The Truth,
the Whole Truth
and
Nothing But the Truth

~~or~~

Designing circuits to implement digital logic

Review of Objectives

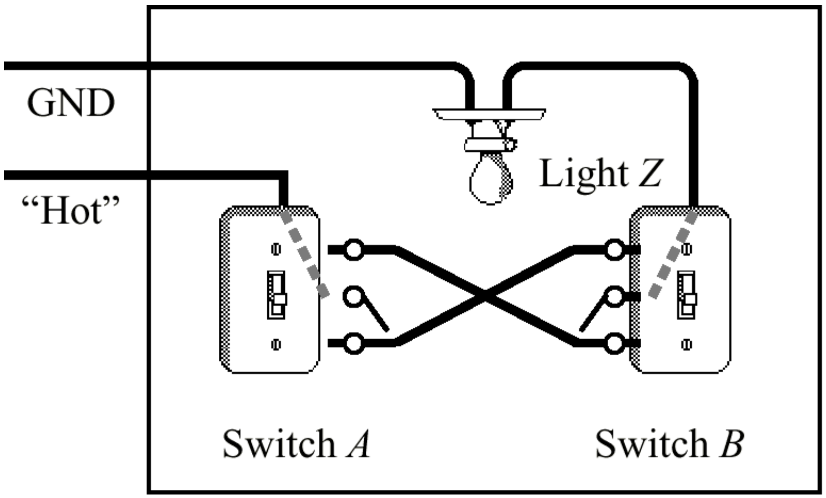
After this lecture, you should be able to…….

- **Compile a truth table for all possible functions of two binary variables**
- **Associate these logic functions with their schematic symbols**
  - **AND, OR, NOT, BUFFER**
  - **NAND, NOR, XOR, XNOR**
- **Draw simple electronic circuits that implement**
  - **NOT, NAND and NOR gates**
- **Define the following terms**
  - **Logic Threshold**
  - **Buffer**

Truth Tables

- Developed in 1854 by George Boole
- further developed by Claude Shannon (Bell Labs)
- Outputs are computed for all possible input combinations (how many input combinations are there?

Consider a room with two light switches. How must they work†?



Inputs		Output
A	*B*	*Z*
0	0	0
0	1	1
1	0	1
1	1	0

†This breaks the electrical wiring code - Do you know why?

Can you identify this logic function, mentioned in the previous lecture?
NB how number of odd and even inputs A,B effect output Z.

Alternate Assignments of Outputs to Switch Settings

- Logically identical truth table to the original (see previous slide), if the switches are configured up-side down.

Inputs		Output
A	*B*	*Z*
0	0	1
0	1	0
1	0	0
1	1	1

Truth Tables Showing All Possible Functions of Two Binary Variables

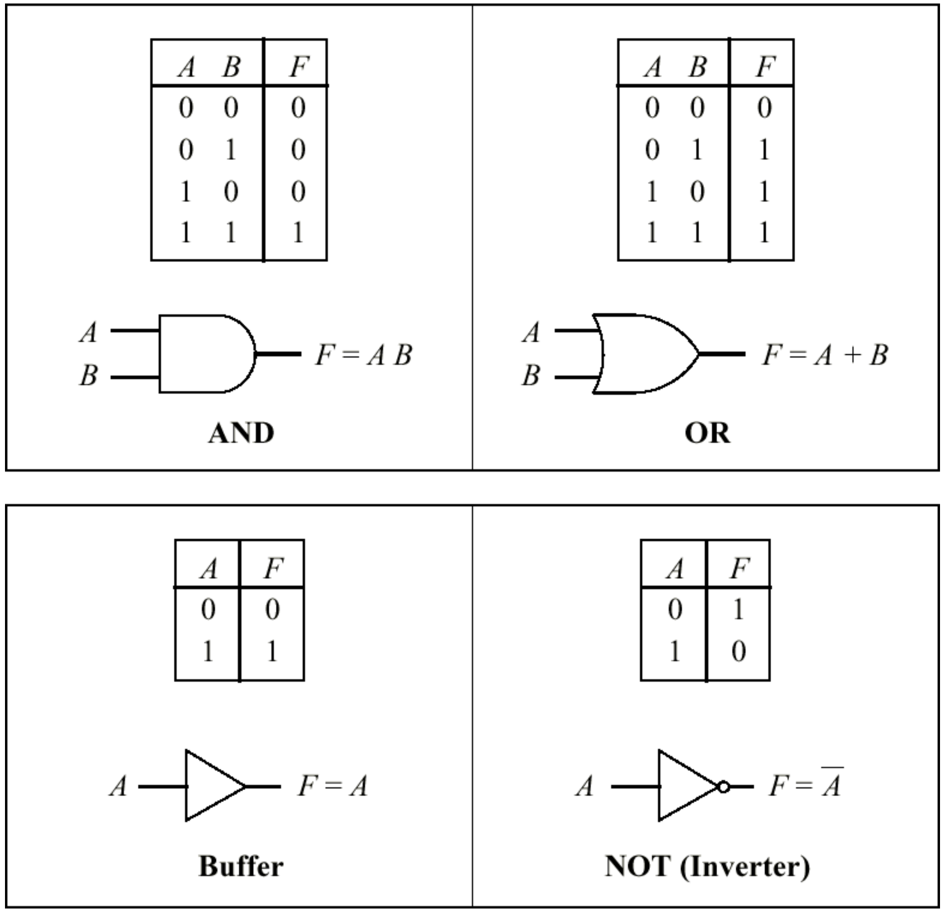
Inputs		Outputs							
A	B	*False*	*AND*	$A\overline{B}$	A	$\overline{A}B$	B	*XOR*	*OR*
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Inputs		Outputs							
A	B	*NOR*	*XNOR*	$\overline{B}$	$A + \overline{B}$	$\overline{A}$	$\overline{A} + B$	*NAND*	*True*
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

- **The more frequently used functions have names: AND, XOR, OR, NOR, XOR, and NAND. (Always use upper case spelling.)**

Logic Gates and Their Symbols

Logic symbols for AND, OR, buffer, and NOT Boolean functions



AND:

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

OR:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Buffer:

A	F
0	0
1	1

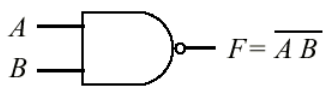
NOT (Inverter):

A	F
0	1
1	0

- **Note the use of the “inversion bubble” mentioned in prior lecture.**
- **(Be careful about the “nose” of the gate when drawing AND vs. OR.)**

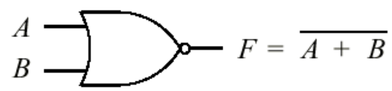
Logic symbols for NAND, NOR, XOR, and XNOR Boolean functions

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



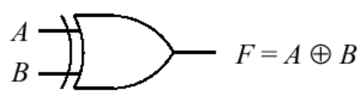
NAND

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



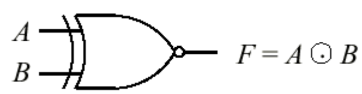
NOR

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



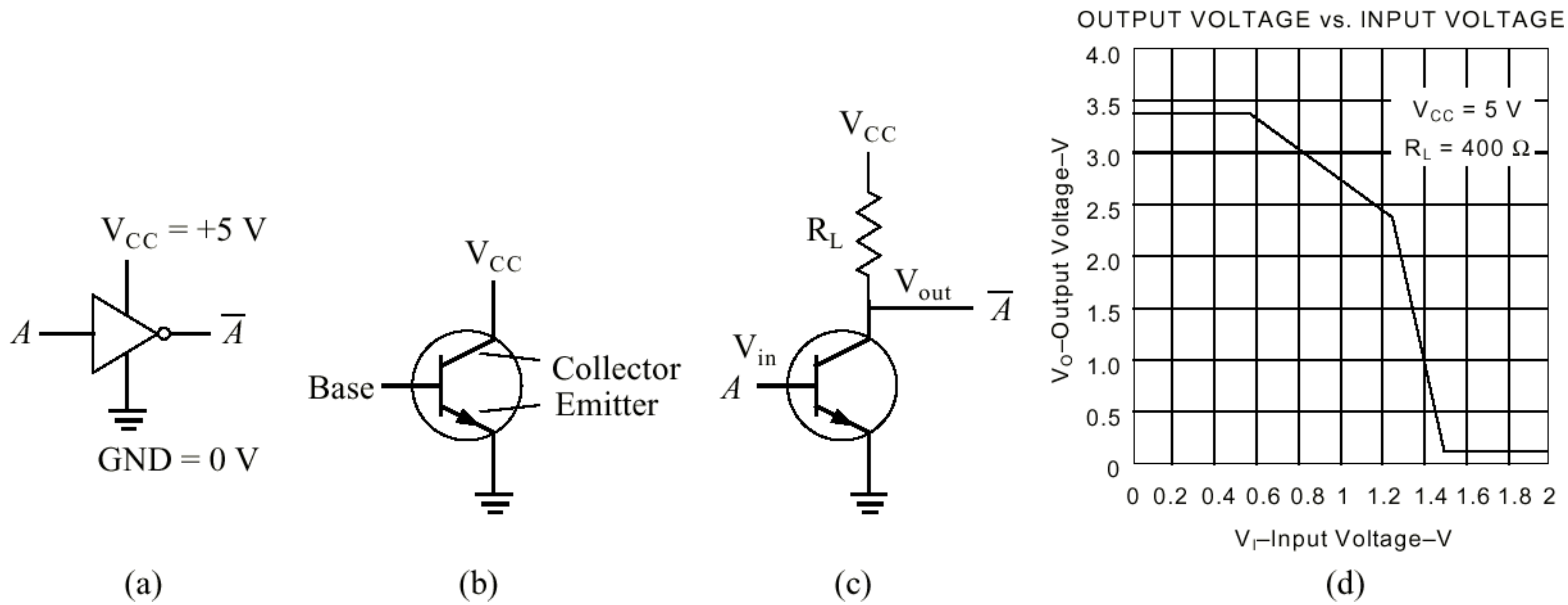
Exclusive-OR (XOR)

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



Exclusive-NOR (XNOR)

Transistor Level Inverter Revisited



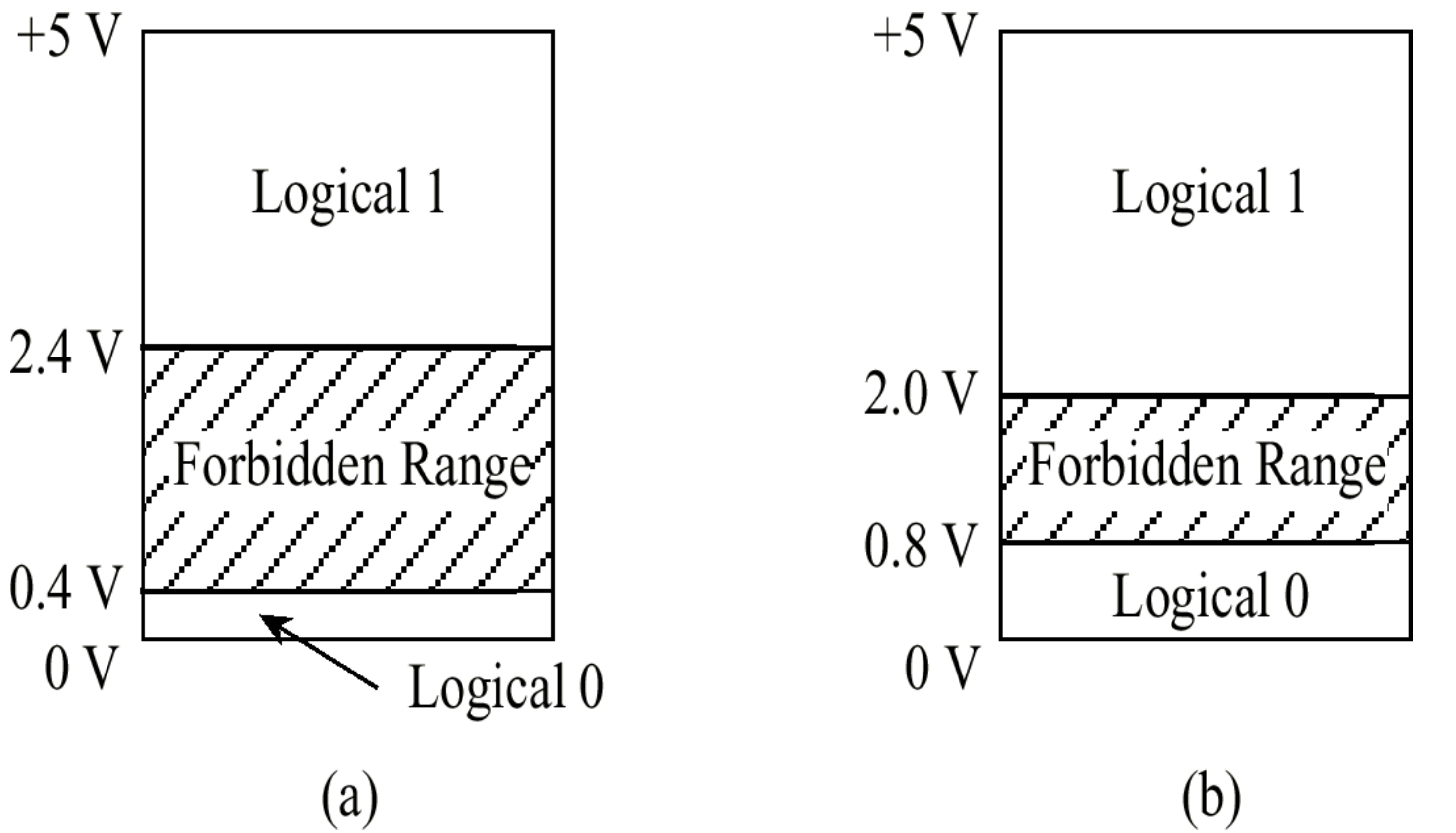
Power Terminals

Transistor Symbol

A Transistor Used as an Inverter

Inverter Transfer Function

Examples of Logic Thresholds at gate output (a) and gate input (b)



Why Use Buffers?

- **Logic gate input and outputs can only source or sink a given amount of current.**
- **The limit to how much current a device can source or sink determines the number of inputs that can be attached to a logic gate's output. This is called the gate's FANOUT.**
- **If necessary a buffer can be used to drive heavier loadings than the original fanout rating of a single gate.**

Technology Progression

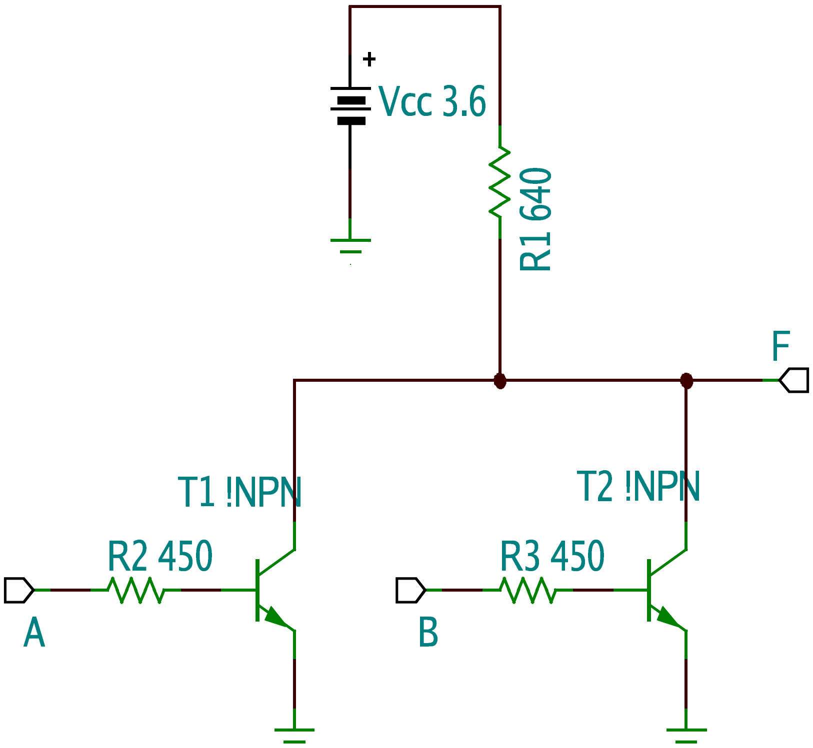
- **Logic circuits examined so far use bipolar junction transistors and resistors: It's an old technology called RTL (resistor-transistor logic).**

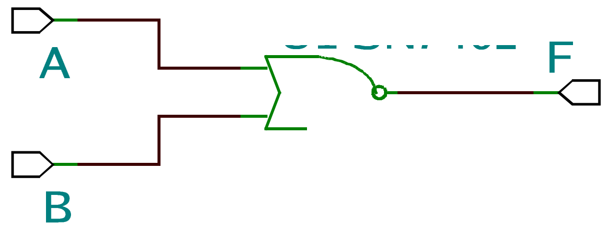
- **RTL was superceded with another technology called NMOS (n-type metal-oxide semiconductor) and then by CMOS (complementary metal oxide semiconductor).**

- **CMOS technology offers**
  - **Smaller on-chip geometries**
  - **Tiny steady-state supply currents**

Moving Toward CMOS Gates #1

- **Resistor-Transistor Logic**



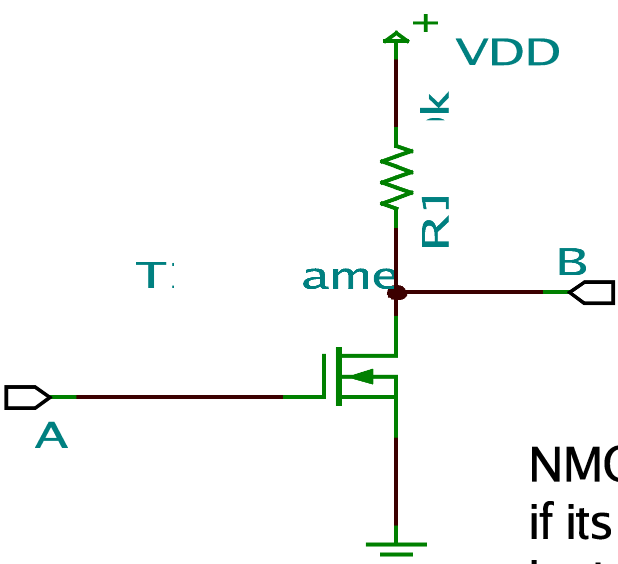


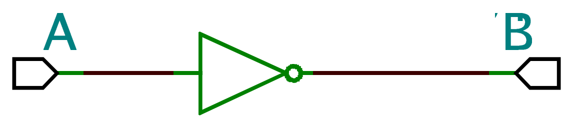
If a logic ‘high’ voltage is applied to an input the transistor turns on.

A small base current makes the transistor act as if a switch has been closed between its collector and emitter pins.

Moving Toward CMOS Gates #2

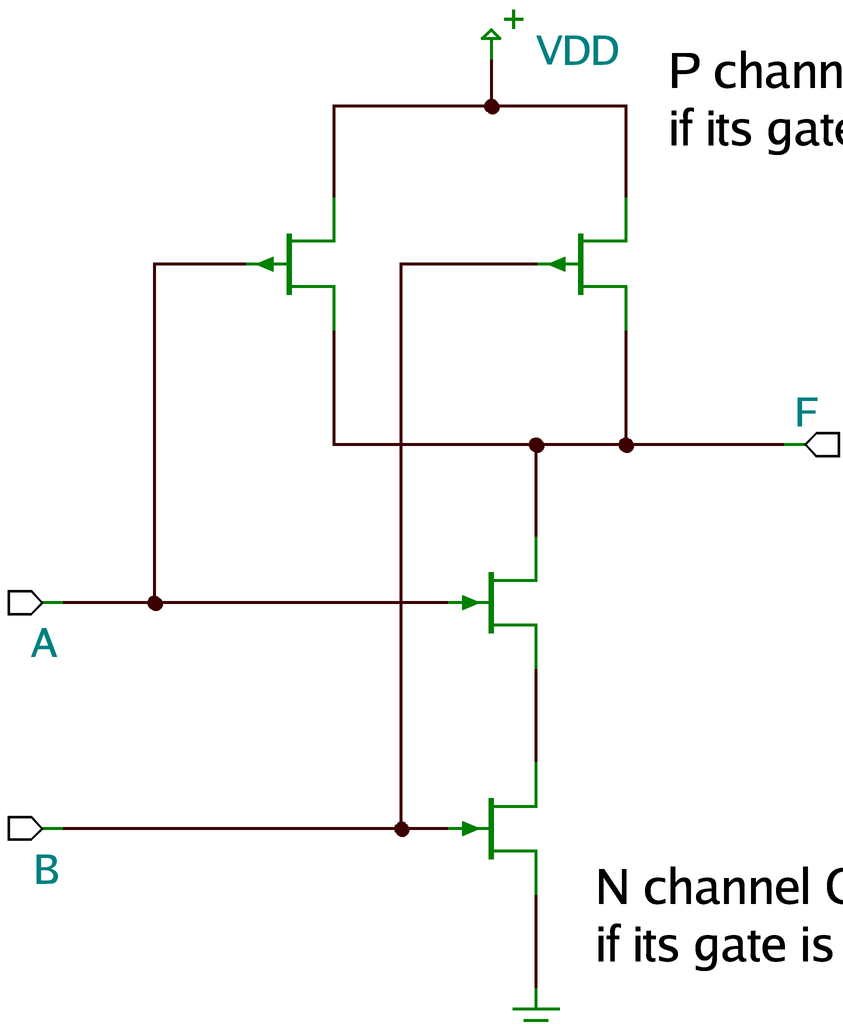
- **NMOS (a step in the transition to CMOS)**



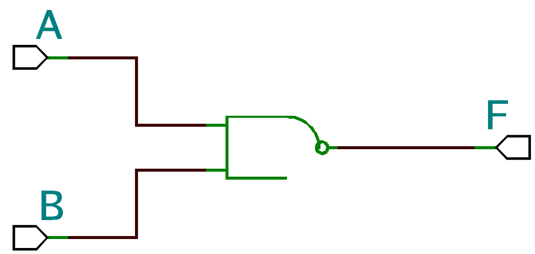


NMOS device conducts if its gate is subject to a logic ‘high’ voltage

Moving Toward CMOS Gates #3



P channel CMOS device conducts
if its gate is at logic ‘low’ voltage

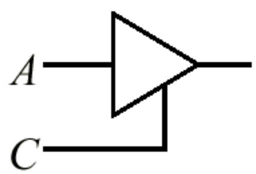


N channel CMOS device conducts
if its gate is subject to a logic ‘high’ voltage

Tri-State Buffers

- **Outputs can be 0, 1, or “electrically disconnected.”**

C	A	F
0	0	ø
0	1	ø
1	0	0
1	1	1



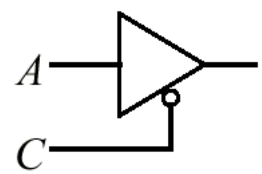
$F = A\,C$

or

$F = ø$

Tri-state buffer

C	A	F
0	0	0
0	1	1
1	0	ø
1	1	ø



$F = A\,\overline{C}$

or

$F = ø$

Tri-state buffer, inverted control

Objectives Again

- **Compiled a truth table showing all 16 i.e. 2^{2^n} possible combinations of n = 2 logic outputs**
- **Associated the following logic functions and symbols**
    - **AND, OR, NOT, BUFFER**
    - **NAND, NOR, XOR, XNOR**
- **Drew simple circuits that implement**
    - **NOT, NAND and NOR gates**
- **Defined terms logic threshold, buffer**

Next time

- **Review rules of Boolean algebra**

- **Investigate two combinational logic forms:**
  - **Sum of Product : SOP**
  - **Product of Sum : POS**

- **Distinguish between positive and negative logic**

- **Review a datasheet for a SSI logic IC**

SSI : small scale integrated circuit