Presentation 4: Programmable Combinational Devices

Asst. Prof Dr. Ahmet ÖZKURT
DEUEEE

Based on the Presentation by
Prof. Kim, Young Ho
Dept. of Information Computer Engineering
E-mail : yhkim@hyowon.cs.pusan.ac.kr

6.1 Memory & Programmable Logic Device Definitions

- Memory
  - a collection of cells capable of storing binary information
  - memory contains electronic circuits for storing & retrieving info

- a digital computer
  - consist of three major units
    - processing unit (registers + combinational logic)
    - memory unit
    - input-output unit

6.1 Memory & Programmable Logic Device Definitions

- two types of memories
  - RAM (Random-Access Memory)
    - can perform both read & write operation
  - ROM (Read-Only Memory)
    - can perform only the read operation (cannot write)
    - the existing information cannot be altered
    - a programmable logic device (PLD)
      (programming: a H/W procedure that specifies the bits
      that are inserted into the H/W configuration of the device)
- Programmable Logic Device (PLD)
  - ROM, PLA, PAL, CPLD, & FPGA
  - IC with internal logic gates (connected by a programmable process)
    - initial state: all the fuses are intact
    - programming by blowing those fuses along the paths

6.6 Programmable Logic Technologies

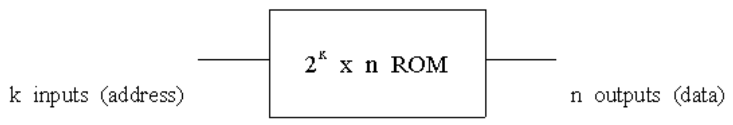
- Five programmable logic devides (PLDs)
  - ROM, PLA, PAL, CPLD, & FPGA
- Programming technologies
  - fuse
    - oldest of the programming technologies
    - each of programmable points consists of a connection,
      formed by a fuse
    - 2 connection states, CLOSED & OPEN
  - mask programming
    - by semiconductor manufacturer
  - antifuse
    - the opposite of a fuse
  - static RAM bit
    - drive the gate of an MOS transistor at the programming point

6.6 Programmable Logic Technologies

- use of programming technologies
  - control connections
  - implement logic by using lookup tables
    - input: address inputs for reading the SRAM
    - output: stored values for the addressed word
  - control transistor switching

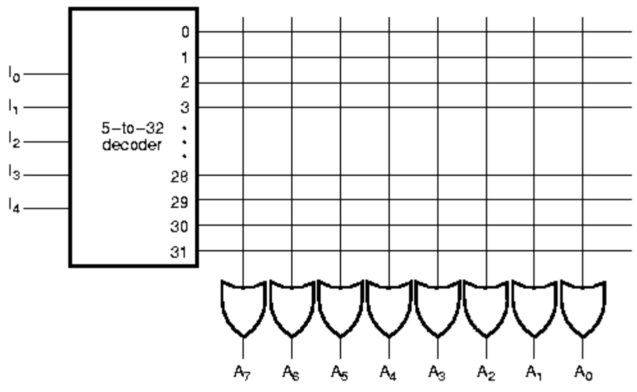
6.7 Read-Only Memory

- a memory device in which permanent binary info is stored
- once a pattern is established, it stays even when power is off
- consist of k address inputs and n data outputs



6.7 Read-Only Memory

(ex) a 32 x 8 ROM

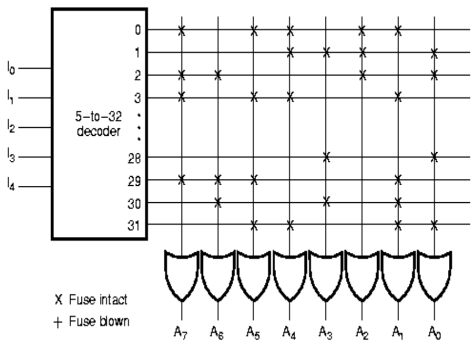


6.7 Read-Only Memory

(ex) the contents of a 32 x 8 ROM

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		.							.			
		.							.			
		.							.			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

programming the ROM
according to the truth table



6.7 Read-Only Memory

- Types of ROMs
  - mask programming (ROM)
  - fuse (PROM)
  - erasable floating gate technology (EPROM)
  - electrically erasable technology (EEPROM, E2PROM)

- Combinational Circuit Implementation
  - a decoder generates the 2^k minterms of the k input variables
  - inserting OR gates to sum the minterms of Boolean functions
    => can generate any desired combinational circuit
  - ROM essentially includes both decoder & OR gates
  - ROM outputs can be programmed to represent the Boolean functions in a combinational circuit
  - ROM may be considered as a comb circuit with (8) outputs,
    each is a function of the (5) input variables
    $A_7(I_4,I_3,I_2,I_1,I_0) = \Sigma\, m(0,2,3,...,29)$
  - widely used to implement complex combinational circuits directly

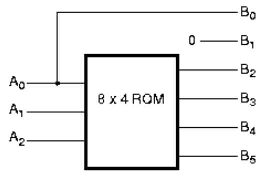
6.7 Read-Only Memory

Ex 6.1 Design a comb circuit using a ROM

- accepts a 3-bit number & generates an output binary No. equal to the square of the input No

Inputs			Outputs						
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

- 3 inputs & 6 outputs
  but B0 = A0; B1 = 1;
  ☒ only need 4 outputs
  ☒ ROM must be 8 x 4



(a) Block diagram

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

(b) ROM truth table

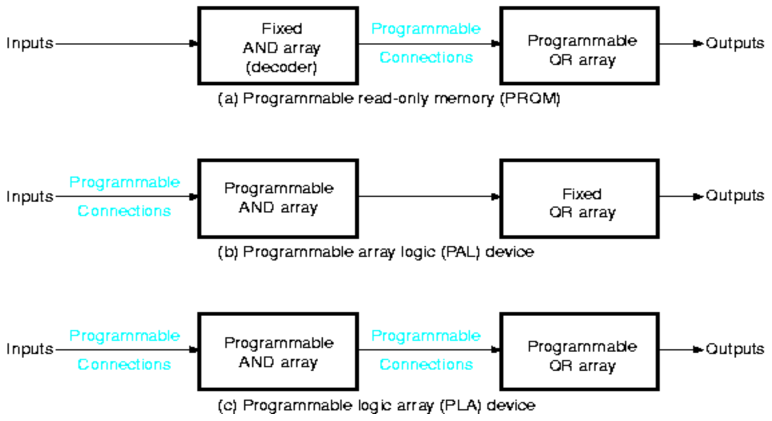
6.7 Read-Only Memory

- Programmable Logic Device
  - an integrated circuit with an array of gates that are connected by programmable fuses
  - the gates in a PLD are divided into AND array & OR array to provide and AND-OR sum of products implementation
- PROM
  - a fixed AND array constructed as a decoder & programmable connections for the output OR gates
  - implements Boolean functions in sum-of-minterms form
- PAL
  - a programmable connection AND array & a fixed OR array
  - AND gates are programmed to provide the product terms, which are logically summed in each OR gate

6.7 Read-Only Memory

- PLA
  - most flexible PLD
  - both AND & OR arrays can be programmed
  - product term in the AND array may be shared by any OR gate to provide the required sum of products implementation
- advantage of using the PLD
  - can be programmed to incorporate a complex logic function within one IC

6.7 Read-Only Memory

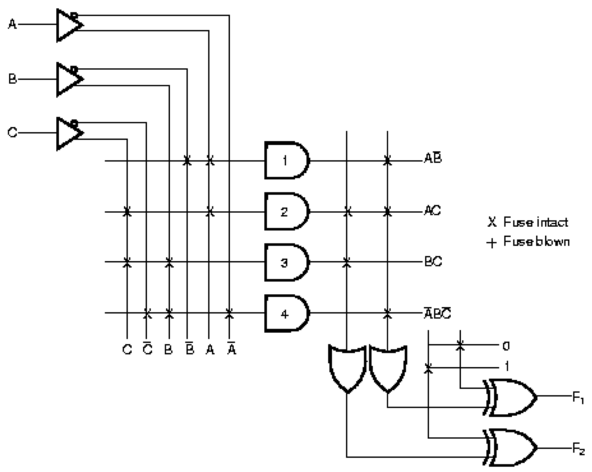


6.8 Programmable Logic Array (PLA)

- PLA is similar to the PROM in concept
- but the PLA doesn't provide full decoding of the variables & doesn't generate all the minterms
- decoder is replaced by an array of AND gates to generate any product term of the input variables
- the product terms are then connected to OR gates to provide the sum of products

- internal logic of a PLA w/ 3 inputs & 2 outputs
  - each input goes through a buffer and an inverter
  - connected through fuses to the inputs of each AND gate
  - output of AND gates are connected by fuses to OR gate
  - output of OR gates goes to an XOR gate, where the other input can be programmed to receive a signal

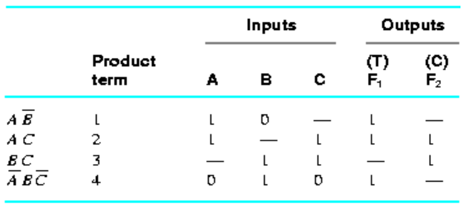
6.8 Programmable Logic Array (PLA)



F_1 = AB' + AC + A'BC'; F_2' = AC + BC

6.8 Programmable Logic Array (PLA)

- the fuse map of a PLA in a tabular form;
  - consists of 3 sections
    - list of the product terms
    - the required paths between inputs & AND gates
    - the path between the AND & OR gates



	Product term	Inputs			Outputs	
		A	B	C	(T) F_1	(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	—

- a careful investigation must be undertaken to reduce the number of distinct product terms (since PLA has a finite number of AND gates)
  - simplify each Boolean function to a minimum No of terms
    - obtain both the true & complement of the function
    - select a comb that gives a minimum No of product terms

6.8 Programmable Logic Array (PLA)

Ex 6.2 Implement the following functions with a PLA

$F_1(A,B,C) = \Sigma\, m(0,1,2,4)$;

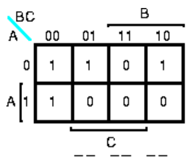
$F_2(A,B,C) = \Sigma\, m(0,5,6,7)$

1) true & complement of the functions are simplified in sum of products

2) select a combination that gives a minimum No of product terms

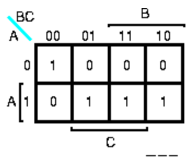
$F_1 = (AB + AC + BC)'$;

$F_2 = AB + AC + A'B'C'$



$F_1 = \overline{A}\overline{B} + \overline{A}\overline{C} + \overline{B}\overline{C}$

$\overline{F_1} = AB + AC + BC$



$F_2 = AB + AC + \overline{A}\overline{B}\overline{C}$

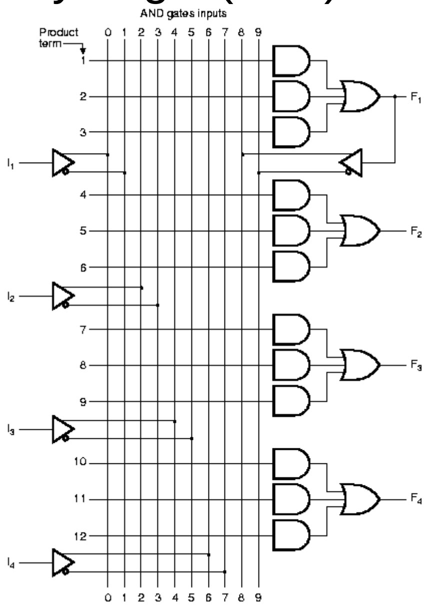
$\overline{F_2} = \overline{A}C + \overline{A}B + A\overline{B}\overline{C}$

PLA programming table

	Product term	Inputs A B C	Outputs (C) F_1	Outputs (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

6.9 Programmable Array Logic (PAL)

- PLD with a fixed OR array & programmable AND array
- easier to program (only AND gates are programmable),
- not flexible as the PLA
- logic configuration of a typical PAL
  - 4 inputs & 4 outputs



6.9 Programmable Array Logic (PAL)

- 4 section in the unit,
  - each composed of a 3-wide AND-OR array
    - 3 programmable AND gates in each section
- output terminals are sometimes bidirectional
  - F/Fs are often included in a PAL device
    - outputs of F/F are fed back through a buffer-inverter gate (sequential circuits !!)
- in designing with a PAL,
  - the Boolean functions must be simplified to fit into each section
  - a product term cannot be shared
  - number of product terms in each section is fixed
    - may be necessary to use 2 sections to implement 1 function

6.9 Programmable Array Logic (PAL)

(Ex) $W(A,B,C,D) = \Sigma\, m(2,12,13)$;
$X(A,B,C,D) = \Sigma\, m(7,8,9,10,11,12,13,14,15)$
$Y(A,B,C,D) = \Sigma\, m(0,2,3,4,5,6,7,8,10,11,15)$
$Z(A,B,C,D) = \Sigma\, m(1,2,8,12,13)$

after simplification
$W= ABC' + A'B'CD'$;
$X = A + BCD$;
$Y = A'B + CD + B'D'$;
$Z = ABC' + A'B'CD' + AC'D' + A'B'C'D$
$= W + AC'D' + A'B'C'D$;

6.9 Programmable Array Logic (PAL)

- PAL programmable table

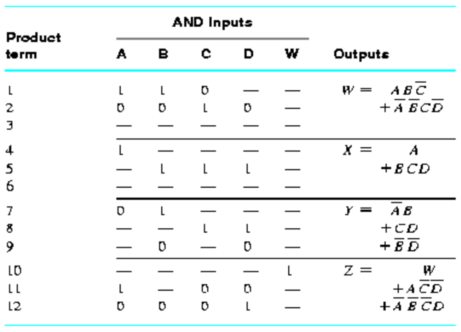
W= ABC' + A'B'CD';

X = A + BCD;

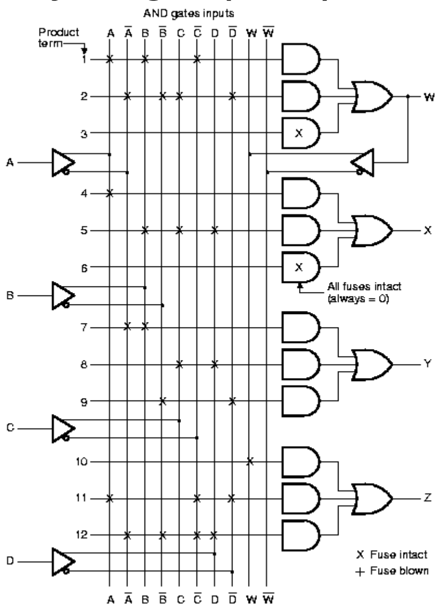
Y = A'B + CD + B'D';

Z = ABC' + A'B'CD' + AC'D' + A'B'C'D

= W + AC'D' + A'B'C'D;



Product term	AND Inputs					Outputs
	A	B	C	D	W	
1	1	1	0	—	—	$W = AB\overline{C}$
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	—	$+\overline{A}\,\overline{B}\,\overline{C}D$


AND gates inputs
Product term
A Ā B B̄ C C̄ D D̄ W W̄
1
2
3
4
5
6
7
8
9
10
11
12
A
B
C
D
W
X
Y
Z
All fuses intact (always = 0)
X Fuse intact
+ Fuse blown