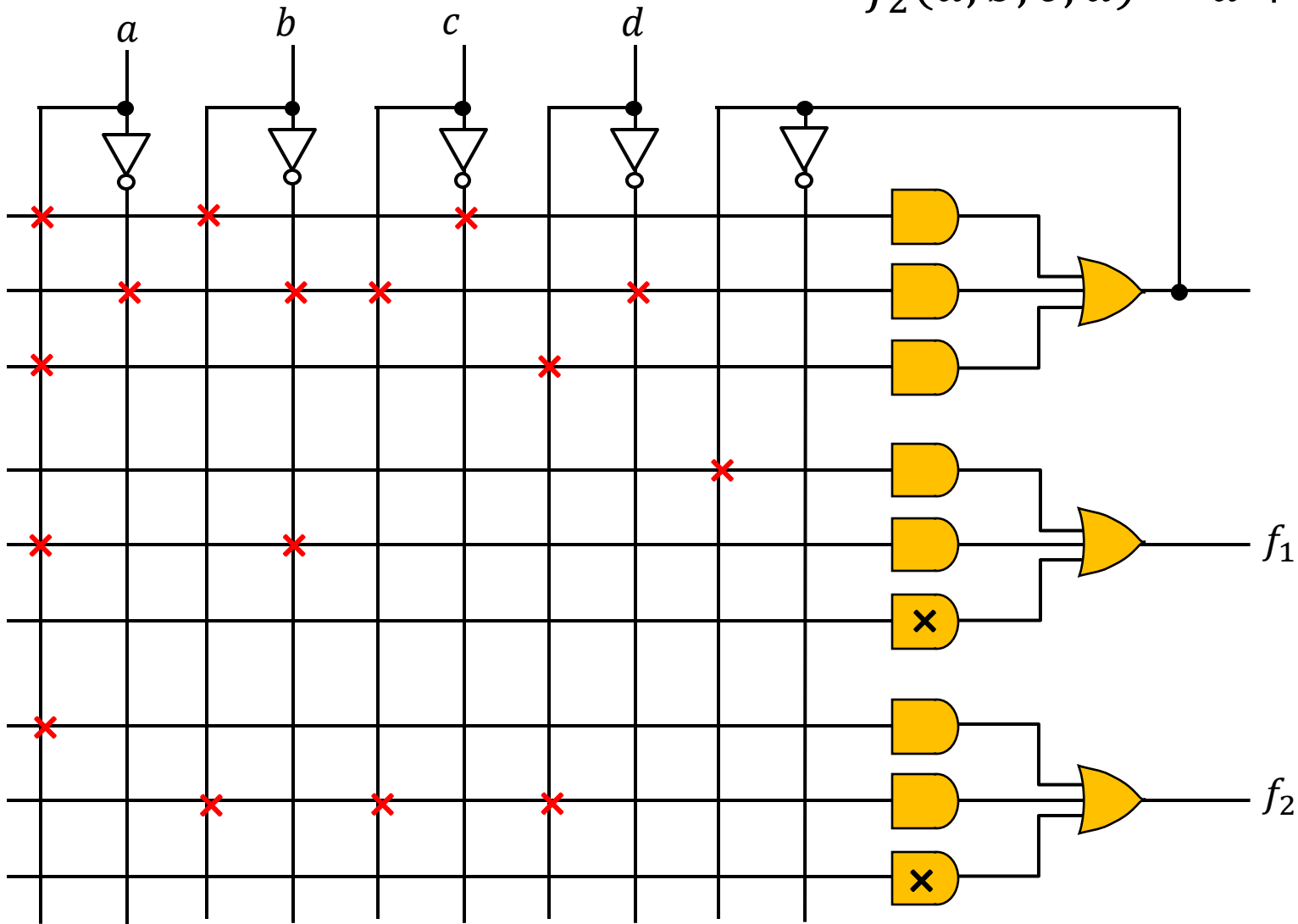$$f_2(a, b, c, d) = a + bcd$$

What if we need more than 3 product terms?

f_1

f_2

f_3

15

Solution: Feedback

$$f_1(a, b, c, d) = abc' + a'b'cd' + ad + ab'$$

$$f_2(a, b, c, d) = a + bcd$$

Exercise

Implement the following functions with the PAL shown in next slide.

$$x(a, b, c, d) = \sum m(2, 12, 13)$$

$$y(a, b, c, d) = \sum m(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$z(a, b, c, d) = \sum m(1, 2, 8, 12, 13)$$

cd \ ab	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

x

cd \ ab	00	01	11	10
00				
01				
11				
10				

y

cd \ ab	00	01	11	10
00				
01				
11				
10				

z

cd \ ab	00	01	11	10
00				
01				
11				
10				

Exercise

Exercise

Implement the following functions with the PAL shown in next slide.

$$x(a, b, c, d) = \sum m(2, 12, 13)$$

$$y(a, b, c, d) = \sum m(2, 5, 8, 9, 10, 11, 12, 13)$$

$$z(a, b, c, d) = \sum m(1, 2, 8, 12, 13)$$

cd \ ab	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

x

cd \ ab	00	01	11	10
00				
01				
11				
10				

y

cd \ ab	00	01	11	10
00				
01				
11				
10				

z

cd \ ab	00	01	11	10
00				
01				
11				
10				

Exercise

PAL16R6

PAL16R6'
J OR W PACKAGE
(TOP VIEW)

PAL16R6'
FK PACKAGE
(TOP VIEW)
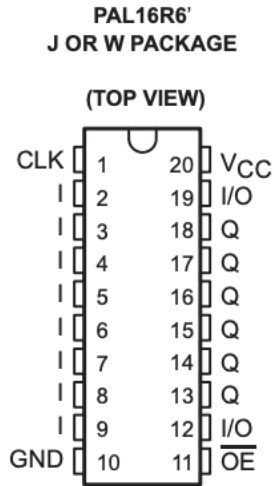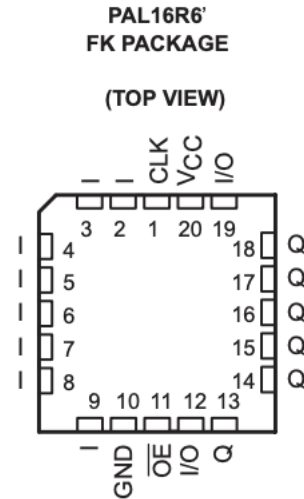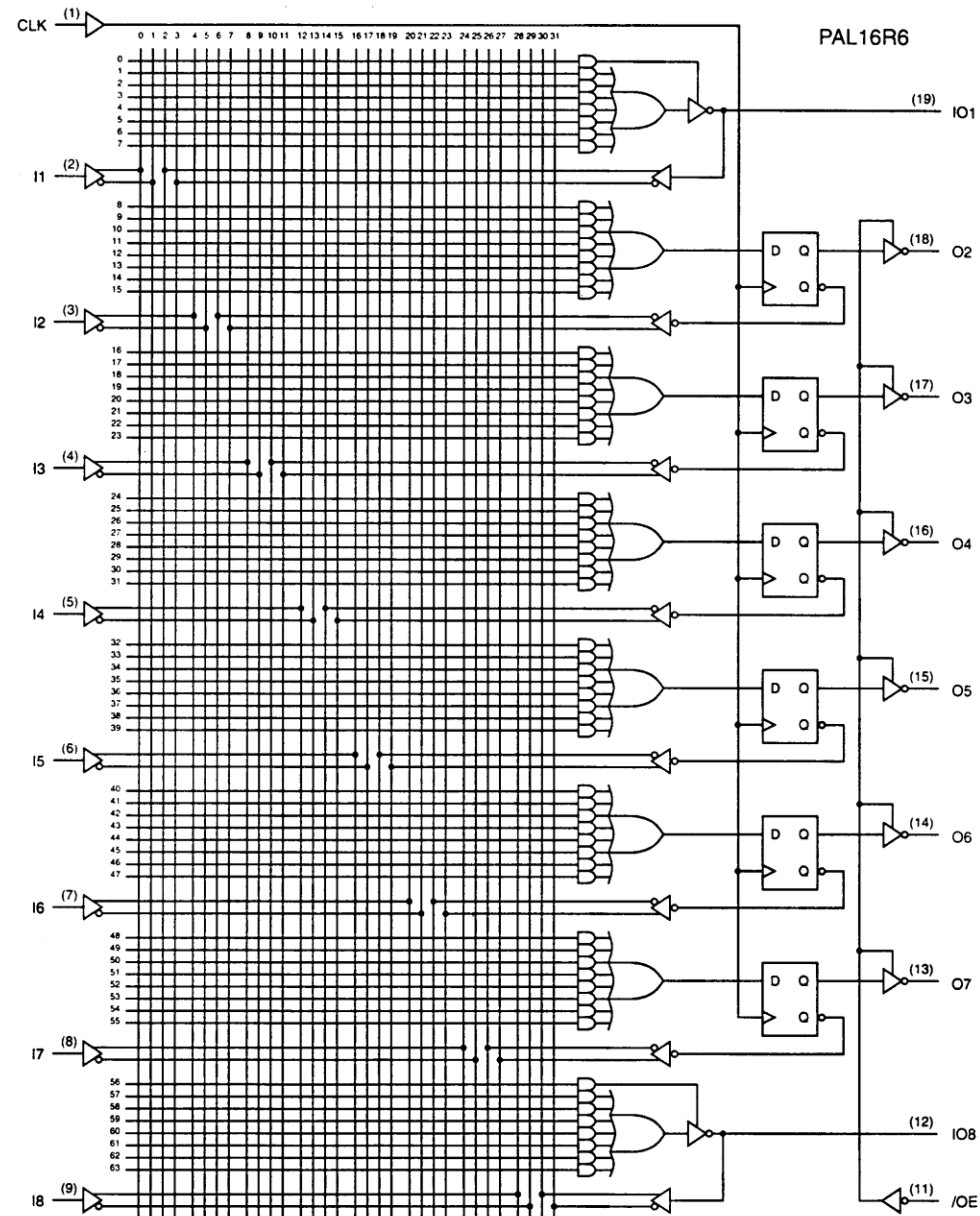
27 mm × 7.6 mm   9 mm × 9 mm

- A PAL device with 16 inputs, 6 outputs with D-latch (1-bit memory device), and 2 un-buffered outputs

- *8 inputs are feedback from outputs*

- This device can be used for designing a sequential circuit (Lecture 7)

PAL16R6

21

6.4 Programmable Logic Array (PLA)

- PLA: Both AND and OR array are programmable
- For PROM, realize a Boolean Function based on minterm canonical expression. No minimization!
- For PLA, AND gates are programmable
  - Based on sum-of-product but may not be canonical
  - Logic designer is bounded by the number of product terms available in the AND-array
  - Simplification is necessary!

Example

$$f_1(a, b, c) = \sum m(0, 1, 2, 6)$$
$$f_2(a, b, c) = \sum m(1, 2, 3, 4, 6)$$

c \ ab	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

f_1

c \ ab	00	01	11	10
0	1	1	1	
1	1			

f_2

c \ ab	00	01	11	10
0		1	1	1
1		1		

$$f_1(a, b, c) = a'b' + bc'$$
$$f_2(a, b, c) = a'c + bc' + ac'$$

Four distinct product term

23

Example

$$f_1(a, b, c) = a'b' + bc'$$
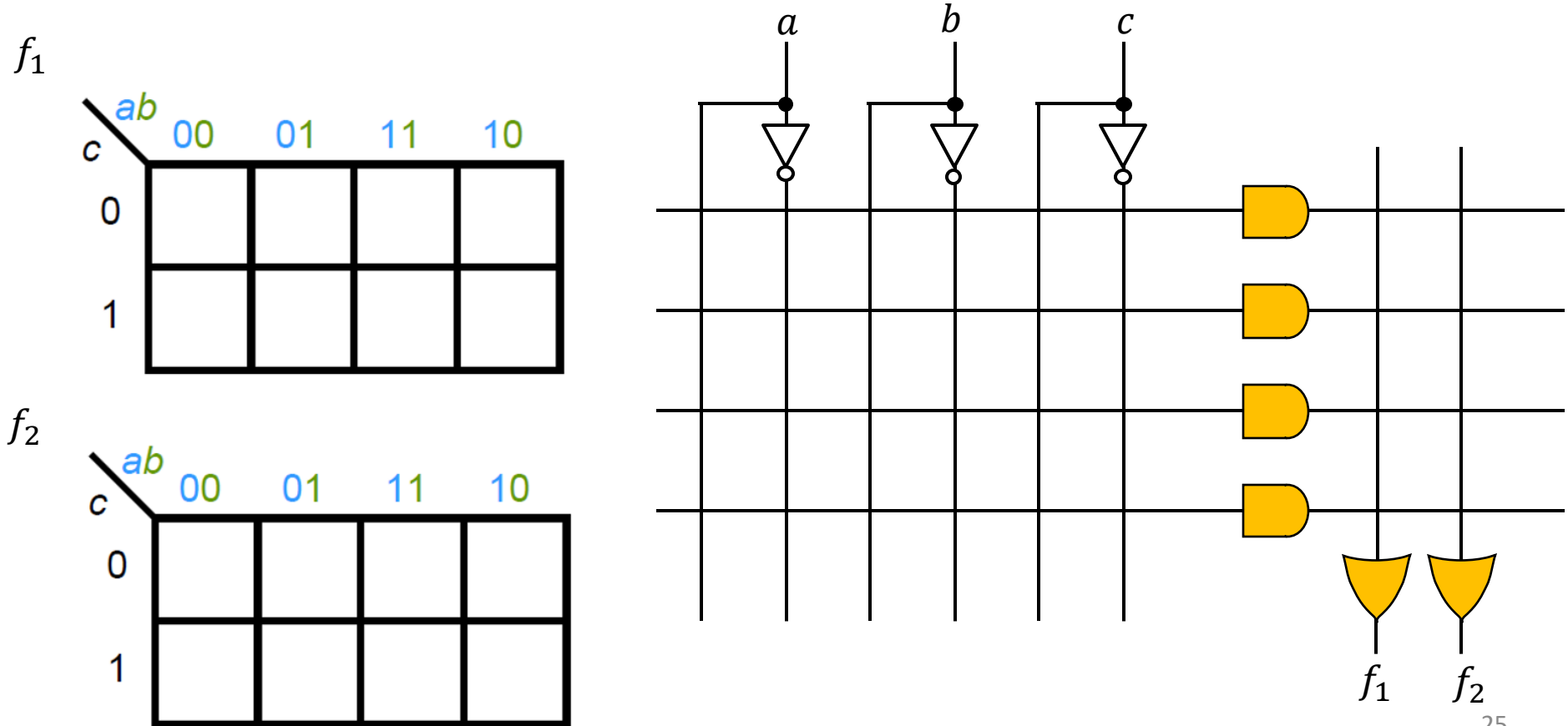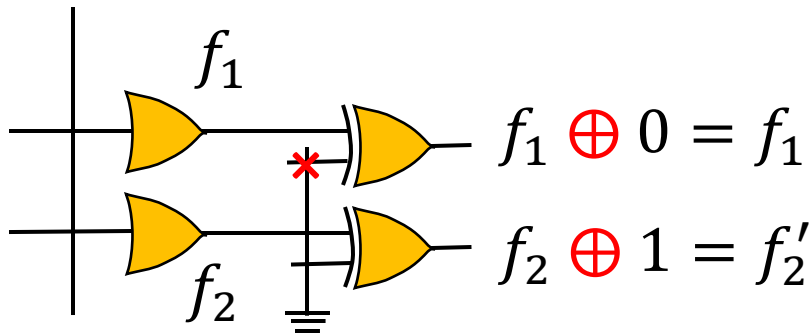$$f_2(a, b, c) = a'c + \textcolor{red}{bc'} + ac'$$
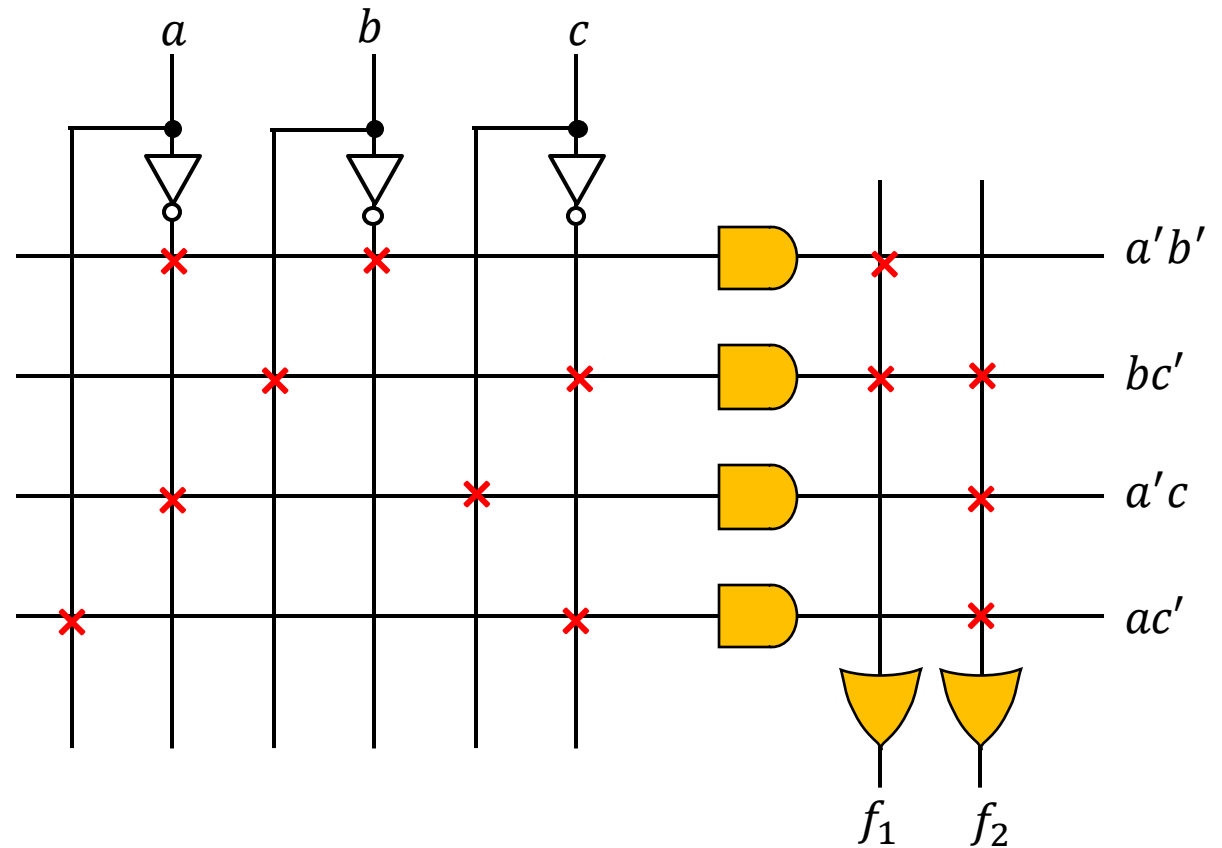
Exercise

$$f_1(a, b, c) = \sum m(0, 2, 3, 6)$$
$$f_2(a, b, c) = \sum m(3, 6, 7)$$

Additional Features

- Addition of XOR gate after OR gate
- To provide flexibility for either a true output or a complemented output
- One of the inputs is connected to the ground when the fuse is intact
- If blown, the input is 1

$$f_1 \oplus 0 = f_1$$

$$f_2 \oplus 1 = f_2'$$

Example

Implement the following functions with a PLA

$$f_1(a, b, c) = ab' + ac + a'bc' \qquad f_2'(a, b, c) = ac + bc$$
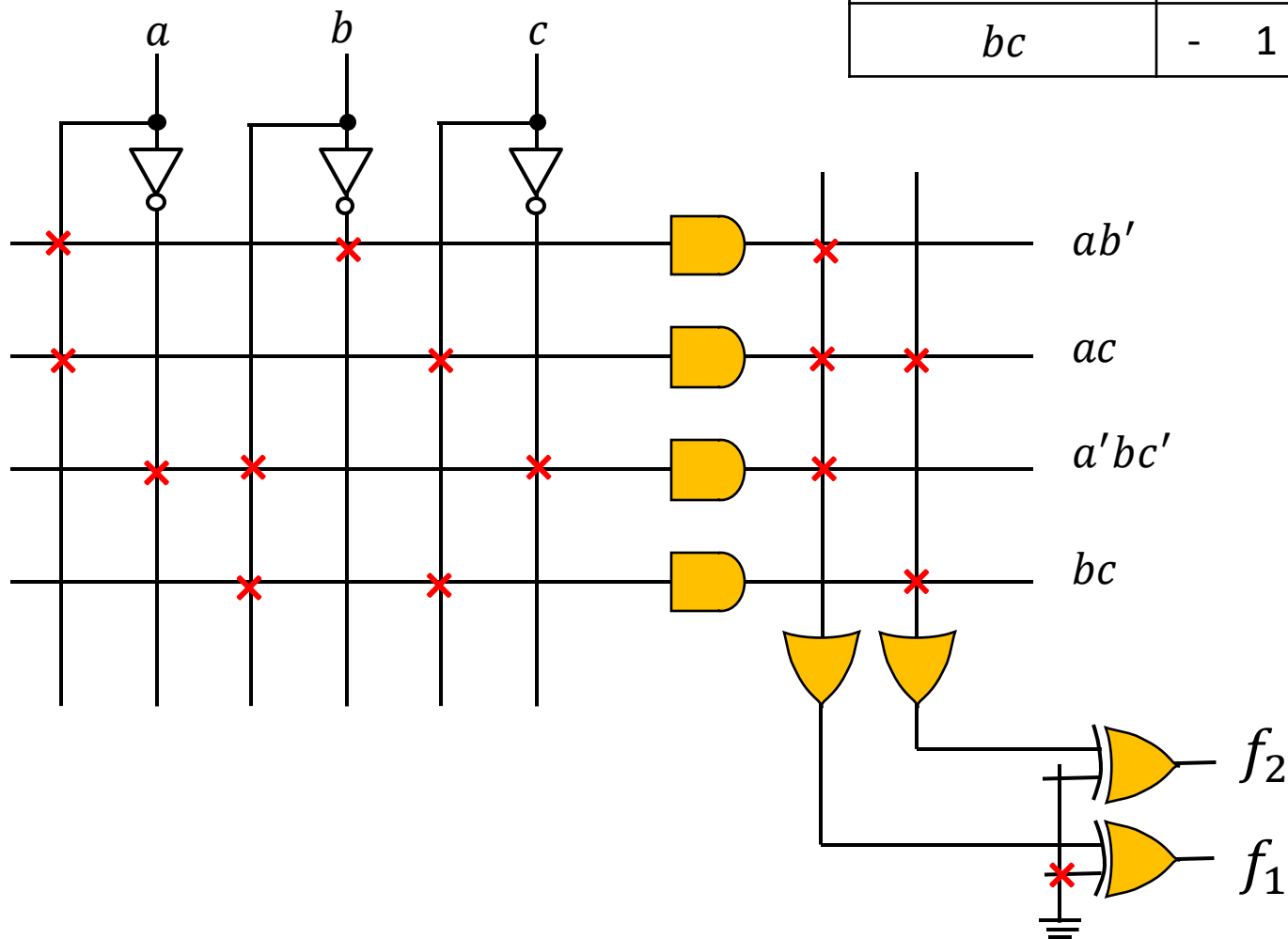
PLA Programming Table

Product terms	Input			Output	
	a	b	c	f_1	f_2
ab'	1	0	-	1	-
ac	1	-	1	1	1
$a'bc'$	0	1	0	1	-
bc	-	1	1	-	1
			T/C	T	C

T: Uncomplemented output

C: Complemented output

Example

Product terms	Input			Output	
	a	b	c	f_1	f_2
ab'	1	0	-	1	-
ac	1	-	1	1	1
$a'bc'$	0	1	0	1	-
bc	-	1	1	-	1
T/C				T	C

Exercise

Implement the following functions using the PLA in next slide.

$$f_1(a, b, c, d) = \sum m(0, 2, 3, 5, 7, 8, 10, 12, 13, 15)$$
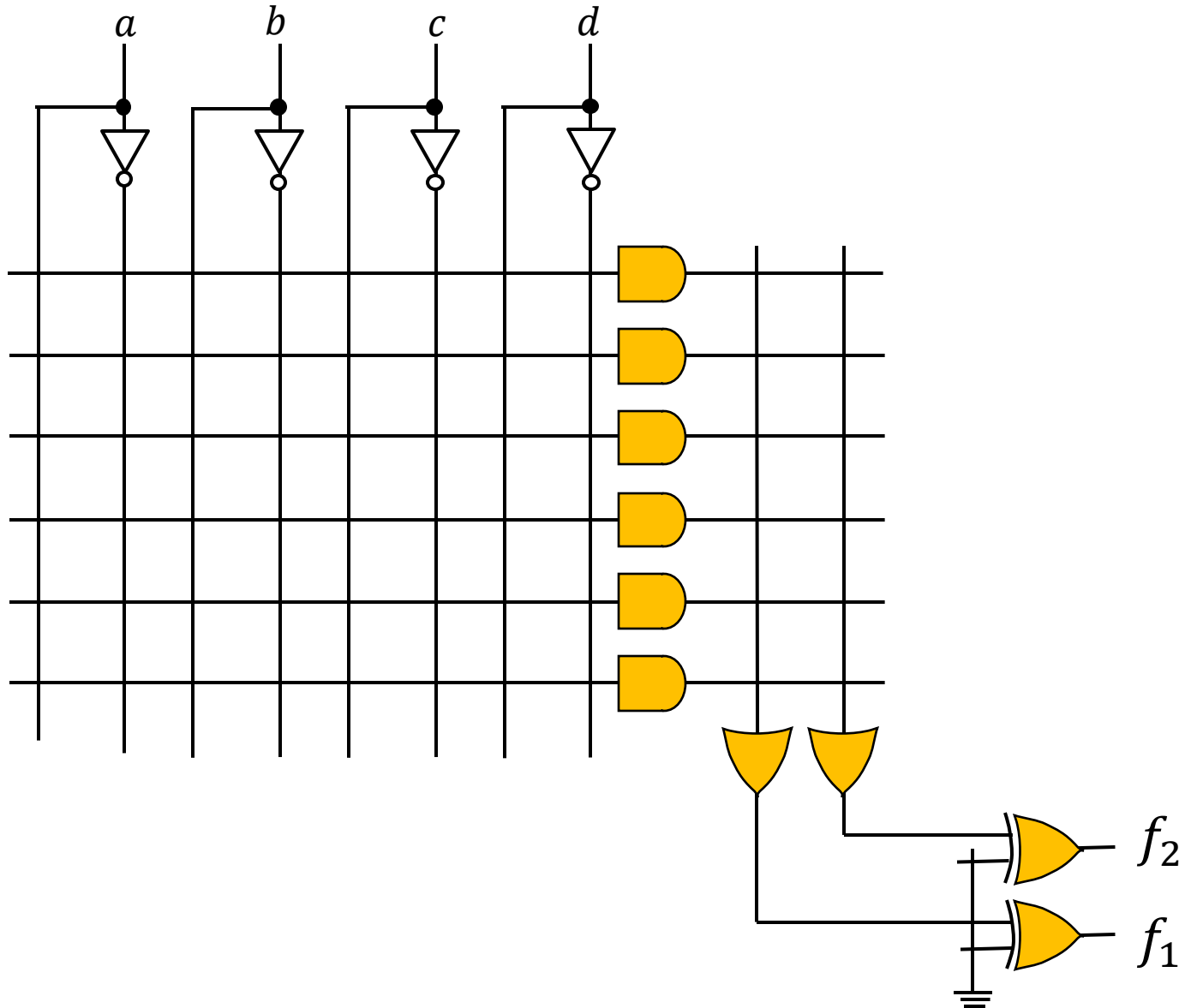$$f_2(a, b, c, d) = \sum m(0, 4, 6, 7, 8, 10, 14, 15)$$

Exercise

Implement the following functions using the PLA in next slide.

$$f_1(a, b, c, d) = \sum m(0, 2, 3, 5, 7, 8, 10, 12, 13, 15)$$
$$f_2(a, b, c, d) = \sum m(0, 4, 6, 7, 8, 10, 14, 15)$$

Exercise

Summary

(a) Programmable read-only memory (PROM)
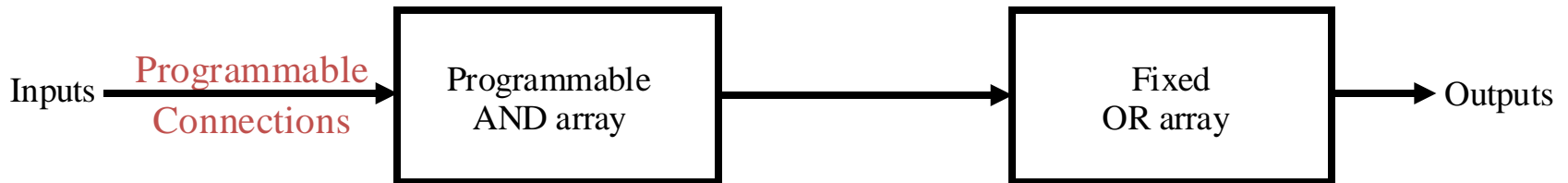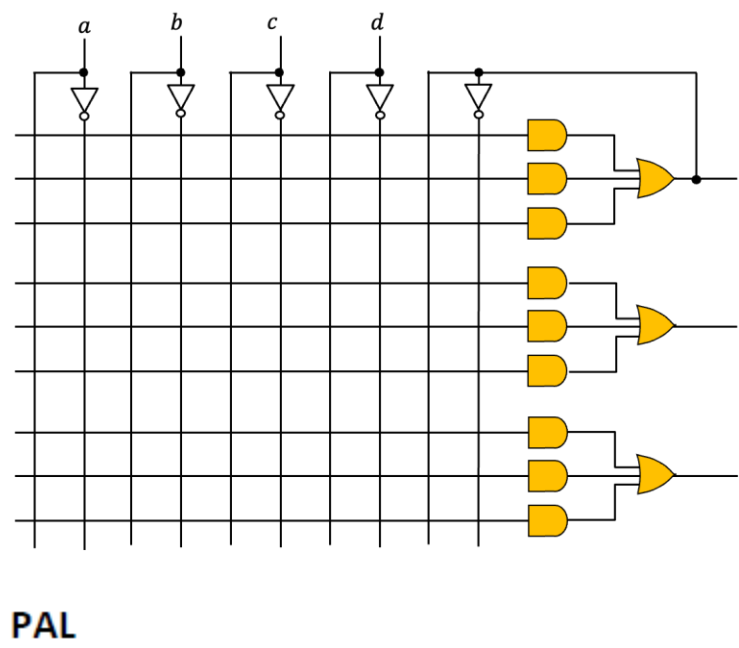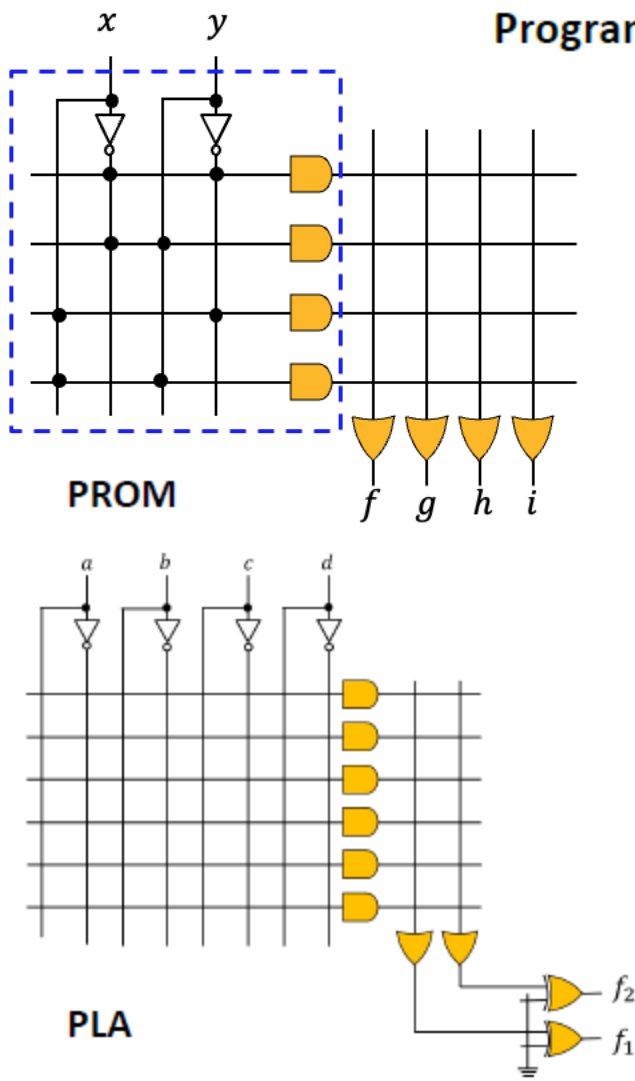
(b) Programmable array logic (PAL) device
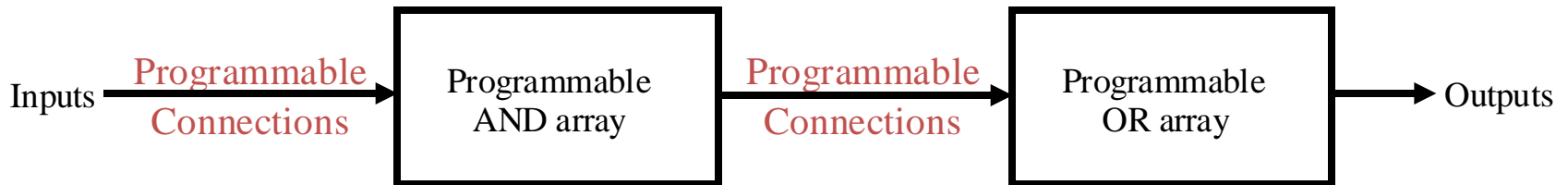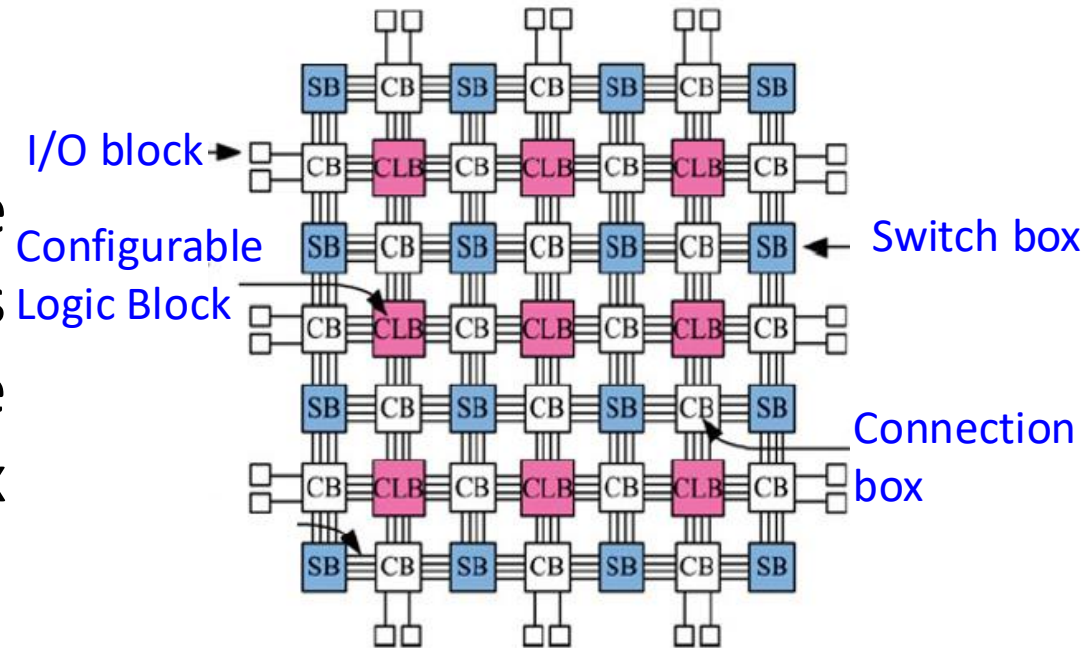
(c) Programmable logic array (PLA) device

Given in Test and Exam

Programmable Logic Devices

PROM

PLA

PAL

6.5 Field Programmable Gate Array Logic

- FPGAs consist of a large number of logic blocks with programmable interconnects in matrix form.

I/O block

Configurable Logic Block

Switch box

Connection box

- Three basic components
  - ➢ **Configurable Logic Block**: To create logic circuit
  - ➢ **I/O Block**: To connect the external inputs/outputs
  - ➢ **Switch/Connection Box**: For interconnection between CLB and I/Os.

Configurable Logic Block

- A configurable logic block (CLB) consists of a **look-up-table (LUT)**

- The LUT is for implementing the **truth table** of a logic function (*e.g.*, 4-input Boolean function)

- Additional elements are for control or other functions: D-FF (store output), MUX (select output)
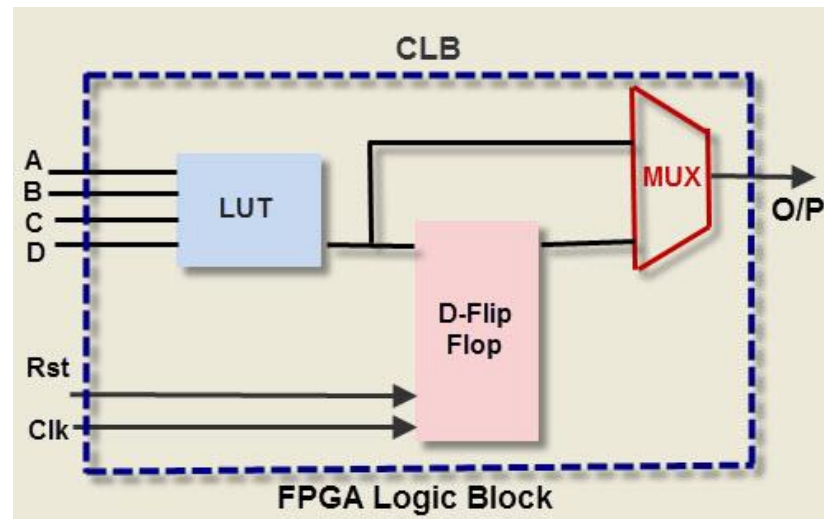
Look-up Tables (LUTs)
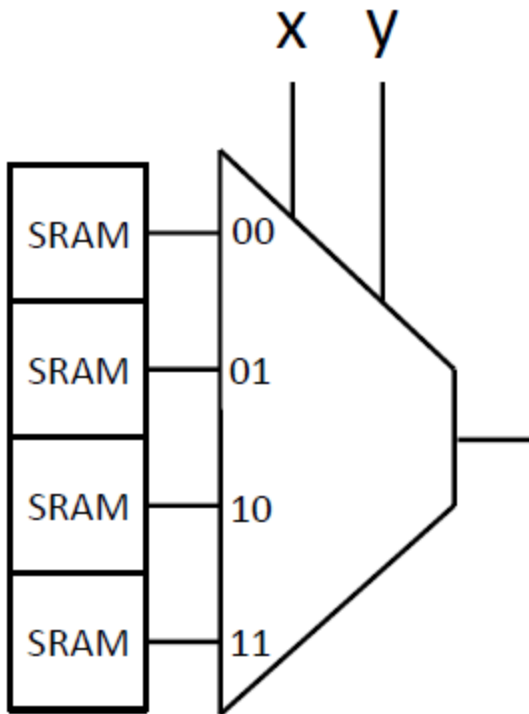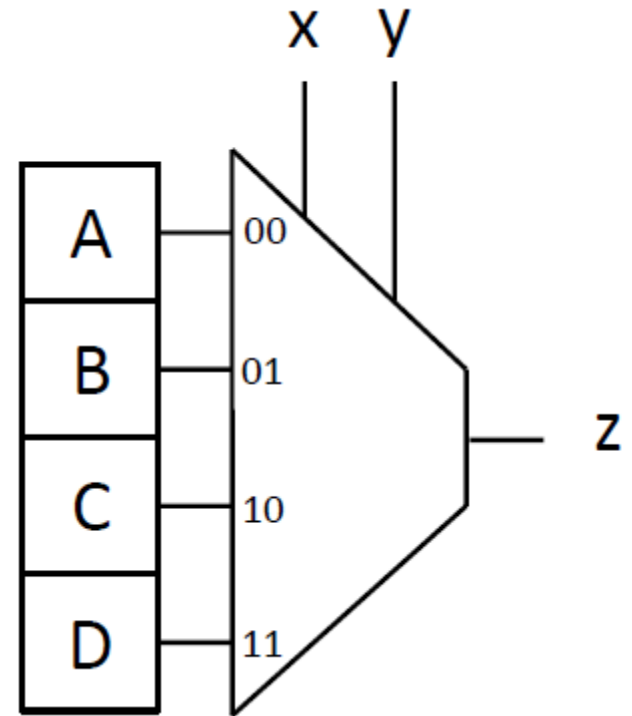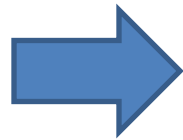

Commercial FPGAs
- Xilinx: 6-LUT
- Altera: 6-LUT
- Microsemi: 4-LUT

For x-input LUT, the number of memory locations is 2^x
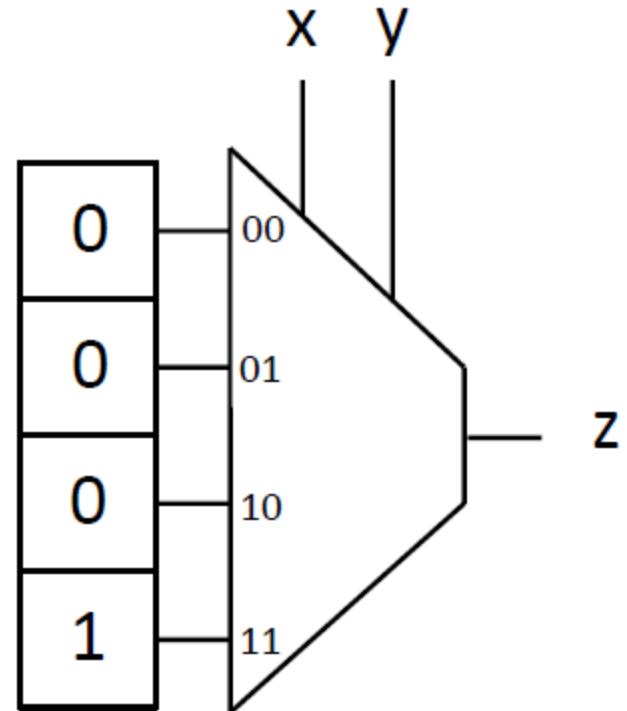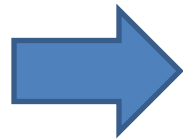- 2-LUT: $2^2 = 4$
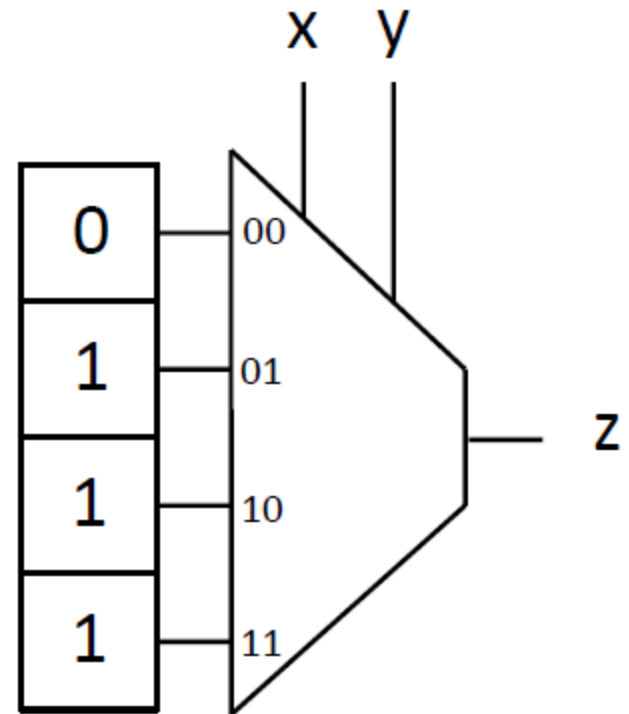- 4-LUT: $2^4 = 16$

Look-up Tables (LUTs)

x	y	z
0	0	A
0	1	B
1	0	C
1	1	D

Example (AND gate)

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

Example (OR gate)

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1
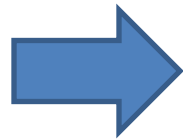
Exercise (XOR)

x	y	z
0	0	
0	1	
1	0	
1	1	
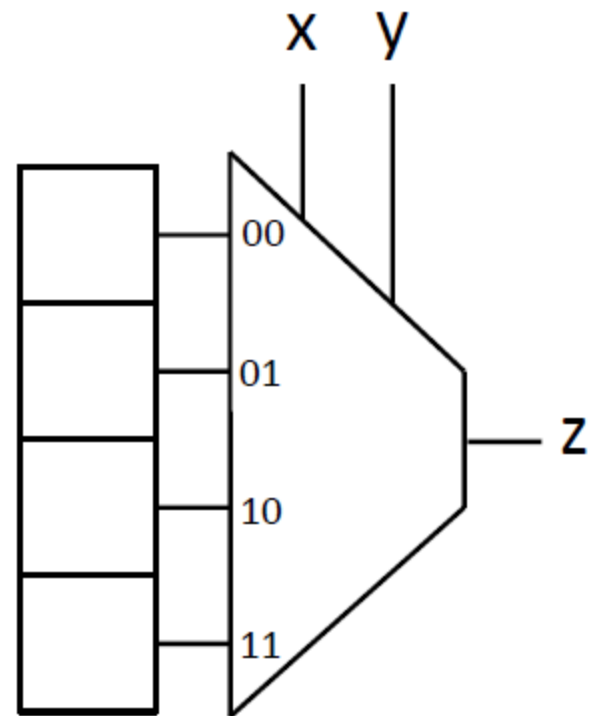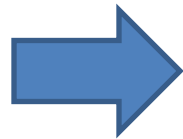
Features of LUTs

- An *n*-LUT can implement any *n*-input logic functions
- Logic minimization: Reduce the number of inputs

Why not a large LUT?
- LUTs grow exponentially based on the number of inputs
  - ➢ 64 inputs and 1 output → 2^{64} memory bits
- Large LUT → long delay, less flexible
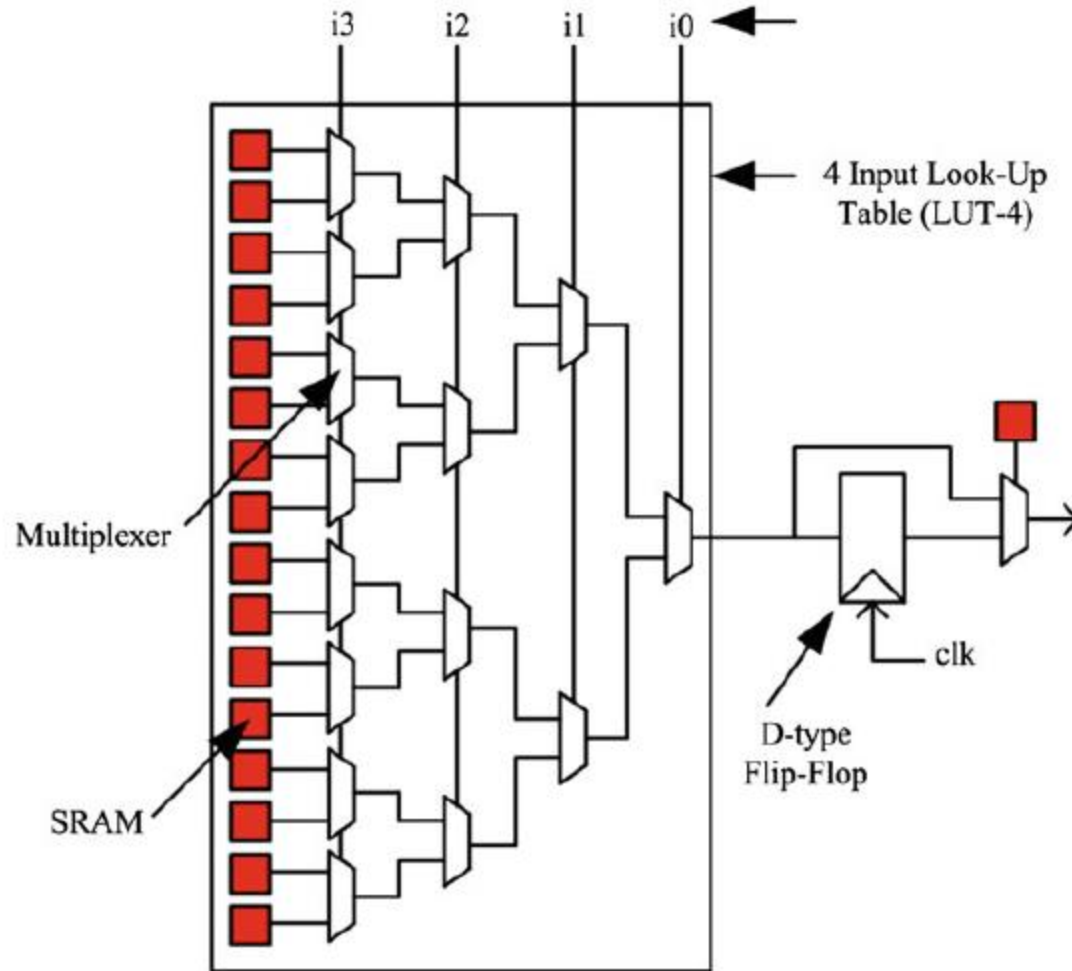- Small LUTs: Design components that can be reused (transforming a complex circuit into simpler circuits)

Solution for Large Inputs

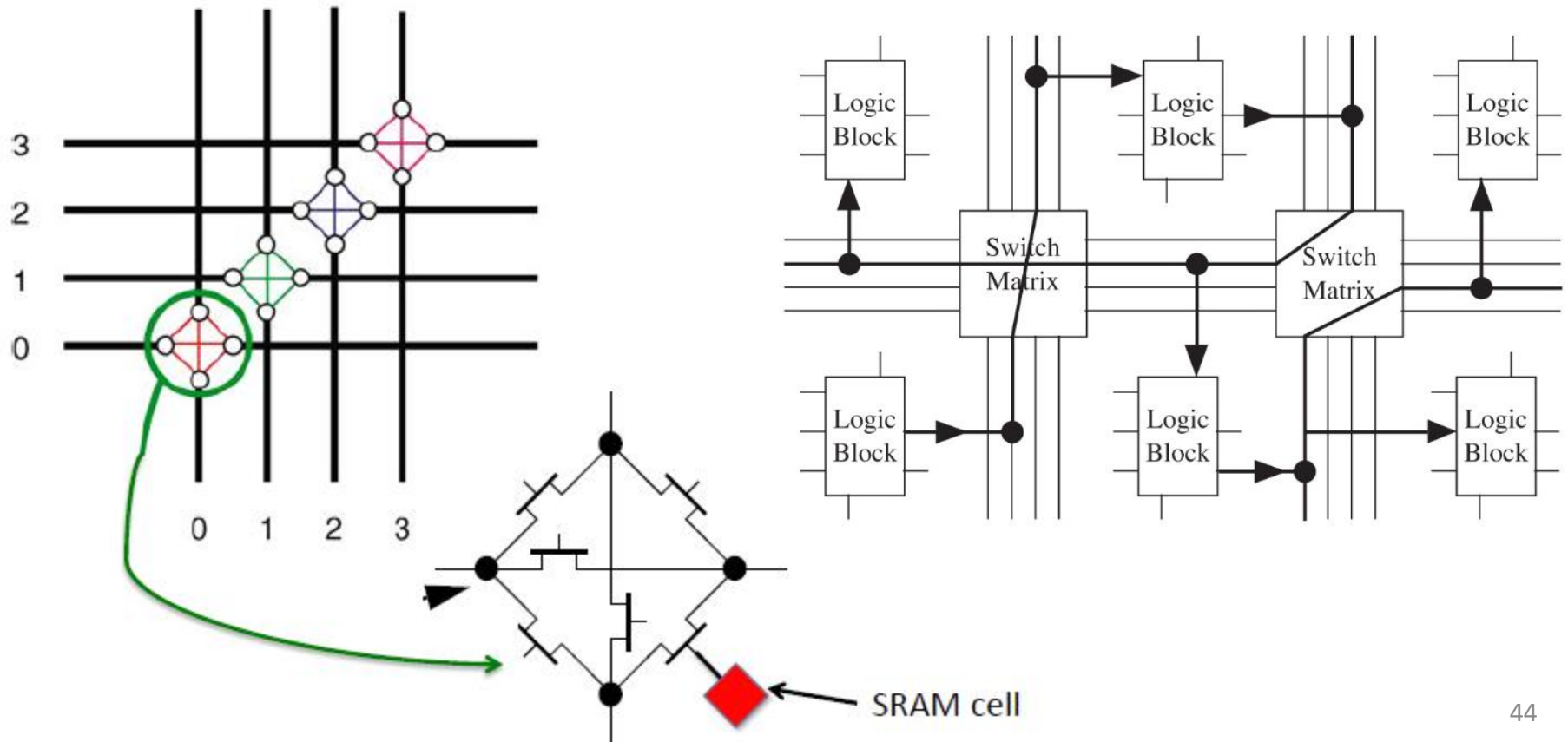Map circuits onto multiple LUTs

➢ Example: Use 2-input LUTs
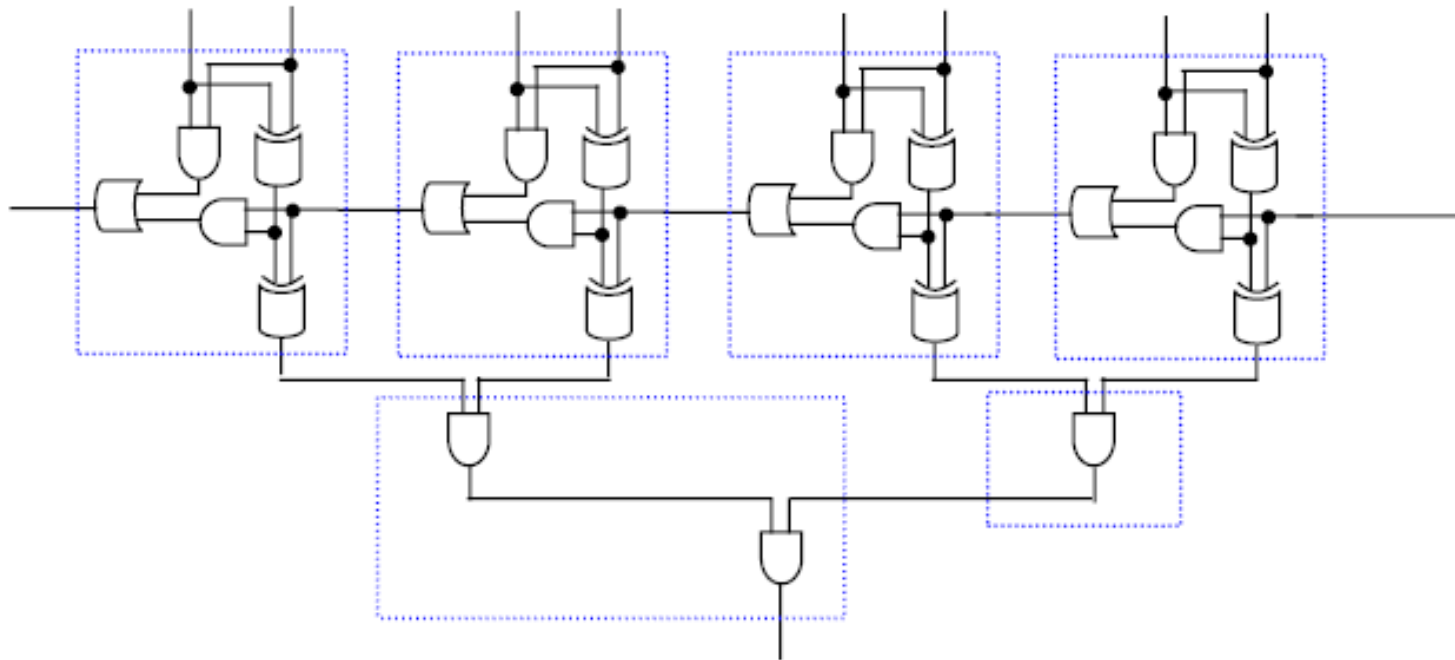
Example (4-input LUT)

Interconnects

- FPGAs use **switch matrices** to provide interconnects for logic blocks and I/Os

- Each switch is controlled by the SRAM cell

I/O Standards

- FPGA output may be connected to device with different electrical requirements

- The FPGA pin I/O standard must be specified

- I/O Standards:
  - LVTTL: low-voltage transistor-transistor logic; 3.3-V standard that can tolerate 5-V signals.

  - LVCMOS: low-voltage complementary metal-oxide semiconductor; LVCMOS2, a 2.5-V standard that can tolerate 5-V signals.

  - …

FPGA Design Flow
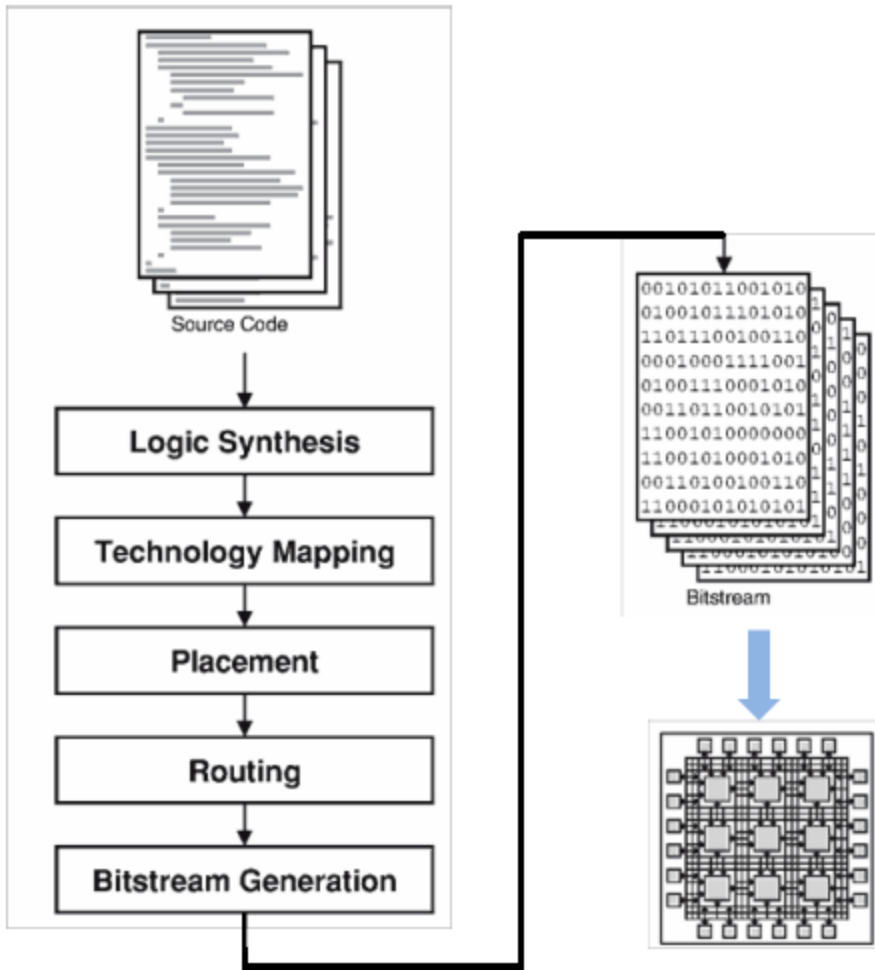
1. **Source Code:** Create a model of the design in a hardware description language such as VHDL or Verilog

2. **Simulation:** Simulate and debug the design

3. **Synthesis:** Analyze the VHDL code and implement the described logic

4. **Technology Mapping:** Transform the logic network to the LUT network

FPGA Design Flow

Source Code → Logic Synthesis → Technology Mapping → Placement → Routing → Bitstream Generation → Bitstream
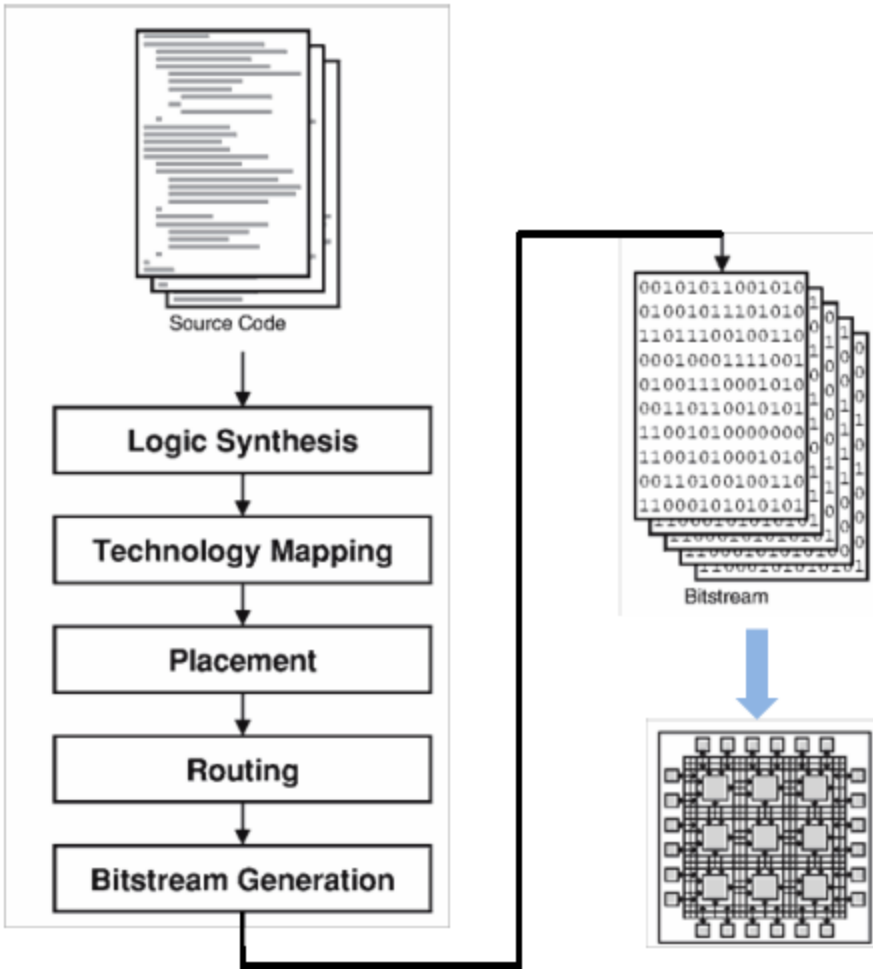
5. **Placement/Routing:** Place and interconnect all LUTs, I/Os, etc.

6. **Bitstream Generation:** Create a configuration file for all SRAM cells (logic and interconnect)

7. **Program** the FPGA based on the bitstream configuration file

Input: VHDL Source Codes

Output: Configuration bitstream