

---

## Module 4

# Combinational Logic Technologies: Supplemental Slides

---

Louis G. Johnson  
School of Electrical and Computer Engineering  
Oklahoma State University

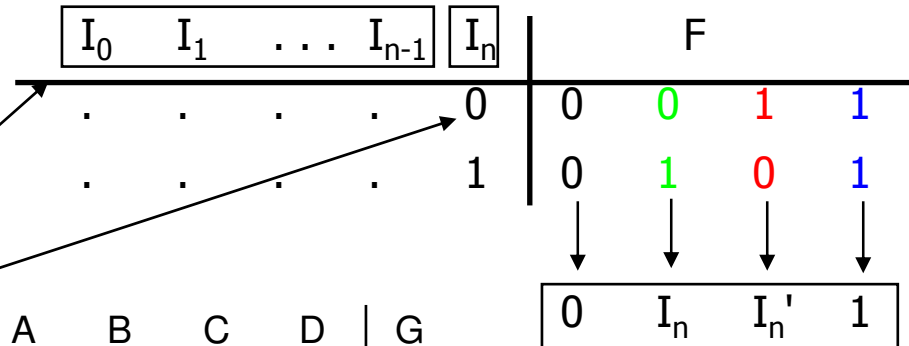
Fall 2007

# Multiplexers as general-purpose logic

- Generalization

n-1 mux control variables

single mux data variable

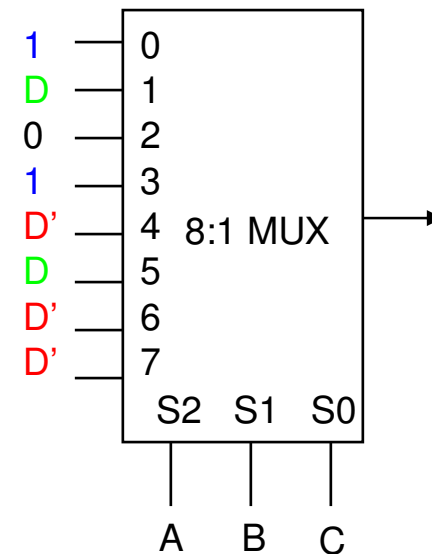


four possible configurations of truth table rows can be expressed as a function of  $I_n$

- Example:  
G(A,B,C,D)  
can be realized by an 8:1 MUX

A	B	C	D	G
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

choose A,B,C as control variables



## Activity

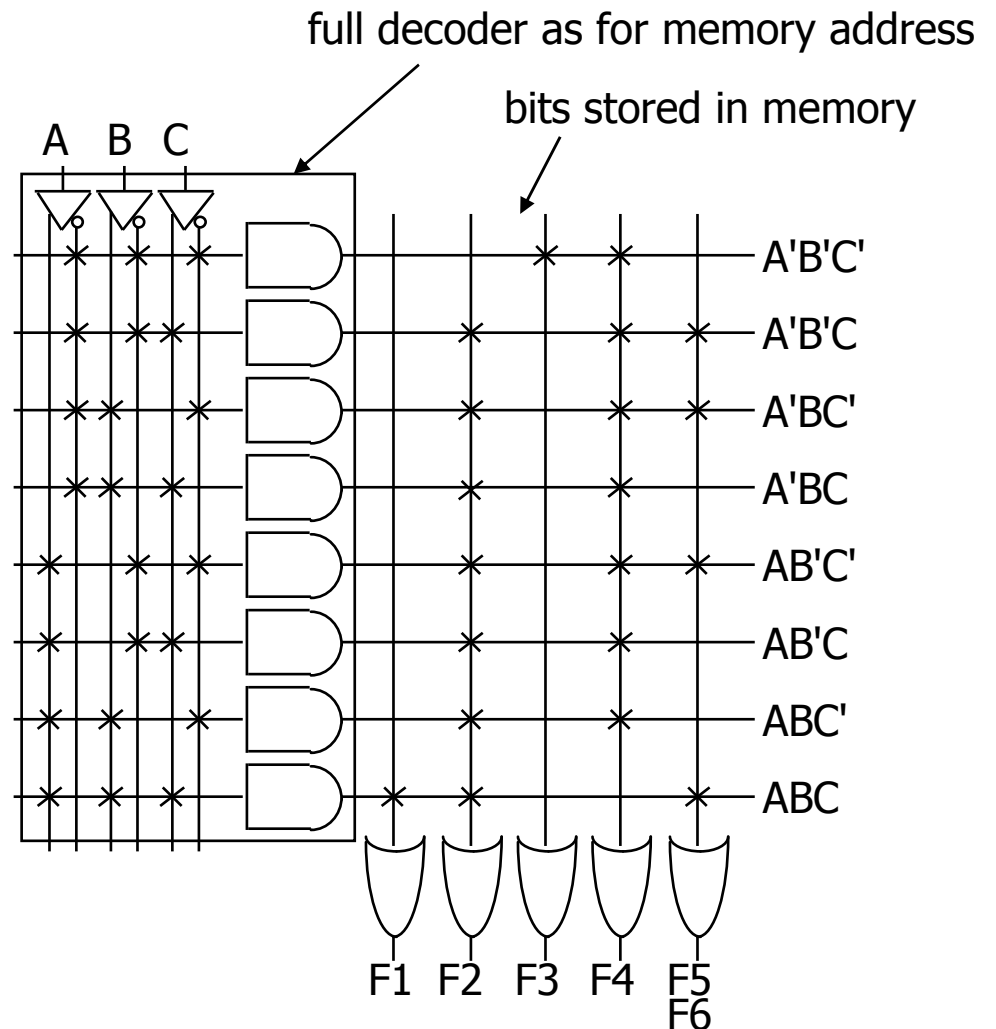
- Realize  $F = B'CD' + ABC'$  with a 4:1 multiplexer and a minimum of other gates:

# Programmable logic array example

- Multiple functions of A, B, C

- $F1 = A B C$
- $F2 = A + B + C$
- $F3 = A' B' C'$
- $F4 = A' + B' + C'$
- $F5 = A \text{ xor } B \text{ xor } C$
- $F6 = A \text{ xnor } B \text{ xnor } C$

A	B	C	F1	F2	F3	F4	F5	F6
0	0	0	0	0	1	1	0	0
0	0	1	0	1	0	1	1	1
0	1	0	0	1	0	1	1	1
0	1	1	0	1	0	1	0	0
1	0	0	0	1	0	1	1	1
1	0	1	0	1	0	1	0	0
1	1	0	0	1	0	1	0	0
1	1	1	1	1	0	0	1	1



# PALs and PLAs: design example

- BCD to Gray code converter

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	1	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	0	0	1	1	0	0	0
1	0	1	—	—	—	—	—
1	1	—	—	—	—	—	—

minimized functions:

$$W = A + BD + BC$$

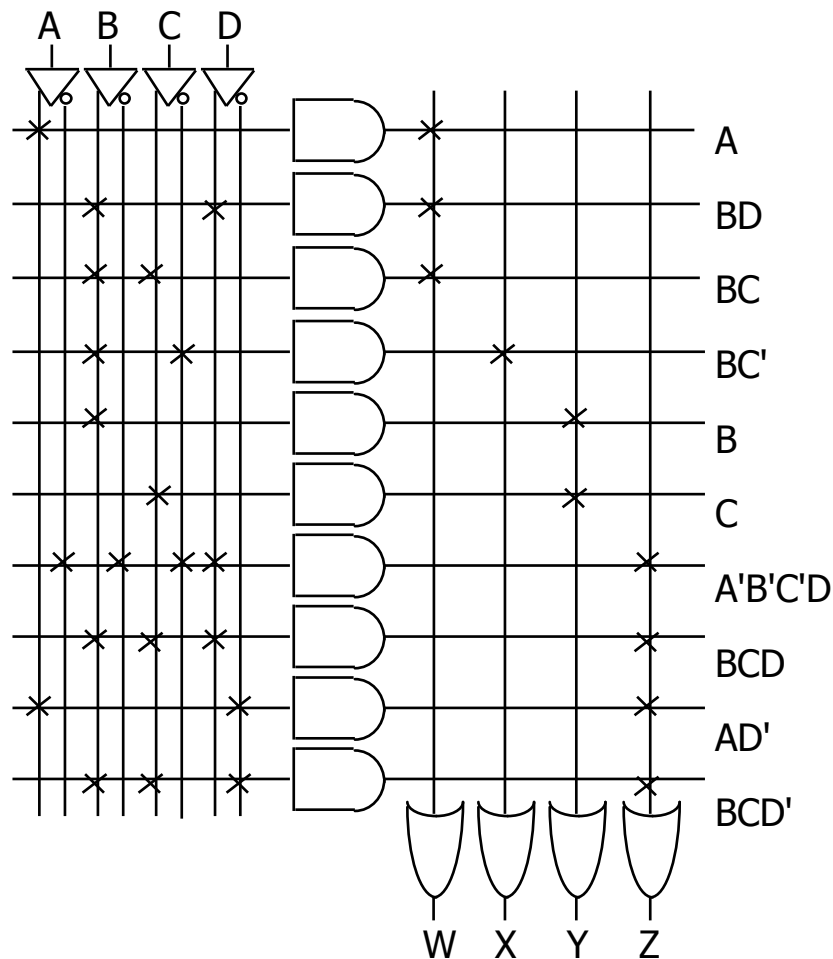
$$X = BC'$$

$$Y = B + C$$

$$Z = A'B'C'D + BCD + AD' + B'CD'$$

# PALs and PLAs: design example (cont'd)

- Code converter: programmed PLA



minimized functions:

$$W = A + BD + BC$$

$$X = B C'$$

$$Y = B + C$$

$$Z = A'B'C'D + BCD + AD' + B'CD'$$

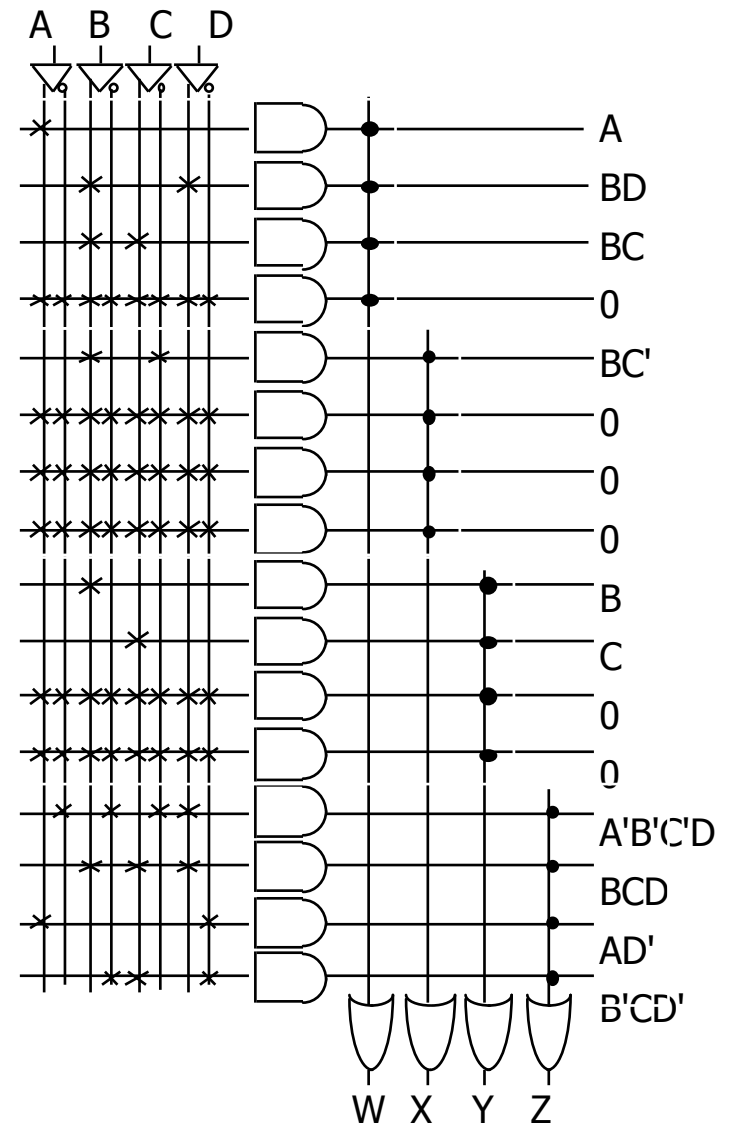
not a particularly good candidate for PAL/PLA implementation since no terms are shared among outputs

however, much more compact and regular implementation when compared with discrete AND and OR gates

# PALs and PLAs: design example (cont'd)

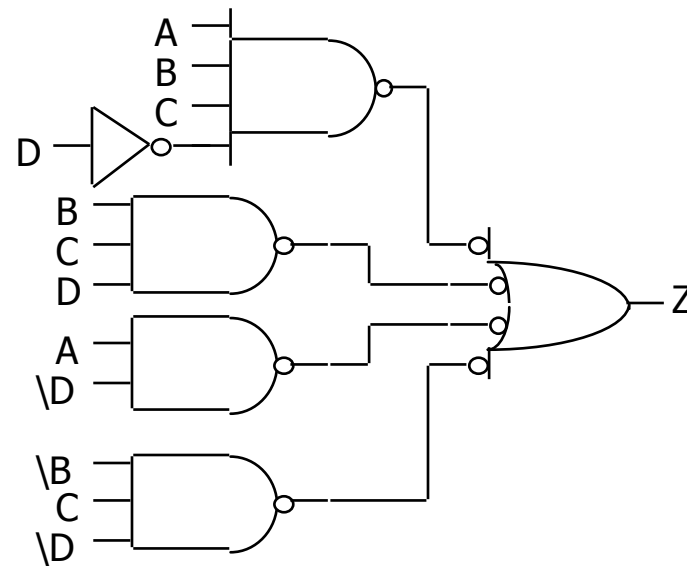
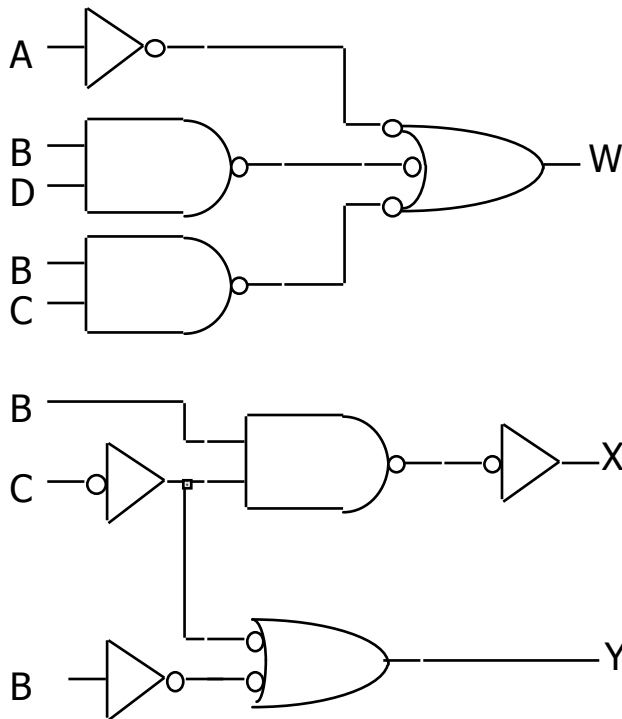
- Code converter: programmed PAL

4 product terms  
per each OR gate



# PALs and PLAs: design example (cont'd)

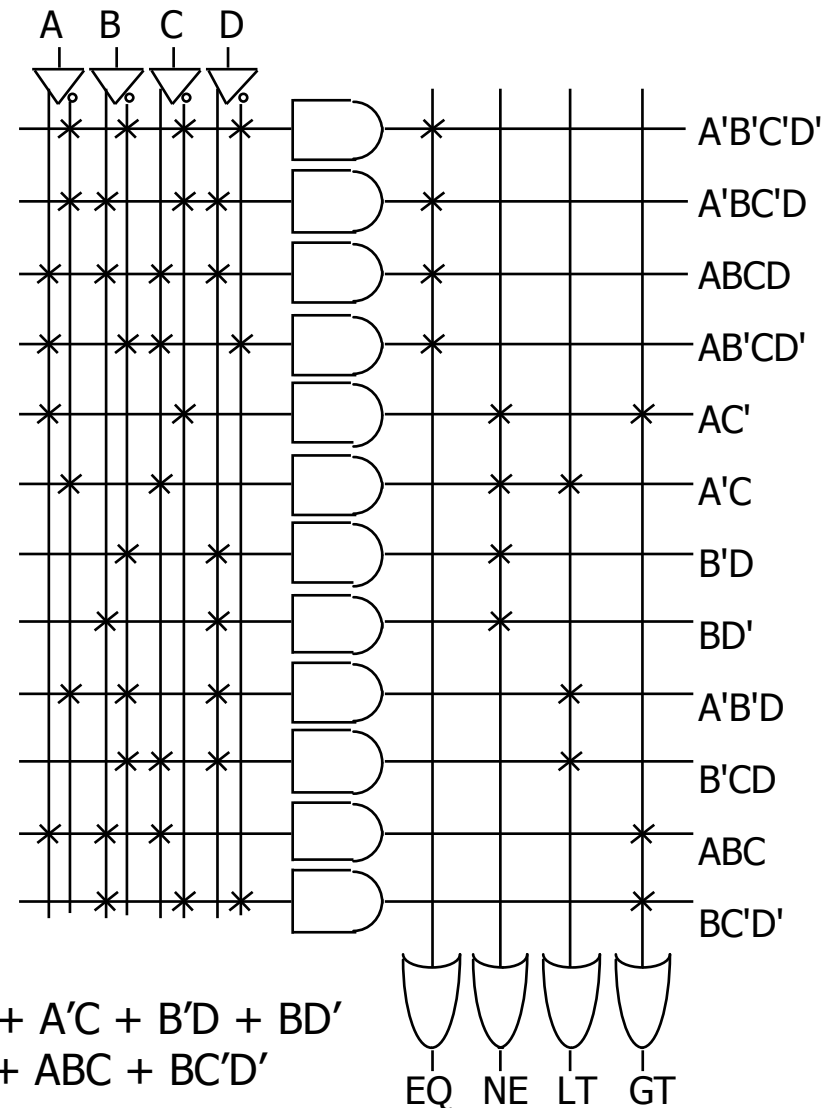
- Code converter: NAND gate implementation
  - ❑ loss of regularity, harder to understand
  - ❑ harder to make changes



# PALs and PLAs: another design example

## ■ Magnitude comparator

A	B	C	D	EQ	NE	LT	GT
0	0	0	0	1	0	0	0
0	0	0	1	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	1	0	0	0
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1
1	0	1	0	1	0	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	1	0	1
1	1	0	1	0	1	0	1
1	1	1	0	0	1	0	1
1	1	1	1	1	0	0	0



minimized functions:

$$EQ = A'B'C'D' + A'BC'D + ABCD + AB'CD'$$

$$LT = A'C + A'B'D + B'CD$$

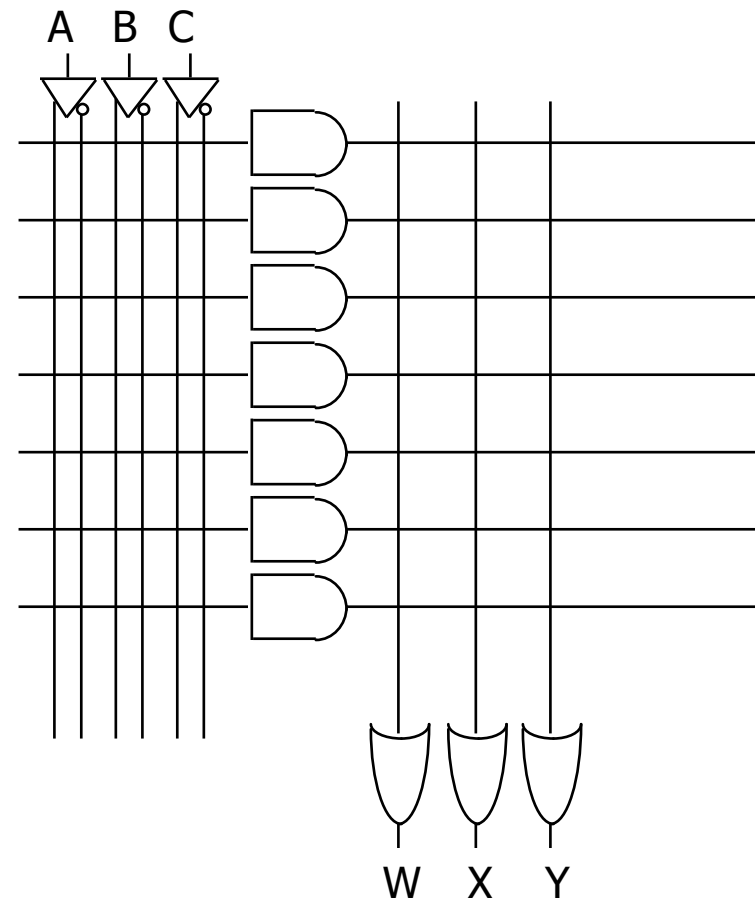
$$NE = AC' + A'C + B'D + BD'$$

$$GT = AC' + ABC + BC'D'$$

# Activity

- Map the following functions to the PLA below:

- $W = AB + A'C' + BC'$
- $X = ABC + AB' + A'B$
- $Y = ABC' + BC + B'C'$



---

## Activity (cont'd)