



# Digital Logic Gates

---

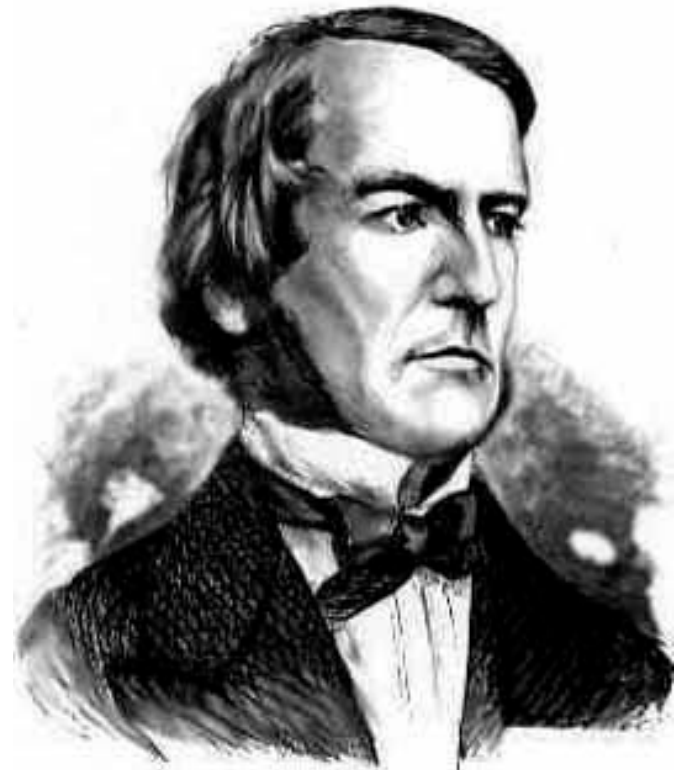
Origins  
Families  
Symbols  
Truth Tables  
Boolean Expressions



# The Origins of Logic Gates

---

**It was George Boole (1815 – 1864) that approached logic in a new way, reducing it to simple algebra, thus incorporating logic into mathematics.**





## ORIGINS CONTINUED...

---

Boole, a mathematician, wrote two papers. The first, "*Mathematical Analysis*" discussed the encoding of logical arguments so that they could be manipulated and solved mathematically.

He came up with a type of linguistic algebra, the three most basic operations of which were (and still are) AND, OR and NOT. These were the only operations necessary to perform comparisons or basic mathematical functions.



# ORIGINS CONTINUED...

---

Boole's second paper, "*An Investigation of the Laws of Thought, on Which Are Founded the Mathematical Theories of Logic and Probabilities*" was based on a binary approach, processing only two objects - the now famous yes-no, true-false, on-off, zero-one.

These ideas made their way into the curriculum of a university logic-philosophy course where a student named Claude Shannon took it.

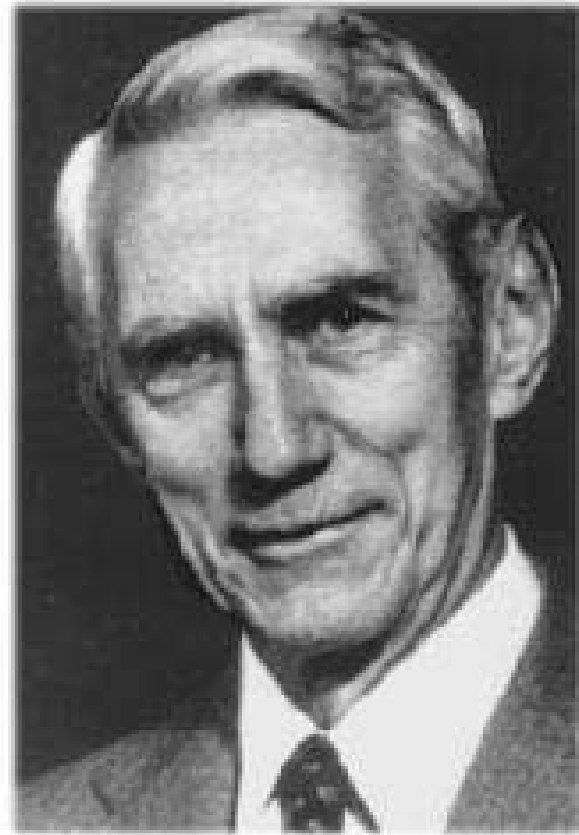


## ORIGINS CONTINUED...

---

**In 1936, graduate student Claude Shannon (1916 - 2001) arrived at the Massachusetts Institute of Technology.**

**To earn money, Shannon worked on his professor's project, a mechanical computing device - the Differential Analyser.**

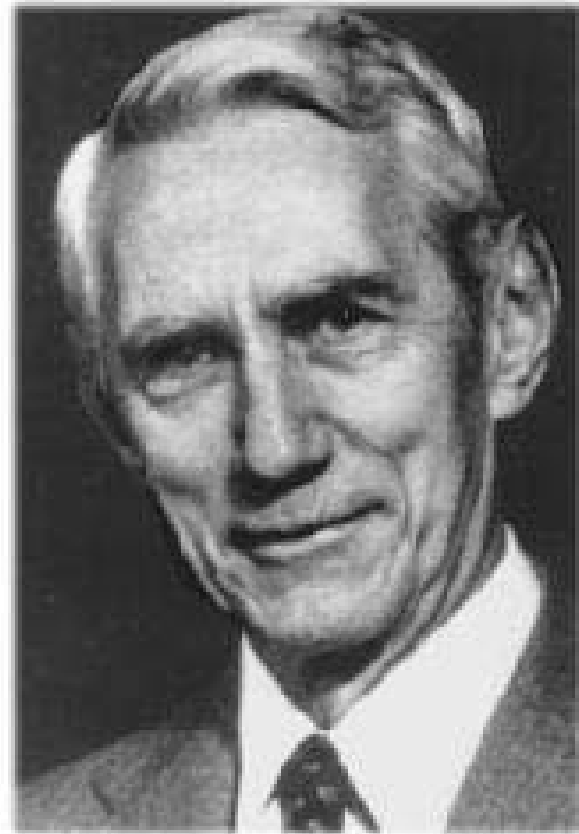




## ORIGINS CONTINUED...

---

For his master thesis he considered ways of improving the Differential Analyser, perhaps by using electrical circuits instead of the present cumbersome collection of mechanical parts.

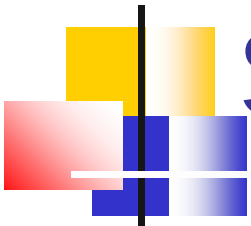




## ORIGINS CONTINUED...

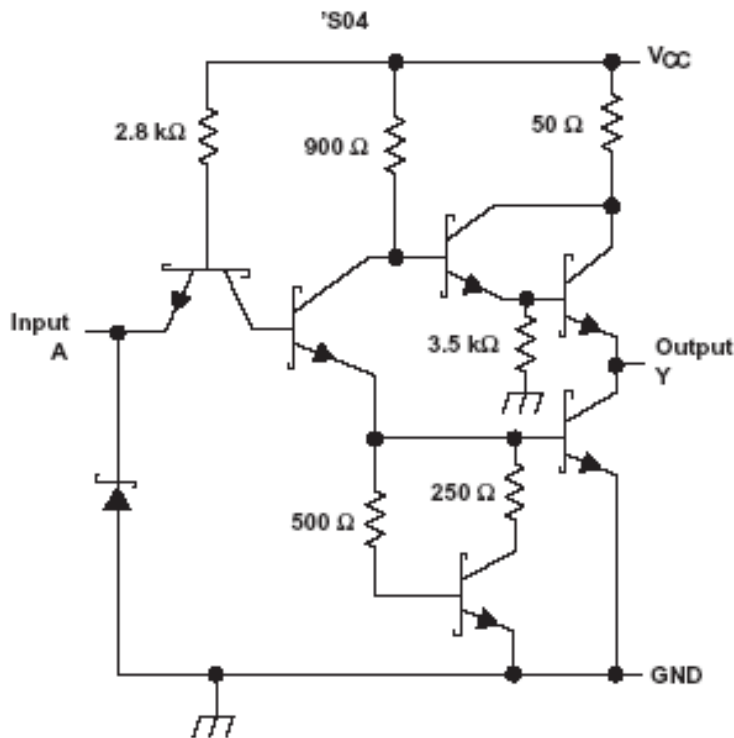
---

**Then Shannon remembered about the boolean algebra he'd learned as an undergraduate was in fact very similar to an electric circuit. The next obvious step would be to lay out circuitry according to Boolean principles.**

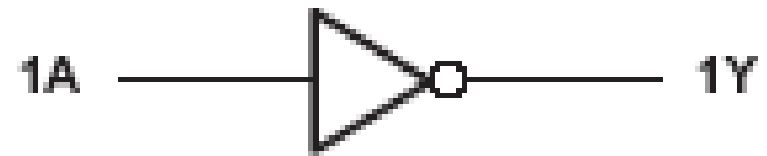


# Schematic to Symbol

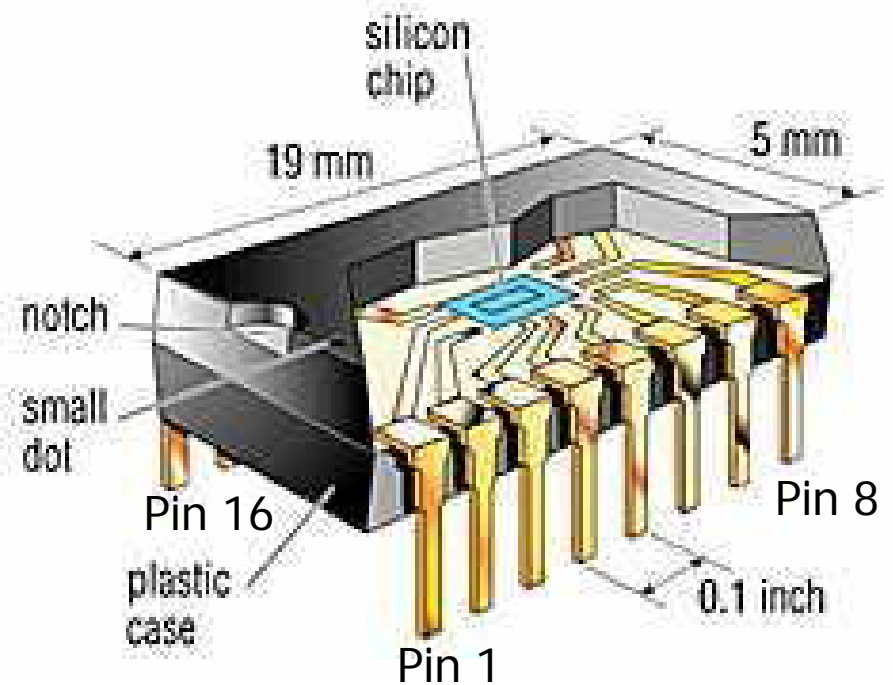
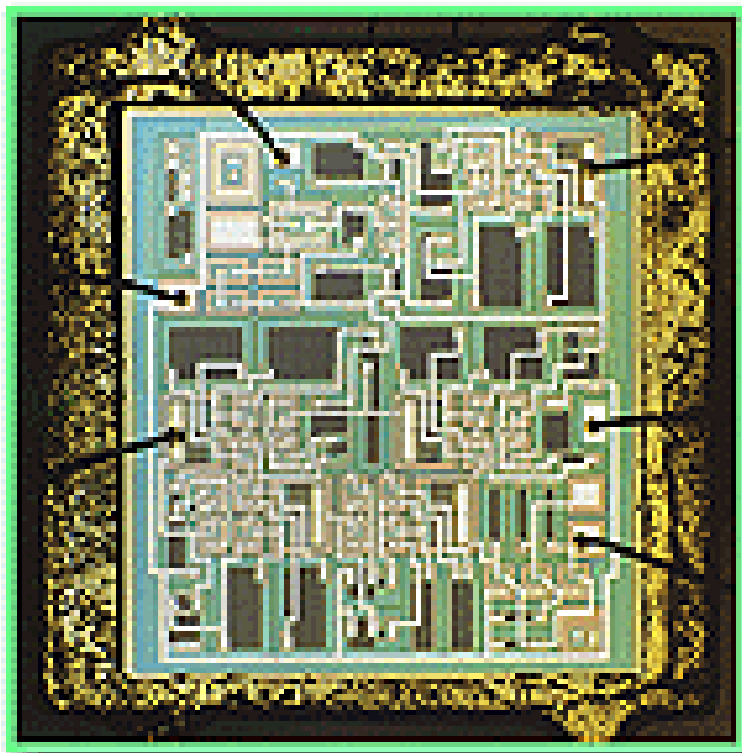
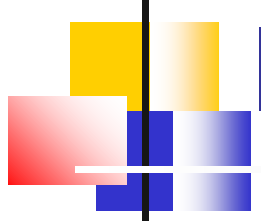
## NOT Gate Schematic



## Not Gate Symbol



# Logic Gates and the Integrated Circuits





# LOGIC GATE FAMILIES

---

7400

TTL (transistor-transistor logic)

Operates on 5V (4.75 – 5.25V)

Fast but requires uses more power

5400

Military version

Operates on 5V (4.5 – 5.5V)

CD4000

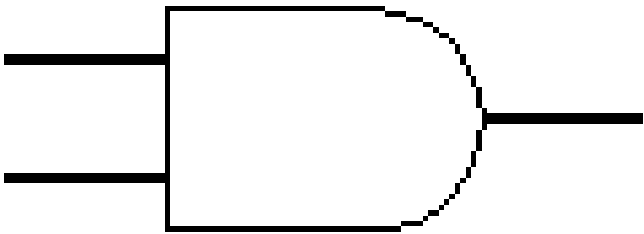
CMOS

Operates between 3 – 18V

Requires less power but not as fast.

# AND Gate - 7408

Symbol



Boolean Expression

$$Y = A \bullet B$$

**Note:** If and A and B are hi, then the output is hi.

1 = Hi / TRUE / +5V

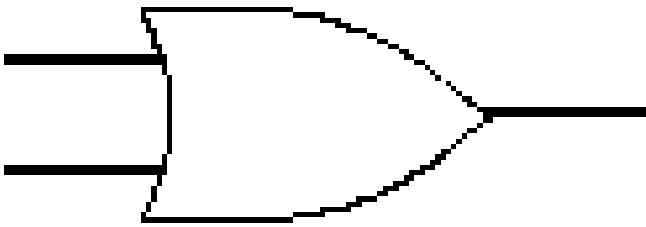
Truth Table

| INPUTS |   | OUTPUT |
|--------|---|--------|
| A      | B | Y      |
| 0      | 0 | 0      |
| 0      | 1 | 0      |
| 1      | 0 | 0      |
| 1      | 1 | 1      |

0 = Lo / FALSE / 0V

# OR Gate - 7432

Symbol



Boolean Expression

$$Y = A + B$$

**Note:** If A or B is hi, then the output is hi.

1 = Hi / TRUE / +5V

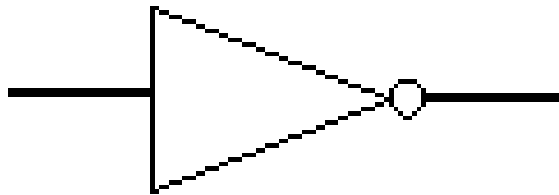
Truth Table

| INPUTS |   | OUTPUT |
|--------|---|--------|
| A      | B | Y      |
| 0      | 0 | 0      |
| 0      | 1 | 1      |
| 1      | 0 | 1      |
| 1      | 1 | 1      |

0 = Lo / FALSE / 0V

# NOT Gate – 7404 (INVERTER)

Symbol



Boolean Expression

$$Y = \overline{A}$$

Truth Table

| INPUT | OUTPUT |
|-------|--------|
| A     | Y      |
| 1     | 0      |
| 0     | 1      |

1 = Hi / TRUE / +5V

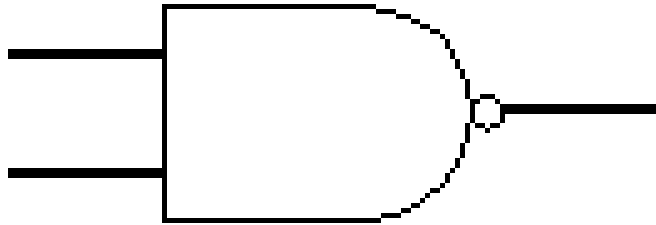
0 = Lo / FALSE / 0V



# NAND Gate - 7400

---

## Symbol



## Boolean Expression

$$Y_* = \overline{A \bullet B}$$

\* Can use either y or f to represent output.

## Truth Table

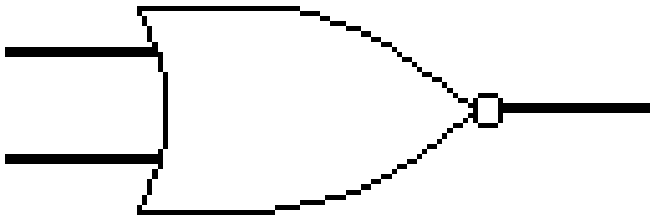
| INPUTS |   | OUTPUT |
|--------|---|--------|
| A      | B | Y      |
| 0      | 0 | 1      |
| 0      | 1 | 1      |
| 1      | 0 | 1      |
| 1      | 1 | 0      |

1 = Hi / TRUE / +5V

0 = Lo / FALSE / 0V

# NOR Gate - 7402

Symbol



Boolean Expression

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

Truth Table

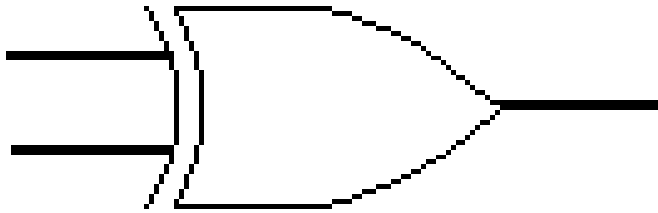
| INPUTS |   | OUTPUT |
|--------|---|--------|
| A      | B | Y      |
| 0      | 0 | 1      |
| 0      | 1 | 0      |
| 1      | 0 | 0      |
| 1      | 1 | 0      |

1 = Hi / TRUE / +5V

0 = Lo / FALSE / 0V

# EXOR Gate - 7486

Symbol



Boolean Expression

$$Y = A + B$$

○

1 = Hi / TRUE / +5V

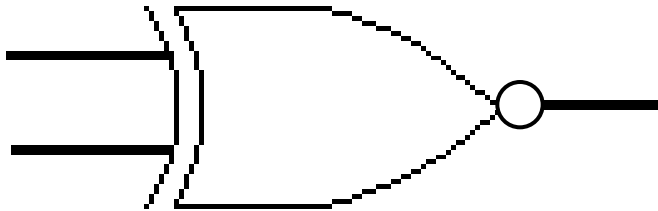
Truth Table

| INPUTS |   | OUTPUT |
|--------|---|--------|
| A      | B | Y      |
| 0      | 0 | 0      |
| 0      | 1 | 1      |
| 1      | 0 | 1      |
| 1      | 1 | 0      |

0 = Lo / FALSE / 0V

# EXNOR Gate

Symbol



Boolean Expression

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

1 = Hi / TRUE / +5V

Truth Table

| INPUTS |   | OUTPUT |
|--------|---|--------|
| A      | B | Y      |
| 0      | 0 | 1      |
| 0      | 1 | 0      |
| 1      | 0 | 0      |
| 1      | 1 | 1      |

0 = Lo / FALSE / 0V



# COMBINATION GATES

---

The order of operations are as follows:

↓  
Brackets  
AND  
OR

This AND operation is done after the OR operation.

Example:  $Y = (A + B) \cdot C$

Because of the brackets, this operation is done first. Otherwise, the OR operation would be done last.

# HOOKING UP THE CHIP

