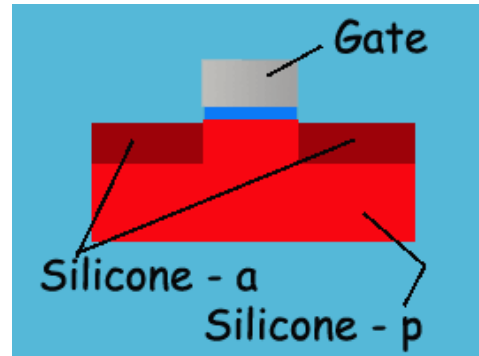


## INTRODUCTION

Imagine for a moment that you are reduced to the size of a molecule and were standing on an Intel processor. What would you see? Mostly, you would see gates. A gate is a kind of transistor that controls the flow of electricity.

What you see might see could look a bit like the image at the right, but naturally it makes no sense, yet. But understanding gates and why they occur in computers is useful for understanding the need for binary and hexadecimal number systems. If you look at the on/off switch on your computer, you will probably notice that the icon on the switch looks like a circle with a line passing through the top. That is actually a zero and a one. Zero means "off," and one means "on." When you switch your computer on, you are switching it to a state of "1." Gates work much the same way. Collections of gates working in combination can create a binary number (e.g., 01001011) that becomes a meaningful message. By putting billions of gates into a processor, Intel and other chip manufacturers make it possible to create general purpose, programmable computers.

This chapter explains how it all works.

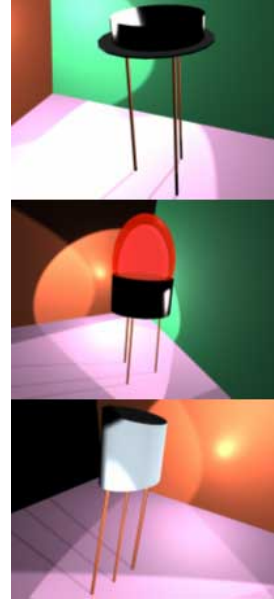


The above image is much like Intel would use to explain how gates work on their processors. Presently, it makes no sense. By the end of this chapter, however, it should all come together.

### WHAT ARE GATES?

"Gates" are another name for transistors. Transistors come in a variety of configurations and serve a variety of purposes. Some keep the flow of electricity going in only one direction while others permit electricity to flow only under special conditions. Why, one might ask, does anybody need these qualities? If we look at a variety of conditions, we may see.

Suppose you are the floor supervisor of a perfume factory. Suppose one position causes you a great deal of trouble. The person who puts the bottles in the boxes typically gets tired of the job after only a few days and quits after only a few weeks. You want to automate.

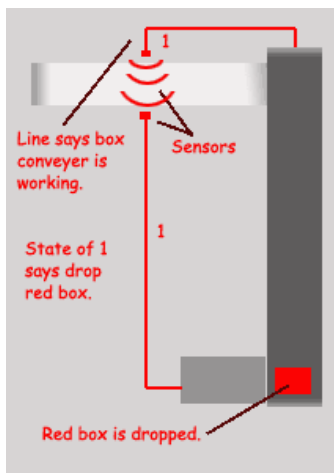


**The three examples above represent only a few examples of the dozens of sizes, shapes, and capabilities found in transistors. Some transistors direct electrical current under special conditions; some shed light, others boost power. In the past fifty years they have evolved from a novelty to a critical tool for our daily lives.**

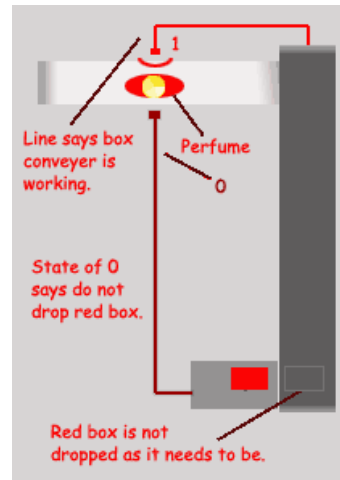
## INTRODUCTION

If you wanted to automate the process, you might put a sensor on one belt that caused a box to be placed on the other belt at a predetermined time. You might run a line from the box belt to the sensor to let the bottle belt know that the box belt was running (don't want bottles dropping onto the floor). The sensor would know when a bottle passed and would cause a box to be dropped on its conveyor belt.

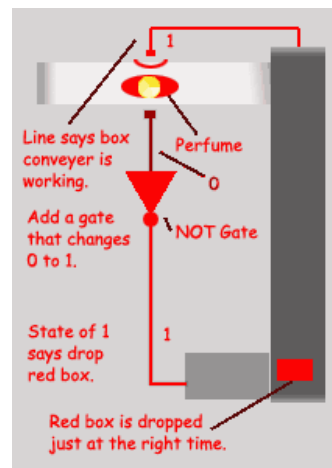
Problem is, when the sensor is interrupted the electricity to the box dropper and does the opposite of what you want unless you find a way to reverse what happens at the sensor. Fortunately, there is a gate called a "not gate." If the not gate is getting electricity, it forwards none. If it is not getting electricity, it forwards electrical current. With a not gate in the line between the sensor and the box dispenser, the system works, and your problems are solved.



The idea is to have a box automatically drop onto the conveyor belt on the right so that the perfume bottle falls into it as it passes.



A signal could be sent from the box conveyor to sensors that recognized the passing of the bottle, but that turns off the bottom line going to the box dispenser.



AThe cure is to place a "NOT Gate in the line. NOT Gates reverse the signal in the line, making a state of 0 become a state of 1 and as the bottle passes the sensor a box is dropped onto its conveyor.

## INTRODUCING "AND GATES"

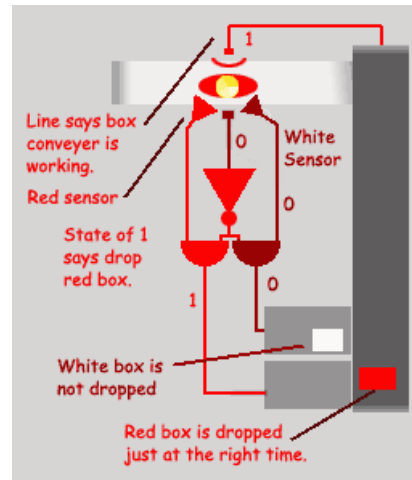
Suppose your profits are so great that you decide you can produce a second line of perfume packaged in a much more elegant (white) box. You need your sensor to be able to differentiate between bottles and send the proper signal to the proper box dispenser.

One solution is to produce one perfume in a red bottle and one in a white bottle and have a sensor that recognizes red and one that recognizes a white bottle. But you also still need to tell the box dispenser that the box conveyor is running.

To do the above, you tie the line from the box dispenser to two sensors that can identify the bottles in such a manner that the correct signal is sent to the correct dispenser. To do that you need a NOT Gate and two "AND Gates."

First tie the NOT Gate to one input lead on each AND Gate and one of the color sensors to the other input leads (one sensor for each gate). Then tie one output lead to the appropriate box dispenser. Now, when a white bottle goes by, the white sensor sends a signal to input B of the white AND GATE. Since it is already getting a signal from the NOT GATE, it will send a command to the white box dispenser.

If you wanted to produce dozens of different bottles and different boxes, you would simply expand the process.



You may add as many AND Gates as you need. You need a sensor to identify color or other special characteristic plus a line from the NOT Gate to add dispensers. If a white bottle arrives, conditions reverse; the white sensor is in a state of 1 while the red sensor is in a state of 0.

### INTRODUCING "OR GATES"

Suppose you need to add one more bottle, but you wish to place it in the same box you use for the red bottle. Suppose the bottle is clear and the sensor will look for the brown liquid in the bottle.

You would add a gate where if  $A$  or  $B = 1$ ,  $C = 1$ .

The gate you add is called an "OR Gate."

### INTRODUCING "NOR GATES"

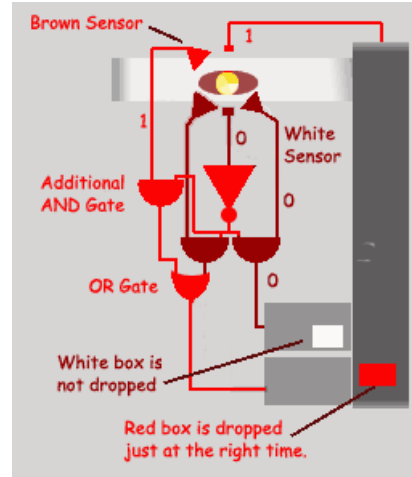
The **NOR Gate** is the opposite of an OR Gate. If either lead  $A$  or lead  $B$  is in a state of 1,  $C$  will be in a state of 0. If  $A$  and  $B$  is in a state of 1,  $C$  will be in a state of 0. Only if both  $A$  and  $B$  are in a state of 0 is  $C$  in a state of 1.

Although the above sounds very confining, it really makes it possible to use a NOR Gate to create any other kind of gate (meaning that the NOR Gate is a **universal gate**. This means that all of the gates on the image to the left could be replaced by a single gate.

### INTRODUCING "NAND GATES"

The **NAND Gate** is the opposite of an AND Gate. If either lead  $A$  or lead  $B$  is in a state of 1,  $C$  will be in a state of 1. If  $A$  and  $B$  is in a state of 1,  $C$  will be in a state of 0. Only if both  $A$  and  $B$  are in a state of 0 is  $C$  in a state of 1.

Like the NOR Gate, the NAND Gate is **Universal**. This means that like the NOR Gate, you can use NAND Gates to make all of the decisions on the right.

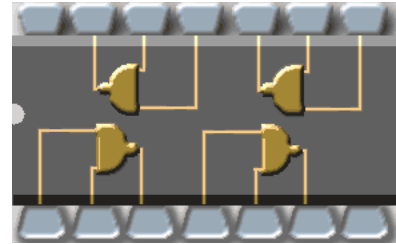


To make it possible to have either one color or the other drop the same box, you add an OR Gate.

## PROGRAMMABLE NAND GATE CHIP

The chip to the right is a 14 pin, programmable, NAND Gate chip. Because both NAND Gates and NOR Gates are both universal, chips can be made using either of them.

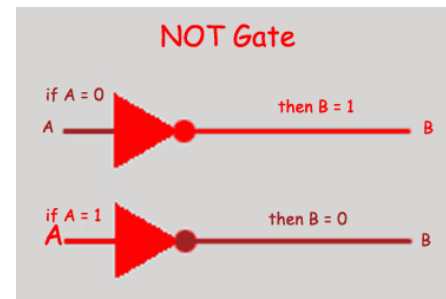
To The transistors that determine how electricity flows can be reduced to microscopic size and placed in computer chips. This is an example of a NAND Gate chip. To see an example of a NOR Gate chip loomake it possible to have either one color or the other drop the same box, you add an OR Gate.



## What the NOT Gate does

As you may see in the image to the right, the NOT Gate changes a state of 0 to 1 or a state of 1 to 0 in a circuit.

The NOT Gate reverses the state of a line. NOTE: it does not change from + to -, It changes from 1 to 0.



## INTRODUCING "OR GATES"

Or gates permit C to be in a state of 1 if either A or B is in a state of 1.

Syllogisms follow:

- If A = 1 and B = 1 then C = 1
- If A = 1 and B = 0 then C = 1
- If A = 0 and B = 1 then C = 1
- If A = 0 and B = 0 then C = 0

OR Gates permit C = 1 if A or B = 1

## "AND GATES"

AND Gates permit lead C to be in a state of 1 only if both A and B = 1.

Syllogisms are as follows:

- If A = 1 and B = 1 then C = 1
- If A = 1 and B = 0 then C = 0
- If A = 0 and B = 1 then C = 0
- If A = 0 and B = 0 then C = 0

AND Gates permit C = 1 only if both A and B = 1.

