CSCI 210: Computer Architecture
Lecture 15: Boolean Algebra

Stephen Checkoway
Slides from Cynthia Taylor

CS History: Augustus de Morgan

- British, born in 1871
- Published De Morgan’s Laws in 1947
- Introduced the term induction
- Didn’t receive an MA from Cambridge because it required passing a theological test and he was an atheist
- Ada Lovelace was his student
- Dedicated to making scientific knowledge available to the public – wrote numerous articles about many topics

Boolean Algebra

- Branch of algebra in which all variables are 1 or 0 (equivalently true or false)
- Introduced by George Boole in 1847
- Multiple notations
  - $x \land y$ $\quad$ $x \lor y$
  - xy $\quad$ $x + y$

Boolean laws

• Commutativity	$x + y = y + x$	$xy = yx$
• Associativity	$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$
• Distributivity	$x + yz = (x + y)(x + z)$	$x(y + z) = xy + xz$
• Idempotence	$x + x = x$	$xx = x$

Which Identity Laws Are True?

A. $x + 0 = x$, $x0 = x$

B. $x + 0 = x$, $x1 = x$

C. $x + 1 = x$, $x0 = x$

D. $x + 1 = x$, $x1 = x$

E. None of the above

Which Complementation Laws Are True?

A. $\bar{x} + x = 0, \quad \bar{x}x = 0$

B. $\bar{x} + x = 0, \quad \bar{x}x = 1$

C. $\bar{x} + x = 1, \quad \bar{x}x = 0$

D. $\bar{x} + x = 1, \quad \bar{x}x = 1$

E. None of the above

Which Annihilator Laws Are True?

A. $x + 0 = 0$, $x0 = 0$

B. $x + 1 = 1$, $x0 = 0$

C. $x + 0 = 0$, $x1 = 1$

D. $x + 1 = 1$, $x1 = 1$

E. None of the above

Simplifying Expressions

$$F = XYZ + XY\overline{Z} + \overline{X}Z$$

A. $F = XY + \overline{X}Z$
B. $F = X(YZ + Y\overline{Z} + \overline{Z})$
C. $F = XY(Z + \overline{Z}) + \overline{X}Z$
D. This cannot be simplified further

- Identity law: $A + 0 = A$ and $A \cdot 1 = A$
- Zero and One laws: $A + 1 = 1$ and $A \cdot 0 = 0$
- Inverse laws: $A + \overline{A} = 1$ and $A \cdot \overline{A} = 0$
- Commutative laws: $A + B = B + A$ and $A \cdot B = B \cdot A$
- Associative laws: $A + (B + C) = (A + B) + C$ and $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- Distributive laws: $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ and $A + (B \cdot C) = (A + B) \cdot (A + C)$

Simplifying Expressions

$$F = XYZ + XY\overline{Z} + \overline{X}Z$$

- Identity law: $A + 0 = A$ and $A \cdot 1 = A$
- Zero and One laws: $A + 1 = 1$ and $A \cdot 0 = 0$
- Inverse laws: $A + \overline{A} = 1$ and $A \cdot \overline{A} = 0$
- Commutative laws: $A + B = B + A$ and $A \cdot B = B \cdot A$
- Associative laws: $A + (B + C) = (A + B) + C$ and $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- Distributive laws: $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ and $A + (B \cdot C) = (A + B) \cdot (A + C)$

De Morgan's Law

- De Morgan's Law
  - Use to obtain the complement of an expression

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

$$\overline{xy} = \overline{x} + \overline{y}$$

What is $\overline{AB + A\overline{C}}$?

A. $\overline{A}\,\overline{B} + \overline{A}\,C$

B. $(\overline{A}\,\overline{B})(\overline{A}\,C)$

C. $(A + B)(A + \overline{C})$

D. $(\overline{A} + \overline{B})(\overline{A} + C)$

E. None of the above

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

$$\overline{xy} = \overline{x} + \overline{y}$$

Questions on Boolean Algebra?

Sum of Products form of Boolean function f

- Developed from the truth table for $f(x_1, \ldots, x_n)$
- Find all the rows of the truth table in which f = 1
- By definition, $f(x_1, \ldots, x_n) = 1$ if and only if the input $x_1, \ldots, x_n$ match one of these rows

- We can write f as an OR (sum) of expressions checking if the input matches one of the rows:
  - f = (input matches row 1) OR (input matches row 4) OR …

Sum of Products

- Developed from the truth table
  - Each product term contains each input exactly once, complemented or not.
  - Need to OR together set of AND terms to satisfy table
  - One product for each 1 in F column

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

What is the Sum of Products of F?

A. $\overline{A} + B\overline{C}$

B. $A\overline{B}\,\overline{C} + A\overline{B}C + \overline{A}\,\overline{B}\,\overline{C}$

C. $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C}$

D. $ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$

E. None of the above

A	**B**	**C**	**F**
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Product of Sums

- Express the same function as the AND of ORs
- Write out the sum of products for $\bar{F}$ and then take the complement using DeMorgan's law

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

Product of Sums

- Simplified: Select the rows where F is 0 and take the complements of the inputs to form the ORs

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

What is the Product of Sums of F?

A. $F = (A + \overline{B} + C)(A + \overline{B} + \overline{C})(A + B + C)$

B. $F = (\overline{A} + B + C)(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + \overline{C})$

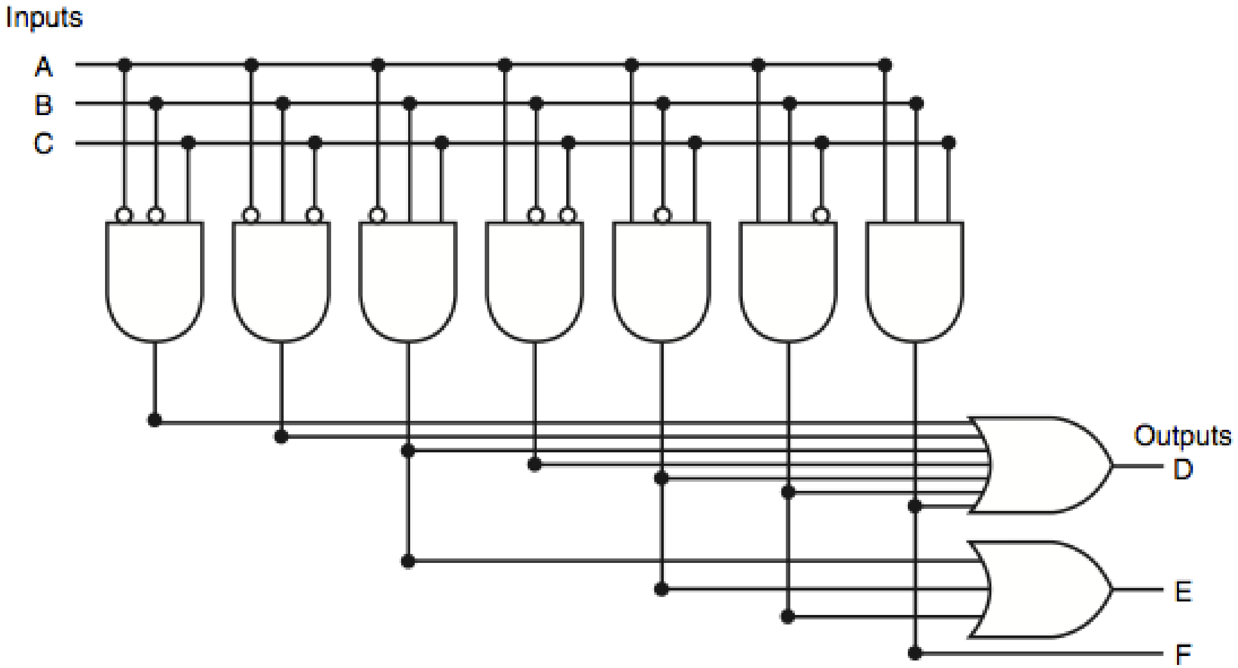
C. $F = (A + \overline{B} + \overline{C})(A + \overline{B} + C)(A + B + C)$

D. $F = (A + B + C)(A + B + \overline{C})(A + \overline{B} + C)(A + \overline{B} + \overline{C})(A + B + \overline{C})$

A	**B**	**C**	**F**
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Programmable Logic Array

- Simple way to create a logical circuit from a truth table, using sum of products
- Set of inputs and inverted inputs
- Array of AND gates
  - Form set of product terms
- Array of OR gates
  - Logical sum of product terms



Uses

- Either programmed during manufacture, or can be reprogrammed

- Used in CPUs, microprocessors

Creating a PLA

- Prepare the truth table
- Write the Boolean expression in sum of products form.
- Decide the input connection of the AND matrix for generating the required product term.
- Then decide the input connections of OR matrix to generate the sum terms.
- Program the PLA.

Size

- Only truth table entries that have a True (1) output are represented
- Each different product term will have only one entry in the PLA, even if the product term is used in multiple outputs

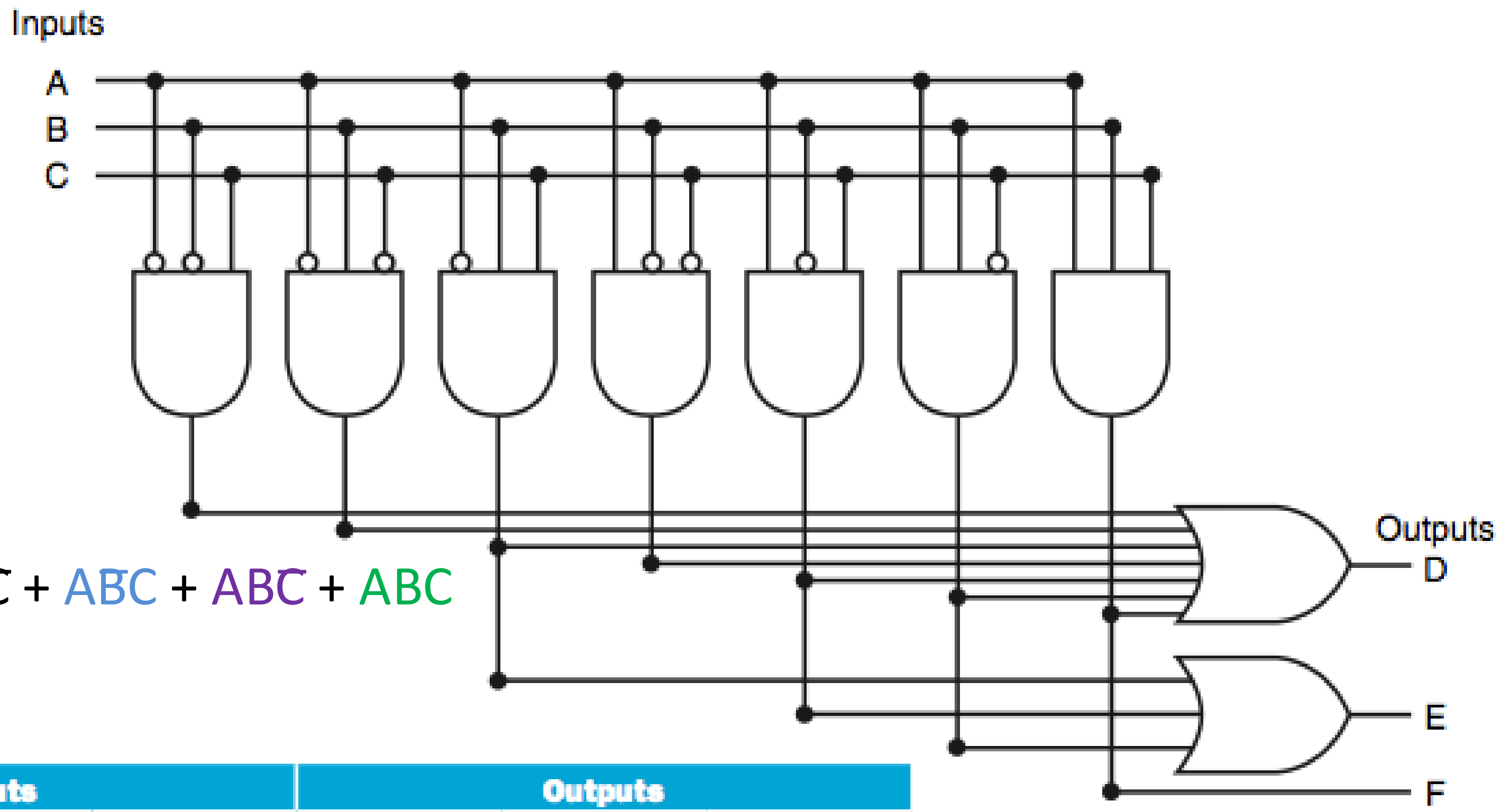
Multiple outputs

Inputs			Outputs		
A	*B*	*C*	*D*	*E*	*F*
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

Output functions: D(A, B, C), E(A, B, C), F(A, B, C)

Inputs			Outputs		
A	*B*	*C*	*D*	*E*	*F*
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

	Sum of Products for output D
A	$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$
B	$\bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$
C	$(A+B+\bar{C})(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$
D	$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$



$D = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

$E = \bar{A}BC + A\bar{B}C + AB\bar{C}$

$F = ABC$

Inputs			Outputs		
A	***B***	***C***	***D***	***E***	***F***
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

Field Programmable PLAs

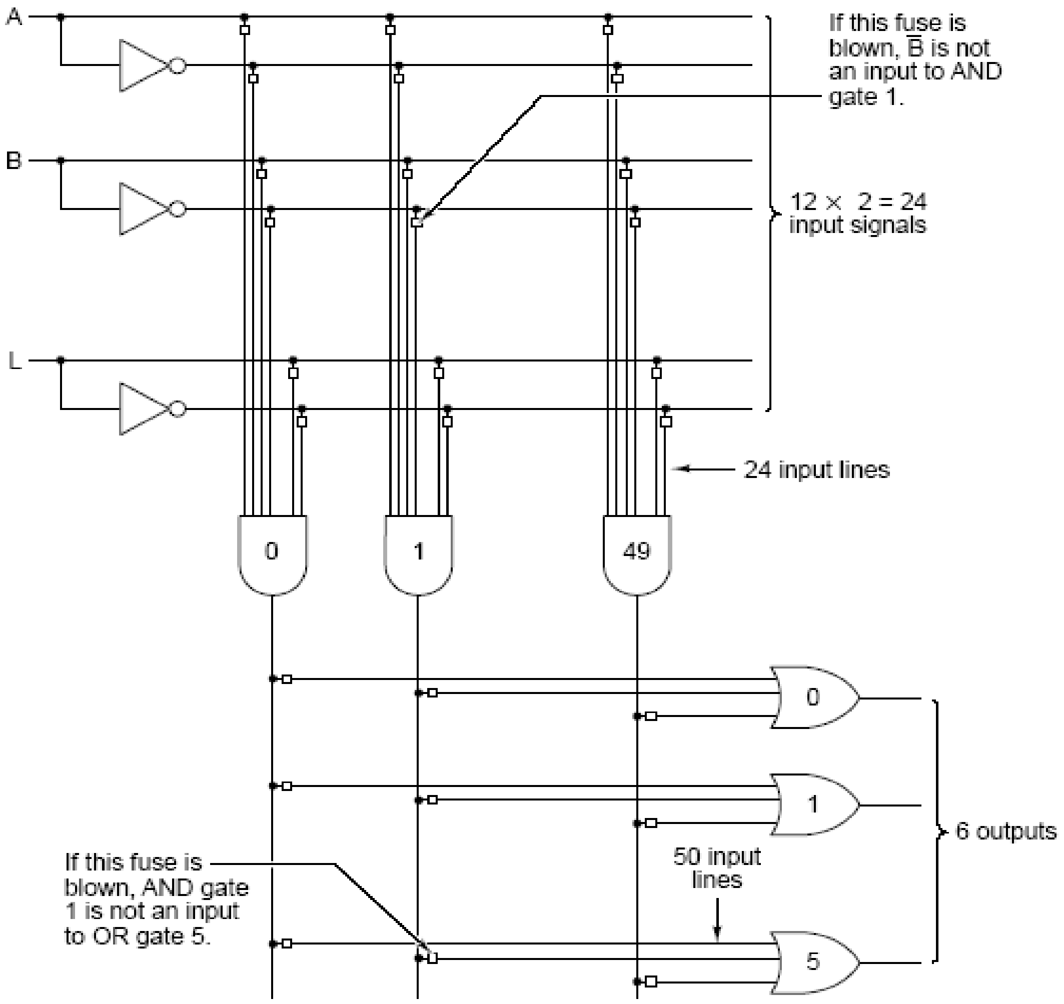


Figure 3-15. A 12-input, 6-output programmable logic array.

Reading

- Next lecture: Combinational Logic
  – Section 3.3 (Skip Don’t Cares section)