Converting from truth tables

- Converting *to* a truth table is straightforward
- How do we convert from a truth table to a circuit or Boolean expression?
  - We will find that we can still "read it off", in a sense
  - There are infinitely many correct circuits or Boolean expressions for a given truth table
  - We might not find the most efficient answer (we'll see a better way later that only works for small problems)

Boolean Algebra Terminology

1. Inverse
   - Complement of a variable
   - Denoted by a bar over a variable
2. Literal
   - A variable or its complement
3. Product/Implicant
   - AND of one or more literals
4. Minterm
   - A product involving all inputs to a function
5. Sum
   - OR of one or more literals
6. Maxterm
   - Sum of all inputs to a function

Order of Operations

1. NOT
2. AND
3. OR

Key: Products are performed before sums!

$$Y = A + BC$$

$$Y = (A + B)(B + C) + ABC$$

$$A = 1$$

$$B = 0$$

$$C = 1$$

Sum of Products (SOP) Form

A Boolean equation for any truth table can be written by summing minterms where output Y is true

Each row has its own minterm that is TRUE for that row

In this case, Y = A'B

A	B	Y	minterm	minterm name
0	0	0	$\overline{A}\ \overline{B}$	m_0
0	1	1	$\overline{A}\ B$	m_1
1	0	0	$A\ \overline{B}$	m_2
1	1	0	$A\ B$	m_3

SOP Form

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

1. Find the outputs that are 1
    1. Second and fourth rows
2. Find the corresponding minterms
    1. A'B
    2. AB
3. OR the minterms
    1. A'B + AB

For any combination of inputs, our expression above matches the output of the truth table

A	*B*	*C*	*Y*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Product of Sums

Reverse of SOP

Join FALSE rows with ANDs on the maxterms (ORs)

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

Product of Sums Form

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

1. Find the outputs that are 0
    1. First and third rows
2. Find the corresponding maxterms that make the row False
    1. A + B
    2. A′ + B
3. AND the maxterms
    1. (A + B)(A′ + B)

For any combination of inputs, our expression above matches the output of the truth table

Product of Sums

Reverse of SOP

Join FALSE rows with ANDs on the maxterms (ORs)

Simpler than SOP in cases where there are only few outputs that are false

Both forms are equal to one another!

(A + B)(A′ + B) = = A′B + AB

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

- Function is true if
  - Any of the true conditions are met (SOP)
  - ALL of the false lines are false (product-of-sums)
  - (A + B + C′)(A + B′ + C)(A + B′ + C′)(A′ + B′ + C)(A′ + B′ + C′)

Sum of Products Practice

A	B	C	Out
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Sum of Products Practice

A	B	C	Out
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$(A + B' + C')\ (A' + B + C)$

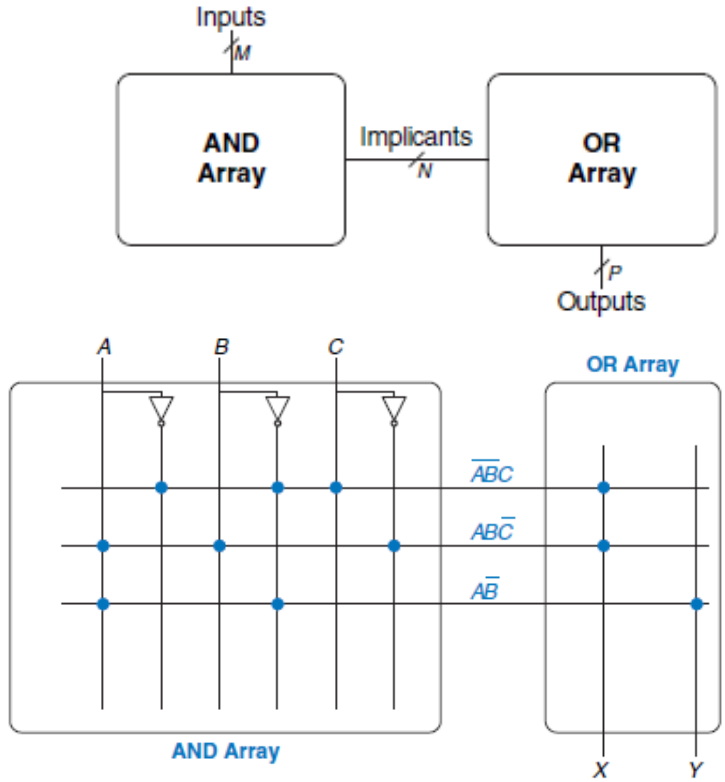
Programmable Logic Array (PLA)

Two-level combinational logic in sum-of-products form

- Can use SOP to represent an arbitrary truth table
- Logically, it follows then that we can build a logic array to represent that truth table
- SOP is just AND and ORs!

PLA

- Built from an AND array followed by an OR array



Programmable Logic Array (PLA)

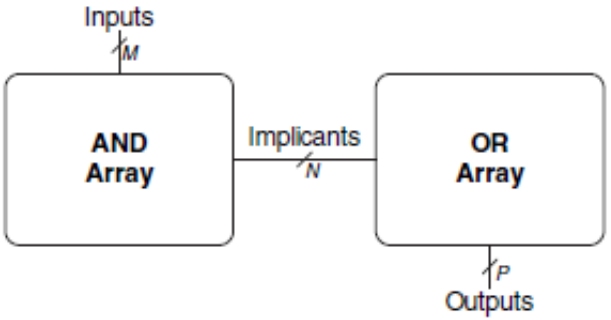
Two-level combinational logic in sum-of-products form

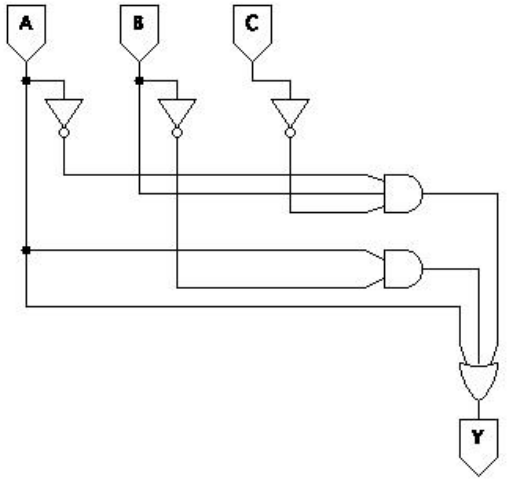
- Can use SOP to represent an arbitrary truth table
- Logically, it follows then that we can build a logic array to represent that truth table
- SOP is just AND and ORs!

PLA

- Built from an AND array followed by an OR array

- Example PLA-style circuit
  - A'BC' + AB' + A





Pros and Cons – Truth Tables

- Pros:
  - the only unique representation
  - (relatively) easy to convert from and to

- Cons:
  - size grows exponentially with number of inputs
  - may not give intuitive feel for what is happening

Pros and Cons – Logic Gates

- Pros:
  - Most closely match our intuition of physical circuits
  - Good for building large, hierarchical systems

- Cons:
  - Not always easy to tell whether two circuits are equivalent
  - Not as easy to simplify as Boolean algebra

Pros and Cons – Boolean Algebra

- Pros:
  - May give simple intuition in some cases
  - Often easier to simplify than other forms

- Cons:
  - Not unique
  - Not good for building large systems