DIGITAL TECHNICS I

Dr. Bálint Pődör

Óbuda University, Microelectronics and Technology Institute

11. LECTURE: COMBINATIONAL CIRCUITS WITH MEMORY ELEMENTS AND PLDs

1st year BSc course 1st (Autumn) term 2018/2019

1

11. LECTURE

1. Memory elements as combinational circuits
2. Programmable logic devices (PLD)

    Programmable logic array (PLA)

    Programmable array logic (PAL)

2

SYNTHESIS USING LOGIC GATES

The traditional process of logic synthesis is based on the application of logic gates.

Its more modern variant makes use of programmable logic devices too.

However in many case it is more advantageous to use a logic synthesis procedure based on the application of logic functional blocks.

IMPLEMENTING COMBINATIONAL LOGIC

The different steps involved in the design of a combinational logic circuit are as follows:

1. Statement of the problem.
2. Identification of input and output variables.
3. Expressing the relationship between the input and output variables.
4. Construction of a truth table to meet input–output requirements.
5. Writing Boolean expressions for various output variables in terms of input variables.
6. Minimization of Boolean expressions.
7. Implementation of minimized Boolean expressions.

IMPLEMENTING COMBINATIONAL LOGIC

These different steps are self-explanatory. One or two points, however, are worth mentioning here.

There are various simplification techniques available for minimizing Boolean expressions, which have been discussed in the previously. These include the use of theorems and identities, Karnaugh mapping, the Quine–McCluskey tabulation method and so on. Also, there are various possible minimized forms of Boolean expressions.

The following guidelines should be followed while choosing the preferred form for hardware implementation:

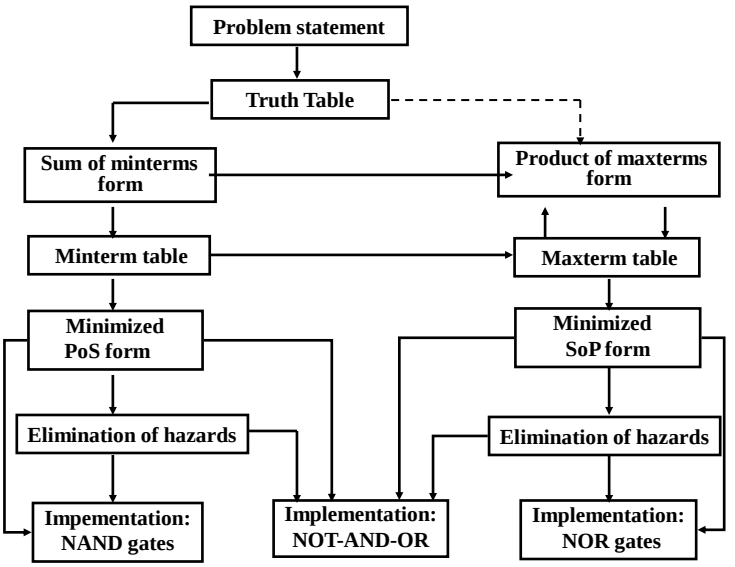
IMPLEMENTING COMBINATIONAL LOGIC

1. The implementation should have the minimum number of gates, with the gates used having the minimum number of inputs.

2. There should be a minimum number of interconnections, and the propagation time should be the shortest.

3. Limitation on the driving capability of the gates should not be ignored.

It is difficult to generalize as to what constitutes an acceptable simplified Boolean expression. The importance of each of the above-mentioned aspects is governed by the nature of application.



Flow diagram of logic synthesis procedure using gates

IMPLEMENTATION OPTIONS

Ready-made catalog-order (modular) devices (gates, functional blocks, etc.)

Custom-design devices

Gate-array devices

Programmable logic devices (PLD), e.g. programmable logic array (PLA), programmable array logic (PAL), etc.

Table look-up (ROM)

Microcomputer

PRGRAMMABLE LOGIC

Now we will discuss a new category of logic devices called *programmable logic devices* (PLDs).

The function to be performed by a programmable logic device is undefined at the time of its manufacture.

These devices are programmed by the user to perform a range of functions depending upon the logic capacity and other features offered by the device.

9

Realization of combinational networks with memory elements

10

PROPERTIES OF MEMORY ELEMENTS

Memory elements with permanent contents:

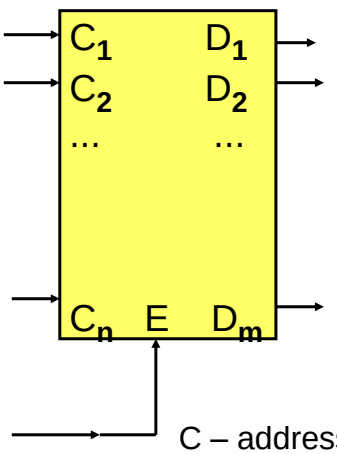
Read Only Memory (ROM),

They can be used to implement combinational logic networks.

The data (word) stored in a memory is a binary combination (D_1, D_2, ... D_m), which can be made available by an address (C_1, C_2, ... C_n), which in turn is also a binary combination.

11

MEMORY ELEMENT: FUNCTIONAL DIAGRAM



In response to an n-bit combination at the input, the m-bit combination stored in the n-th cell will be available on the output.

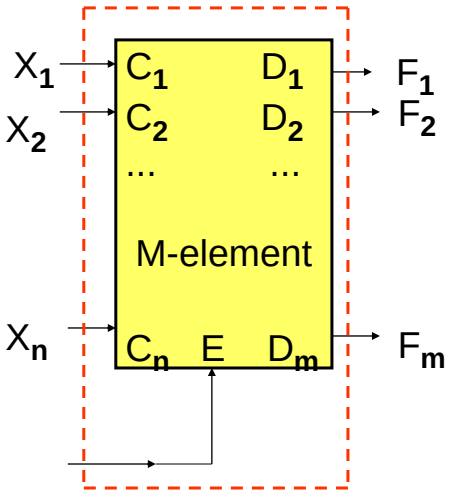
The logic signal on the enable input serves for enabling/disenabling the data output.

Using the enable input more memory elements can connected to form a common unit.

C – address input
D – data output
E - (enable input)

12

MEMORY ELEMENT AS A COMBINATIONAL NETWORK



Combinational network

The memory element realizes a combinational network having n input variables and m output functions.

The loading of data can be performed directly on the basis of the truth table (i.e. the disjunctive canonic form), or extended sum-of product form.

NO MINIMIZATION IS NECESSARY !

ROM DEMO: UNIVERSAL COMBINTIONAL CIRCUIT

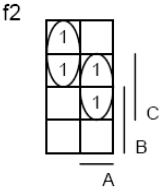
$f_1 = A\,B\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C}$

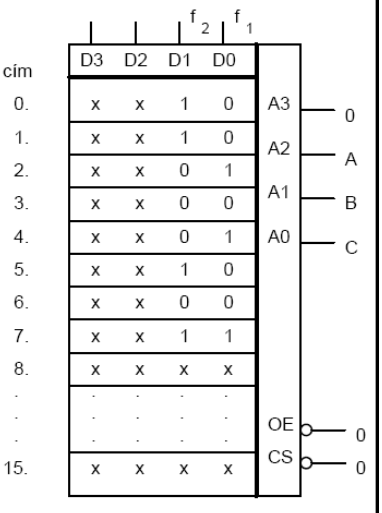
$f_2 = \bar{A}\,\bar{B} + A\,C = \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + A\,\bar{B}\,C + A\,B\,C$

Minterms: f1: 1, 4, 7, and f2: 0, 1, 5, 7

Example:

implementation of two 3-variable function using a memory element



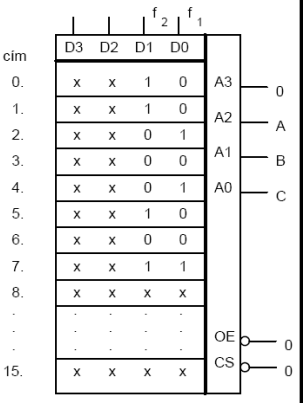


ROM: UNIVERSAL COMBINTIONAL CIRCUIT

To implement the two 3-variable function a 8x2 bit ROM would be enough, but the smallest available on the market 16x4 bit .

A2, A1, A0 address – A, B, C variables
D0, D1 outputs for f1, f2 functions.

Generation of truth table and programming:
A3 address line connected to 0 (only the lower 8 words of the ROM is used, the remaining area is not relevant)
D3, D2 – not relevant
CS (*chip select*), OE (*output enable*) connected to active level, permanent enabling.



15

ROM APPLICATIONS: CODE CONVERSIONS

One important application of ROMs in combinational logic is the code conversion.

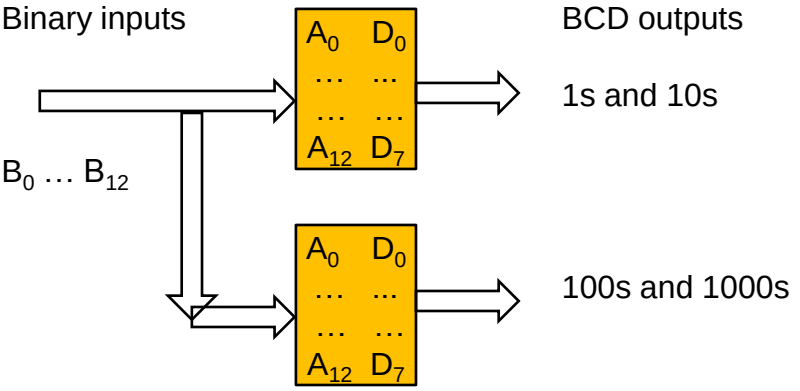
n-bit code → m-bit code → necessary memory capacity: $m \times 2^n$.

8-bit binary code → 8-bit Gray code: 256x8 ROM.

13-bit binary code → 4 tetrade BCD code:
Two 8-kbyte kapacity EPROM, 1s, and 10s, and 100s and 1000s respectively, (13 bit: 0-8191).

16

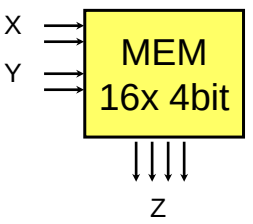
CODE CONVERSION USING ROM



13-bit binary code to 4 tetrade BCD code converter

APPLICATION EXAMPLE: MULTIPLICATION BY ROM READOUT

	X	Y	Z
0*0	00	00	0000
0*1	00	01	0000
0*2	00	10	0000
0*3	00	11	0000
1*1	01	01	0001
1*2	01	10	0010
1*3	01	11	0011
2*0	10	00	0000
2*1	10	01	0010
2*2	10	10	0100
2*3	10	11	0110
3*0	11	00	0000
3*1	11	01	0011
3*2	11	10	0110
3*3	11	11	1001



Fast, simple, cheap,
Can generate any function
(look-up-table, LUT)

18

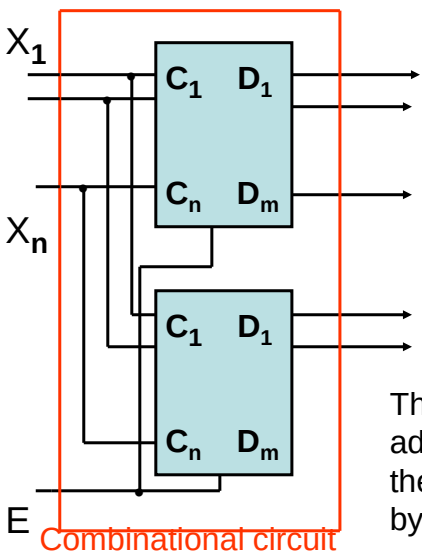
DYNAMIC BEHAVIOUR, HAZARDS

For memory elements the concepts of static and dynamic hazards are not applicable as for the gate circuits, therefore the usual hazard elimination procedures are not applicable.

However the important aspect is that after an appropriate cycle time following the address change, the data will be valid available on the output. The effects of transients during the cycle period can be eliminated using synchronization or external control (e.g. using the ENABLE input).

19

INTERCONNECTION OF TWO MEMORY ELEMENTS



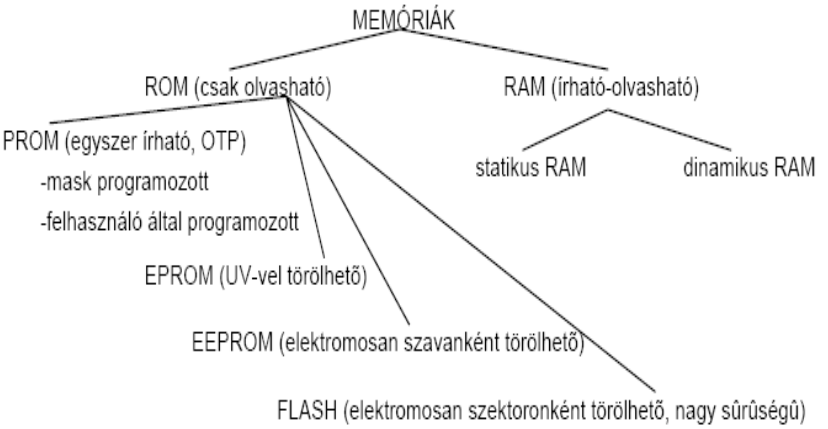
Appropriately connecting the address and enable inputs of memory elements having m outputs m+k outputs can be realized (in the figure k = 2).

The input variables are connected to address inputs of both elements, and the two additional outputs are ensured by the second element.

20

MEMORY ELEMENTS: SEMICONDUCTOR MEMORIES

Two groups:
1. Changeable content(read-write) memories,
2. Permanent (read only) memories.



PERMANENT (READ ONLY) MEMORIES

ROM - Read Only Memory. IC chip storage element, retains its contents if supply voltages is switched off. Contents can only be read, not possible to write them under normal operational conditions.
Entering (writing) information: programming.

Data stored in a ROM can be accessed randomly (random access), however because of historic reasons the random access memory (RAM) usually designates the read/write memoties (RWM).

22

PROGRAMMABLE PERMANENT MEMORIES (PROM)

PROM Programmable Read Only Memory

Mask-programmed PROM: the manufacturer determines the contents of the ROM by appropriate layout of the interconnection metallization.

Field-programmable (burn-in) PROM (not erasable): the programming of the individual cells is accomplished by burning-out of the metal alloy or poly-silicon fuses belonging to the individual cells.
Original (virgin) state logic 1, burn-ot of the fuse programs-in logic 0.

Enabling technology: Si bipolar diodes/transistors.

23

REPROGRAMMABLE PROM: EPROM AND EEPROM

EPROM (Erasable PROM) circuit technology: Si MOS FET.

Programming: injection of charges.

Erasing: Illumination with UV light, charges are released, information content is erased.
After erasure the remaining state is either full 0 or full 1.

EEPROM Electrically Erasable PROM: can be erased electrically. Also MOS technology.

Representative parameters: programming/loading and erasing 10 – 50 msec.
Number of erasing processes:10^4 -10^6.

24

Realization of combinational networks with programmable logic devices (PLDs)

25

WHY PROGRAMMABLE LOGIC?

- Facts:
  - It is most economical to produce an IC in large volumes
  - Many designs require only small volumes of ICs
- Need an IC that can be:
  - Produced in large volumes
  - Handle many designs required in small volumes
- A programmable logic part can be:
  - made in large volumes
  - programmed to implement large numbers of different low-volume designs

26

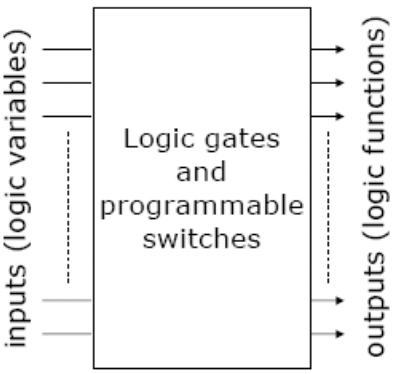
PROGRAMMABLE LOGIC DEVICES (PLD) FUNDAMENTALS

A PLD is a general purpose chip for implementing logic circuitry

Contains a collection of logic circuit elements that can be customized in different ways

Can be viewed as a black box containing logic gates and programmable switches that allow for different connections between the logic elements

Can implement whatever logic circuit is needed – subject to limitations of the device



PROGRAMMABLE LOGIC: ADDITIONAL ADVANTAGES

Many programmable logic devices are *field- programmable*, i.e., can be programmed outside of the manufacturing environment.

Most programmable logic devices are *erasable* and *reprogrammable*. Allows "updating" a device or correction of errors. Allows reuse the device for a different design - the ultimate in re-usability!

Programmable logic devices can be used to prototype design that will be implemented for sale in regular ICs.

Complete Intel Pentium designs were actually prototype with specialized systems based on large numbers of VLSI programmable devices!

PROGRAMMABLE CONFIGURATIONS

Programmable Array Logic (*PAL*)® - 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” .

PROGRAMMABLE LOGIC DEVICES: PLD

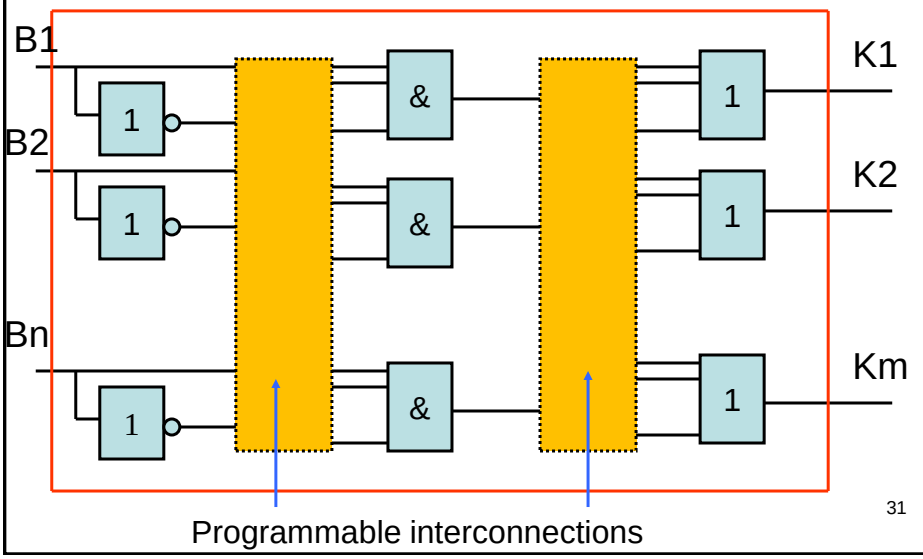
Programmable Logic Device (PLD) – one of the most common forms is the programmable logic matrix or Programmable Logic Array, PLA: certain number of AND and OR gates, and a number of inverters (to generate the negated values of the input variables).

The internal connections are realized on the basis of the conceptual logic diagram of the two-level multiple output combinational network.

In generating the conceptual logic diagamm the usual minimization and hazard elimination methods can be used.

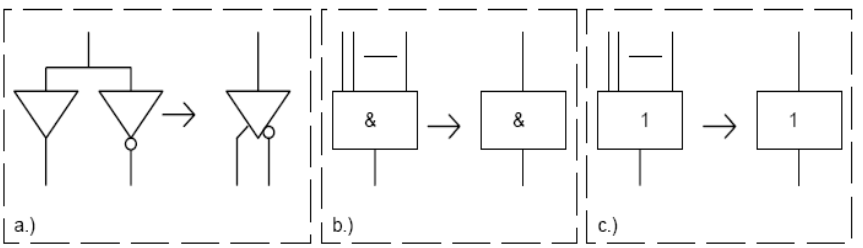
30

PLA ARCHITECTURE



31

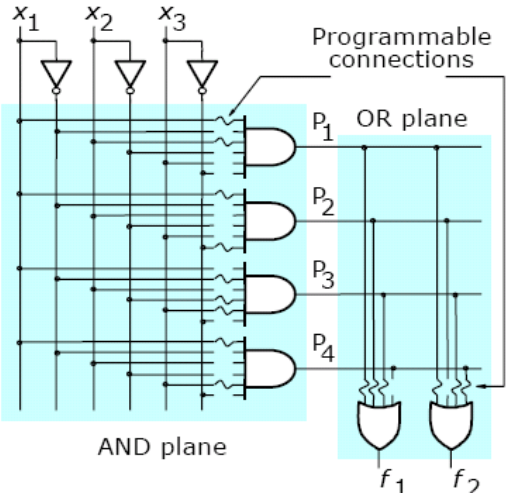
PLA: USUAL NOTATIONS



32

PLA FUNCTIONAL DIAGRAM

Gate-level diagram of a PLA

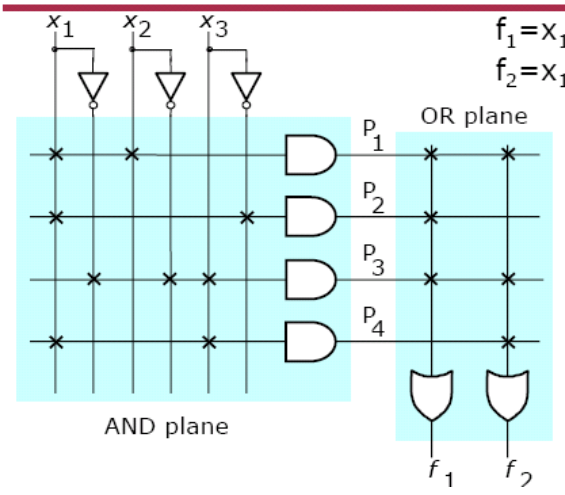


PLA SCHEMATIC CIRCUITRY

Customary schematic of a PLA

$f_1 = x_1x_2 + x_1x_3' + x_1'x_2'x_3$

$f_2 = x_1x_2 + x_1'x_2'x_3 + x_1x_3$



PLA ELEMENTS: PROPERTIES AND USE

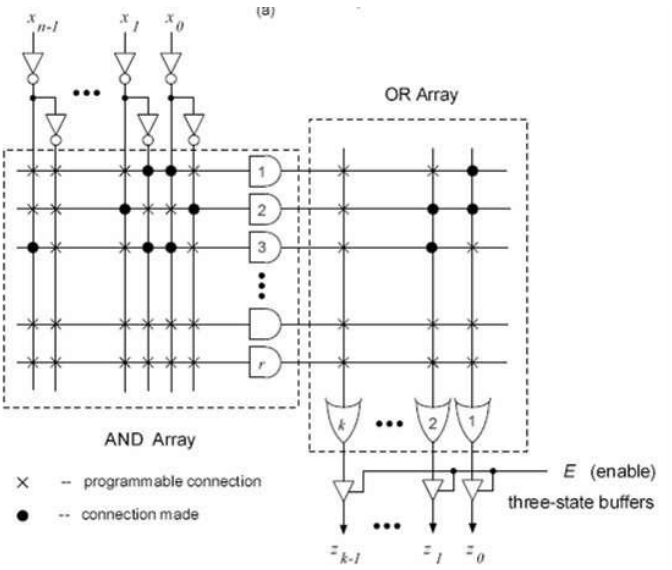
The technical characteristics of PLA elements determine whether one or more chips are necessary to implement a given combinational network:

- the number PLA inputs;
- the number of PLA outputs;
- the number of internal AND gates.

The number of inputs of internal AND gates usually agrees with the number of inputs of the whole PLA.
The number of inputs of the internal OR gates usually corresponds to the number of internal AND gates.
PLAs are also viable without incorporated input INVERTERs, the each input variable uses up two PLA inputs assisted by an external INVERTER.

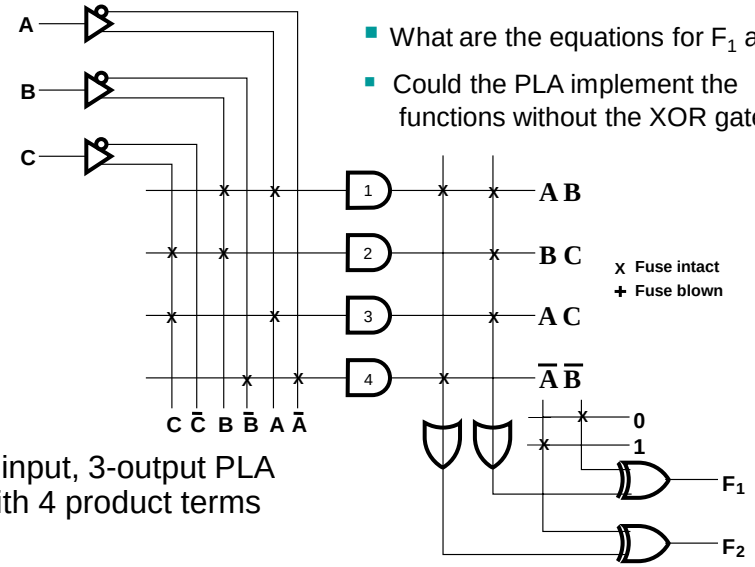
35

PROGRAMMED PLA EXAMPLE



36

PROGRAMMABLE LOGIC ARRAY EXAMPLE



- What are the equations for F_1 and F_2?
- Could the PLA implement the functions without the XOR gates?

- 3-input, 3-output PLA with 4 product terms

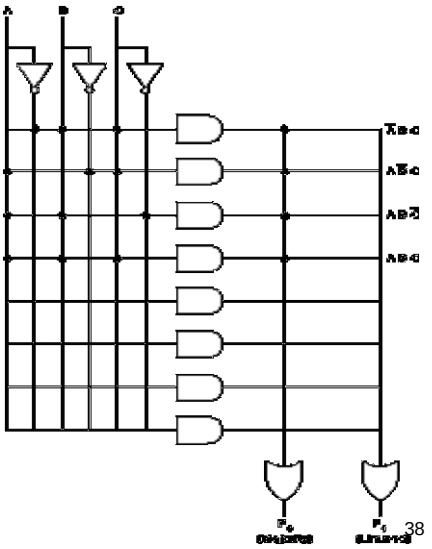
37

EXAMPLE: IMPLEMENTATION OF THREE-VARIABLE MAJORITY FUNCTION

3-variable majority function
(number of 1s is greater than the number of 0s)

$M = \Sigma^3(3,5,6,7) =$

$\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

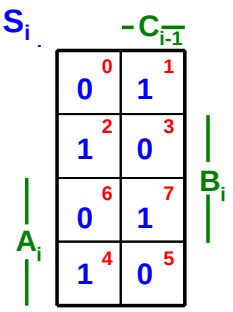


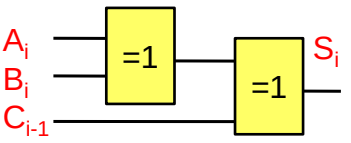
38

EXAMPLE: S_i (SUM) FUNCTION OF THE 1 BIT FULL ADDER

„chessboard pattern”
symmetric function

i	(4) A_i	(2) B_i	(1) C_{i-1}	S_i
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1



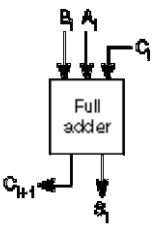


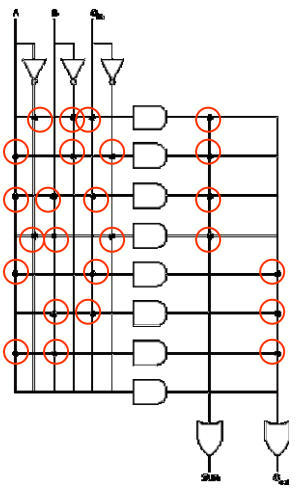
Sum function $S_i = A_i \oplus B_i \oplus C_{i-1}$

Alternative implementation: two-level AND-OR using PLA [39]

EXAMPLE: FULL ADDDER USING PLA

A_i	B_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1





S: direct from the truth table
(4 minterms)

C: minimized (3 terms)

40

INTERCONNECTION OF PLA'S (1)

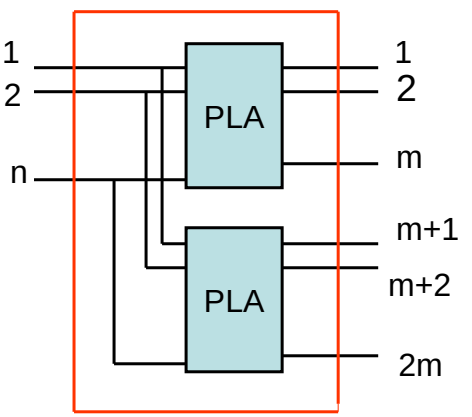
If the logic function to be implemented in its minimized form cannot be implemented on a single PLA chip, the two or more PLA elements should be appropriately interconnected.

The method and way of interconnection depends on which parameter of the PLA is the limiting factor.

41

INTERCONNECTION OF PLA'S (2)

If the number of outputs of the PLA chip is insufficient, then e.g. two PLAs with parallel connected inputs can be used.



42

INTERCONNECTION OF PLA'S (3)

If the number of internal AND gates is less than necessary, then the inputs of two PLA chips are to be connected un parallel, and the corresponding outputs can be connected in pairs as wired-OR (using external resistors).

If for any reason (mostly circuit consideration) the wired-OR logic cannot be used, then external OR gates or extra PLAs are necessary to implement the necessary OR connections.

43

INTERCONNECTION OF PLA'S (4)

Other, here not discussed cases, or cases when several simultaneous limiting factors should be taken into consideration, can be handled analogously.

44

PROGRAMMING OF PLA, FPLA

The programming of PLA can be mask-programming, this can only be done in the manufacturers premises.

Most PLAs can be programmed by the user (Field Programmable Logic Array (FPLA)). The programming is performed by burning-out the built-in ”fuses” with appropriately controlled current pulses.

45

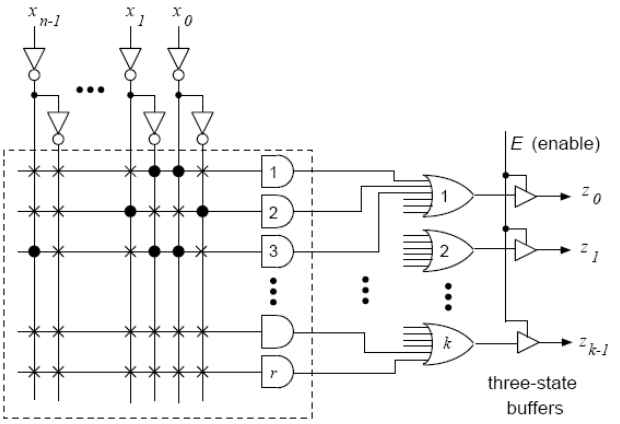
PROGRAMABLE ARRAY LOGIC

The Programmable Array Logic (PAL) element is simpler than the a PLA, because only the AND gates can be programmed, the connections of OR gates are fixed by the manufacturer.

In the AND matrix metal-semiconductor(Schottky) diodes are incorporated, and the diodes are connected with a metal layer to the connecting wires, which can be burned out by appropriate over-voltage.

46

PROGRAMMED PAL EXAMPLE

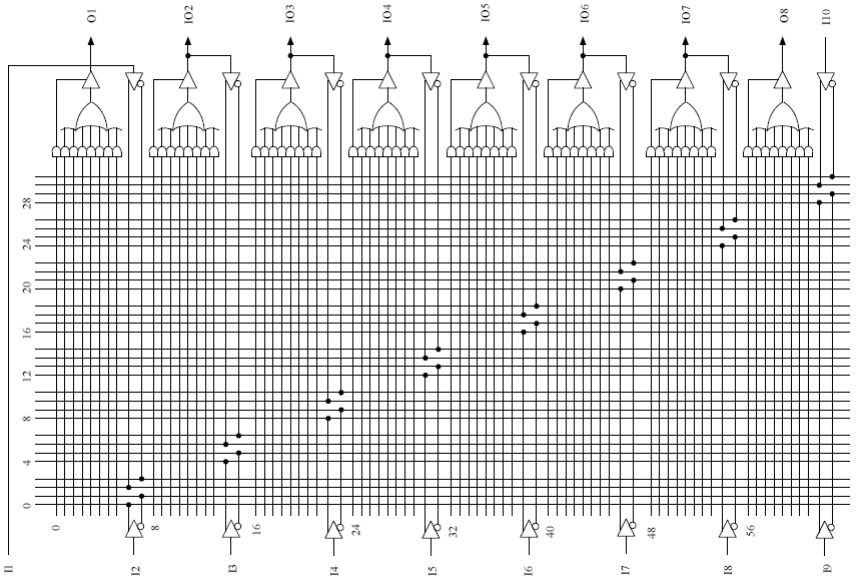


Fixed OR array

Faster, more inputs and product terms compared to PLAs

47

16-INPUT, 8-OUTPUT PAL (P16/H8)



PLA: HAZARDS

In the case of a network implemented on a single PLA module the synthesis is based on the conceptual two-level logic diagram, and the hazard elimination might and should be performed using the standard procedures.

In the case of more complex networks consisting of several PLA elements further checks and analysis might be necessary.

49

COMPARISON OF SOLUTIONS BASED ON MEMORY ELEMENTS AND ON PLA

Speed of operation:

Using PLA higher operational speeds can be achieved.

Reason: the PLA is essentially a two-level logic network, further on because of technological reasons a PLA is intrinsically faster then a memory element.

50

COMPARISON OF SOLUTIONS BASED ON MEMORY ELEMENTS AND ON PLA

Hazards and hazard elimination:

When using PLA the hazard elimination should be performed during the synthesis applying the known procedures.

The behaviour of memory elements within the cycle is not defined ususally.
Solution: external control or synchronization.

51

COMPARISON OF SOLUTIONS BASED ON MEMORY ELEMENTS AND ON PLA

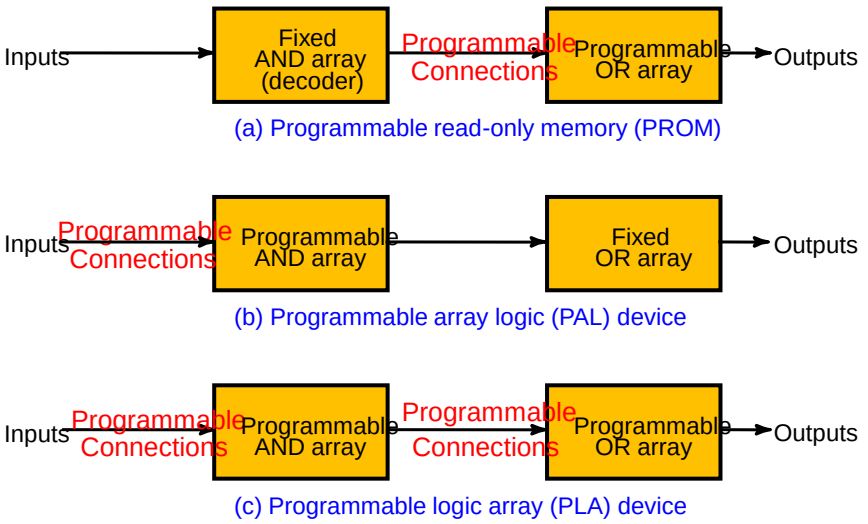
Synthesis procedure:

The synthesis procedure is simpler in the case of ROMs, because it is based directly on the truth table (extended sum-of product form).

In order to program a PLA it is necessary to establish the minimized hazard-free two level conceptual logic diagram.

52

ROM, PAL AND PLA CONFIGURATIONS



53

COMPARISON OF LOGIC OF ROM, PLA, AND PAL IMPLEMENTATATIONS

	AND gate layer	OR gate layer
Memory element	fixed	programmed
PLA	programmed	programmed
PAL	programed	fixed

54

END

55