15 The end of Moore's law

Will it be possible to remove the heat generated by tens of thousands of components in a single silicon chip?
Gordon Moore[1]

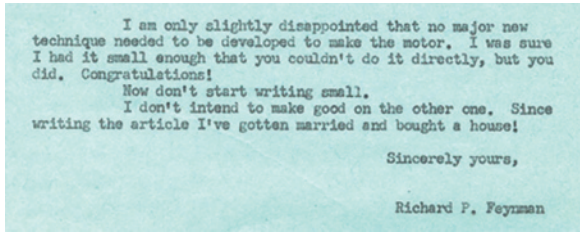
Nanotechnology

In 1959, at a meeting of the American Physical Society in Pasadena, California, physicist Richard Feynman set out a vision of the future in a remarkable after-dinner speech titled "There's Plenty of Room at the Bottom." The talk had the subtitle "An Invitation to Enter a New Field of Physics," and it marked the beginning of the field of research that is now known as *nanotechnology*. Nanotechnology is concerned with the manipulation of matter at the scale of nanometers. Atoms are typically a few tenths of a nanometer in size. Feynman emphasizes that such an endeavor does not need new physics:

> I am not inventing anti-gravity, which is possible someday only if the laws are not what we think. I am telling you what could be done if the laws *are* what we think; we are not doing it simply because we haven't yet gotten around to it.[2]

During his talk, Feynman challenged his audience by offering two $1,000 prizes: one "to the first guy who makes an operating electric motor which is only 1/64 inch cube," and the second prize "to the first guy who can take the information on the page of a book and put it on an area 1/25000 smaller."[3] He had to pay out on both prizes – the first less than a year later, to Bill McLellan, an electrical engineer and Caltech alumnus (Fig. 15.1). Feynman knew that McLellan was serious when he brought a microscope with him to show Feynman his miniature motor capable of generating a millionth of a horsepower. Although Feynman paid McLellan the prize money, the motor was a disappointment to him because it did not require any technical advances (Fig. 15.2). He had not made the challenge hard enough. In an updated version of his talk given twenty years later, Feynman speculated that, with modern technology, it should be possible to mass-produce motors that are 1/40 a side smaller than McLellan's original motor. To produce such micromachines, Feynman envisaged the creation of a chain of "slave" machines, each producing tools and machines at one-fourth of their own scale.

Fig. 15.1. Richard Feynman examining Bill McLellan's miniature electric motor in 1960. The motor could generate a millionth of a horsepower and Feynman paid McLellan the $1,000 prize money.


I am only slightly disappointed that no major new technique needed to be developed to make the motor. I was sure I had it small enough that you couldn't do it directly, but you did. Congratulations!

Now don't start writing small.

I don't intend to make good on the other one. Since writing the article I've gotten married and bought a house!

Sincerely yours,

Richard P. Feynman

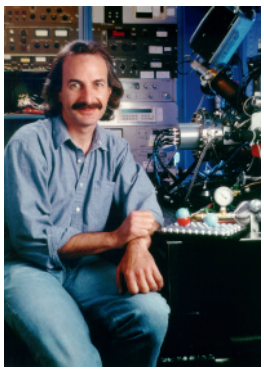
Fig. 15.2. Feynman's letter to McLellan expresses disappointment that McLellan did not need to develop any new technology to build his motor but instead had been able to use tweezers and a microscope.

Fig. 15.3. Stanford graduate student Tom Newman wrote the first page of *A Tale of Two Cities* by Charles Dickens using electron beam lithography to form letters only fifty atoms wide.

It was not until twenty-six years later, in 1985, that Feynman had to pay out on the second prize. The scale of the challenge is equivalent to writing the entire contents of *Encyclopædia Britannica* on the head of a pin (Fig. 15.3). The winner was Tom Newman, a Stanford graduate student who was using electron beam lithography to engrave patterns on silicon to make integrated circuits. A friend showed Newman a copy of Feynman's 1959 talk and pointed out the section offering a prize for "writing small." Newman calculated he would have to reduce individual letters down to a scale only fifty atoms wide. Using an electron beam machine, he thought it should be possible. To check that the prize was still being offered after all that time, Newman sent a telegram to Feynman. He was surprised to receive a telephone call from Feynman confirming that it was. Because Newman was supposed to be working on his thesis, he had to wait until his thesis adviser went to Washington, D.C., for a few days before he made his attempt. He programmed the machine to write the first page of Charles Dickens's novel *A Tale of Two Cities*. The major difficulty turned out to be actually finding the tiny page on the surface after it had been written. Newman duly received a check from Feynman in November 1985.

Researcher Don Eigler (B.15.1) and his colleagues at the IBM Almaden Research Center in California used the scanning tunneling microscope (STM), invented by their colleagues at IBM Zurich, to manipulate individual atoms and create the world's smallest IBM logo in 1989 (Fig. 15.4). They have also made spectacular quantum "corrals" (Fig. 15.5) and created "artificial" molecules, one atom at a time (Fig. 15.6), confirming another speculation of Feynman's:

> It would be, in principle, possible (I think) for a physicist to synthesize any chemical substance that the chemist writes down. Give the orders and the physicist synthesizes it. How? Put the atoms down where the chemist says, and so you make the substance.[4]



B.15.1. Don Eigler achieved many breakthroughs in nanotechnology in his laboratory at IBM's Almaden Research Center. His group produced the smallest IBM logo using an STM to position the individual atoms.

In 2012, IBM researchers announced they had used the same technique to store a single bit of information on a magnetic memory made of just twelve atoms. According to researcher Sebastian Loth, it currently takes about a million atoms to store a bit of information on a hard disk. Loth explains that:

> Roughly every two years hard drives become denser. The obvious question to ask is how long can we keep going. And the fundamental physical limit is the world of atoms. The approach that we used is to jump to the very end, check if we can store information in one atom, and if not one atom, how many do we need? We kept building larger structures until we emerged out

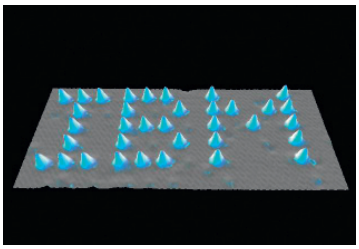


Fig. 15.4. IBM researchers Don Eigler and Erhard Schweizer spelled out the initials of the company in thirty-five individually positioned xenon atoms in 1989.

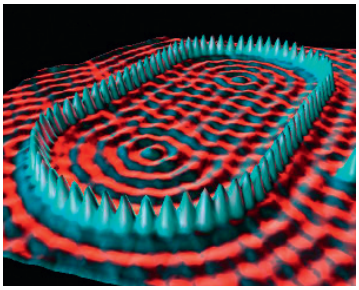


Fig. 15.5. A stadium-shaped "quantum corral" was built by positioning individual iron atoms on a copper surface.

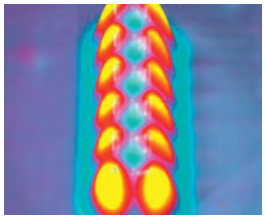


Fig. 15.6. In 2012, IBM researchers built a magnetic memory device consisting of just twelve atoms.

of the quantum mechanical into the classical data storage regime and we reached this limit at 12 atoms.[5]

The groups of atoms were arranged using an STM operating at very low temperatures. By scaling up these twelve-atom bits to a few hundred atoms, it may be possible to make such structures stable at room temperature. Clearly, however, volume production of such memory devices is many years away.

In his 1986 book *Engines of Creation*, the nanotechnology researcher Eric Drexler (B.15.2) envisages a future in which self-replicating nanomachines could be engineered that could create almost any type of matter (Fig. 15.7). In his vision of a nanotechnology-powered future, hunger would be eliminated, all diseases cured, and the human life span extended dramatically. Drexler uses the term *grey goo* to refer to an out-of-control, spreading mass of self-replicating machines that could literally cause the end of the world. Bill Joy, one of the founders of Sun Microsystems, became so concerned about the potentially catastrophic effects of Drexler's nanomachines that he warned against unregulated experimentation with nanotechnology in *Wired* magazine. Fortunately, while Drexler's book certainly excited many people about the future potential of nanotechnology, most scientists believe that we are a long way from actually creating any of Drexler's self-assembling machines.

The near future

As Gordon Moore acknowledged in 2005 (see Chapter 7), the size of transistors is "approaching the size of atoms which is a fundamental barrier"[6] for present-day technology. Each year, a group of semiconductor experts in the five leading chip-manufacturing regions in the world – the United States, Japan, Taiwan, South Korea, and Europe – prepare a report called the *International Technology Roadmap for Semiconductors* (ITRS), which identifies the challenges for the future of semiconductor chip manufacturing. In past years, the roadmap has laid out research and development targets necessary for the continuation of *geometrical scaling*, the continued reduction in size predicted by Moore's law. Now, however, the roadmap also includes *equivalent scaling*, improving performance through innovative design, software solutions, and new materials or structures. The 2012 version of the ITRS looks at both near-term goals, through

B.15.2. Eric Drexler is known for his theoretical work on molecular nanotechnology. He developed the concept of self-assemblers capable of constructing molecules atom by atom. This idea not only captured the imagination of science fiction writers but also created real research interest in this field. There are many skeptics of Drexler's ideas and the research has not demonstrated the possibility of building nanoscale self-assemblers.

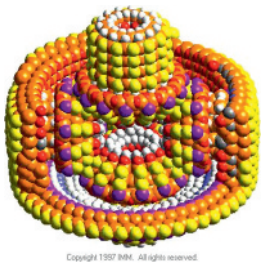


Fig. 15.7. Eric Dexler's vision of nanotechnology included fabricating such things as molecular differential gears.

2018, and long-term goals, 2019 through 2026. On the near-term goal, the ITRS comments:

> Scaling planar CMOS [complementary metal oxide silicon, the technology used to build integrated circuits] will face significant challenges. The conventional path of scaling, which was accomplished by reducing the gate dielectric thickness, reducing the gate length, and increasing the channel doping, might no longer meet the application requirements set by performance and power consumption. Introduction of new material systems as well as new device architectures, in addition to continuous process control improvement are needed to break the scaling barriers.[7]

On the longer-term outlook, the ITRS report highlights the problem of managing the power leakage of CMOS devices:

> While power consumption is an urgent challenge, its leakage or static component will become a major industry crisis in the long term, threatening the survival of CMOS technology itself, just as bipolar technology was threatened and eventually disposed of decades ago. Leakage power varies exponentially with key process parameters such as gate length, oxide thickness, and threshold voltage. This presents severe challenges in light of both technology scaling and variability. Off-currents in low-power devices increase by a factor of 10 per generation, and will emphasize a combination of drain and gate leakage components. Therefore design technology must be the key contributor to maintain constant or at least manageable static power.[8]

In May 2011, Intel announced the most radical shift in semiconductor technology in fifty years. The new Intel technology uses the latest fabrication process to produce three-dimensional transistors that allow microprocessors to operate faster and use less power than conventional two-dimensional transistors. According to Moore:

> For years we have seen limits to how small transistors can get. This change in the basic structure is a truly revolutionary approach, and one that should allow Moore's Law, and the historic pace of innovation, to continue.[9]

B.15.3. The FinFET transistor team at Berkeley. From left to right: Ali Javey, Vivek Subramanian, Ali Niknejad, Jeff Bokor, Chenming Hu, and Tsu-Jae King Liu.

Research that led to this breakthrough started in 1997, in a DARPA-funded project at the University of California, Berkeley. The Berkeley team (B.15.3), Chenming Hu, Jeff Bokor, and Tsu-Jae King Liu, looked at the challenge of building a transistor smaller than twenty-five nanometers, ten times smaller than those in production at the time. (A *nanometer* is a thousand-millionth of a meter.) Two years later, the researchers came up with the idea of a new three-dimensional transistor structure they called a "FinFET" (Fig. 15.8). This is a *field effect transistor* (FET) formed with a narrow silicon "fin" rising from the surface of the chip. A FET operates by creating an electric field that changes how one of the transistor's semiconductor regions, the gate region, conducts electric current. In a standard two-dimensional FET, the current can only be controlled from the top surface of a silicon channel linking the

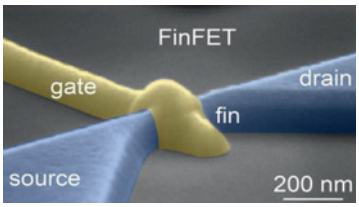


Fig. 15.8. Illustration of a three-dimensional FinFET transistor. Intel began manufacturing twenty-two-nanometer Tri-Gate transistors in 2012.

semiconductor regions. With a fin-shaped silicon channel, the flow of current can be controlled more effectively, using all of the channel's side surfaces. Hu explained the rationale for the fin structure as follows:

> An analogy is to think of this channel like a vein. If you want to stop bleeding, you would pinch the vein from both sides. This would be much better than just pressing from one side.[10]

In 2000, the Berkeley researchers predicted that FinFET technology could be scaled down to at least ten nanometers, and they estimated that such three-dimensional transistors could move into full-scale production in about ten years. Intel started volume production of its new twenty-two-nanometer, three-dimensional Tri-Gate transistors in 2012 with the announcement of the third-generation Intel Core processor family (formerly code-named Ivy Bridge). The new three-dimensional architecture allows for a 37 percent performance increase at low voltage and a 50 percent power reduction, compared to chips made using conventional two-dimensional technology.

What happens after 2020 or so? Physicist Michio Kaku (B.15.4) has predicted the end of the "Age of Silicon":

> But this process cannot go on forever. At some point, it will be physically impossible to etch transistors in this way that are the size of atoms. You can even calculate roughly when Moore's law will finally collapse: when you finally hit transistors the size of individual atoms. Around 2020 or soon afterward, Moore's law will gradually cease to hold true and Silicon Valley may slowly turn into a rust belt unless a replacement technology is found. Transistors will be so small that quantum theory or atomic physics takes over and electrons leak out of the wires. For example, the thinnest layer inside your computer will be about five atoms across. At that point, according to the laws of physics, the quantum theory takes over.… According to the laws of physics, eventually the Age of Silicon will come to a close, as we enter the Post-Silicon Era.[11]

To see what might happen after 2020, we now take a quick look at three possible postsilicon technologies.

B.15.4. Michio Kaku is an American theoretical physicist and popularizer of science. He has written papers on string theory and several popular science books. He has also hosted several TV programs about science. Kaku predicts that the end of silicon-based computing is near.

A postsilicon age?

The ITRS roadmap is looking toward incorporating nanotechnologies onto a CMOS silicon platform. One of the leading technologies will likely involve new allotropes of carbon. An *allotrope* is a specific structural arrangement of the atoms of an element in crystalline form: for carbon, the two most common allotropes are diamond and graphite. In diamond, each carbon atom uses its four outer electrons to bond with four other carbon atoms to form a tetrahedral structure that is extremely rigid (Fig. 15.9). This structure gives diamond its legendary strength and hardness. For any substance to conduct electric current, it must contain charged particles that can move freely through the material, such as electrons in the outer shell of an atom. In diamond, because all four of the outer electrons in each carbon atom are tied up in bonds between the atoms, the electrons cannot move around freely and so diamond cannot conduct electric current. In graphite, each carbon atom uses only three of its

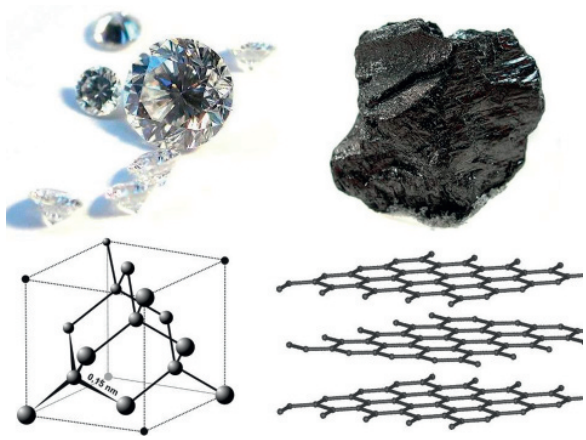


Fig. 15.9. Carbon takes on different forms depending on how its atoms are arranged. The atoms in a diamond form a rigid pyramid shape. In graphite, the atoms are arranged in flat layers.

four outer electrons to bond to three other carbon atoms, forming flat, parallel layers. Graphite consists of many of these layers of atoms, which can easily slide over each other, making graphite soft. In addition, one of the four outer electrons in each of the carbon atoms in these layers remains free to move and as a result, graphite is a very good conductor of electricity (Fig. 15.9).

Interest in new forms of carbon began more than twenty years ago, when researchers Robert Curl, Harry Kroto, and Richard Smalley (B.15.5) discovered a new, stable form of carbon in which sixty carbon atoms formed a closed spheroidal structure. The carbon atoms were connected in the shape of a soccer ball. Because of its similarity to inventor R. Buckminster Fuller's geodesic dome, the discoverers called this new allotrope of carbon *buckminsterfullerene*, soon shortened to *buckyball* in the popular press (Fig. 15.10). In fact, this carbon 60 allotrope was just the first of a whole new family of hollow carbon structures now known as *fullerenes*, which can take the form of spheres or tubes. In 1991, researcher Sumio Iijima in Japan observed threads of pure carbon that were only about a nanometer in diameter. The walls of these *carbon nanotubes* have the same atomic structures as graphite (B.15.6). The ends of the tubes can either be open or closed. The nanotubes can be up to several centimeters long and have extraordinary strength. IBM researchers have used nanotubes to

B.15.5. The team of Sean O'Brien, Richard Smalley, Robert Curl, Harold Kroto, and Jim Heath that discovered a new stable form of carbon the C_{60} in 1985. Smalley, Curley, and Kroto were awarded the 1996 Nobel Prize for chemistry.

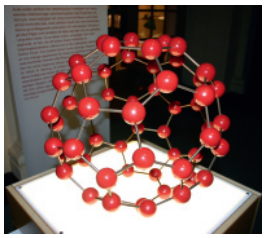
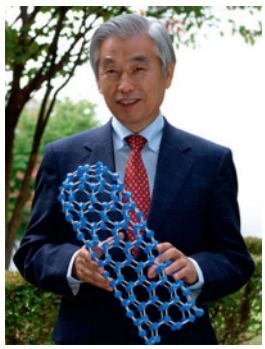


Fig. 15.10. The sixty atom carbon structure discovered by Robert Curl, Harry Kroto, and Richard Smalley. They called this new allotrope of carbon "buckminsterfullerene" in reference to the geodesic domes of American architect and inventor Buckminster Fuller. Inevitably this name is usually shortened to "buckyball."



B.15.6. Sumio Iijima, discoverer of carbon nanotubes, pictured with a model of a nanotube.

B.15.7. Russian physicists, Andre Geim and Konstantin Novoselov, working at the University of Manchester, England, received the 2010 Nobel Prize in Physics for their discovery of graphene.

make very small, fast transistors. At IBM's Thomas J. Watson Research Center, researchers have constructed an array of carbon nanotubes on the surface of a silicon wafer and used this silicon to build chips with more than ten thousand working transistors (Fig. 15.11).

In 2004, in Manchester, England, Andre Geim and Konstantin Novoselov (B.15.7) showed how to use graphite to produce a new form of carbon called *graphene*, which consists of an individual sheet of carbon atoms. A single sheet of graphene is just one atom thick and has some remarkable properties. It is the strongest two-dimensional material ever found, able to withstand stress two hundred times greater than steel without tearing apart. In addition, graphene conducts heat better than any metal, and electrons in the two-dimensional layer can move at speeds much faster than in silicon. Geim and Novoselov were awarded the 2010 Nobel Prize in physics "for groundbreaking experiments regarding the two-dimensional material graphene."[12] Researchers all around the world are looking at all sorts of applications of this new form of carbon – from lightweight, flexible display screens to new types of electronic circuits. In 2010, researchers at IBM used graphene to create transistors that can amplify signals about ten times faster than any silicon transistor.

Our last example of nanotechnology introduces a new type of electronic component. As long ago as 1971, Professor Leon Chua from the University of California, Berkeley, wrote a paper titled "Memristor – The Missing Circuit Element." In his paper, Chua argued that in addition to the familiar resistor, capacitor, and inductor there was a "missing" two-terminal circuit element. The name *memristor* is derived from *memory resistor*, because the component can change its resistance according to the current flowing but can also "remember" its final state when the voltage is switched off. A memristor is analogous to an unusual sort of pipe whose diameter can expand or shrink according to the amount of water flowing through it. In a memristor, if electrical charges flow in one direction, the resistance of the component will increase, and if charges flow in the opposite direction, the resistance will decrease. If the flow of charges stops, the component remembers the last resistance that it had, and when the flow starts again, the resistance of the circuit will be the same as it was when last active.

Stan Williams (B.15.8) and his colleagues at Hewlett Packard (HP) Labs in Palo Alto, California, have pioneered fabrication of nanoscale memristors (Fig. 15.12). The devices have advantages over conventional silicon-based memory in terms of access speed, power, and density, and can be fabricated using conventional silicon lithography techniques. HP's process to create an array of memristors consists of laying down a set of parallel "nanowires" – less than about ten nanometers wide – coated with a layer of titanium dioxide a few nanometers thick. A second set of wires is then laid down at right angles to the first set, and the crossover points of these wires are the memristors. Commercial versions of memristor memory chips will likely appear in the next few years, but it will be some time before such technologies present a significant challenge to *flash memory*, the durable, rewriteable memory chips used in digital cameras, smart phones, and other portable devices.

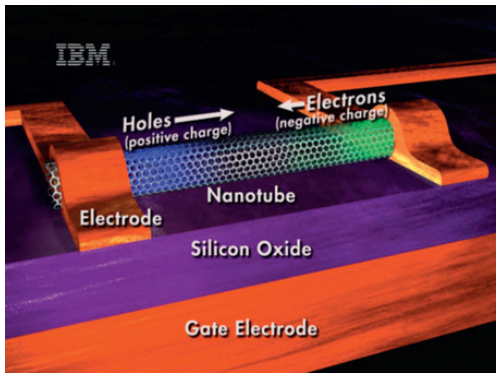


Fig. 15.11. An illustration of a carbon nanotube transistor. IBM has built chips with more than ten thousand nanotube transistors.

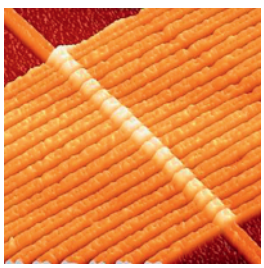


Fig. 15.12. New electronic components called *memristors* have the potential to transform the market for solid-state memory devices. A memristor has resistance to electrical current, but the resistance changes as the current changes. When the current is removed, the memristor preserves the memory of its last resistance. In this image, each of the white spots is a memristor only fifty nanometers in diameter.

Quantum computing

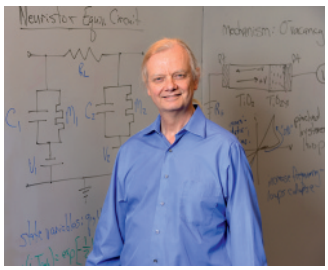
The study of the limits imposed by quantum mechanics on computers probably became respectable as an academic field after physicist Richard Feynman gave a keynote talk at a conference on the "Physics of Computation" at MIT in 1981. In his speech, Feynman talked about the problem of performing a computer simulation of physics:

> I'm not happy with all the analyses that go with just the classical theory, because Nature isn't classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.[13]

Feynman proposed building a computer out of elements that obey quantum mechanical laws:

> Can you do it [simulate quantum mechanics] with a new kind of computer – a quantum computer? … It's not a Turing machine, but a machine of a different kind.[14]

As we have seen, the basic principles of a Turing machine – that simple, theoretical computational device devised by Alan Turing in 1936 – underlie the operation of all conventional computers. Yet, as Feynman pointed out, a computer operating according to the laws of quantum mechanics would be a new kind



B.15.8. Stan Williams received a doctorate in physical chemistry from Berkeley. He is director of the Memristor Research Group at HP Labs in Palo Alto. In 2000, Williams was awarded the Feynman Prize in Nanotechnology.

B.15.9. David Deutsch developed the theory of a universal quantum Turing machine. In a famous paper published in1985, he argued that if a quantum computer could be built, it would have some remarkable properties due to quantum parallelism.

of computer, one that might be able to do calculations that conventional computers cannot do. Feynman was referring specifically to simulations of quantum systems to calculate quantum wave functions and quantum probabilities. After Feynman's lecture, David Deutsch (B.15.9), a physicist at the University of Oxford, took the next step. In 1985, Deutsch proved that a quantum computer could indeed do some calculations faster than a conventional computer. But it was not until 1994 that interest in quantum computing really exploded when Peter Shor (B.15.10) of Bell Laboratories discovered a *quantum algorithm* that could potentially solve certain mathematical problems much faster than the best algorithm running on a conventional computer.

What are the key elements of a quantum computer? First, we are only allowed to use quantum objects, like electrons or atoms, to input and store information, and to perform logical operations on this information. Quantum algorithms are executed using these fundamental logical operations. Finally, we need to be able to read out the answer to our quantum calculation. In his talk in 1981, Feynman speculated about the possibility of storing a single bit of information using the quantum states of a single electron. As we discussed in Chapter 7, electrons possess a property called *spin*. In quantum mechanics, an electron can exist in one of two possible spin states, which we call *spin up* ↑ and *spin down* ↓. To represent digital information, we can use the spin up state ↑ to represent a 1 and the spin down state ↓ to represent a 0. But this is not the whole story: in quantum mechanics, the electron can be in a *quantum superposition* of both these states. The electron's state is described by a *probability amplitude*. Using the traditional symbol ψ to represent the probability amplitude, this quantum superposition can be written:

$$\Psi = \alpha \uparrow + \beta \downarrow$$

B.15.10. Peter Shor received a PhD in applied mathematics from MIT in 1985. While working at Bell Labs he became famous for his quantum factorization algorithm that he discovered in 1994. Shor has been a professor at MIT since 2003.

where α and β are the amplitudes of the two possible spin states. What happens if we make a measurement of the spin of an electron in such a quantum superposition? According to standard quantum mechanics, we must observe the electron in either a spin up state or a spin down state, but for any given electron in the state ψ it is impossible to predict with certainty which spin state we will see. However, if we were to prepare an *ensemble* collection of many different electrons in exactly the same way, so that each of them is in the same state ψ, then quantum mechanics does make a definite prediction. If we make measurements of the spin state of all of the electrons in this collection, quantum mechanics predicts that we will obtain the spin up result with probability α^2 and the spin down result with probability β^2. The total probability to get any of the possible results must always add up to one, so the sum of these two probabilities must add up to one.

In some sense, we can say that the electron in superposition state ψ is in both spin states at the same time. So now we see that if we use an electron to represent digital information, in addition to being in one of the 1 and 0 states, the electron could also be in a superposition of both the 1 and 0 states with probabilities determined by α and β. After more than half a century studying the fundamentals of computation, physicists had discovered something new about information at the quantum level. Information stored in a quantum

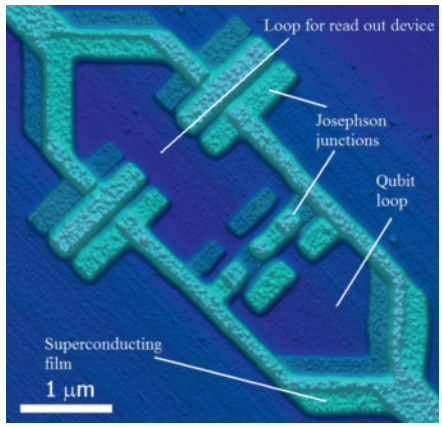


Fig. 15.13. A superconducting Josephson Junction qubit device made by researchers at Delft University of Technology.

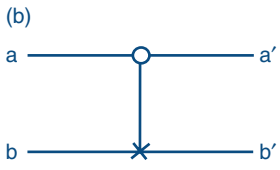
system therefore requires a new name, a *quantum bit* or *qubit* (Fig. 15.13). This superposition property of quantum states is one of the two key properties of quantum mechanics that give quantum computers their remarkable power. In a conventional computer, a bit can have a value of either 0 or 1. In a quantum computer, a qubit can also be in a quantum superposition and so can be both 0 and 1 at the same time. A system with two qubits can hold four values simultaneously – 00, 01, 10, and 11.

When MIT computer scientist Edward Fredkin (B.15.11) visited Feynman at Caltech in 1974, Fredkin was researching the seemingly strange problem of how to build a *reversible* computer. This is a type of computer that would be able to reverse calculations – "uncalculating" – as well as being able to calculate forward in the usual way. In conventional computers, logical operations are performed by logic gates implemented in silicon. The familiar "AND" gate is shown in Figure 2.8 with its two inputs and one output. All the possible inputs and outputs for an AND gate are summarized in the accompanying truth table. From this truth table, we see that an AND gate outputs a 1 only if both its inputs are 1; for the other three possible input combinations, the gate outputs a 0. The AND gate is therefore not reversible in the sense that it is impossible to deduce a unique input signal from just the output signal. Fredkin devised a new set of logic gates that are reversible – that is, gates such that the output signal from the gate uniquely determines the input signal. The simplest example of one of Fredkin's gates is the "Controlled NOT" or CNOT gate. This gate is shown in Figure 15.14 together with a conventional NOT gate and the corresponding truth tables. From the truth table for the CNOT gate, we see that the bottom input either "does nothing" or acts as a conventional NOT gate, reversing a 1 to a 0 and vice versa. Which action is chosen is determined by the signal on the upper input, which acts as a control line. If the upper input is a 0, the lower line does nothing. If it is a 1, the lower line acts as a NOT gate. Fredkin showed that it was possible to perform every logical operation using a complete set of such reversible gates (more than just the CNOT gate).

Why do we need to bother about reversible gates? Such gates are relevant for quantum computing because the laws of quantum mechanics are reversible in time. Reversibility is a property of conventional physical waves, not just

B.15.11. At the age of nineteen, Ed Fredkin left Caltech and joined the U.S. Air Force to serve as a fighter pilot. He became a professor at MIT in 1968 and was director of project MAC from 1971 to 1974. He was a close friend of Richard Feynman's and introduced him to the concept of reversible computing. Fredkin's research interests are wide-ranging and include the physics of computation and cellular automata.

a	NOT a
0	1
1	0



a	b	a′	b′
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Fig. 15.14. Edward Fredkin devised a set of logic gates that are reversible in the sense that the output signal from the gate fully determines the input signal. (a) A classical NOT gate and its truth table, and (b) a controlled NOT or CNOT gate and its truth table.

these strange quantum probability waves. A wave traveling in one direction along a string, for example, can just as easily travel in the reverse direction. This reversibility property of quantum mechanics means that if we wish to construct a quantum computer, we have to use computational elements that are reversible.

We can now write down the essential ingredients of a quantum computer. There must be a physical system in which information can be stored as qubits on individual quantum objects such as electrons, atoms, or photons. The information can be not only the familiar digital 1s and 0s but also quantum superpositions of 1 and 0. A quantum computer must have mechanisms by which these qubits can be made to interact so that we can perform Fredkin's reversible logic operations. Note that because we could choose to start off our quantum computer in a quantum superposition of all the possible initial states, in principle the quantum computer would calculate results for all the possible logical paths at the same time. David Deutsch, who first proved that quantum computers can be more powerful than conventional computers, called this property *quantum parallelism*. But how to exploit this property is not so obvious. According to standard quantum theory, making a measurement on a quantum superposition will result in only one of the possible states being selected, so how can quantum parallelism actually be useful? Shor's great contribution was to find a way to extract just a little information from all these quantum paths.

There is a second key feature of quantum mechanics that we must now explain, called *quantum entanglement*. Entanglement is a feature of certain types of two-particle quantum states that we can think of as having some invisible wiring to share information between the two particles (Fig. 15.15). We can illustrate the strange nature of entanglement by considering a thought experiment from particle physics. There is an unstable particle called a *neutral pion* that most of the time spontaneously decays into two photons (a photon being a particle-like bundle of light energy). On some occasions, however, the pion decays into an electron (e^-) and its antiparticle, a positron (e^+), instead of two photons. This is a rare occurrence for the pion, but it gives us the simplest experiment to illustrate what is meant by quantum entanglement. As in classical physics, *angular momentum* must be conserved in any quantum mechanical process. The

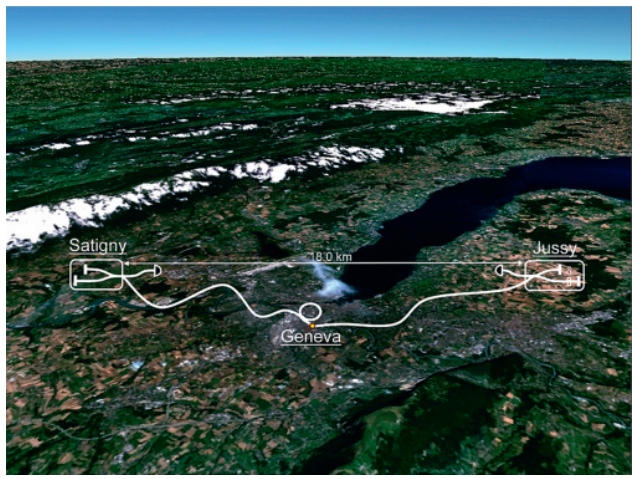


Fig. 15.15. Experiments with quantum entanglement were carried out using optical fibers running under Lake Geneva in Switzerland by Nicolas Gisin from the University of Geneva.

pion has zero spin, and because angular momentum must be the same before and after the decay, the spins of the electron-positron pair must be in opposite directions for conservation of angular momentum. If we also start with the pion sitting at rest, conservation of linear momentum dictates that the electron and positron must fly off in opposite directions (Fig. 15.16). If we just focus on the spin state of the two particles, there is probability ½ for the positron to be in the spin up state ↑ with the electron going in the opposite direction in the spin down state ↓. Similarly there is probability ½ for the positron to be in the spin down state ↓ and the electron in the spin up state ↑. What this means is that if we measure the spin of the positron to be spin up ↑ even though the particles may be widely separated in space, we know instantly that the spin of the electron traveling in the opposite direction must be spin down ↓. Similarly, if the positron is measured to be spin down ↓, we know instantly that the electron is spin up ↑. The spin information is shared – "entangled" – between the two particles.

It was the physicist Erwin Schrödinger who came up with the wave equation that determines how quantum probability waves evolve with time. Schrödinger was familiar with superposition and the physics of waves from classical physics. From the earliest days of quantum mechanics, he used the term *entangled* to describe such two-particle states and said of this entanglement property:

> I would not call that *one* but rather *the* characteristic trait of quantum mechanics, the one that enforces its entire departure from classical lines of thought. By the interaction the two representatives (or ψ-functions) have become entangled.[15]

We can now do experiments to verify these spin measurement predictions in situations where the information about the measurement of the first spin could not have influenced the second measurement on its separated partner – unless the information traveled faster than the speed of light. Albert Einstein

greatly disliked what he called the "spooky action at a distance"[16] effect needed to explain these surprising quantum spin correlations.

Because entanglement is entirely nonclassical, it may not be surprising that a quantum computer acting on entangled states can lead to results beyond the power of a classical computer. We can easily see how such entangled states can arise in quantum computation. Consider the action of a quantum CNOT gate on a two-qubit state (see box on Quantum Entanglement). When the two-qubit states are just simple products of single-particle 1 and 0 states, we obtain the exact analog of the classical result. But if one of the qubits is in a superposition state of 1 and 0, acting on this state with a quantum CNOT gate yields an entangled two-qubit state just like the example of the pion decaying into an electron and positron. It is this nonclassical feature of quantum mechanics that gives quantum computers their extraordinary properties.

Quantum entanglement

For a pion decaying at rest to a positron-electron pair (e^+e^-), the positron and the electron move away from each other in opposite directions as shown in Fig. 15.16 (a). Since the pion has zero spin, the net spin of the positron-electron pair must also be zero because of conservation of angular momentum. However, the spin state of either the positron or the electron is not definitely known and the spin state of the pair is said to be entangled. The entangled wave function for the pair is shown in Fig. 15.16 (b). If we measure the positron spin to be $\uparrow_{e^+}$ then we know immediately that the spin of the electron must be $\downarrow_{e^-}$ and vice versa. Since the positron and the electron are moving apart, this quantum correlation of spin measurements can be over long distances.

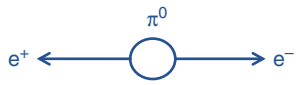


Fig. 15.16. Addition – Pion decay: $\pi^o \rightarrow e^+e^-$

a) Pion decaying at rest to a positron-electron pair (e^+e^-).

$$\Psi_{e^+e^-} \sim \left(\uparrow_{e^+}\downarrow_{e^-} - \downarrow_{e^+}\uparrow_{e^-}\right)$$

b) Entangled spin state of the positron-electron pair resulting from the pion decay.

Quantum computation often involves entangled states. These can arise from the action of a quantum CNOT gate on a two qubit state. The equations a, b, and c below show the action of a CNOT gate on three different two qubit states. With qubit input 1 on the upper control line of the gate, a qubit input 0 to the bottom line is flipped to a 1 as shown in (a). With qubit input 0 to the upper control line, a qubit input 0 to the bottom line is left unchanged as in (b). However, if the qubitinput on the control line is a superposition (1 + 0) the action of the CNOT gate on a 0 qubit input to the bottom line produces the entangled state shown in (c).

For cases (a) and (b) the action is straightforward and each particle is in a definite spin state before and after the CNOT gate. Acting with a CNOT gate on a two qubit superposition input state on the upper control line produces an entangled state in which neither particle is in a definite spin state as shown in (c).

$$\text{(a)}\ \ 1_1O_2 \xrightarrow{CNOT} 1_11_2$$
$$\text{(b)}\ \ O_1O_2 \xrightarrow{CNOT} O_1O_2$$
$$\text{(c)}\ \ (1_1 + O_1)\,O_2 \xrightarrow{CNOT} (1_11_2 + O_1O_2)$$

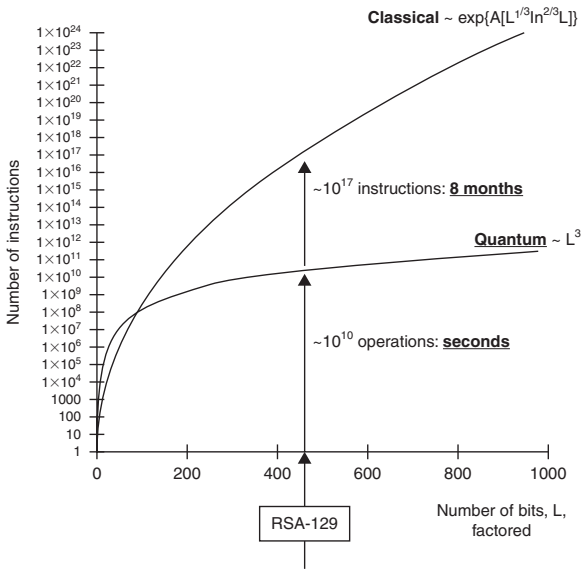


Fig. 15.17. Factorizing RSA-129. This graph shows the increase in computing power, measured in numbers of computer instructions, required to factorize larger and larger numbers, measured in numbers of bits. For a classical computer, the required power grows exponentially with the number of bits in the number to be factorized. The importance of Peter Shor's quantum algorithm was that it showed that with a quantum computer, the required power grows only as the cube of the number of bits. Also shown is the 129-digit number RSA-129 that was factorized in 1994 by volunteers using about 1,600 computers over several months. A quantum computer operating at the same speed as just one of these machines could factorize the number in only a few seconds.

Conventional computers are very good at multiplying two numbers together. For example, the time taken to multiply two N digit numbers grows as the square of N. By contrast, the time needed to *factorize* an N-digit number – that is, to resolve the number into two smaller numbers that when multiplied together form the larger number – grows faster than any power of N. This is an example of a *one-way function*, as explained in our discussion of public-key cryptography in Chapter 12. A one-way function is a mathematical problem that is easy to solve in one direction, but difficult or even impossible to solve in the other. For example, it is easy to multiply together two large *prime numbers* (numbers divisible only by themselves and 1). However, if you give the huge number resulting from that multiplication to someone else and ask him or her to tell you what numbers you started with, this problem is very hard. Shor showed that a quantum computer could, in principle, factorize numbers just as easily as it multiplied them, without the computing time increasing unreasonably as the size of the number to be factorized grows. This ability is astonishingly powerful. As we have seen, the whole basis of the RSA cryptosystem – named for its inventors, the computer scientists Ronald Rivest, Adi Shamir, and Leonard Adleman – is the computational difficulty of factorizing large numbers. For example, in 1994, the 129-digit number known as RSA-129 required eight months to factorize, using more than 1,600 computers (Fig. 15.17). If we could build a quantum computer that was roughly the same speed as just one of the computers used in this trial, Shor's algorithm could factorize RSA-129 in less than ten seconds. For this reason alone, many government agencies around the world are now funding attempts to build a quantum computer.

The computer scientist Lov Grover discovered another interesting class of algorithms in 1997. Grover's quantum search algorithm showed that a quantum computer could greatly increase the speed of searching a database. An example would be trying to find the name of a person in a telephone directory if you only know their telephone number. For a database with N items, Grover's

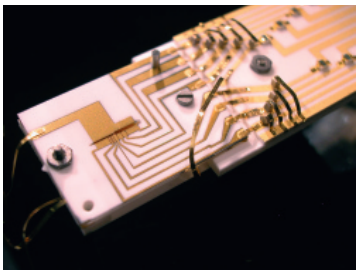


Fig. 15.18. Physicists at NIST in Boulder, Colorado, have demonstrated sustained, reliable quantum information processing in the ion trap at the left center of this photograph. The ions are trapped inside the dark slit – 3.5 millimeters long and 200 microns wide – between the gold-covered alumina wafers. By changing the voltages applied to each of the gold electrodes, scientists can move the ions between six zones of the trap.

algorithm reduces the number of steps needed to find the answer from N to the square root of N. So, for a database with a million entries, a quantum computer could find the correct entry in only one thousand steps.

So how much progress has been made toward actually building a quantum computer? It is a fast-moving field, and many groups around the world are exploring different ways to store and manipulate qubits. In 1995 Ignacio Cirac and Peter Zoller from the University of Innsbruck showed how the energy levels of *trapped ions* could be used to store qubits and how a quantum CNOT gate could operate on these qubits. In ion traps, ions (electrically charged atoms) are confined by an arrangement of electric fields so that the ions are kept suspended in space. The whole system needs to be in an almost complete vacuum, and the ions must be cooled to near absolute zero to remove their vibrational energy. The ions then arrange themselves in a linear array. After Cirac and Zoller's paper, Nobel Prize recipient David Wineland's (B.15.12) team at the National Institute of Standards and Technology (NIST) became the first to demonstrate quantum logic operations on qubits stored on trapped ions. Two energy levels of the ion are used as the qubit states, which are prepared and measured by directing laser beams at specific ions. Coupling between the ions is provided by the vibrational states of the ions in the ion trap. Using these techniques, the researchers were able to isolate systems containing a few qubits and to construct a quantum gate. More recently, Wineland's group stored qubits using two beryllium ions that can be moved between different zones of the ion trap by applying electric fields (Fig. 15.18). They were able to initialize and store the qubits on the ions in any desired starting state and then perform logic operations on the qubits. They were also able to transfer quantum information between the different zones in the trap. Using these techniques, Wineland's team successfully performed a sequence of four single-qubit operations, one two-qubit operation, and ten transport operations (Fig. 15.19). To scale beyond ten to one hundred trapped ion qubits, Wineland and his group have proposed using what they call a *quantum charge coupled device* (QCCD) (Fig. 15.20). Its operation will require very precise control of the ion positions as they are shuttled from region to region. Wineland's team notes that "scaling to thousands or more qubits in the QCCD may be challenging."[17]

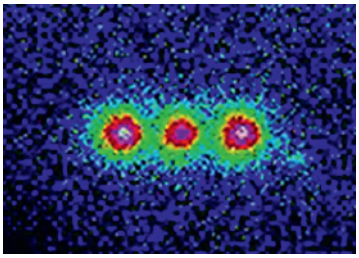


Fig. 15.19. David Wineland's research group at NIST designed and built this trap to confine three magnesium ions. A team of researchers in Innsbruck, Austria, have now been able to store an array of fourteen entangled qubits in an ion trap.

Although operation of such complex ion-trap systems is still very delicate, ion-trap technology does allow the use of simple quantum algorithms. But how close are we to creating a quantum computer that could factorize a number with hundreds of qubits? To factorize RSA-129 with 426 bits, we would need to build a quantum computer with close to a thousand qubits of memory that can execute about a billion quantum gate operations. There are other problems for would-be builders of quantum computers. Conventional computer memories suffer from the problem that individual bits can occasionally get "flipped." Cosmic rays, for example, are one cause of such errors. To counter this problem, the computer industry has developed a wide range of error detection and correction techniques. A simple example is a *parity check* in which the 1s and 0s are added before and after sending a message. If a 1 has been corrupted to a 0, or vice versa, a parity check will reveal the error. Computer engineers have devised more complicated techniques to handle situations where more than one error has occurred and also ways of detecting which bit has flipped and then correcting it. For qubits, we have all these problems and more. Not only

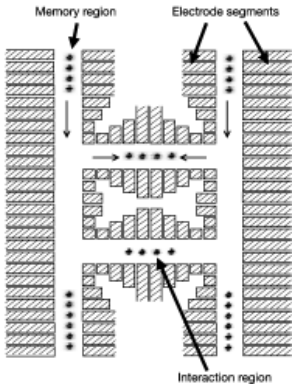
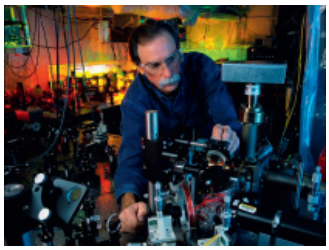


Fig. 15.20. David Wineland and his team at NIST propose using a quantum charge-coupled device for scaling up ion trap qubits.



B.15.12. Nobel Prize recipient David Wineland adjusts a laser beam used to manipulate ions in a low-temperature, high-vacuum ion trap. Wineland's group at the NIST laboratory in Boulder, Colorado, have demonstrated all the key elements required to build a quantum computer.

can we have random bit flips, but also the phase relationship between the different states in a quantum superposition can be affected by interactions with the surrounding environment. Surprisingly, it turns out to be possible in principle to detect and correct such quantum errors. Andrew Steane from Oxford and Peter Shor at Bell Labs independently devised schemes that use quantum entanglement to protect and correct quantum data. Such quantum error-correction techniques will require an order of magnitude more qubits, and it remains to be seen whether such methods will be feasible in practice.

Ion traps are only one technology that researchers are investigating for developing a quantum computer. Several research groups are investigating building qubit systems using a *Josephson junction*, an insulating barrier separating two superconducting materials. Superconducting electron pairs can travel through the barrier by tunneling. Other groups are working to manipulate spins on electrons bound to atoms embedded in a silicon chip. Another exciting approach is exploring the possibility of *topological quantum computing*. The Russian-born physicist Alexei Kitaev first suggested looking for quantum systems with a *topological excitation*. We can illustrate the idea of a topological excitation in terms of the vibrations of an elastic band stretched between two points and anchored at both ends. A topological excitation is analogous to vibrations of a band with a twist. There is no way for the band to untwist itself. Similarly, information stored in a topological qubit would be automatically protected against errors caused by interactions with its surroundings, which eliminates the need for quantum error correction. Research into such solid-state systems is still at an early stage but topological quantum computers may be the best bet to deliver quantum computers that can handle large numbers of qubits.

Synthetic biology and DNA computing

The intersection of computing, nanotechnology, and biology is an exciting area of research (B.15.13 and B.15.14). The research field of *synthetic biology* is attempting to produce standard biological components using the principles of computer science and engineering. Tom Knight (B.15.15), an MIT researcher who studies the intersection between computing and biology, says:

B.15.13. Nadrian "Ned" Seeman is a professor at New York University and one of the founders of structural DNA nanotechnology. He studied biochemistry and crystallography and since the 1980s he has been researching the structural properties of DNA molecules. In 1991 he managed to construct a cube from DNA molecules by orienting them in an electric field. In 1995 he was awarded the Feynman Prize in Nanotechnology. Together with Don Eigler from IBM he won the Kavli Prize in Nanoscience in 2010 "for their development of unprecedented methods to control matter on the nanoscale."[B1]

B.15.14. Randy Rettberg studied electrical engineering and computing. In the 1990s as a major career change, he decided to quit his job with Sun Microsystems and apply his engineering knowledge to molecular biology. Rettberg is president of the International Genetically Engineered Machine Foundation. The organization runs a global competition for undergraduates and high school students in designing brand new biological parts for "genetically engineered machines."

> The key ideas of modern engineering – modularity, modeling, hierarchical design, isolation of concerns, abstraction, reusable parts, defined interfaces, design rules, flexibility – promise to be just as applicable to biological systems as they are to computers or aircraft.…
>
> But the real challenge is learning to engineer with unique characteristics of biological systems: their self-reproducing capability, their evolutionary capacity to adapt, and their remarkable robustness in the face of damage and imperfect or failing components. These organizational engineering principles will play an important role not just in engineering biological systems, but in engineering in our existing disciplines.[18]

One key goal of synthetic biology is to produce a catalog of standard biological devices that biological engineers can put together to design new life systems. We will look at another avenue of research based on DNA sequences.

Engineering with DNA is an extreme example of molecular nanotechnology. Humans have about one hundred trillion cells, and most human cells are between one and one hundred micrometers in diameter. Each cell contains a nucleus, where most of our genetic material is stored as DNA. Inside the nucleus, the DNA is organized in linear molecules called *chromosomes*. The genetic information in DNA is stored as a code consisting of four nitrogen-containing compounds called *bases*: adenine (A), guanine (G), cytosine (C), and thymine (T). Sequences of these bases determine the genetic instructions for maintaining and replicating cells. Because every base must be one of these four types, each base encodes two bits of information. There about 3.5 billion bases in human DNA, so the entire human genome, a complete set of all our genetic instructions, corresponds to about seven billion bits of information or less than a gigabyte to store the whole human genetic code. The bases pair up in a specific way with each other – A with T and C with G – to form what are known as *base pairs*. The DNA molecule is shaped like a twisted ladder, a structure called a double helix. Each rung of the ladder consists of a base pair. The base pairs are attached to the sides of the ladder, which consist of sugar and phosphate molecules (Fig. 15.21). This is the famous double helix of Francis Crick and James Watson.

An important property of DNA is that it can be exactly copied so that the cell can divide into two new cells, each with an exact copy of the original DNA. The fundamental unit of heredity for individuals is a gene, a sequence of DNA that provides instructions for making a specific protein. These gene sequences range from a few hundred to more than two million base pairs in length, and

B.15.15. Tom Knight studied electrical engineering at MIT and worked on the ARPANET in the 1960s and 1970s. He was a graduate student in the Artificial Intelligence Lab at MIT and received a doctorate in integrated circuit design in 1983. In the 1990s he became interested in biology and started working with simple bacteria called mycoplasmas. By modifying the DNA, Knight managed to assemble a synthetic bacterial cell. From his research he developed the concept of BioBricks – standard sections of DNA that can be joined together in different ways to create organisms that can perform some specific functions. There are now more than ten thousand parts in the BioBricks registry.

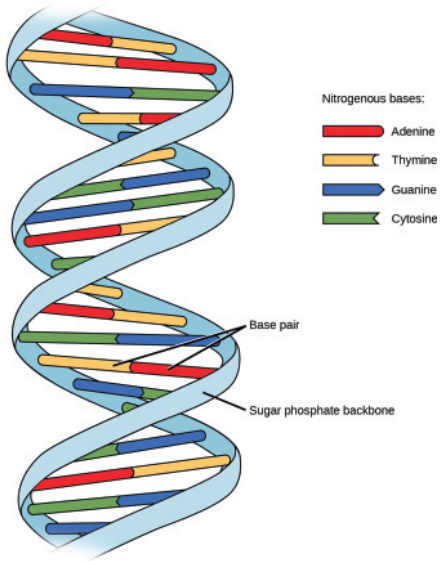


Fig. 15.21. Illustration of the double helix of the DNA molecule.

Fig. 15.22. Researchers at the University of California, San Diego, genetically engineered bacteria to glow and blink in unison, acting as a tiny "neon" sign.

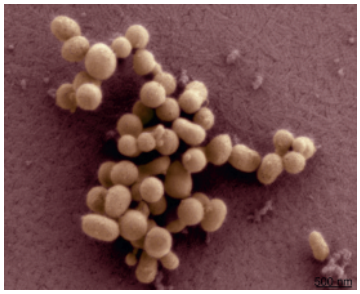


Fig. 15.23. The first self-replicating synthetic cells from the J. Craig Venter Institute.

a chromosome contains many such genes. The Human Genome Project estimated that humans have between twenty thousand and twenty-five thousand genes stored in twenty-three pairs of chromosomes. Thus a cell stores a gigabyte of genetic information in a volume as small as about a millionth of a cubic millimeter (Fig. 15.22).

Understanding the genetic basis of cells means that it is possible in principle to manufacture a DNA sequence to order and produce a synthetic version of a cell's genome. In 2010, researchers at the J. Craig Venter Institute in Rockville, Maryland, announced the creation of the first synthetic life form, a self-replicating bacterial cell (Fig. 15.23). The team synthesized the genome of an existing bacterium consisting of more than one million base pairs and inserted this synthetic genome into a different bacterium with its own genome removed. The new genome took over the cell's machinery, changing the cell's appearance and behavior, and the modified cell was able to divide and multiply. To prove that the cell's genome was artificially manufactured, the Venter group inserted four markers into the DNA sequence. The markers included the names of the researchers; a paraphrased quotation from Richard Feynman, "What I cannot create, I do not understand" (words found on his blackboard after he died); and a message congratulating the decoder.

The first use of DNA in computing was an experiment performed by Len Adleman, whom we met earlier in the discussion of the RSA encryption scheme. Adleman invented a way of using single-stranded DNA – one side of a DNA ladder with its sequence of bases not paired with a partner DNA strand – to solve a puzzle called the *seven-city directed Hamiltonian path problem*, a variation of the traveling salesman problem we discussed in Chapter 5. To solve a seven-city directed Hamiltonian path problem, you must find the shortest route between seven cities, beginning at one designated city and ending at another, passing through each of the other five cities exactly once. In Adleman's experiment, a single strand of DNA represented each city, with a corresponding unique

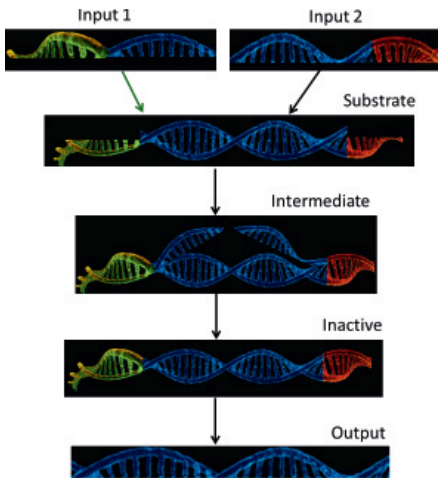


Fig. 15.24. A DNA logic gate. This gate takes two DNA strands as input and only produces an output if both input strands are present.

sequence of A, T, C, and G. All the possible paths were represented by complementary DNA sequences consisting of the last half of a strand corresponding to a departure city and the first half of another strand representing a possible arrival city. Adleman mixed the DNA strands together and all the possible paths were generated, created by the complementary A-T and C-G bonding between the strands. He then had to perform a manual analysis to separate out molecules representing paths that did not start or end with the right city or paths that did not go through all the different cities. This DNA-based computation produced all possible paths very quickly due to the large number of molecules involved, but the manual analysis to separate out the strands for valid paths took several days. So although Adleman's work was an interesting experimental approach to computing with DNA, it was not a practical method for solving large-scale problems in a reasonable time.

An alternative direction for DNA computing has been to focus on creating more general-purpose computational circuits using both single-stranded and double-stranded DNA. The technique is called *strand displacement*, and the interactions are specified by the choice of complementary DNA sequences. A strand displacement reaction is initiated when a short portion of an incoming single strand binds to a complementary exposed portion of a double-stranded complex. If the remaining sequence of the incoming strand matches the sequence of a neighboring strand in the complex, the incoming strand will displace the existing strand through strand displacement. In this way, researchers have been able to create logic gates from DNA (see Fig. 15.24). Several research groups are experimenting to scale up such strand displacement gates to perform complex computations. The ultimate goal of this research is to make programming DNA circuits as straightforward as programming computers.

Key concepts

- Nanotechnology
- Memristor

- 3D transistor
- Carbon nanotubes
- Quantum computer
- Quantum entanglement
- DNA computing
- Strand displacement