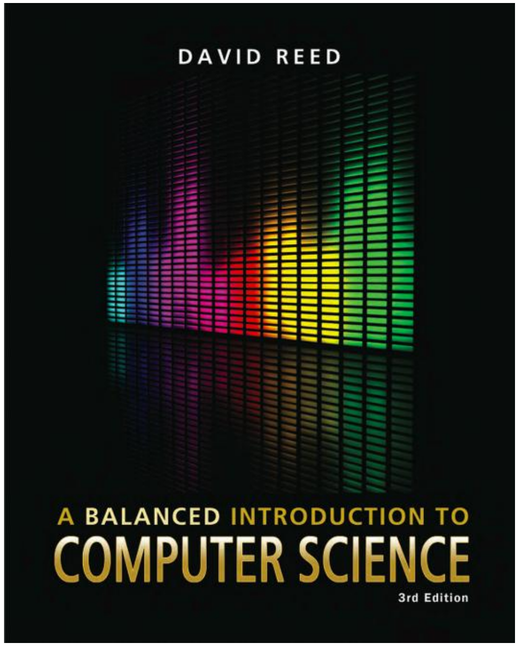
A Balanced Introduction to Computer Science, 3/E

David Reed, Creighton University

ISBN 978-0-13-216675-1



Chapter 16
Inside the Computer –
Transistors and Integrated Circuits

Electricity and Switches

modern computers are powered by electricity, using electrical signals to store and manipulate information

the components of a computer require electrical power to carry out their assigned task

- electricity generates the light that shines through a computer screen, illuminating the individual pixels that make up images and letters
- electricity runs the motor that spins the hard-drive disk, allowing information to be accessed
- main memory and CPU employ electrical signals to store and manipulate data
- bit patterns are represented by the presence or absence of electrical current along a wire

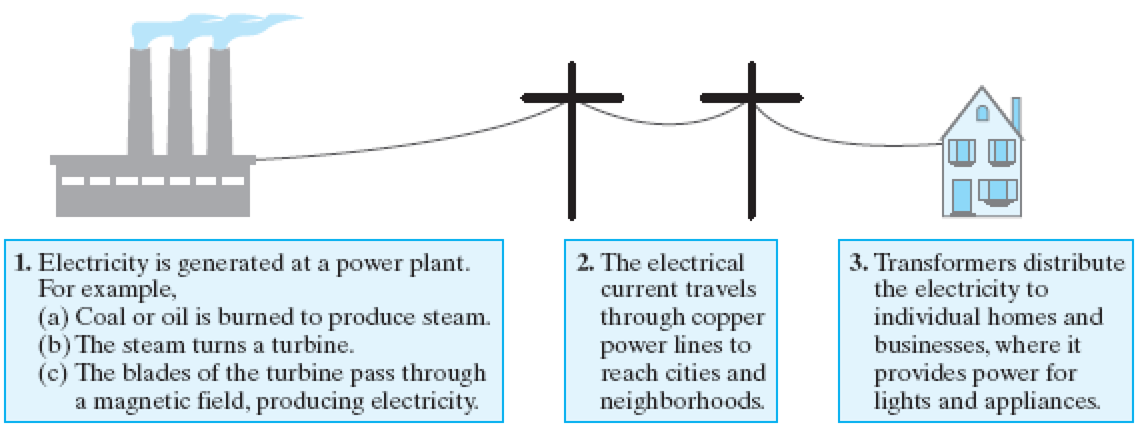
Electricity Basics

electricity is a flow of *electrons,* the negatively charged particles in atoms, through a medium

- good conductors of electricity allow for the flow of electrons with little resistance (e.g., copper, silver, gold)
- other elements, especially nonmetals, are poor conductors (e.g., carbon, oxygen)

electricity can be quantified in *amperes* or *voltage*

- *amperes* gauge electron flow: 1 amp is equal to 6.24 quintillion electrons flowing past a given point each second
- *voltage* measures the physical force produced by the flow of electrons: standard household in United States has 110 to 120 volt outlets



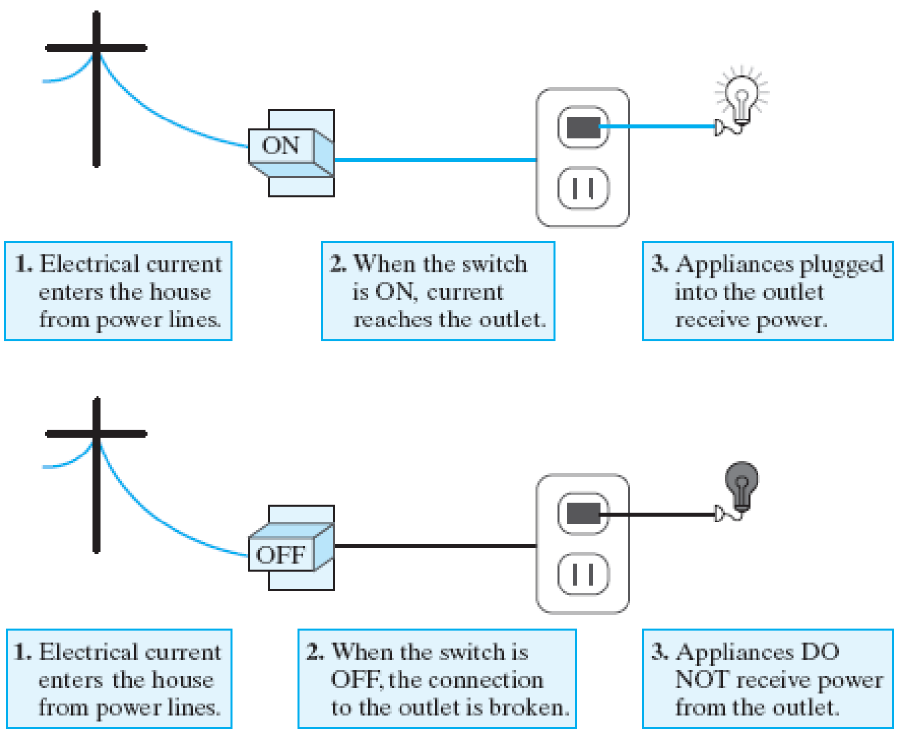
Switches

the most basic tool for controlling the flow of electricity is a *switch*

- a switch can be flipped to connect or disconnect two wires, thus regulating the flow of electricity between them

example: a light switch on a wall serves as an intermediary between the power line entering your home and the outlet that operates a lighting fixture

- if the switch is turned on, then the wires that link the outlet to the power line are connected, and the lighting fixture receives electricity
- if the switch is turned off, then the connection is interrupted, and no power reaches the outlet



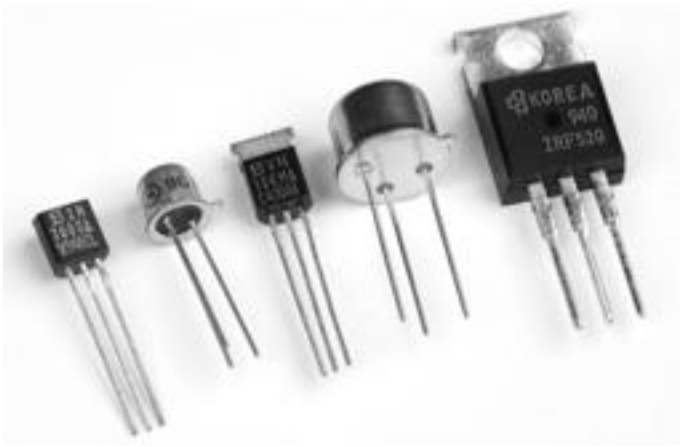
Transistors

as we saw in Chapter 6, advances in switching technology have defined the generations of computers

- 1930’s – electromagnetic relays served as physical switches, whose on/off positions were controlled by the voltage to a magnet
- 1940’s – vacuum tubes replaced relays, which were faster (since no moving parts) but tended to overheat and burn out frequently
- 1948 – the transistor was developed by Bardeen, Brattain, and Shockley
  - a transistor is a solid piece of metal attached to a wire that serves as a switch by alternatively conducting or resisting electricity
  - transistors allowed for the development of smaller, faster machines at a lower cost

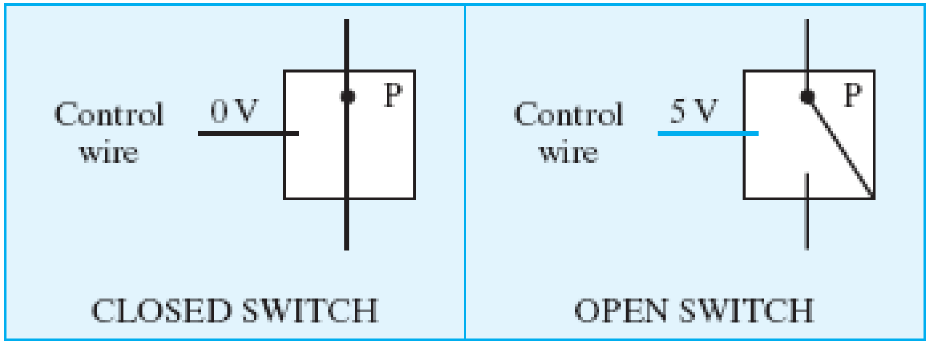
semiconductors are metals that can be manipulated to be either good or bad conductors of electricity

- the first transistors were made of germanium and gold, but modern transistors are constructed from silicon
- through a process known as *doping*, impurities are added to a slab of silicon, causing the metal to act as an electrical switch

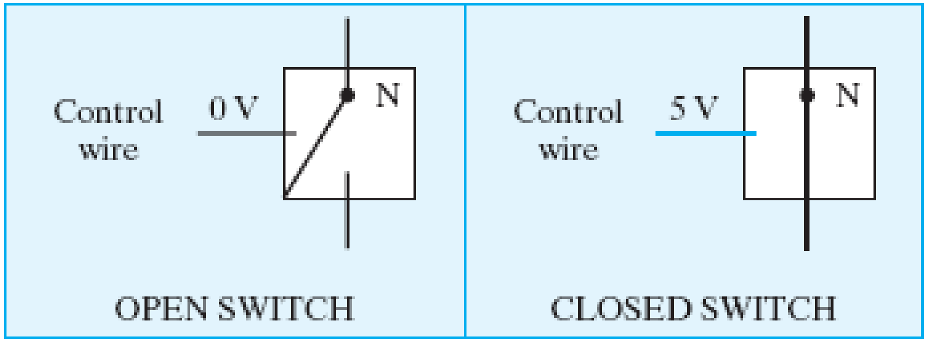


Transistors as Switches

a *PMOS transistor* is positively doped, so that the switch is "closed" when there is no current on the control wire, but "opens" when current is applied



an *NMOS transistor* is negatively doped, so that the switch is "open" when there is no current, but "closes" when there is current

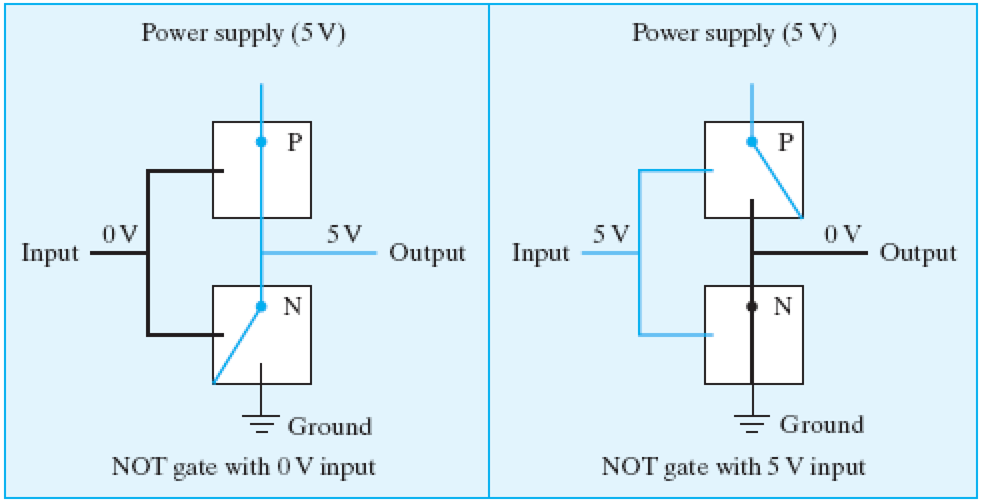


From Transistors to Gates

transistors can be combined to form a *circuit,* which controls the flow of electricity in order to produce a particular behavior

example: consider the following circuit combining two transistors

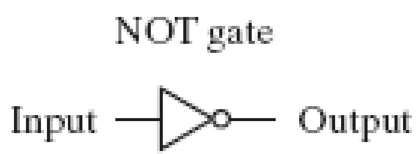
- if no current (0 volts) is applied to the input wire, the PMOS transistor will close to allow current to travel on the output wire, and the NMOS transistor will open to disconnect the ground
- if current (5 volts) is applied to the input wire, the PMOS transistor will open to disconnect the output wire, and the NMOS transistor will close to ground the input
- the end result is that the output is the opposite of the input
- this circuit known as a *NOT gate*



Gates and Binary Logic

the term “gate” suggests a simple circuit that controls the flow of electricity

- in the case of a NOT gate, the flow of electricity is manipulated so that the output signal is always opposite of the input signal
- we can think of a gate as computing a function of binary values
  - 0 represents no current; 1 represents current



input	NOT output
0	1
1	0

  - the symbol to the left (triangle w/ circle) is often used to denote a NOT gate
  - the *truth table* to the right describes the mapping of input to output

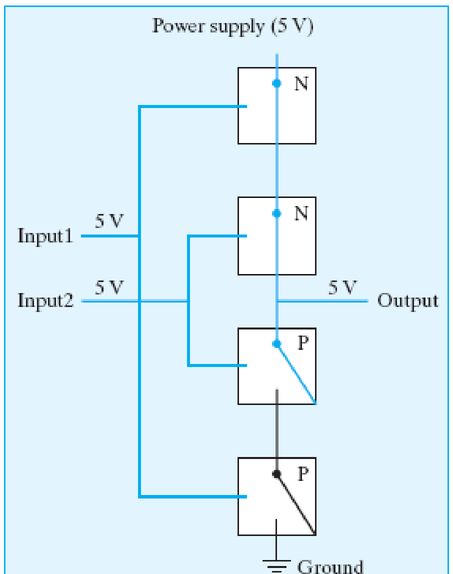
note: NOT gates invert voltages in the same way that the JavaScript NOT operator (!) inverts Boolean values

  - 0 corresponds to false; 1 corresponds to true

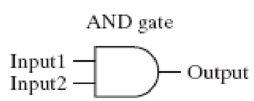
Gates and Binary Logic

many other simple circuits can be defined to perform useful tasks

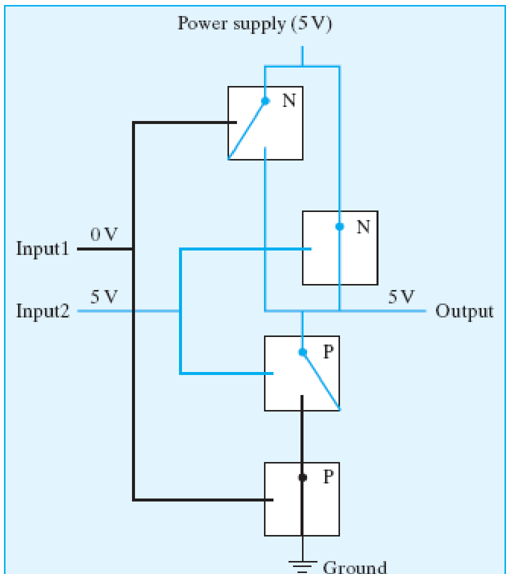
- AND gate – produces voltage on its output wire if both input wires carry voltage
- OR gate – produces voltage on its output wire if either input wire carries voltage



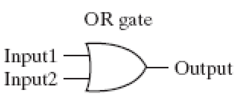
AND gate with inputs 5 V and 5 V



input1	input2	AND output
0	0	0
0	1	0
1	0	0
1	1	1



OR gate with inputs 0 V and 5 V



input1	input2	OR output
0	0	0
0	1	1
1	0	1
1	1	1

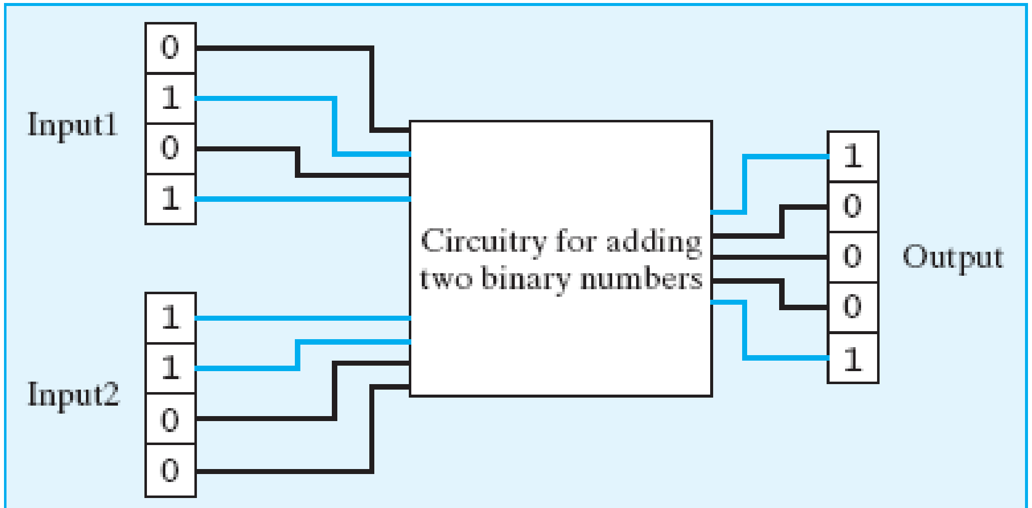
- AND, OR, and NOT gates can be combined to construct all the circuitry required to store and manipulate information within a computer

From Gates to Circuits

transistors are connected to form basic logic gates, which are then combined to build more advanced circuitry

example: adding two binary numbers

- we can represent a 4-bit binary number using 4 wires
- current on a wire signifies a 1 bit for that place; no current signifies 0



Abstract circuitry demonstrating that 0101 + 1100 = 10001

Half-adder Circuit

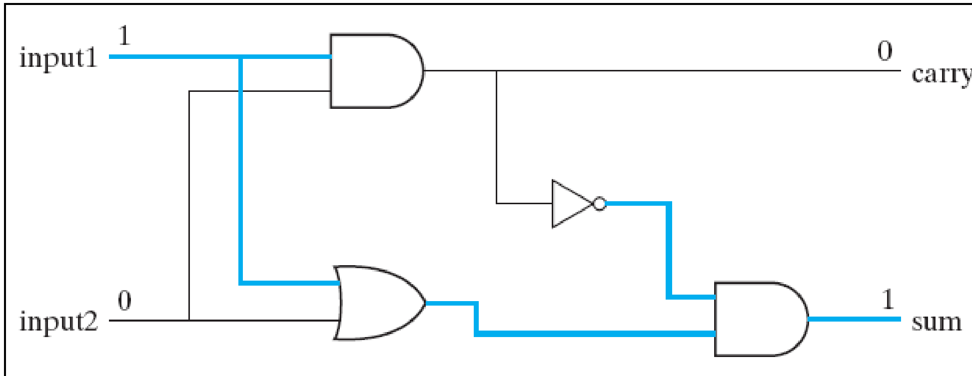
recall the rules of binary addition:

$$\begin{array}{r} {\scriptstyle 1\,1\;\;} \\ 1\,0\,1\,1_2 \\ +\quad\;\; 1_2 \\ \hline 1\,1\,0\,0_2 \end{array} \qquad\qquad \begin{array}{r} {\scriptstyle 1\,1\;\;\;\;\;\;\;\;} \\ 1\,1\,0\,0_2 \\ +\;\; 1\,0\,1_2 \\ \hline 1\,0\,0\,0\,1_2 \end{array}$$

although binary addition is relatively straightforward, designing a circuit for adding binary numbers is quite complex

- instead of starting at the transistor level, we can use AND, OR, and NOT gates

- focus first on the addition of 2 bits
  - requires two input lines, two output lines (sum of inputs and possible carry)
  - the circuit consist of four gates (known as a *half-adder*)

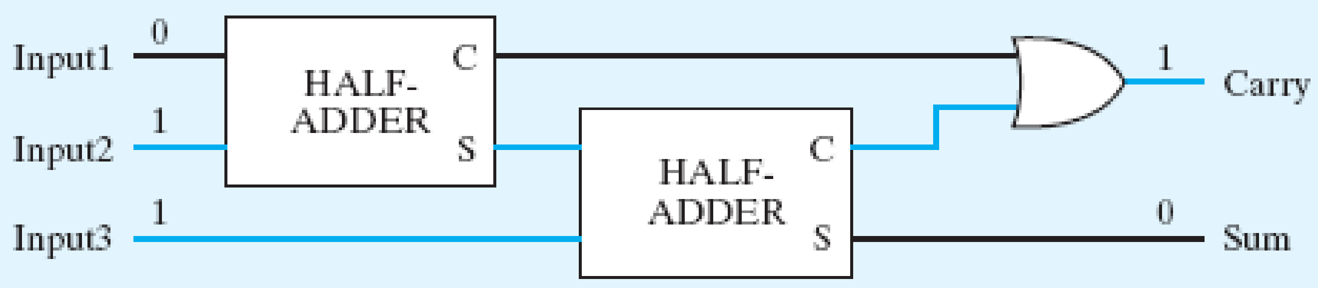


half-adder demonstrating that 1 + 0 = 01

Full-adder Circuit

the term “half-adder” refers to the fact that when you add binary numbers containing more than one bit, summing the corresponding bit pairs by column is only half the job

- you must also consider that a bit might be carried over from the previous addition
- using half-adders and logical gates as building blocks, we can design a circuit that takes this into account (known as a *full-adder*)

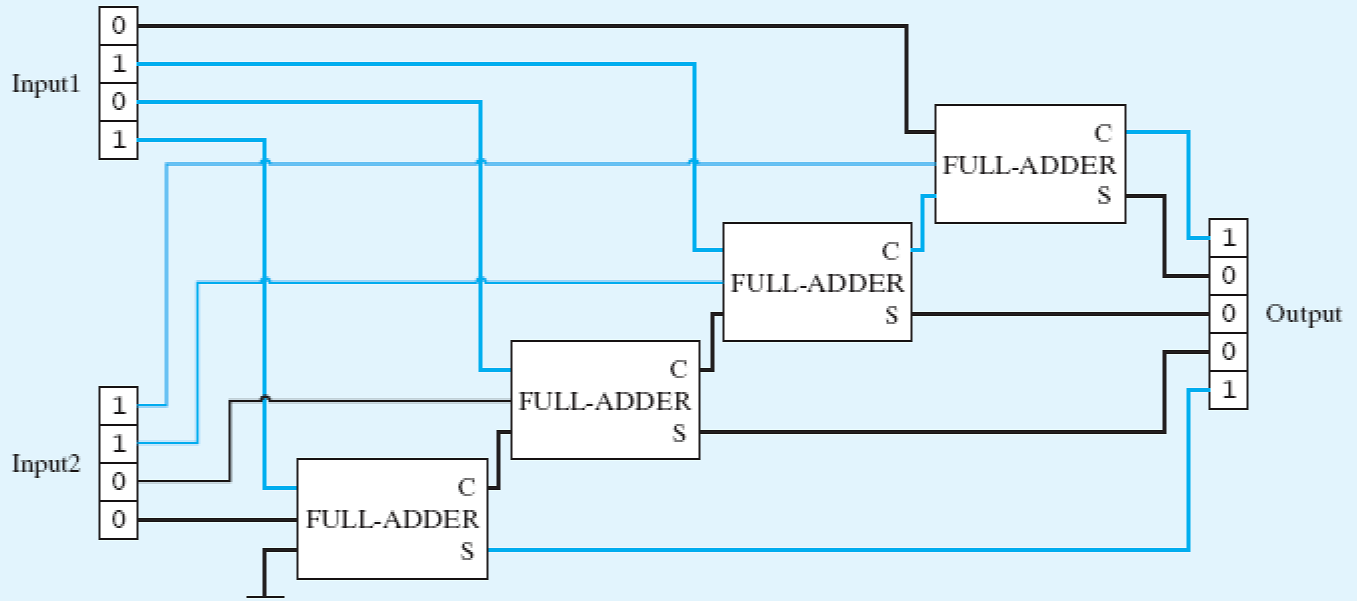


Full-adder demonstrating that 0 + 1 + 1 = 10

4-bit Adder Circuit

using full-adders as building blocks, we can design a more complex circuit that sums two 4-bit numbers

- since a full-adder is required to add each corresponding bit pair together (along with possible carry), the circuit will need four full-adders wired together

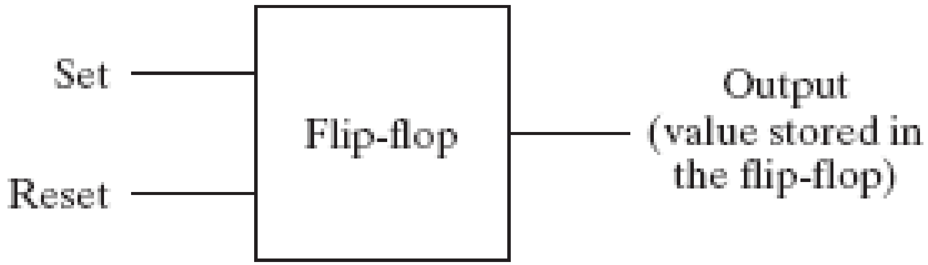


4-bit adder demonstrating that 0101 + 1100 = 10001

Designing Memory Circuitry

main memory and registers within the CPU are composed of circuitry

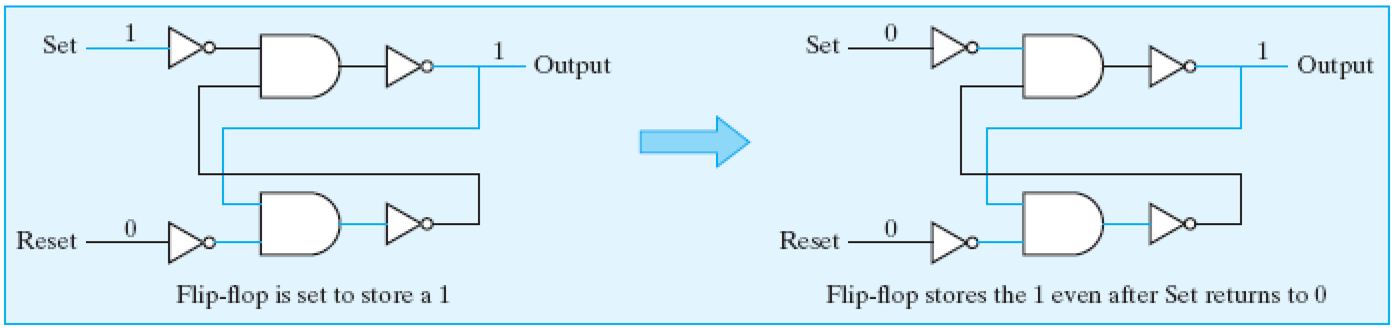
- whereas adders manipulate inputs to produce outputs, memory circuits must maintain values over time
- the simplest circuit for storing a value is known as a *flip-flop*
  - it can be set to store a 1 by applying current on an input wire
  - it can be reset to store a 0 by applying current on another input wire



Flip-flop Circuit

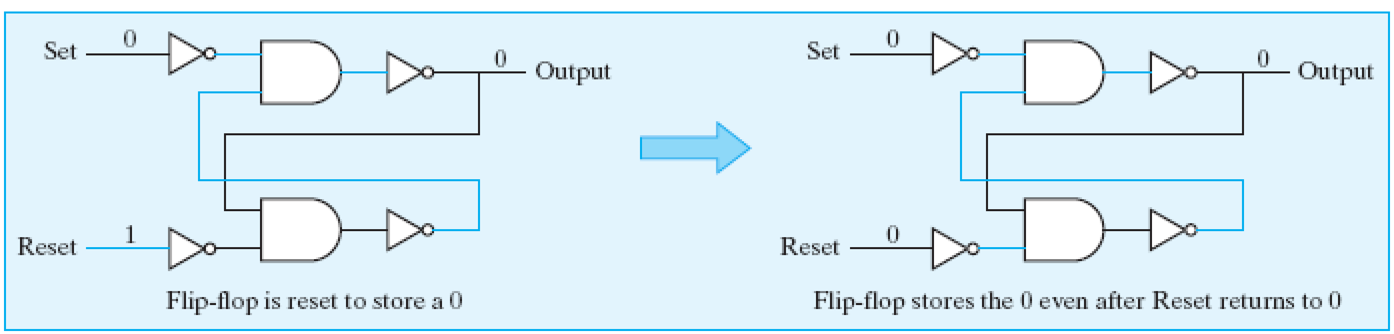
a flip-flop stores a value by feeding the output currents back into the circuit

- the value is maintained by current flowing around and around the circuit
- a current on the Set wire produces current on the output, which then cycles



Flip-flop is set to store a 1
Flip-flop stores the 1 even after Set returns to 0

- a current on the Reset wire produces no current on the output



Flip-flop is reset to store a 0
Flip-flop stores the 0 even after Reset returns to 0

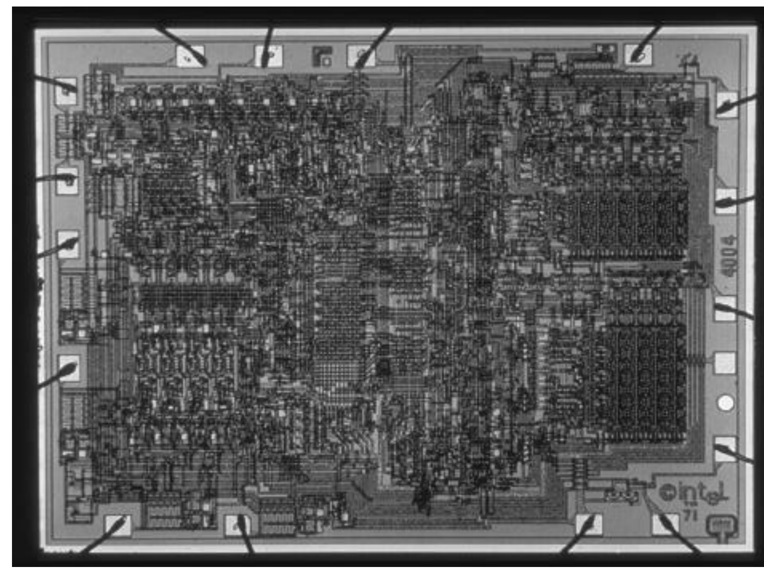
From Circuits to Microchips

initially, circuits were built by wiring together individual transistors

- this did not lend itself to mass production
- it also meant that even simple circuits consisting of tens or hundreds of transistors were quite large (to allow space for human hands)

in 1958, two researchers (Jack Kilby and Robert Noyce) independently developed techniques that allowed for the mass-production of circuitry

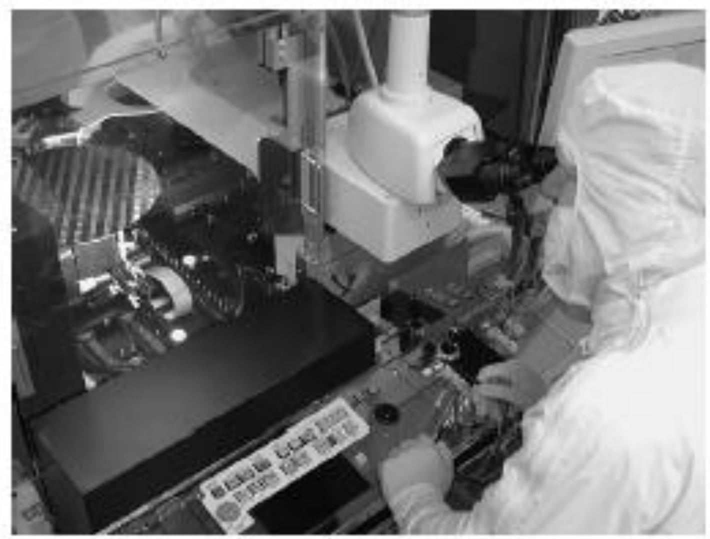
- circuitry (transistors + connections) is layered onto a single wafer of silicon, known as a *microchip*
- since every component is integrated onto the same microchip, these circuits became known as *integrated circuits*



Manufacturing ICs

the production of integrated circuits is one of the most complex engineering processes in the world

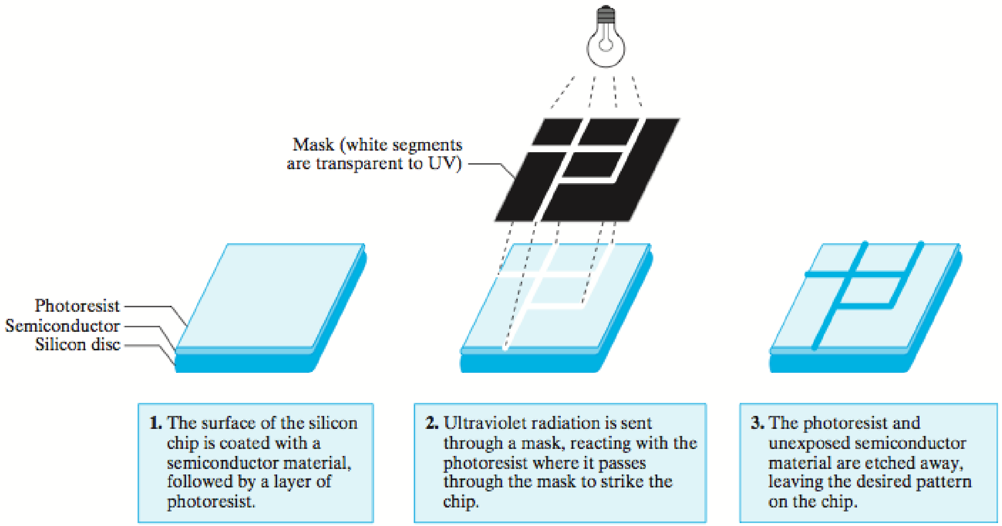
- transistors on chips can be as small as .065 microns (roughly $1/1{,}500^{th}$ the width of human hair)
- since a hair or dust particle can damage circuitry during manufacture, chips are created in climate-controlled "clean rooms"



Manufacturing ICs

to produce the incredibly small and precise circuitry on microchips, manufacturers use light-sensitive chemicals

- initially, the silicon chip is covered with a semiconductor material, then coated with a layer of photoresist (a chemical sensitive to UV light)
- transistors are then printed onto a mask (transparent surface on which an opaque coating has been applied to form patterns)
- UV light is filtered through the mask, passing through the transparent portions and striking the surface of the chip in the specified pattern
- the photoresist that is exposed to the UV light reacts, hardening the layer of the semiconductor below it
- the photoresist that was not exposed and the soft layer of semiconductor below are etched away, leaving only the desired pattern of semiconductor material on the surface of the chip
- the process can be repeated 20-30 times depositing multiple layers



Packaging Microchips

since a silicon chip is fragile, the chip is encased in plastic for protection

- metal pins are inserted on both sides of the packaging, facilitating easy connections to other microchips

impact of the microchip

- lower cost due to mass production
- faster operation speed due to the close proximity of circuits on chips
- simpler design/construction of computers using prepackaged components

Moore’s Law describes the remarkable evolution of manufacturing technology

- Moore noted that the number of transistors that can fit on a microchip doubles every 12 to 18 months
- this pattern has held true for the past 30 years
- industry analysts predict that it will continue to hold for the near future