Digital Electronics
COE328

Lecture 5

Dr. Shazzat Hossain

Logic circuits Implementation: Standard Chips

Logic circuit implementation

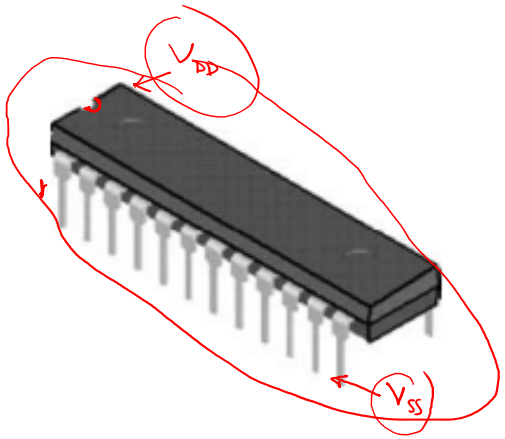
- ❑ Standard Chips (old technique)
- ❑ Programmable Logic Devices (PLD)
  - ❑ Programmable Logic Array (PLA)
  - ❑ Programmable Array Logic (PAL)
- ❑ Complex Programmable Logic Devices (CPLD)
- ❑ Field-Programmable Gate Array (FPGA)

$f = x_1 x_2 + \bar{x}_2 x_3$

Chips

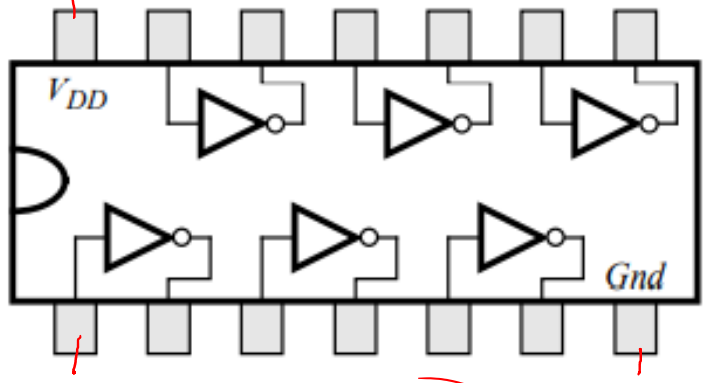
7400

1. NOT gate → 1
2. AND gate → 2
3. OR gate – 1



(a) Dual-inline package

DIP



(b) Structure of 7404 chip

Figure 3.21 A 7400-series chip.

Implement a function with chips

Let us implement $f = x_1x_2 + \bar{x}_2x_3$

We need one gate from 7404 (NOT gate), 2 gates from 7408 (AND), and one gate from 7432 (OR).

Power up the chips.

Connect wires to the inputs and output of each gate accordingly.

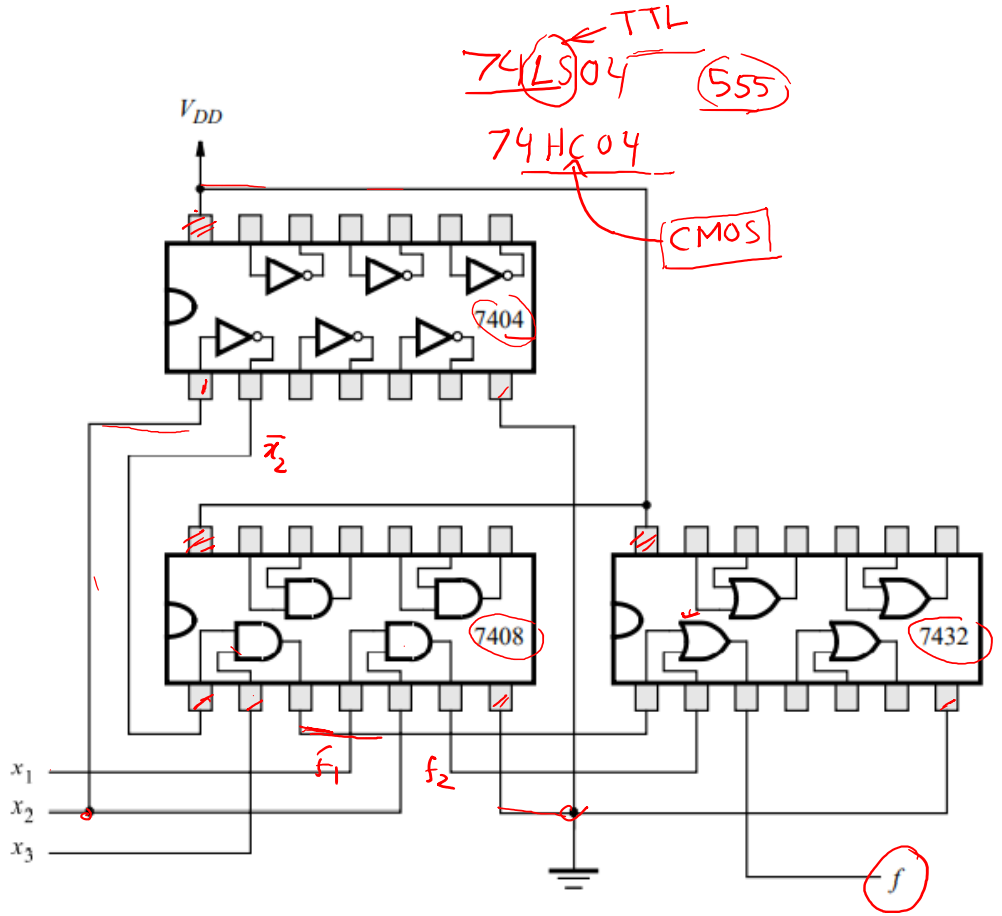


Figure 3.22 An implementation of $f = x_1x_2 + \bar{x}_2x_3$.

Programmable Logic Device (PLD)

- A **programmable logic device** (**PLD**) is an electronic component used to build reconfigurable digital circuits. Unlike digital logic constructed using discrete logic gates with fixed functions, a PLD has an undefined function at the time of manufacture. Before the PLD can be used in a circuit it must be programmed to implement the desired function. Compared to fixed logic devices, programmable logic devices simplify the design of complex logic and may offer superior performance. Unlike microprocessors, programming a PLD changes the connections made between the gates in the device.
- PLDs can broadly be categorized into, in increasing order of complexity, Simple Programmable Logic Devices (SPLDs), comprising programmable array logic, programmable logic array, and generic array logic; Complex Programmable Logic Devices (CPLDs) and Field-Programmable Gate Arrays (FPGAs).

$f = x_1 x_2 + \bar{x}_2 x_3$

$(x_1 \text{ and } x_2)$ or (not x_2 and x_3)

Programmable Logic Array (PLA)

PLA has programmable AND plane and OR plane. Implement SOP functions.

Let us implement, $f = x_1\bar{x}_2 + x_1\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + x_1x_3 = P_1 + P_2 + P_3 + P_4$

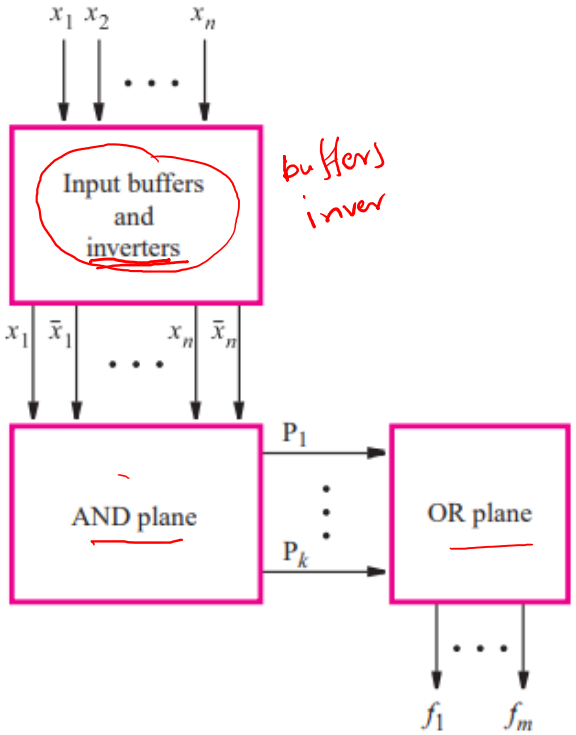


Figure 3.25 General structure of a PLA.

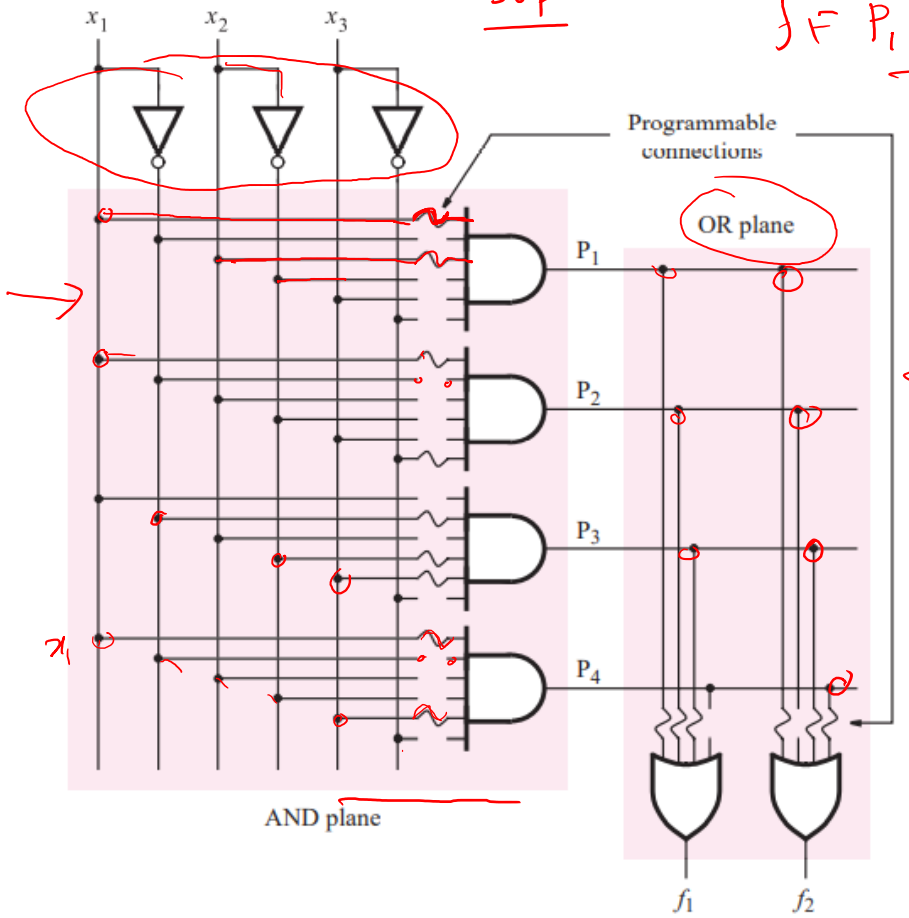


Figure 3.26 Gate-level diagram of a PLA.

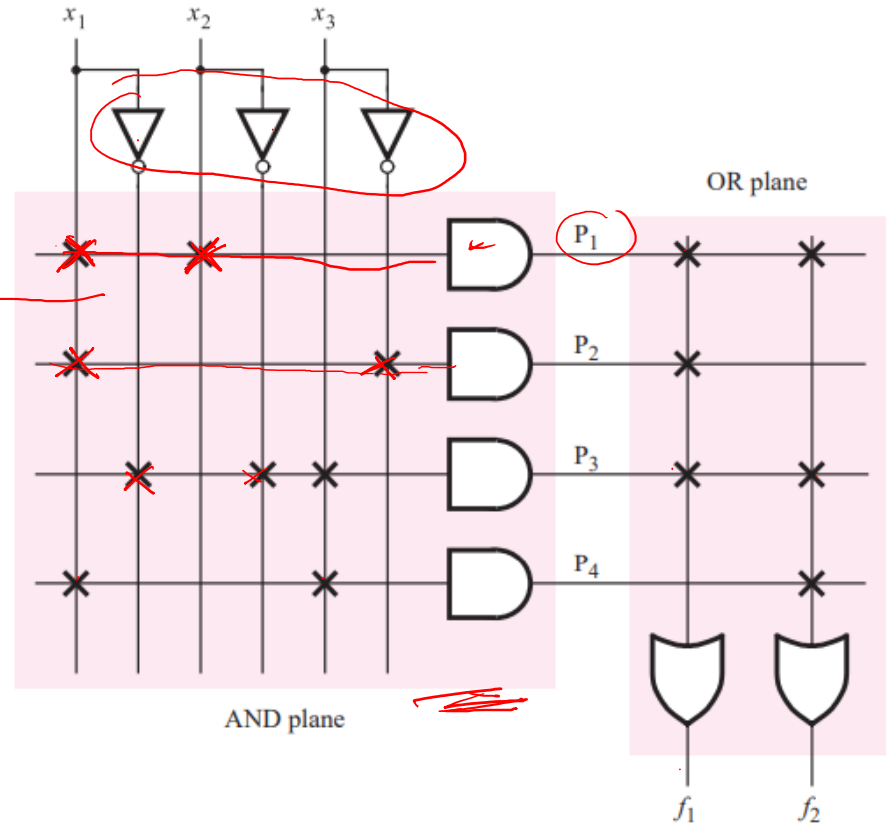


Figure 3.27 Customary schematic for the PLA in Figure 3.26.

Programmable Array Logic (PAL)

Programmable Array Logic (PAL) is a commonly used programmable logic device (PLD). It has programmable AND array and fixed OR array. Because only the AND array is programmable, it is easier to use but not flexible as compared to Programmable Logic Array (PLA).
The functions performed by the PAL shown here are:

$$f_1 = P_1 + P_2$$
$$f_2 = P_3 + P_4$$
$$P_1 = x_1 x_2 \bar{x}_3$$
$$P_2 = \bar{x}_1 x_2 x_3$$
$$P_3 = \bar{x}_1 \bar{x}_2$$
$$P_4 = x_1 x_2 x_3$$

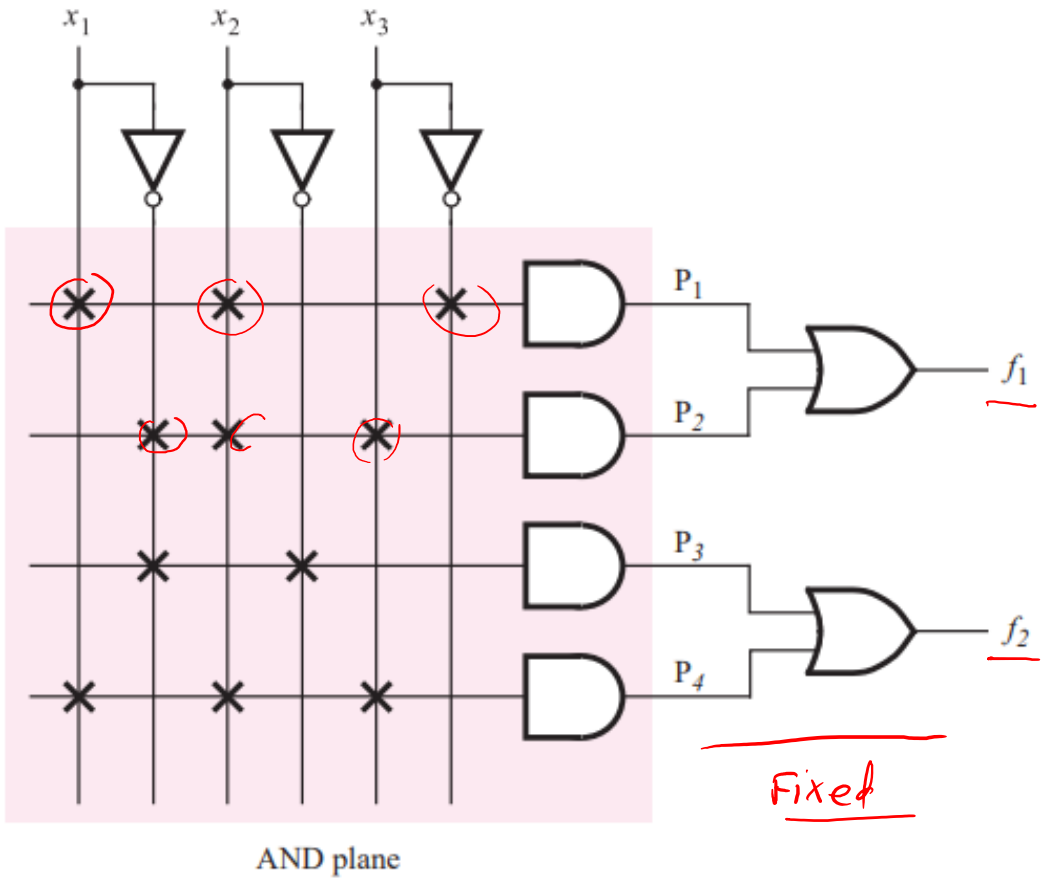


Figure 3.28 An example of a PAL.

Complex Programming Logic Device (CPLD)

A complex programmable logic device (CPLD) is a logic device with completely programmable AND/OR arrays and macrocells. Macrocells are the main building blocks of a CPLD, which contain complex logic operations and logic for implementing disjunctive normal form expressions. AND/OR arrays are completely reprogrammable and responsible for performing various logic functions. Macrocells can also be defined as functional blocks responsible for performing sequential or combinatorial logic

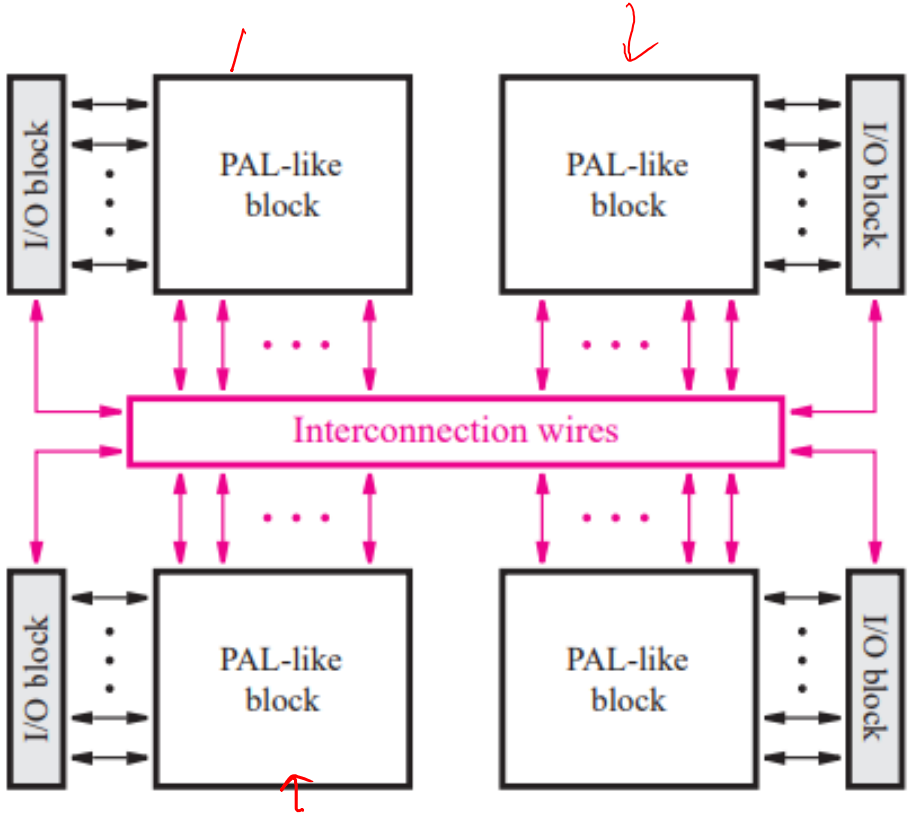
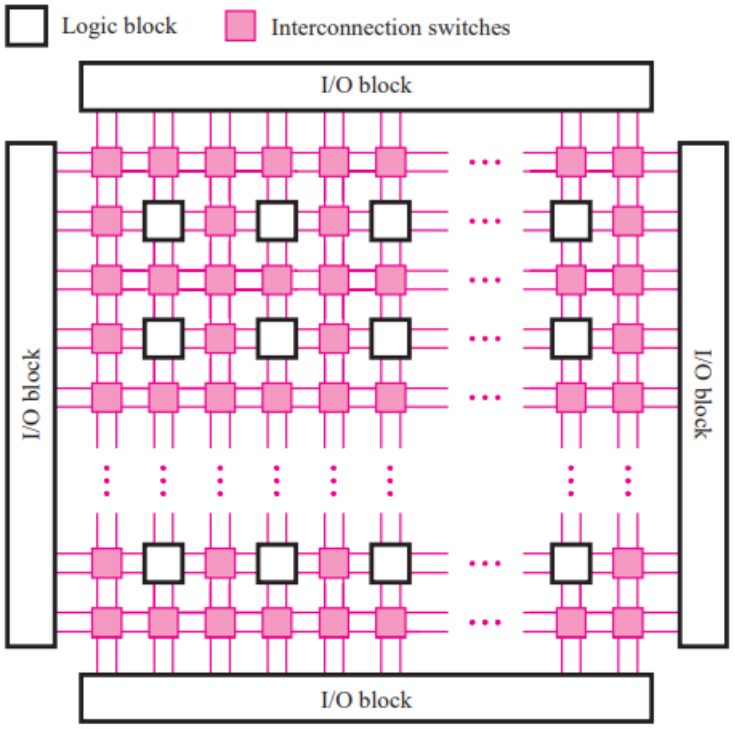
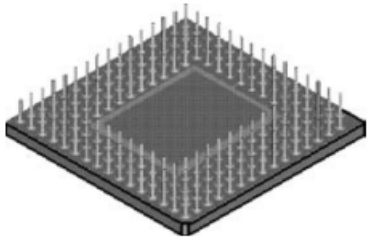


Figure 3.32 Structure of a complex programmable logic device (CPLD).

Field-Programmable Gate Array (FPGA)



(a) General structure of an FPGA

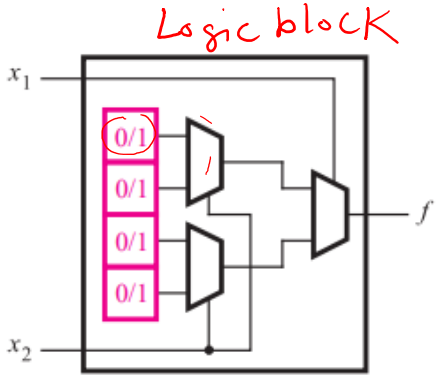


(b) Pin grid array (PGA) package (bottom view)

Figure 3.35 A field-programmable gate array (FPGA).

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term field-programmable. The FPGA configuration is generally specified using a hardware description language (HDL).

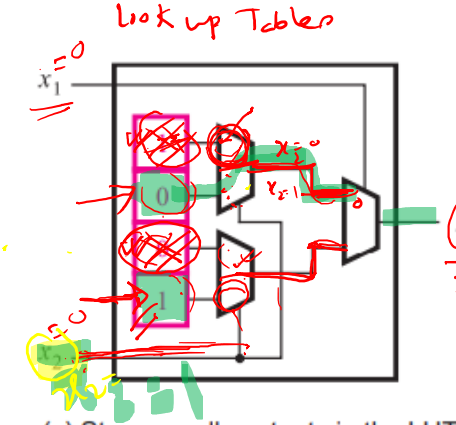
The most common FPGA architecture consists of an array of logic blocks (called configurable logic blocks, CLBs, or logic array blocks, LABs, depending on the vendor), I/O pads, and routing channels.



(a) Circuit for a two-input LUT

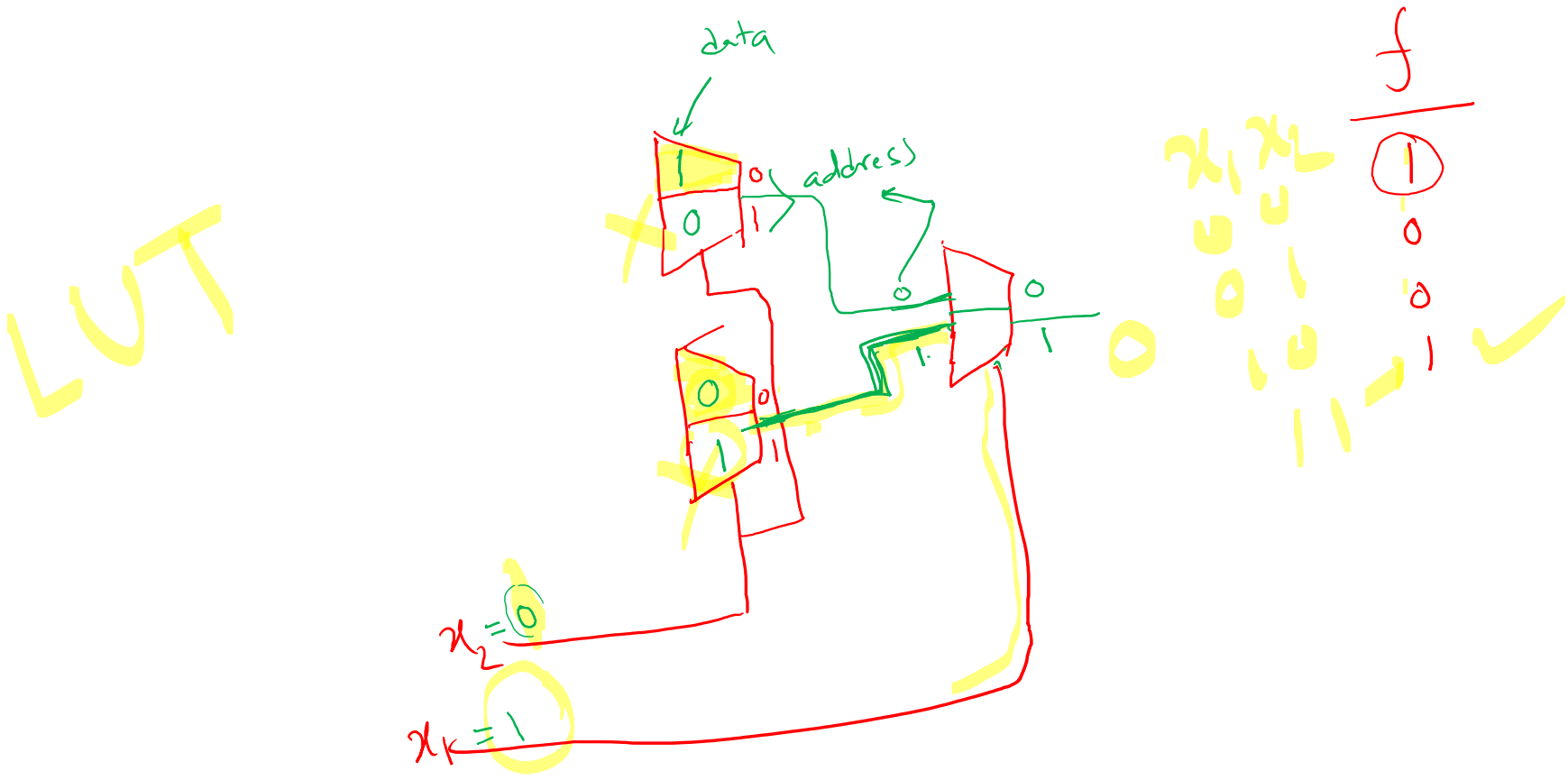
x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

(b) $f_1 = \bar{x}_1\bar{x}_2 + x_1x_2$



(c) Storage cell contents in the LUT

Figure 3.36 A two-input lookup table (LUT).


LUT
data
address
1
0
0
1
0
1
0
1
0
1
0
1
$x_2 = 0$
$x_1 = 1$
f
x_1 x_2
1
0
0
1

FPGA

Let us implement a function $f = f_1 + f_2 = x_1x_2 + \bar{x}_2x_3$

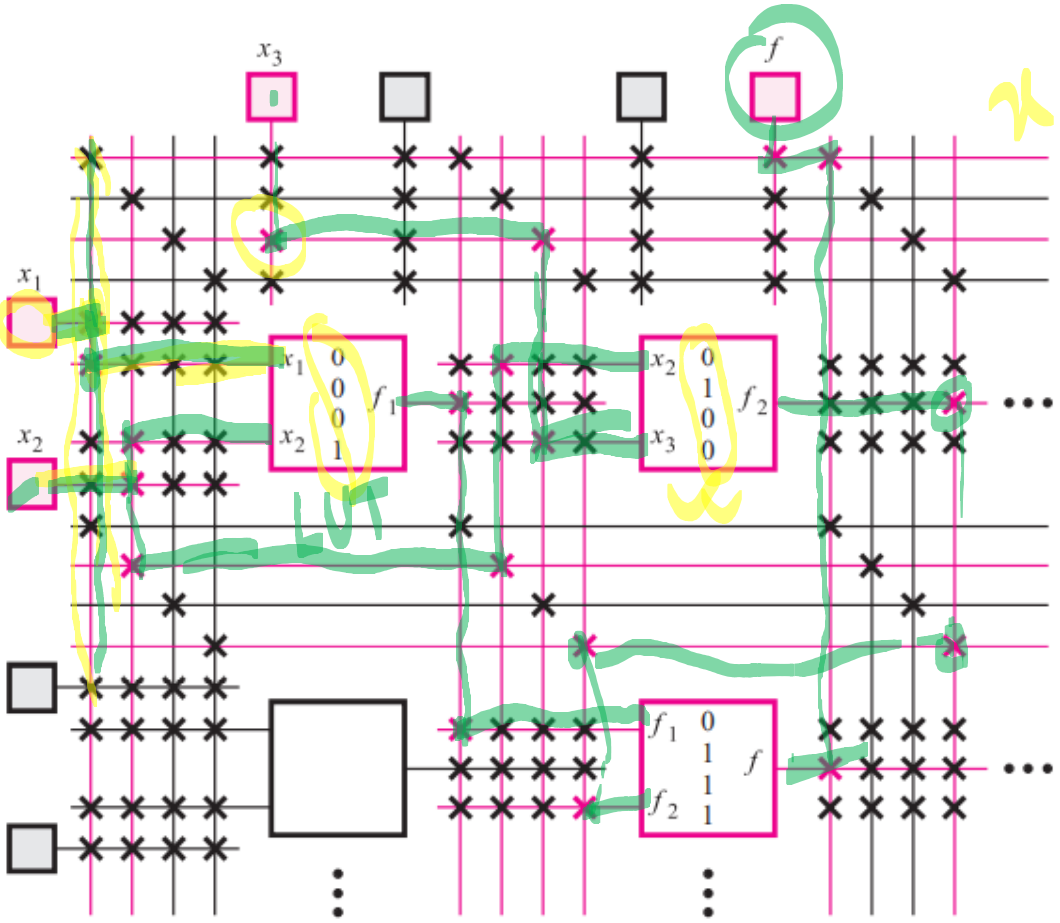
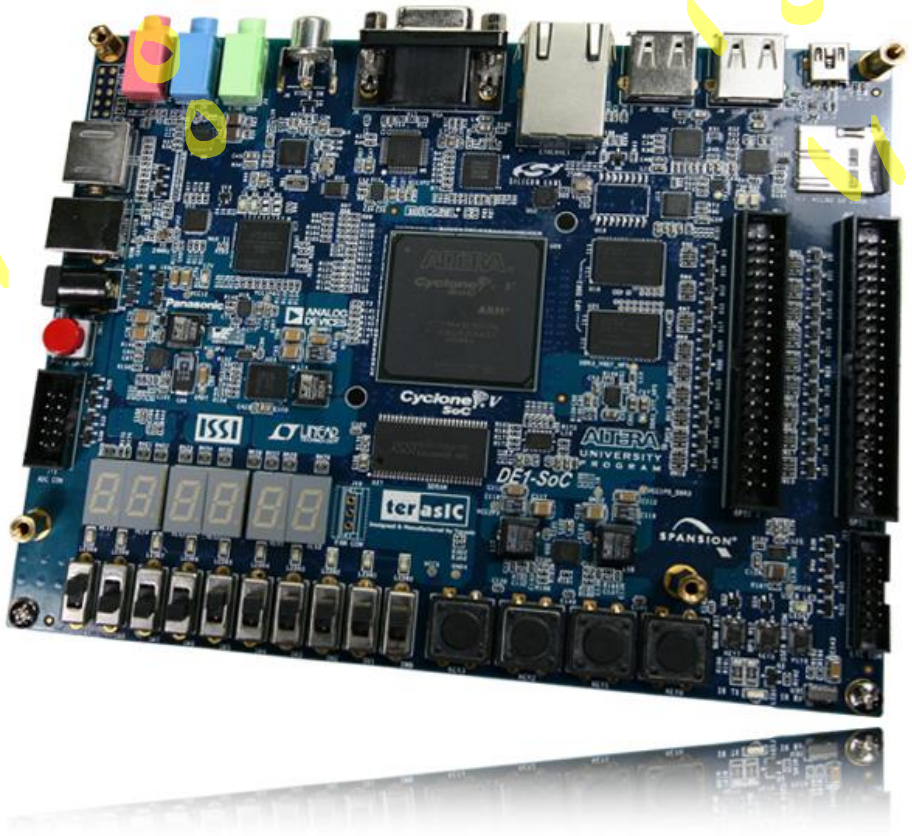


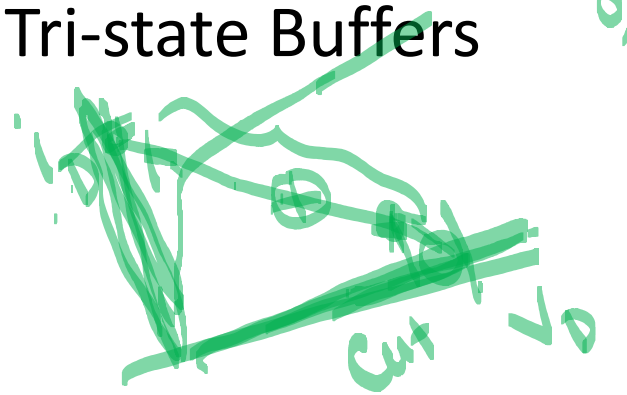
Figure 3.39 A section of a programmed FPGA.

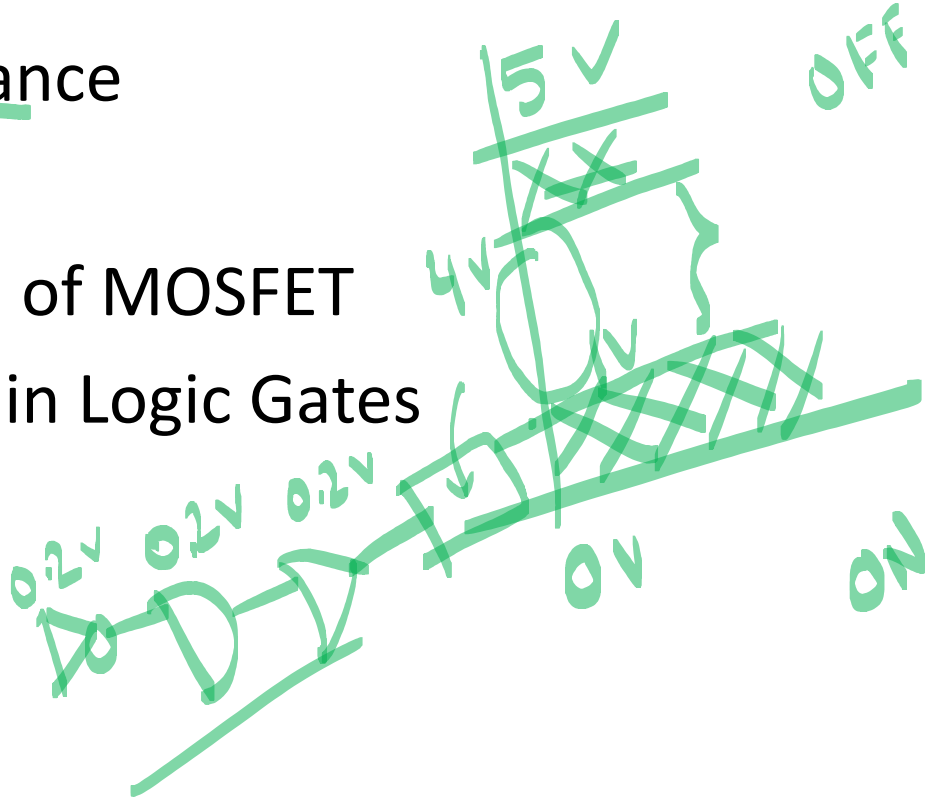


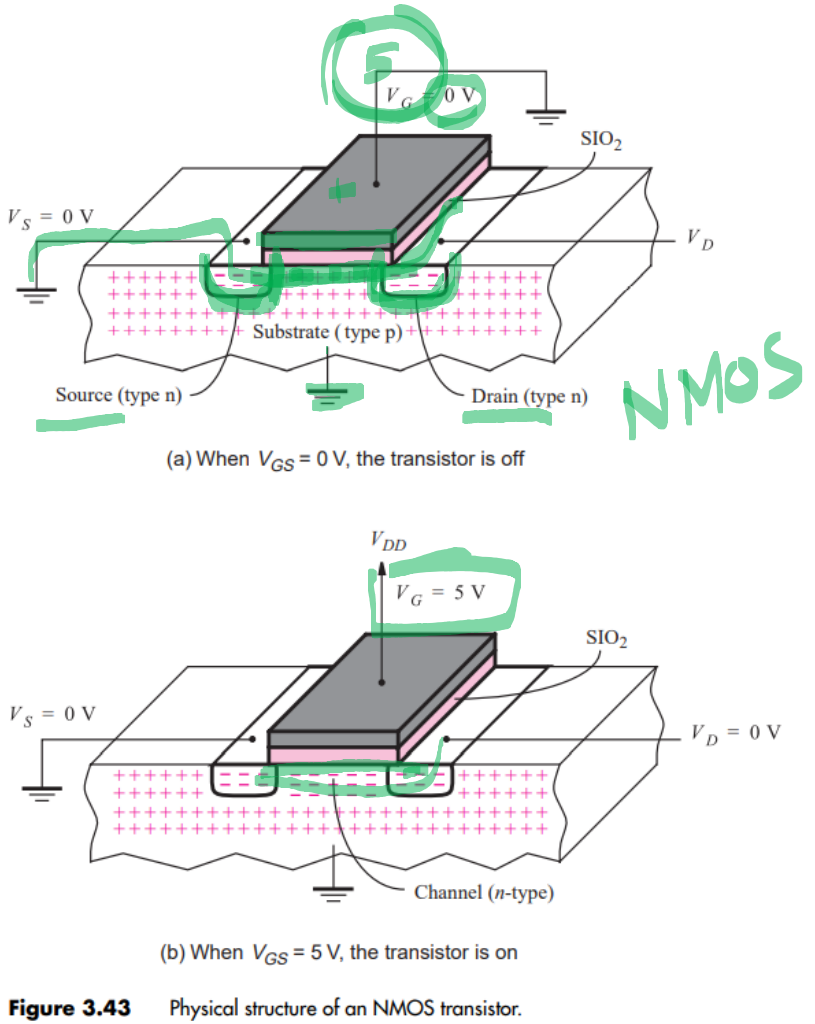
ALTERA PREFERRED BOARD FOR OpenCL™

Practical Aspect

- MOSFET Fabrication and Behavior
- MOSFET ON Resistance
- Noise Margin
- Dynamic Operation of MOSFET
- Fan-in and Fan-out in Logic Gates
- Buffers
- Tri-state Buffers







(a) When $V_{GS} = 0$ V, the transistor is off

(b) When $V_{GS} = 5$ V, the transistor is on

Figure 3.43 Physical structure of an NMOS transistor.

Practical Aspects

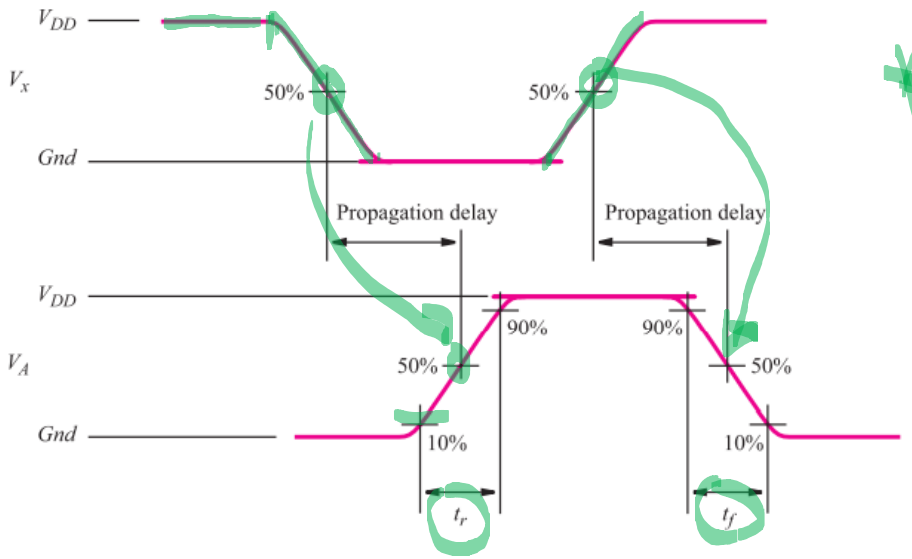


Figure 3.48 Voltage waveforms for logic gates.

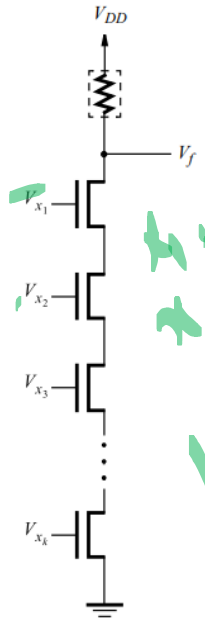
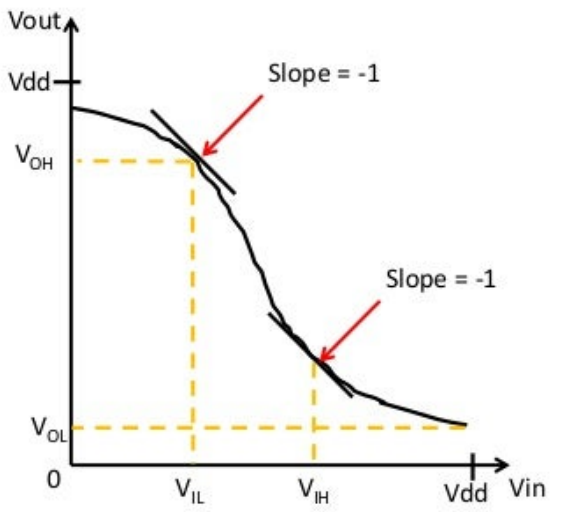
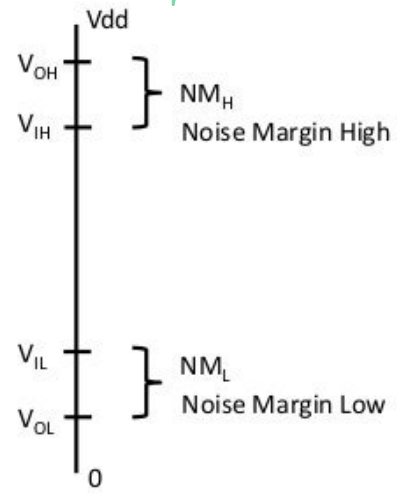


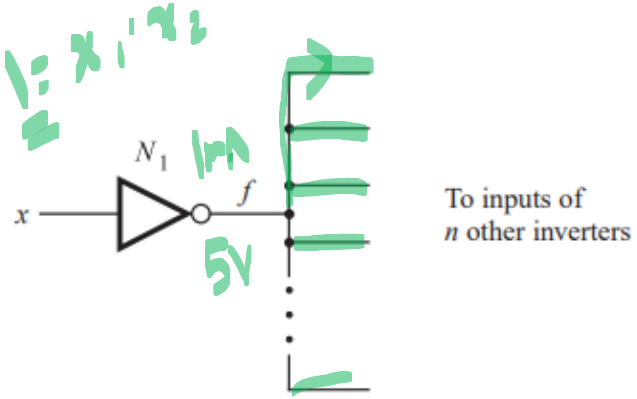
Figure 3.53 High fan-in NMOS NAND gate.





$NM_H = V_{OH} - V_{IH}$

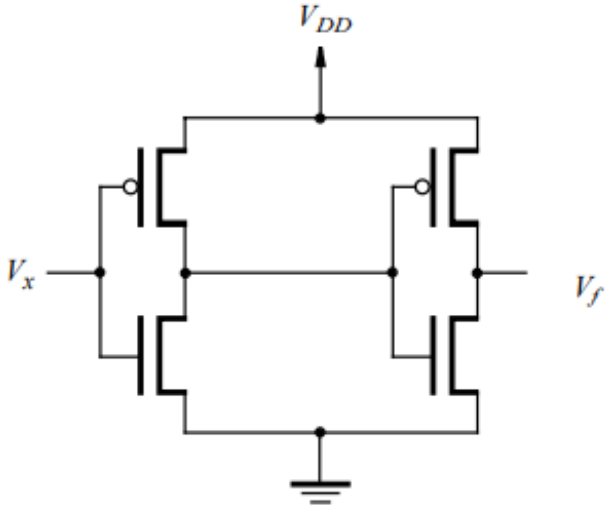
$NM_L = V_{IL} - V_{OL}$



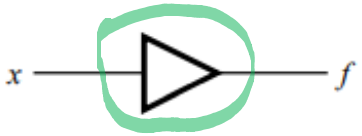
(a) Inverter that drives n other inverters

Buffer

A temporary memory area in which data is stored while it is being processed or transferred, especially one used while streaming video or downloading audio.



(a) Implementation of a buffer



(b) Graphical symbol

Figure 3.56 A noninverting buffer.

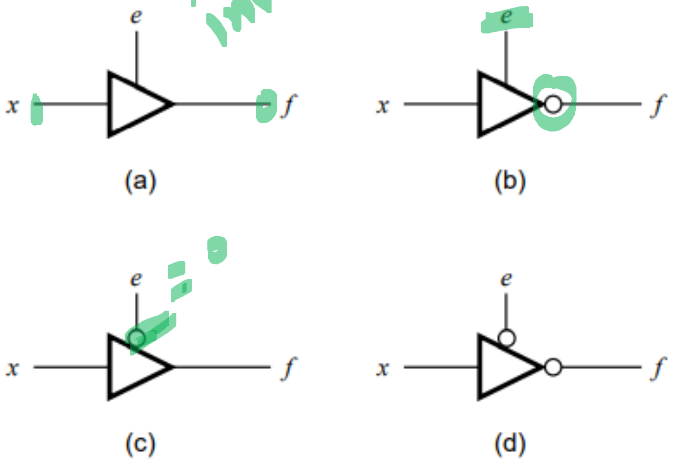
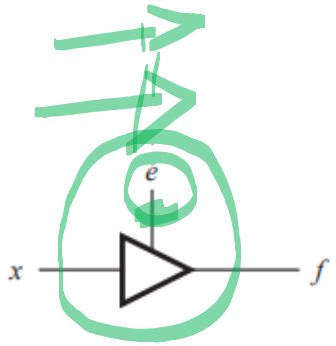
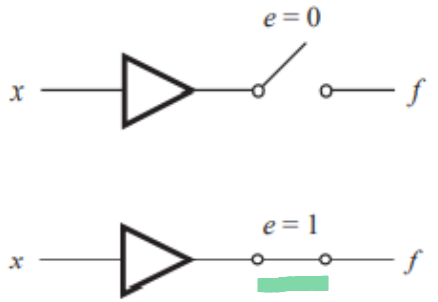


Figure 3.58 Four types of tri-state buffers.



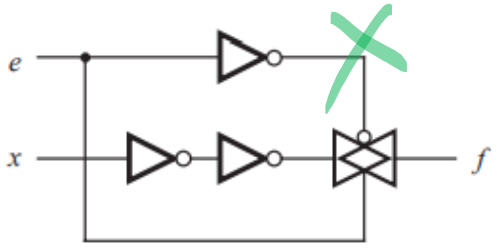
(a) A tri-state buffer



(b) Equivalent circuit

e	x	f
0	0	Z
0	1	Z
1	0	0
1	1	1

(c) Truth table



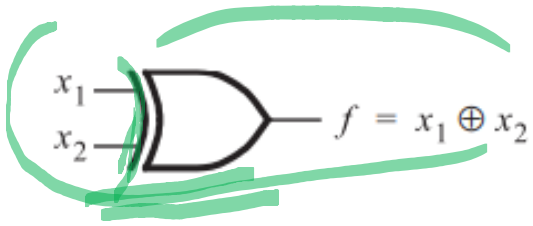
(d) Implementation

Figure 3.57 Tri-state buffer.

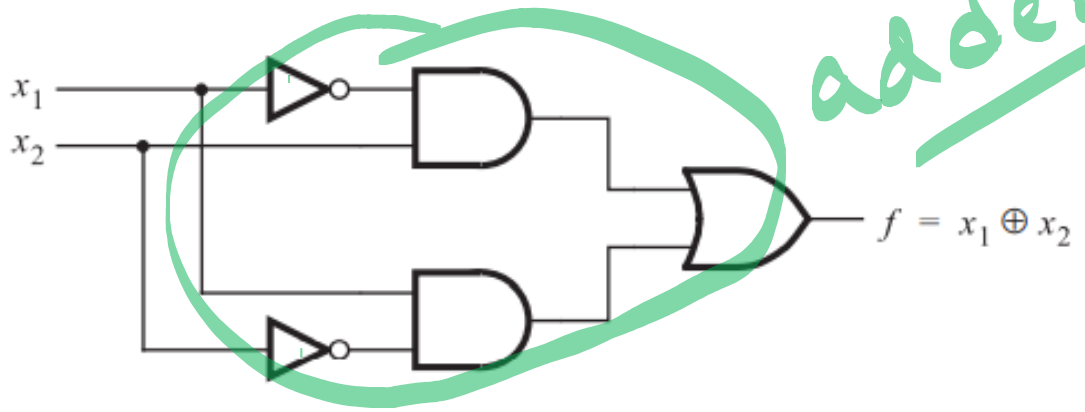
Exclusive-OR Gate

x_1	x_2	$f = x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

(a) Truth table



(b) Graphical symbol



(c) Sum-of-products implementation

Examples

Determine the function performed by the PLA shown in the figure.

$$f_1 = S_1 + S_2 + S_3$$
$$f_2 = S_4 + S_5 + S_2$$

$$S_1 = x_1 x_3$$
$$S_2 = x_1 \bar{x}_2$$
$$S_3 = \bar{x}_1 x_2 \bar{x}_3$$
$$S_4 = x_1 \bar{x}_3$$
$$S_5 = \bar{x}_1 x_2$$

6-8

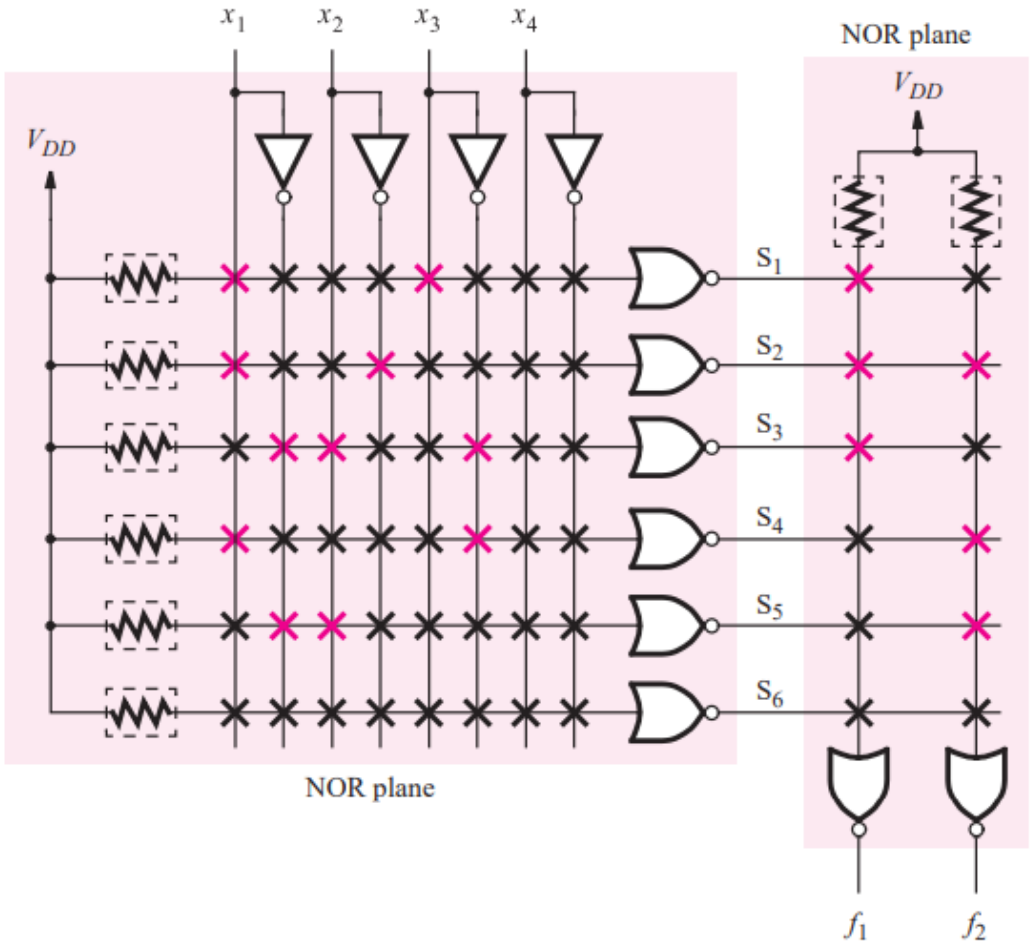


Figure 3.65 Programmable version of the NOR-NOR PLA.