Reminder

Midterm 1: Thursday, Oct. 5th

- In class: 1 hour and 15 minutes
- Chap 1 – 2.6
- Closed book, closed notes
- No calculator
- Boolean Theorems & Axioms document will be attached as last page of the exam for your convenience

ELSEVIER

Multiplying Out: SOP Form

An expression is in simplified **sum-of-products (SOP)** form when all products contain literals only.

- SOP form: Y = AB + BC' + DE
- **NOT** SOP form: Y = DF + E(A'+B)
- SOP form: Z = A + BC + DE'F

ELSEVIER

Multiplying Out: SOP Form

Example:

$Y = (A + C + D + E)(A + B)$

Apply T8' first when possible: W+XZ = (W+X)(W+Z)

ELSEVIER

Multiplying Out: SOP Form

Example:

Y = (A + C + D + E)(A + B)

Apply T8' first when possible: W+XZ = (W+X)(W+Z)

Make: X = (C+D+E), Z = B and rewrite equation

Y	= (A+X)(A+Z)	substitution (X=(C+D+E), Z=B)
	= A + XZ	T8': Distributivity
	= A + (C+D+E)B	substitution
	= A + BC + BD + BE	T8: Distributivity

or

Y	= AA+AB+AC+BC+AD+BD+AE+BE	T8: Distributivity
	= **A**+AB+AC+AD+AE+BC+BD+BE	T3: Idempotency
	= **A** + BC + BD + BE	T9': Covering

ELSEVIER

Canonical SOP & POS Form

- SOP – sum-of-products

O	C	E	minterm
0	0	0	$\overline{O}\ \overline{C}$
0	1	0	$\overline{O}\ C$
1	0	1	$O\ \overline{C}$
1	1	0	$O\ C$

$$E = O\overline{C}$$
$$= \Sigma(m_2)$$

- POS – product-of-sums

O	C	E	maxterm
0	0	0	$O + C$
0	1	0	$O + \overline{C}$
1	0	1	$\overline{O} + C$
1	1	0	$\overline{O} + \overline{C}$

$$E = (O + C)(O + \overline{C})(\overline{O} + \overline{C})$$
$$= \Pi(M0, M1, M3)$$

ELSEVIER

Factoring: POS Form

An expression is in simplified **product-of-sums (POS)** form when all sums contain literals only.

- POS form: Y = (A+B)(C+D)(E'+F)
- **NOT** POS form: Y = (D+E)(F'+GH)
- POS form: Z = A(B+C)(D+E')

ELSEVIER

Factoring: POS Form

Example 1:

Y = (A + B'CDE)

Apply T8' first when possible: W+XZ = (W+X)(W+Z)

ELSEVIER

Factoring: POS Form

Example 1:

Y = (A + B'CDE)

Apply T8' first when possible: W+XZ = (W+X)(W+Z)

Make: X = B'C, Z = DE and rewrite equation

Y	= (A+XZ)	substitution (X=B'C, Z=DE)
	= (A+B'C)(A+DE)	T8': Distributivity
	= (A+B')(A+C)(A+D)(A+E)	T8': Distributivity

ELSEVIER

Factoring: POS Form

Example 2:

Y = AB + C'DE + F

Apply T8' first when possible: W+XZ = (W+X)(W+Z)

ELSEVIER

Factoring: POS Form

Example 2:

Y = AB + C'DE + F

Apply T8' first when possible: W+XZ = (W+X)(W+Z)

Make: W = AB, X = C', Z = DE and rewrite equation

Y	= (W+XZ) + F	substitution W = AB, X = C', Z = DE
	= (W+X)(W+Z) + F	T8': Distributivity
	= (AB+C')(AB+DE)+F	substitution
	= (A+C')(B+C')(AB+D)(AB+E)+F	T8': Distributivity
	= (A+C')(B+C')(A+D)(B+D)(A+E)(B+E)+F	T8': Distributivity
	= (A+C'+F)(B+C'+F)(A+D+F)(B+D+F)(A+E+F)(B+E+F)	T8': Distributivity

ELSEVIER

Boolean Thms of Several Vars: Duals

#	Theorem	Dual	Name
T6	B•C = C•B	B+C = C+B	Commutativity
T7	(B•C) • D = B • (C•D)	(B + C) + D = B + (C + D)	Associativity
T8	B • (C + D) = (B•C) + (B•D)	B + (C•D) = (B+C) (B+D)	Distributivity
T9	B • (B+C) = B	B + (B•C) = B	Covering
T10	(B•C) + (B•$\overline{C}$) = B	(B+C) • (B+$\overline{C}$) = B	Combining
T11	(B•C) + (B•$\overline{D}$) + (C•D) = (B•C) + (B•$\overline{D}$)	(B+C) • (B+$\overline{D}$) • (C+D) = (B+C) • (B+$\overline{D}$)	Consensus

Axioms and theorems are useful for *simplifying* equations.

ELSEVIER

Simplification methods

- **Distributivity (T8, T8')**

  $B(C+D) = BC + BD$

  $B + CD = (B + C)(B+D)$

- **Covering (T9')**

  $A + AP = A$

- **Combining (T10)**

  $PA + P\overline{A} = P$

- **Expansion**

  $P = PA + P\overline{A}$

  $A = A + AP$

- **Duplication**

  $A = A + A$

- **A combination of Combining/Covering**

  $PA + \overline{A} = P + \overline{A}$

ELSEVIER

DeMorgan's Theorem

Number	Theorem	Name
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots} = \overline{B_0} + \overline{B_1} + \overline{B_2} \ldots$	DeMorgan's Theorem

ELSEVIER

COMBINATIONAL LOGIC DESIGN

DeMorgan's Theorem

Number	Theorem	Name
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots} = \overline{B_0} + \overline{B_1} + \overline{B_2} \ldots$	DeMorgan's Theorem

The **complement of the product**
is the
sum of the complements

ELSEVIER

DeMorgan's Theorem: Dual

#	Theorem	Dual	Name
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots} = \overline{B_0} + \overline{B_1} + \overline{B_2} \ldots$	$\overline{B_0 + B_1 + B_2 \ldots} = \overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2} \ldots$	DeMorgan's Theorem

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

DeMorgan's Theorem: Dual

#	Theorem	Dual	Name
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots} = \overline{B_0} + \overline{B_1} + \overline{B_2} \ldots$	$\overline{B_0 + B_1 + B_2 \ldots} = \overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2} \ldots$	DeMorgan's Theorem

The complement of the product
is the
sum of the complements

ELSEVIER

DeMorgan's Theorem: Dual

#	Theorem	Dual	Name
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots} = \overline{B_0} + \overline{B_1} + \overline{B_2} \ldots$	$\overline{B_0 + B_1 + B_2 \ldots} = \overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2} \ldots$	DeMorgan's Theorem

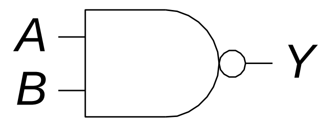
The complement of the product
is the
sum of the complements.

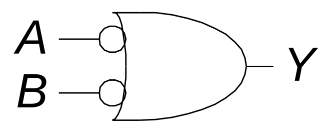
Dual: The **complement of the sum**
is the
product of the complements.

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

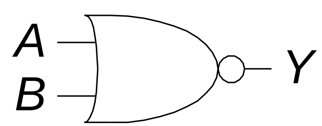
DeMorgan's Theorem

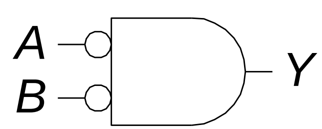
- $Y = \overline{AB} = \overline{A} + \overline{B}$





- $Y = \overline{A + B} = \overline{A} \cdot \overline{B}$





DeMorgan's Theorem Example 1

$$Y = \overline{(A+\overline{BD})\overline{C}}$$

DeMorgan's Theorem Example 1

$$Y = \overline{(A+\overline{BD})\overline{C}}$$

$$= \overline{(A+\overline{BD})} + \overline{\overline{C}}$$

$$= (\overline{A}\bullet(\overline{\overline{BD}})) + C$$

$$= (\overline{A}\bullet(BD)) + C$$

$$= \overline{A}BD + C$$

ELSEVIER

DeMorgan's Theorem Example 2

$$Y = \overline{(\overline{AC}E+\overline{D}) + B}$$

DeMorgan's Theorem Example 2

$$Y = \overline{(\overline{ACE}+\overline{D}) + B}$$

$$= \overline{(\overline{ACE}+\overline{D})} \bullet \overline{B}$$

$$= (\overline{\overline{ACE}} \bullet \overline{\overline{D}}) \bullet \overline{B}$$

$$= ((\overline{\overline{AC}}+\overline{E}) \bullet D) \bullet \overline{B}$$

$$= ((AC+\overline{E}) \bullet D) \bullet \overline{B}$$

$$= (ACD + D\overline{E}) \bullet \overline{B}$$

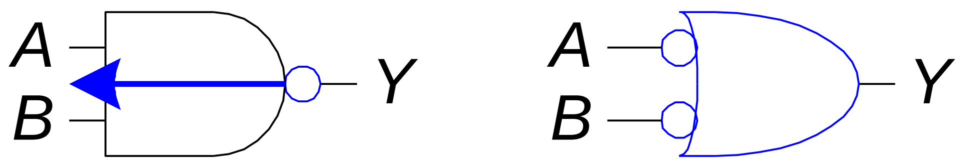
$$= A\overline{B}CD + \overline{B}D\overline{E}$$

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

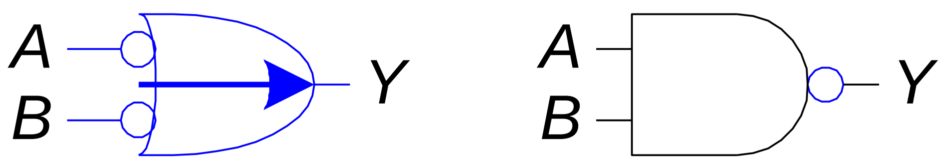
ELSEVIER

Bubble Pushing

- **Backward:**
  - Body changes
  - Adds bubbles to inputs

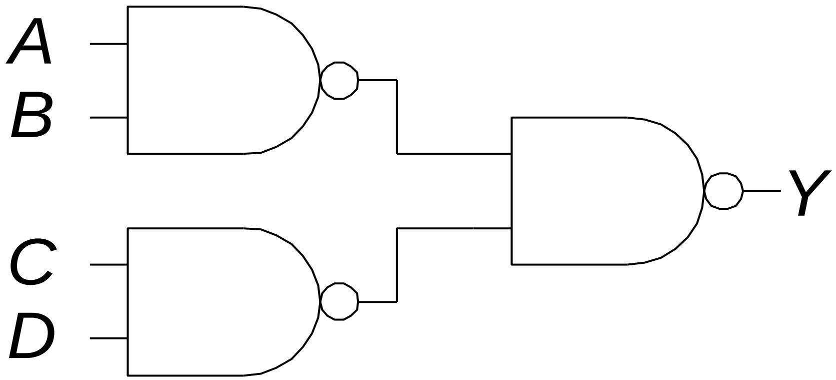


- **Forward:**
  - Body changes
  - Adds bubble to output



Bubble Pushing

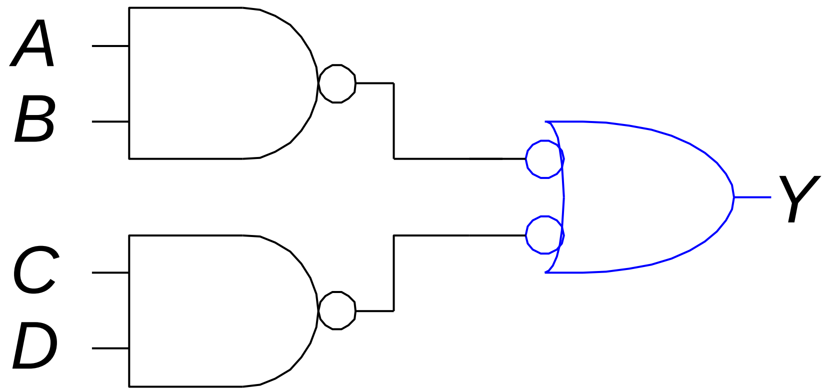
- What is the Boolean expression for this circuit?



ELSEVIER

Bubble Pushing

- What is the Boolean expression for this circuit?

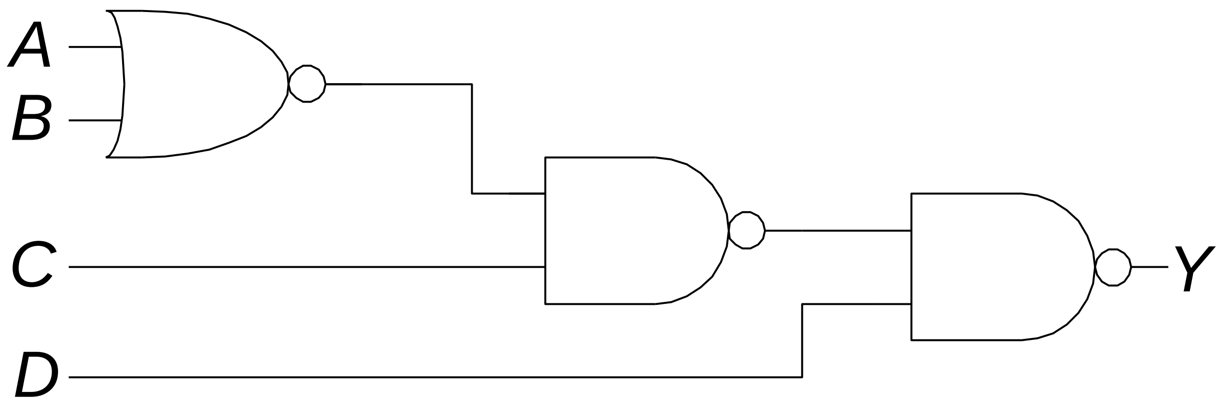


$$Y = AB + CD$$

ELSEVIER

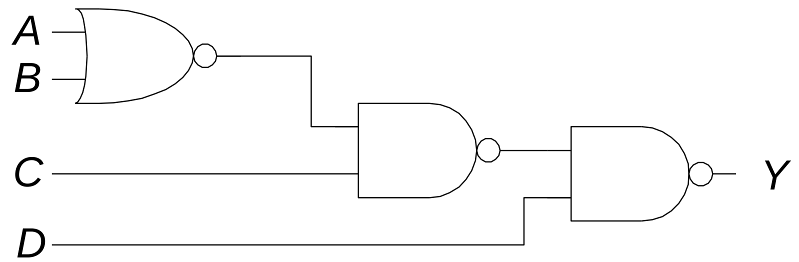
Bubble Pushing Rules

- Begin at output, then work toward inputs
- Push bubbles on final output back
- Draw gates in a form so bubbles cancel



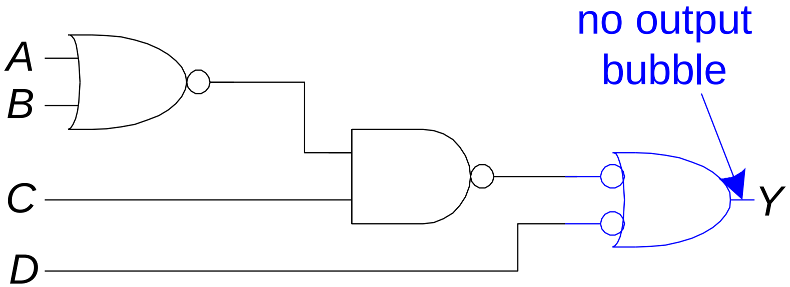
ELSEVIER

Bubble Pushing Example

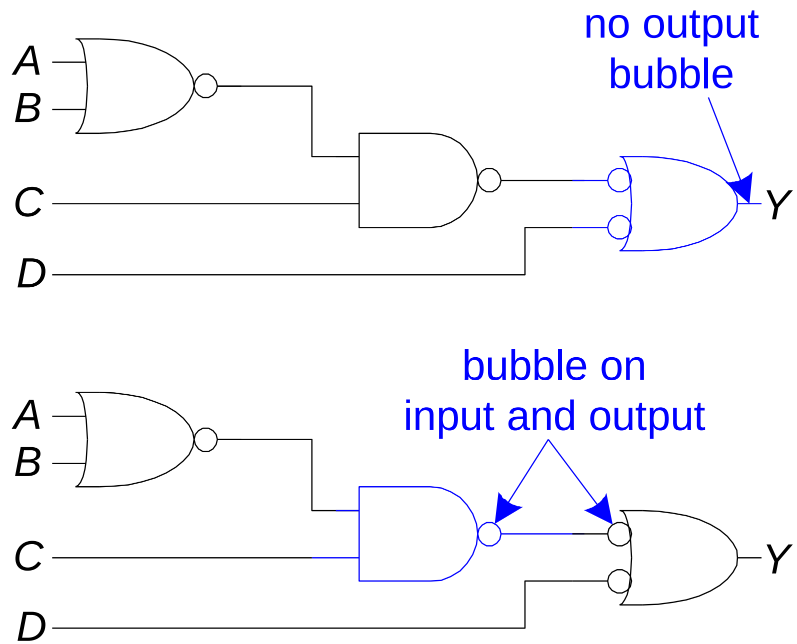


ELSEVIER

Bubble Pushing Example

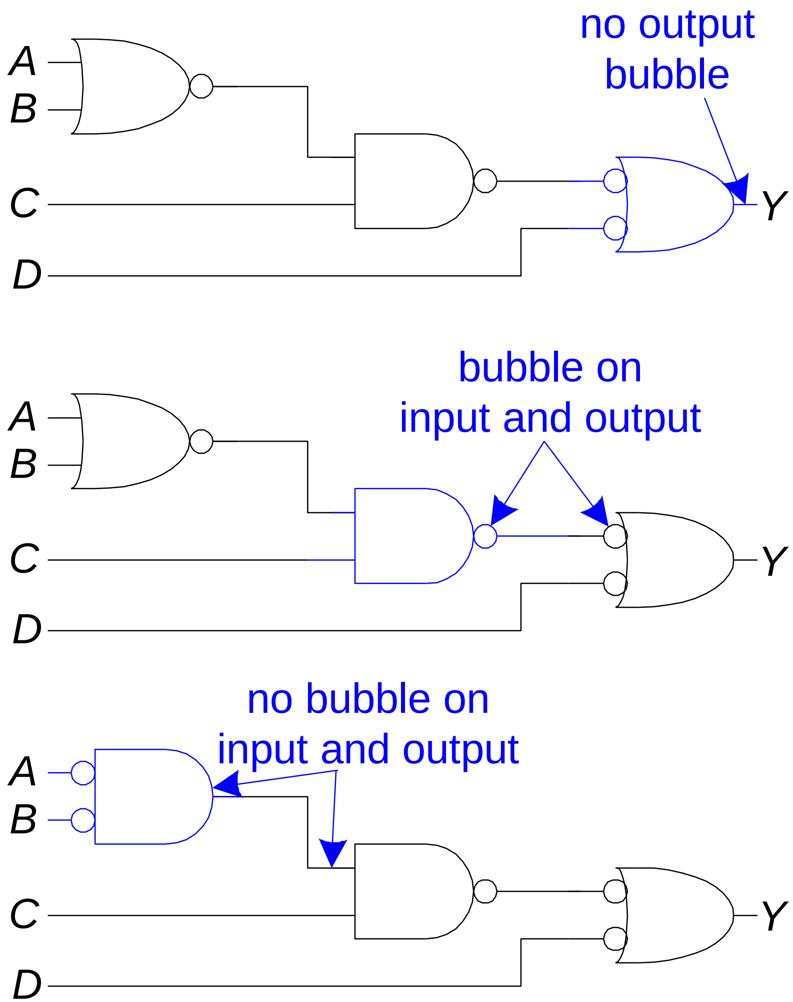


Bubble Pushing Example



ELSEVIER

Bubble Pushing Example



$$Y = \overline{A}\,\overline{B}C + \overline{D}$$

Canonical SOP & POS Form Revisited

- SOP – sum-of-products

O	C	E	minterm
0	0	0	$\overline{O}\ \overline{C}$
0	1	0	$\overline{O}\ C$
1	0	1	$O\ \overline{C}$
1	1	0	$O\ C$

How do we implement this logic function with gates?

$$E = O\overline{C}$$
$$= \Sigma(m_2)$$

- POS – product-of-sums

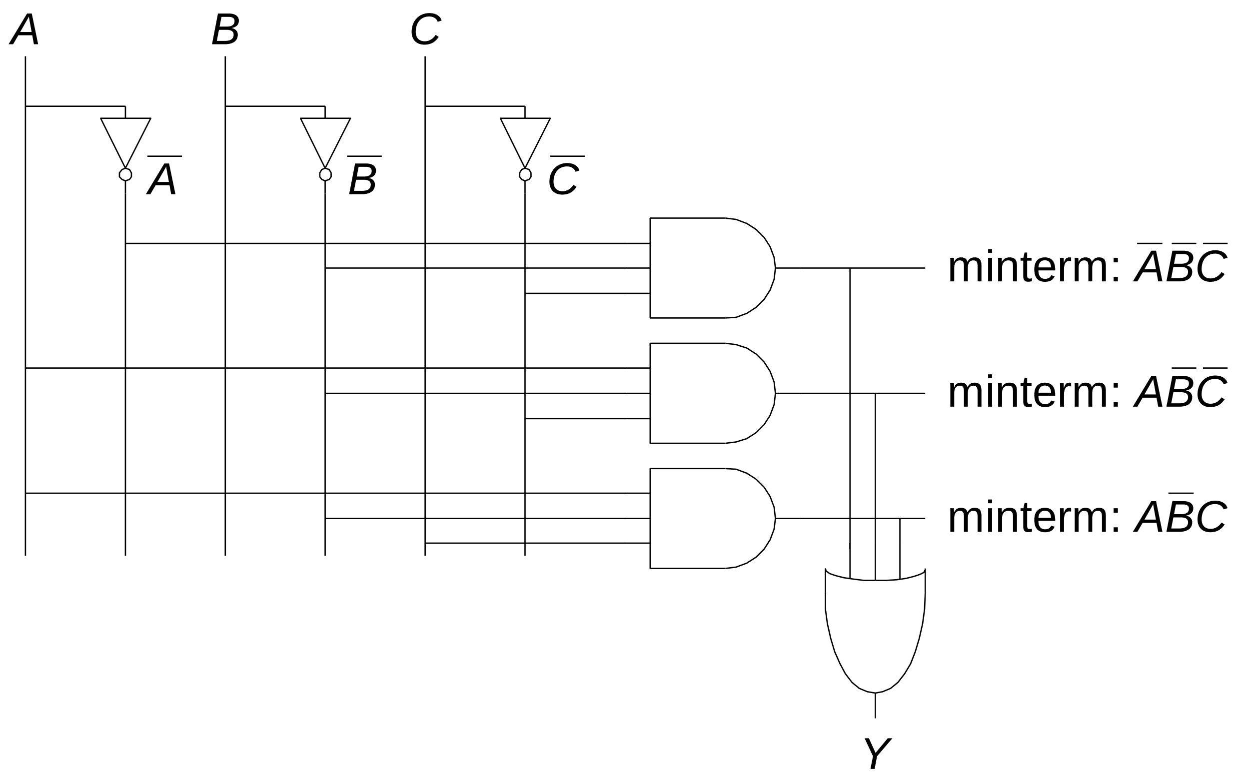
O	C	E	maxterm
0	0	0	$O + C$
0	1	0	$O + \overline{C}$
1	0	1	$\overline{O} + C$
1	1	0	$\overline{O} + \overline{C}$

$$E = (O + C)(O + \overline{C})(\overline{O} + \overline{C})$$
$$= \Pi(M0, M1, M3)$$

ELSEVIER

From Logic to Gates

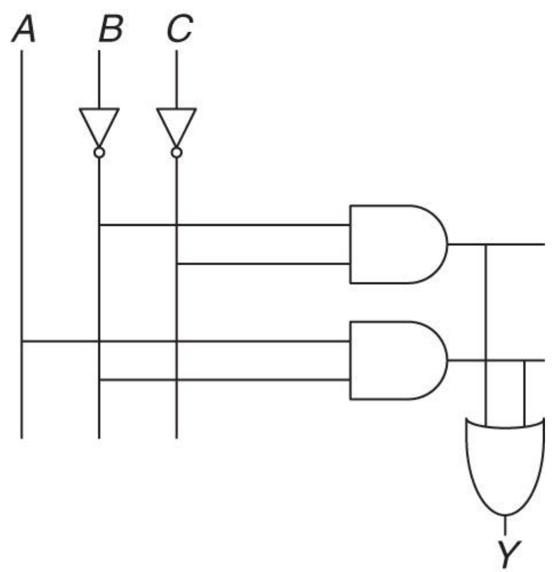
- Two-level logic: ANDs followed by ORs
- Example: $Y = \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$



ELSEVIER

Circuit Schematics Rules

- Inputs on the left (or top)
- Outputs on right (or bottom)
- Gates flow from left to right
- Straight wires are best

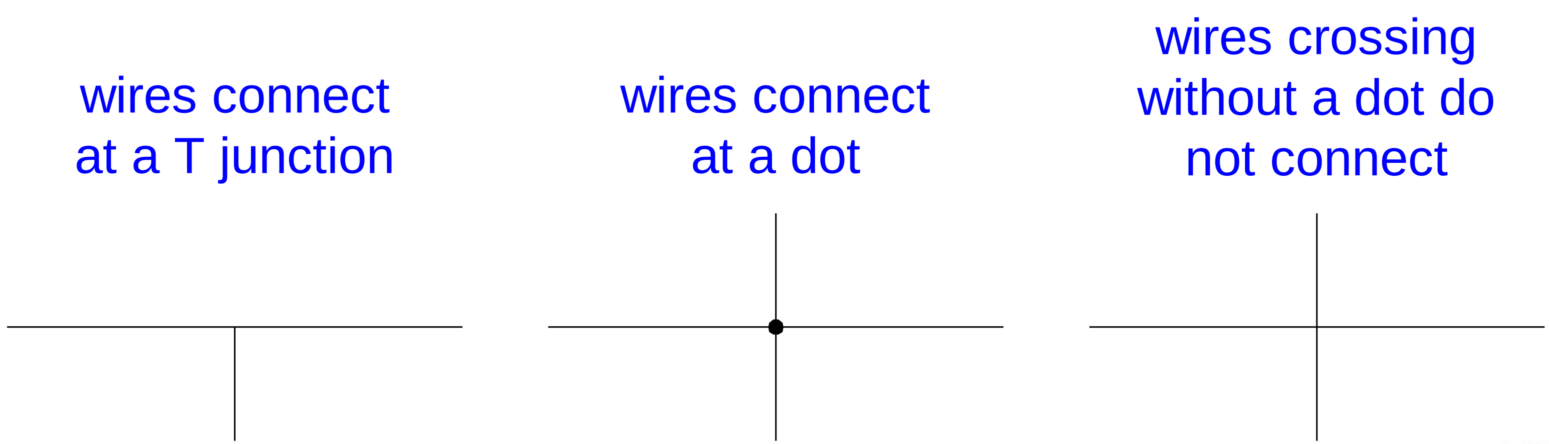


$$Y = \bar{B}\bar{C} + A\bar{B}$$

ELSEVIER

Circuit Schematic Rules (cont.)

- Wires always connect at a T junction
- A dot where wires cross indicates a connection between the wires
- Wires crossing *without* a dot make no connection

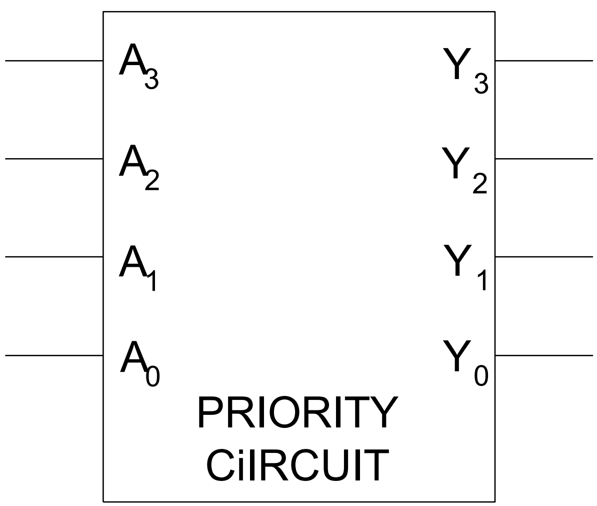


ELSEVIER

Multiple-Output Circuits

- **Example: Priority Circuit**

  Output asserted corresponding to most significant TRUE input



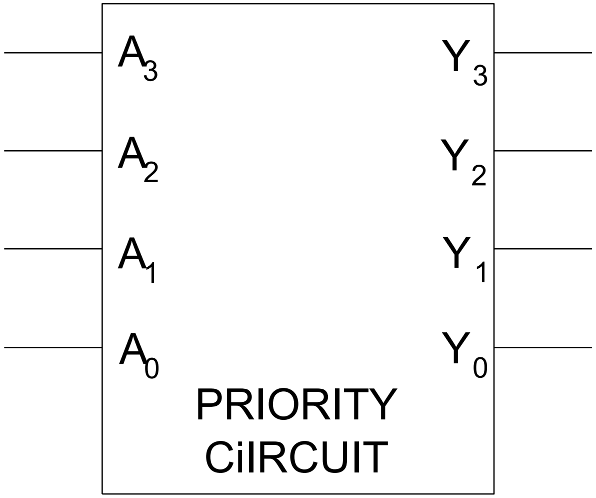
A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

ELSEVIER

Multiple-Output Circuits

- **Example: Priority Circuit**

  Output asserted corresponding to most significant TRUE input

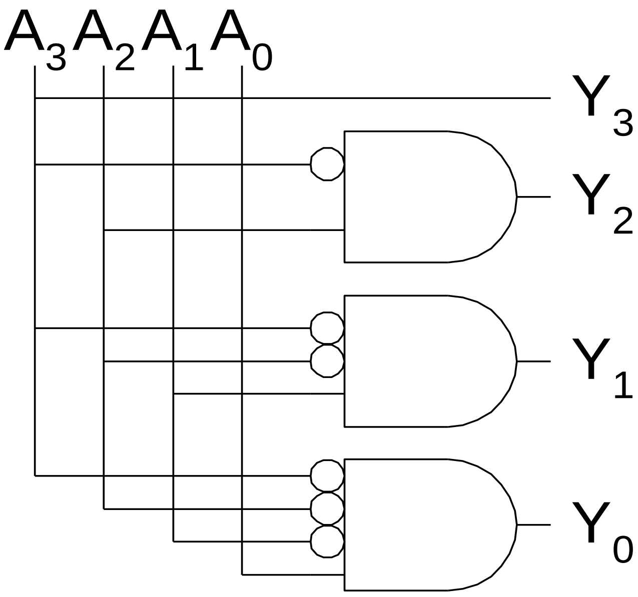


A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

ELSEVIER

Priority Circuit Hardware

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0



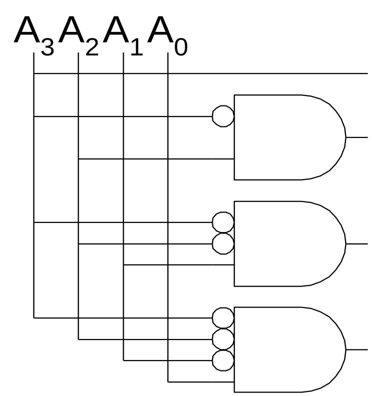
Don’t Cares

- Simplify truth table by ignoring entries

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	X	0	0	1	0
0	1	X	X	0	1	0	0
1	X	X	X	1	0	0	0

Much easier to read off Boolean equations



$Y_3 = A_3$

$Y_2 = \overline{A_3}A_2$

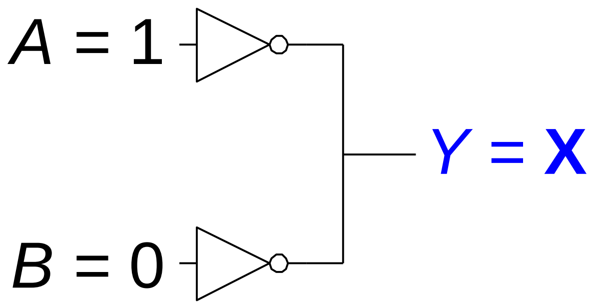
$Y_1 = \overline{A_3}\,\overline{A_2}\,A_1$

$Y_0 = \overline{A_3}\,\overline{A_2}\,\overline{A_1}\,A_0$

ELSEVIER

Contention: X

- Contention: circuit tries to drive output to 1 **and** 0
  - Actual value somewhere in between
  - Could be 0, 1, or in forbidden zone
  - Might change with voltage, temperature, time, noise
  - Often causes excessive power dissipation



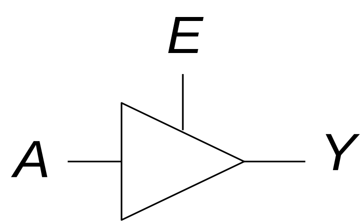
- **Warnings:**
  - Contention usually indicates a **bug**.
  - **X is used for "don't care" and contention** - look at the context to tell them apart

ELSEVIER

Floating: Z

- Floating, high impedance, open, high Z
- Floating output might be 0, 1, or somewhere in between
  - A voltmeter won't indicate whether a node is floating

Tristate Buffer



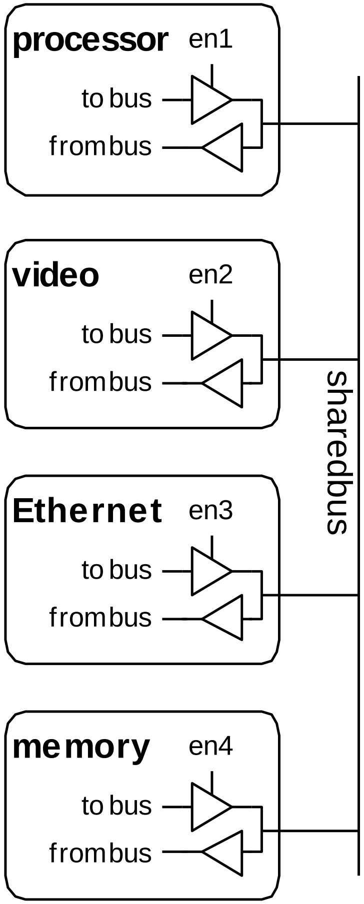
E	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

Note: tristate buffer has an enable bit (E) to turn on the gate

ELSEVIER

Tristate Busses

- Floating nodes are used in tristate busses
  - Many different drivers
  - Exactly one is active at once



ELSEVIER