

Understanding Computers, Smartphones, and the Internet

Ernie Dainow

Who is this book for?

Most introductory books about computers are either long, detailed technical books such as those used in a computer science course or else tutorials that provide instructions on how to operate a computer with little description of what happens inside the machine.

This book fits in the large gap between these two extremes. It is for people who would like to understand how computers work without having to learn a lot of technical details. Only the most fundamental things about computers are covered. There is no math except some simple arithmetic. The only prerequisite is knowing how to use a web browser.

You can also get videos for each chapter in the book by going to [youtube.com](https://www.youtube.com) and searching for “Understanding Computers, Smartphones and the Internet”.

Only current day technology is covered. People who are interested in learning about how computers evolved from the earliest machines can read the book “A Concise History of Computers, Smartphones and the Internet”.

While originally intended for people who are not in the computer field, this book is also useful as adjunct reading for those taking a coding course or an introductory computer course. Even people already in the computer field will find things of interest in this book.

Ernie Dainow
edainow@gmail.com
February 2017

All rights reserved © 2016 Ernest Dainow

ISBN: 978-0-9952144-0-8

Contents

1. What is a Computer?

Start with a problem
Write a program for a solution
Machine Code
Machine Instruction Set
Integrated Circuits
Moore's Law
The Future

2. How Does Software Work?

Computer Languages
Databases
The Layers of Software
Software Development
Software Applications

3. How Does the Internet Work?

What is the Web and how does it work?
HTML (Hyper Text Markup Language)
Other Protocols
What is the Domain Name System (DNS)?
How is the Internet managed?

4. How Do Smartphones Work?

Smartphone Hardware
Smartphone Software
The Cellular Phone Network
How does radio work?
How does a smartphone connect to the cellular network?

Appendix 1. CalculateTax Program

Appendix 2. CalculateTax Machine Code

Appendix 3. Binary Numbers, Bits and Bytes

Appendix 4. Machine Instruction Sets

Appendix 5. Internet Routing

Appendix 6. How Does Email Work?

About the Author

1. What is a Computer?

In the context of this book, the word “computer” can refer to many different sorts of devices. Familiar computers are laptops, desktops and the big mainframe computers used in large organizations. But computers are found in many other places, such as smartphones, tablets, video game consoles, telephone networks, automobiles, medical equipment, television broadcasting and factories (to name a few). All these computers are fundamentally the same. They are just different sizes and use specialized hardware and software. The description of computers in this book applies to all of these devices.

Basic computer architecture

This diagram is a high level view of a computer that shows the main components.

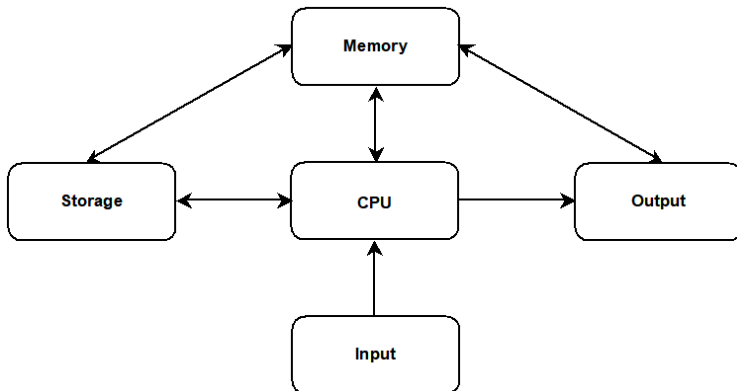


Figure 1-1. Basic computer architecture

Hardware refers to all the physical components in a computer – the CPU (Central Processing Unit), memory chips, disk storage, and various input and output devices.

Software refers to programs that are run on the computer. They are “soft” because they can be changed without having to modify any of

the hardware, which is much more time consuming and expensive.

A special software component is the **Operating System (OS)** that controls the computer hardware. A computer generally can't do anything without an OS. When you power on a computer, one of the first things it does is copy the OS from files in Storage into Memory so that they can run in the CPU. Some of the familiar operating systems in use today are Microsoft Windows for PCs, Linux for servers, and Apple iOS or Google Android for Smartphones. Once the OS is running, you can then run applications (usually called apps on smartphones), such as a web browser, a text messaging app or a game.

Computer hardware is built by assembling electronic components onto one or more **printed circuit boards**. The boards have sockets for the components and wired lines that make the necessary electrical connections between them. There are small basic components such as diodes, resistors, capacitors and transistors as well as larger silicon chips, such as memory chips and a CPU chip (marked Amlogic).



Figure 1-2. Printed Circuit Board

CPU is the Central Processing Unit. This is the brain of the computer. It controls the operations in all the other components and it makes all the decisions. Many computers use an Intel processor chip for the

CPU. Smartphones and tablets use a variety of smaller CPU chips.

The CPU generally needs two things: a program (code) and data.

1. A program, or application, is the list of instructions that tells the computer what to do.
2. Data is information that a program processes, typically information provided from an Input device or from a file in Storage.

All computations done by a computer can only be done on data that is in a “register” in the CPU. These registers are very expensive to build so even large computers do not have very many.

Input includes such familiar devices as a keyboard, mouse or a touchscreen.

Output is where the computer’s results are sent. On a PC or mobile phone, this is typically your display screen. But it can also be a printer or a network, like the Internet. A network is also an input device.

Storage is where programs and data are saved. Data includes documents created by different programs, music files, video files and system files. Storage is typically a hard disk drive but on Smartphones it is a flash drive. On most computers you can manage Storage and create folders, copy files from one folder to another and delete folders and files. Most Smartphones do not provide access to Storage by default. You need to download a third party app to manage Storage.

Memory is a high speed storage area. But there are some important differences between Memory and Storage. On a typical midrange laptop you might have 8 GB of Memory (8 billion bytes) but 1 TB of disk storage (1 trillion bytes).

Here’s why.

The CPU needs to get the program instructions one at a time. The CPU operates very quickly, so it needs to be able to copy program instructions into its registers as quickly as possible. So computers use a special type of high speed storage called Memory. Memory is very expensive compared to a disk drive, so it generally has less capacity.

But Memory needs constant power to operate. When you turn off the

computer, all the information in Memory disappears. However, a disk drive provides permanent copies of information that is put there. This is why many programs (such as a word processor) require you to “save” your work before ending the program or turning off the computer. A save command generally writes data in Memory to a permanent location such as a file on disk storage.

While there are various types of Memory, it is generically referred to as RAM (Random Access Memory). It is typically on a small circuit board that contains several silicon chips.

Start with a problem

To understand step by step how a computer works, let’s start with a problem. Here is a problem many of us face every year. Calculating our income tax.

First you provide the data for your income and your allowable expenses. By subtracting your expenses from your income, you get a number called your “Taxable Income”. This number is used to calculate your income tax.

$$\text{Taxable Income} = \text{Total Income} - \text{Total Expenses}$$

Many countries have a graduated (also called progressive) income tax, where the higher the income, the higher the tax rate. So the tax calculation depends on your tax bracket.

The U.S. tax rates for each income bracket in 2016 were

Taxable Income	Tax Rate
\$0 - \$9,275	10%
\$9,276 - \$37,650	15%
\$37,651 - \$91,150	25%
\$91,151 - \$190,150	28%
\$190,151 - \$413,350	33%
\$413,351 - \$415,050	35%
\$415,051 or more	39.6%

In other words, your first \$9,275 of taxable income is taxed at 10%, income from \$9,276 to \$37,650 is taxed at 15%, and so on.

This table is for 2016. The numbers are adjusted each year for inflation and may also be changed by legislation passed by Congress, as was done under the Trump administration for taxes beginning in 2018.

Instead of performing a multi-step calculation with this table to find your income tax, The Internal Revenue Service (IRS) provides a worksheet that has step by step instructions to calculate your income tax. Most people actually do not need to use this worksheet. If your taxable income is under \$100,000, you can look up your tax in a tax table. But if your taxable income is over 100,000, then you must calculate your income tax.

Let's calculate your income tax when your taxable income is \$125,000.

Taxable Income If line 43 is —	(a) Enter the amount from line 43	(b) Multiplication amount	(c) Multiply (a) by (b)	(d) Subtraction amount	Tax. Subtract (d) from (c).
At least \$100,000 but not over \$190,150	\$ 125,000	x 28% (.28)	\$ 35,000	\$ 6,963.25	\$ 28,036.75
At least \$190,150 but not over \$413,350	\$	x 33% (.33)	\$	\$ 16,470.75	\$
At least \$413,350 but not over \$415,050	\$	x 35% (.35)	\$	\$ 24,737.75	\$
Over \$415,050	\$	x 39.6% (.396)	\$	\$ 43,830.05	\$

Tax Computation Worksheet from IRS Form 1040 Instructions for 2016

First you find your tax bracket by reading down the first column. Taxable Income of \$125,000 is "at least \$100,000 but not over \$190,150" so you enter it on row 1 in column (a).

Following the instructions in column (b), you multiply it by 28% resulting in 35,000, which you enter in column (c).

Finally you subtract 6,963.25 in column (d) from column (c) to get your Tax 28,036.75 in the last column.

In computer terms, this set of step by step rules is an “algorithm”.
Let’s get a computer to do this.

Write a program for a solution

Once you have an algorithm, you can write a sequence of computer instructions that follow the step by step rules. This sequence of instructions is a “program”, or “code”.

Programs are written using a computer language. The following is an example of some code that calculates the Income Tax for the tax bracket in the first row of the tax form, using the rules highlighted in yellow.

Taxable Income If line 43 is --	(a) Enter the amount from line 43	(b) Multiplication amount	(c) Multiply (a) by (b)	(d) Subtraction amount	Tax. Subtract (d) from (c). Enter the result here
At least \$100,000 but not over \$190,150	\$	x 28% (.28)	\$	\$ 6,963.25	\$

```
Income = 125,000
If Income <= 190,150
    Tax = Income * .28 - 6,963.25
Else
    ... (code for other rows)
```

The first line sets the value of Income for the calculation.

The second line tests whether Income is less than or equal to 190,150. If that is true, then the computer proceeds to the next line and calculates the tax.

Else, if it is not true (in other words, Income is greater than 190,150), the computer continues on to the remaining lines in the program, which contain the code for the other rows in the tax form.

The full program that contains the code for the other rows is in [Appendix 1. CalculateTax Program.](#)

To make this a more usable program with a “user interface”, you could add instructions to read the taxable income as input from the keyboard and write the answer (the Tax) as output to the display screen.

Many different computer languages have been developed to try and make programming easy. This sample code is generic and does not conform exactly to a particular language, but it is very similar to many computer languages such as Java, C, PHP, Pascal, Python and others.

Machine Code

Now the truth must be told. Computer hardware really cannot understand even as simple a program as this. Hardware is not able to perform most of the instructions in any computer language. What computer hardware can do is run machine instructions, one by one. Machine instructions are much simpler and more fundamental than the instructions in programming languages.

So how do you get a computer to run a program written in a computer language?

The trick is to translate the program into machine instructions (or machine code). This is done with a special software tool, called a compiler or an interpreter.

The following shows the result of running the program through a compiler. The compiler reads each line of the program on the left and translates it into one or more machine instructions shown on the right.

Program → Compiler → Machine Code

Income = 125,000	MOV EAX, 125000
If Income <= 190,150	CMP EAX, 190150
	JLE R1
	<i>... (code for other rows)</i>
Tax = Income * .28 - 6,963.25	R1: MUL EAX, 28
	DIV EAX, 100
	SUB EAX, 6963
	MOV TAX, EAX

The first instruction is move (MOV) which simply loads the Income amount into the EAX register. EAX is one of the registers in the CPU. Remember that all computations done by a computer can only be done on data that is in a CPU register.

The second line in our program is translated into two machine instructions. The first instruction CMP compares the number in EAX to 190,150. The second instruction JLE (Jump Less than or Equal) will cause the program to jump to line R1 because EAX is less than 190,150.

The third line in our program is translated into a series of machine instructions which do the tax calculation by applying successive arithmetic operations multiply, divide and subtract to the number in the EAX register, very similar to what you do when using a desk calculator. The last instruction saves the final result into a memory location labelled TAX.

The machine code for the complete program is in [Appendix 2. CalculateTax Machine Code](#).

This code is based on an Intel processor (CPU chip). If it was compiled on a computer with a different CPU processor, it would generate machine code that would be different from this but fundamentally the same.

Note that instead of multiplying by .28 for the tax rate, the calculation multiplies by 28 and then divides by 100. To keep this example simple, it uses integer arithmetic. Input and output is dollars without cents. To handle decimal numbers, it would be necessary to use

machine code instructions for “floating point” numbers, which are somewhat more complicated.

This code represents machine code, but it is not actual machine code. Actual machine code that a computer can execute consists of a series of numbers.

Every machine instruction and CPU register have a numeric code. If 9 is a code number for the MOV instruction and 5 is a code number for the EAX register, then the first line of this program

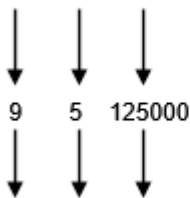
MOV EAX, 125000

Can easily be translated to

9 5 125000

But inside the computer, these numbers would actually be stored as binary numbers, numbers that have only the two digits 0 and 1. That’s because computers are electronic devices composed of electrical circuits. Electricity has only two states: positive or negative. Positive is represented as 1 and negative is represented as 0. So machine code inside the computer is actually a sequence of binary numbers, that is 1’s and 0’s. It’s very easy to convert numbers to binary, so finally here is what a machine instruction looks like inside the computer.

MOV EAX, 125000



1001 0101 000000011110100001001000

Text data

If machine code is a sequence of numbers, what about programs that do not use numbers? When you enter a Google search, create a post on Facebook or type an email, you are using computer programs to process text, not numbers. How is text stored in a computer?

The answer is quite simple. Text is converted to numbers using a standard coding scheme called Unicode. Unicode can represent the characters of all languages as numbers.

For example, “Hello” would be represented as the series of numbers 072 101 108 108 111.

Hello in Chinese 你好 would be represented as the series of numbers 228 189 160 229 165 189.

Running the program

To run a program, the computer must load it into memory. A typical way you do this is by selecting the icon that represents the program (with your mouse, or finger on a touchscreen). Then the operating system copies the program from a file in storage into memory.

The following shows a section of memory with the highlighted area showing where the machine code has been loaded, the first instruction in yellow and the rest of the program in blue.



Memory

```
11010100 11101101 01110110 11101100
01100001 01110100 01100101 01010100
00101110 01101110 11000010 00011000
00101000 00100000 10010101 00000001
11101000 01001000 10101111 10100111
00100100 00110110 01100101 01001001
01101001 01100110 00100000 01010100
01100001 01100010 01101100 01100101
01100011 01101111 01101101 01100101
01111000 01100001 01100010 01101100
01010001 10000100 10110101 11011
```

CPU

Instruction Register:

10010101 00000001 11101000 01001000

EAX Register:

00000001 11101000 01001000

Each binary digit, 0 or 1, is called a bit. A series of 8 bits is called a byte. This picture shows a space after every 8 bits to show the byte

boundaries.

The bytes are numbered sequentially so each byte in memory has a unique address. Addresses can be very large numbers; in a computer with 8GB of memory, a memory address could be a number from 0 to 8 billion. A unique memory address is very important because it allows the CPU to rapidly copy data directly between memory and a register in the CPU.

After the program has been loaded into memory, the first instruction in the program is copied from memory into a special register in the CPU called the Instruction Register, which can process instructions. The first instruction is processed, which moves a number into the EAX register (the first instruction in the program was `MOV EAX, 125000`, as shown earlier). Then the next instruction is copied from memory into the Instruction Register and processed, and so on until the end of the program is reached.

For more advanced details about binary numbers and memory storage, see [Appendix 3. Binary Numbers, Bits and Bytes](#)

Summary

So that's pretty well it. This is how computers work.

1. Start with a problem.
2. Develop an algorithm.
3. Write a program (code).
4. Compile it to generate machine code.
5. Load the machine code into memory so the computer can run the machine instructions.

In the following sections, we look at machine instructions in more detail to see how they are designed and how they are built using transistors.

Machine Instruction Set

Basically a computer processor (the CPU) has instructions that can do only four things. Our small machine code example showed three of them.

Program → **Compiler** → **Machine Code**

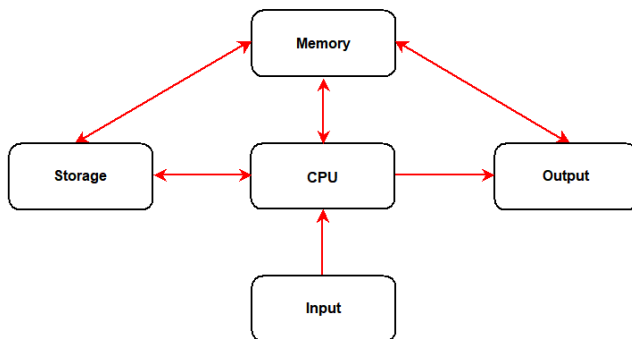
Income = 125,000
If Income <= 190,150

Tax = Income * .28 - 6,963.25

```
MOV EAX, 125000
CMP EAX, 190150
JLE R1
... (code for other rows)
R1: MUL EAX, 28
DIV EAX, 100
SUB EAX, 6963
MOV TAX, EAX
```

1. Move/Copy

The arrows in the basic computer model show a number of the different move instructions.



2. Jump

Computers process instructions in the order in which they occur in the program. A few instructions can change this and allow a program to jump to another part of the program, or to another program. The example code uses one type of Jump instruction that Compares two numbers and then Jumps to a different line in the program depending

on the result. Note that Compare and Jump can also be used with text, since text is coded as numbers with the Unicode standard. It may be hard to believe but all the so-called intelligence of a computer, its ability to make any decision at all, is based on this rather simple type of instruction.

3. Calculate

There are machine instructions for basic arithmetic, logical operations (AND, OR, NOT) and a few mathematical operations such as square roots, logarithms, trigonometry. More advanced mathematical computations all have to be done with software that must be compiled into machine code instructions.

4. Special Instructions

There are a number of special instructions that are needed by the Operating System to control the operation of the computer. But they are privileged instructions and cannot be used by an application program.

For more details, see [Appendix 4. Machine Instruction Sets](#)

It is rather amazing that all programs and software written for a computer get compiled down to only four types of machine instructions. This applies to all computers large and small, smartphones and custom computers in use today and computers being designed for the future such as robots, self-driving cars and space craft.

What is also amazing is that the very earliest computers also had this same simple instruction set. The Univac I, one of the first commercial computers, released in 1951, had fewer instructions than computers today but it had the same basic four types of machine instructions.

How are machine instructions designed?

At the electronic level, machine instructions are designed using **logic gates**. Here is the symbol for an **AND** logic gate, showing two input signals and one output signal.



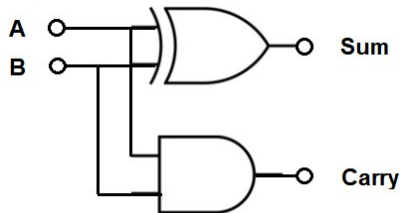
Output is the result of the logical **AND** of the two inputs **A** and **B**. Remember that the statement “A and B” is true only if both A and B are true.

In an electric circuit, **true** represents a current and is indicated by 1, **false** represents no current and is indicated by 0. So current flows through an **AND** gate only if there is current at both inputs.

The machine instructions in the CPU can be built using combinations of the basic logic gates for

- AND
- OR
- NOT
- NAND (not and)
- NOR (not or)
- XOR (exclusive or)
- XNOR (exclusive nor)

For example, here is the logic design for an Add instruction that adds two binary digits **A** and **B**.



The inputs **A** and **B** go through an XOR gate to produce the **Sum** and an AND gate to produce the **Carry**.

For example, when you add $8 + 4$ the sum of the digits is 2 and you carry 1 to the next column.

The **Sum** (XOR, exclusive or) is 1 when either **A** or **B** are 1 but not when both are 1.

There are only four different possibilities for the inputs **A** and **B**. Here are all the binary addition cases this circuit does:

A	B	Sum XOR	Carry AND
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

The first three cases are simple arithmetic.

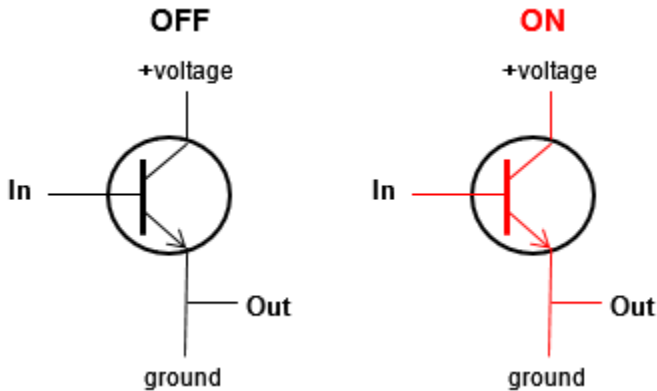
In the last case, $1 + 1 = 2$, but 2 is represented in binary as 10. The addition is $1 + 1 = 0$ and carry 1 to the next column.

This is the same as the decimal number system, where $9 + 1 = 0$ and carry 1 to the next column.

Machine instructions for addition are generally needed for numbers that are larger than one binary digit, called a bit. To design an instruction that adds 16 bit numbers, you can use a sequence of 16 binary adders slightly more complicated than this one where the Carry output from the adder for the first binary digit is input into the adder for the second binary digit, and so on.

How are machine instructions built?

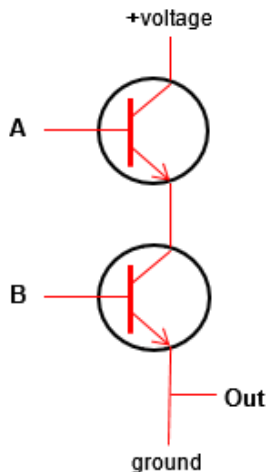
Logic gates are built electronically using **transistors**, which provide the basic on/off function of a switch. There is an electrical voltage differential between the **+voltage** and the **ground**, so a current can flow through the transistor.



In the case on the left, the vertical bar in the transistor is not conducting a current, so the switch is off and no current flows **Out**.

On the right, current applied at **In** turns the switch on and current flows **Out** of the switch.

The circuit diagram below is an **AND** gate constructed with two transistors. There must be current flowing through both **A** and **B** into the transistors for current to flow **Out**. If there is no current at either **A** or **B**, then there is no current at **Out**.



Similarly, transistor circuits can be built for all the other logic gates OR, NOT, NAND, NOR, XOR and XNOR. This is the basis for the hardware design of a CPU chip.

It's also easy to see that transistors can be used to make memory chips. Memory holds a binary value, 0 or 1. So a transistor in the off state is 0 and a transistor in the on state is 1. Memory chips are constructed out of large arrays of transistors that have circuitry to read and write to memory. Reading memory detects whether a transistor is off or on and writing turns individual transistors off or on.

Machine Instruction Summary

At the hardware level, computers have only four types of instructions, Move, Jump, Calculate and a few reserved instructions for the Operating System. Machine instructions can be designed with seven types of basic logic circuits such as AND, OR and NOT. These logic circuits can be built with simple switches. Electronically, these switches are made with transistors.

Integrated Circuits

Transistors are made with a semiconductor material like silicon. Current does not flow through pure silicon but if the silicon is combined with other elements, the electrical balance can be changed to a positive or negative bias so that current does flow.

Assembling transistors and other electronic components individually on a printed circuit board (*Figure 1-2*) can be costly. So silicon chips that contain many components are used as much as possible.

Silicon chips are **integrated circuits**, in which many electrical components are all fabricated on the same piece of silicon. As well as reducing assembly costs, this greatly reduces size, so less power and fewer printed circuit boards are needed to manufacture a computer.



Four memory chips on a circuit card

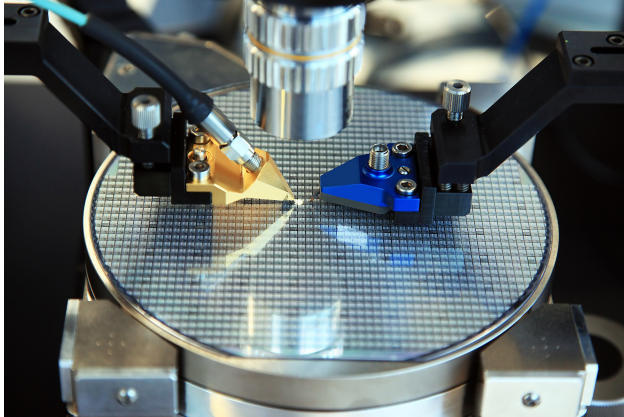
Transistors and integrated circuits are considered by many to be the greatest inventions of the 20th century. Their inventors each won the Nobel Prize in physics. These inventions revolutionized the whole electronics industry and made great advances possible in many areas such as radio, television, telephone systems and consumer electronics as well as computers. Personal computers and cell phones would not be possible without integrated circuits.

The most complicated integrated circuits are CPU processor chips. Designing and manufacturing a CPU chip is a very large and expensive undertaking. It takes a team of highly skilled hardware engineers. Advanced software tools are needed to design the chip, verify the design and generate the information needed for manufacturing.

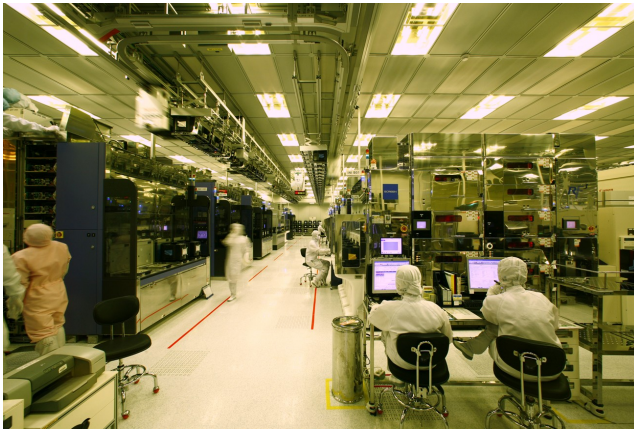
Most computer manufacturers do not design their own CPU chips but buy them from third party suppliers, such as Intel, who have large semiconductor fabrication plants. These plants are extremely costly to build, around \$10 billion. The semiconductor industry is a very large competitive business with plants around the world and a total market of over \$300 billion per year.

Manufacturing a CPU chip is a multi-step process that may require more than 100 steps and take up to 30 days, a process that can cost up to \$3 million.

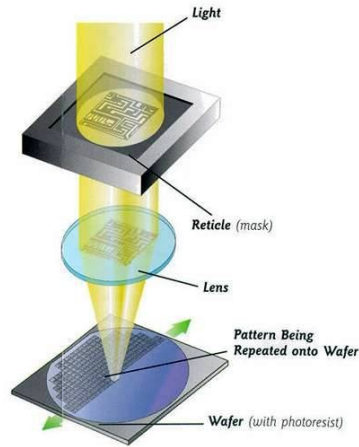
First, pure silicon wafers are created from sand. Each wafer will contain hundreds of copies of the integrated circuit.



This must be done in a clean room, as a slight impurity in the silicon may lead to failures in many of the individual chips.



The design is built on the chip one layer after another. First a special film is applied to the silicon. Then a mask produced by the engineering design software is transferred to the chip with a photographic process.



Etching exposes the silicon in the light areas of the mask that represent the circuitry. Different elements are applied to the exposed silicon, creating a layer that will conduct electricity.

When all layers have been fabricated on the chip, an automated system tests each chip on the wafer. Those that fail are discarded.

Finally, the wafer is cut into individual chips which are assembled into a hard plastic package to protect the chip. This packaging is usually black and makes most silicon chips look similar. It is not possible to see any of the layers or the circuitry on the chip.

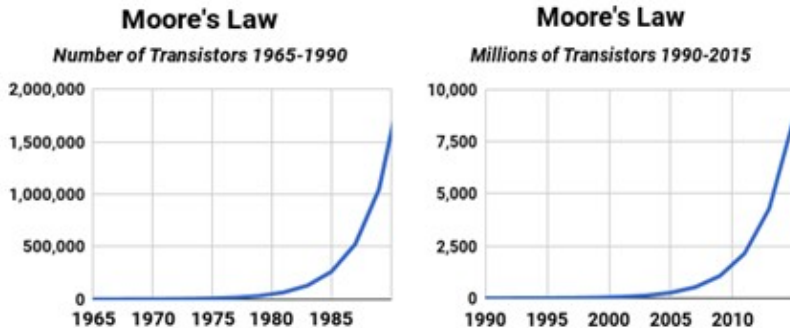
Moore's Law

Ever since the first integrated circuits were fabricated, the semiconductor industry has been able to make transistors smaller and smaller. As electronic circuits are made smaller they get faster, since the electrical signals have less distance to travel. In addition more functions can be put on a single chip, making it possible to build smaller and cheaper computers and other electronic devices.

In 1965 Gordon Moore, an engineer who co-founded Intel, predicted that the number of transistors on integrated circuits would double approximately every two years. This prediction turned out to be quite

accurate for both memory chips and microprocessor chips.

This is exponential growth. The first integrated circuits had less than 20 transistors. By 1990 there were about 2 million transistors on a chip and by 2015 there were chips with 10 billion transistors.



Moore's Law, more than anything else, has been the engine that has powered the huge advances in the computer and electronics industry.

A laptop computer in 2015 compared to a computer in 1965 (a mainframe computer since that was the only type of computer then) was orders of magnitude smaller, faster and cheaper.



1965 Computer
\$15,000,000



2015 Laptop
\$1000
100,000 times faster

Moore's Law has been the engine that has powered the huge advances in the computer and electronics industry.

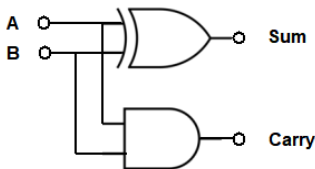
Chapter 1 Summary

1. Computer programs are compiled (translated) into **Machine Instructions**.

ADD EAX, 100 is a machine instruction that adds 100 to the number in the EAX register in the CPU.

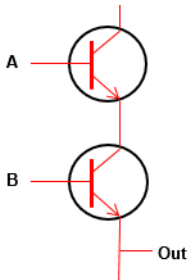
2. **Machine Instructions** are designed with **Logic Gates**.

An Add instruction can be designed by using an XOR gate with an AND gate.



3. **Logic Gates** are built with **Transistors**.

An AND gate can be built with 2 transistors.



4. **Transistors** are manufactured as **Integrated Circuits**.

Billions of transistors can be manufactured on a single Integrated Circuit, or silicon chip.



The Future

There has been a major revolution in computers introducing a new era on average every 14 years since its first invention. Following is a brief summary of this evolution. For a more detailed history, see the book “A Concise History of Computers Smartphones and the Internet”, by Ernie Dainow.

1936 Early Computers — Alan Turing proposed a simple theoretical machine in a mathematics paper. While various calculators existed, this was the theoretical groundwork for a general purpose computing machine. The first working computers were developed during World War II and the following years in government research labs and in universities.

1951 First Commercial Computers — The first general purpose computers became available from many different companies in the U.S., Europe and Japan.

1964 IBM Mainframe Era — the IBM System/360 family of compatible computers was a huge success and drove many computer vendors out of the market, establishing IBM as a dominating force in the computer industry for the next 25 years.

1981 Microcomputers — while there were earlier microcomputers such as the Apple I in 1976, the real revolution in widespread use of microcomputers started with the IBM Personal Computer (PC) that was released in 1981.

1995 Internet — the Internet was first built in 1969 as a research project under a grant from the U.S. Department of Defense, but widespread use of the Internet did not occur until after the World Wide Web was developed in 1991 and Microsoft Windows 95 made it easy for non-technical users to access the Internet with simple point and click.

2007 Smartphones — the first smartphone was the Simon developed by IBM in 1994 followed by the Blackberry in 2002. But the general Smartphone era really began with the revolutionary Apple iPhone in

2007 which began a shift to mobile computing in general.

Following this 14 year pattern, the next revolution will occur around **2021**. What will it be?

There are a number of emerging technologies that could become the next big thing.

- **Wearable computers** — a number of small devices that you can wear are already available, such as fitness monitors, smart watches and virtual reality headsets.
- **Robots and Artificial Intelligence (AI)** — while electronic robots have been used in factories since 1961, they have been special purpose machines. With recent advances in Artificial Intelligence, general purpose robots and other software applications that an individual can use such as personal aids for many things using simple voice commands are now starting to be used by many companies and start-ups.
- **Internet of Things** — the **IoT** is envisioned as a huge extension of the Internet to interconnect small devices, sensors and appliances as well as computers. There are estimates of 26 billion devices on the Internet of Things by 2020. Many great new uses are foreseen, such as health sensors that provide detailed information for diagnosis and warnings of medical conditions like heart attacks before they become emergencies and smart infrastructure such as roads and bridges that report problems needing repairs before there is an accident.

However, there is a physical limit as to how small the transistors in an integrated circuit can become. Transistors are approaching the size where quantum physics will affect the movement of electrons and transistors will not reliably switch on and off. Moore's law will likely end around 2021 and that could be followed by a significant slowdown in computer hardware advances. There is ongoing research into alternatives to silicon for semiconductor manufacturing and alternative computing techniques such as quantum computing, optical computing, and biochemical computing show promise but it seems like it will be a long time before they will be ready for production use.

2. How Does Software Work?

Let's look at software in more detail to see how it extends the power of the computer beyond the rather limited machine instruction set that was described in Chapter 1.

Computer Languages

We saw in Chapter 1 how a programming language is much easier to write than the machine code that a computer needs. Because of this, many computer languages have been developed. There is no one general purpose language that is used for everything. Instead there are different languages that are tailored to specific purposes or environments.

- Some operating systems support only certain languages. For example, developing an app for the iPhone and iPad would be done with the Objective-C or Swift language, but the same app for an Android phone would be programmed in Java.
- For scientific applications, there are preferred languages, such as Fortran or C.
- Many business applications are written in COBOL on mainframe computers, Java on Linux computers and C#.Net on Microsoft Windows computers.
- For work in “Artificial Intelligence”, a number of special languages have been developed, such as Lisp.
- For web applications, there are many other languages such as PHP, Javascript.

Databases

We have largely concentrated on the computational power of computers — their ability to do billions of instructions per second.

Another great power of computers is the ability to store and retrieve huge amounts of information.

Writing software from scratch to manage large amounts of data can be a very big job. Instead, the power of specialized database software is usually used. Many applications use database software that is available from a number of third party vendors.

Databases are used for data that can be represented as a table of rows and columns. A standard language called Structured Query Language (SQL) is used to access such data.

The following report of charges on a credit card is a good example of such a database.

Date	Description	Amount
27/02/2016	AA INFLIGHT MC FACET 3 PHOENIX AZ	\$5.56
27/02/2016	GREAT AMERICAN GRILL T TUCSON AZ	\$55.07
27/02/2016	AMERICAN 000102726190000 PHOENIX ON	\$52.50
03/03/2016	KAISER GRILLE PALM SPR PALM SPRINGS CA	\$94.46
07/03/2016	PARIS LE VILLAGE BUFFE LAS VEGAS NV	\$92.13

To understand how powerful databases are, all the programmer needs to do to retrieve this data is to write this simple command:

```
SELECT Date, Description, Amount FROM Transactions;
```

Databases also make searching for information easy for a programmer. If you had a database of books, you could search for all books containing the word “introduction” in the book title with the following command:

```
SELECT * FROM Books WHERE Title LIKE '%introduction%';
```

The Layers of Software

Part of the power of software is the ability to combine many layers and

pieces of software into very large programs.

Many of these pieces are provided by third parties, so that a large software development can be undertaken that focuses on the particular problem (the application) without having to code all the details of other more generic operations.

The following diagram that shows the layers of software is pretty generic. It applies to the apps you download to your smartphone as well as to custom software developed by say a large financial institution to run on their mainframe computer.

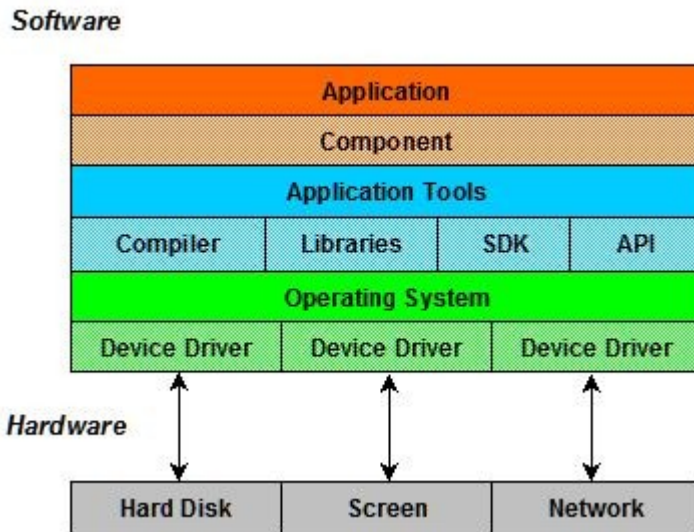


Figure 2-1. Software Layers

Application Layer

The top layer is the Application (or app) that is being developed for a particular use. While this diagram shows only one component, a large application is generally separated into multiple components. This aids in the logical design of the software and makes it easier for a team of developers to work on the application by assigning different

components to different developers or groups of developers.

Application Tools

Below the Application layer are the application tools, most of which are provided by third parties. Some are general purpose such as tools that automate building the software. Some are particular to each programming language, such as a compiler and a debugger.

Libraries provide software that can be called by the Application. Many basic functions are provided, such as reading and writing to files. Extensive additional libraries may also be available, depending on the programming language, to handle complex processing like

- mathematics
- text processing
- networking, Internet protocols
- image and audio files

SDK is a “Software Development Kit”. Many capabilities of the operating system may not be available in the standard libraries. An SDK has to be added to the tool set to provide an interface to a particular operating system. Developing software for Windows applications needs the Windows SDK from Microsoft whereas developing an iPhone app needs the iOS SDK from Apple.

API is an “Application Programming Interface”. It provides a software interface to other software which may be third party services running on another computer. For example, a web application may insert a map to provide the location of a restaurant. The map can be provided by using a Google API to send requests to a Google server and get back the information to display the map.

Operating System

The bottom layer is the Operating System (OS). The OS is a special program that controls the computer hardware. When you first turn on a computer or mobile phone, it needs to load the OS before it can really do anything. Like any program, the OS has to be read from

storage into memory. Since most operating systems are quite large and contain many components, this can take some time. This is why you have to wait when you start or “reboot” a system.

Familiar operating systems in use today are Microsoft Windows, iOS for Apple iPhones/iPads and Google Android for various mobile phones and tablets. Specialized computers have their own custom operating systems, for example Cisco network switches and routers run under Cisco IOS.

The OS provides an interface for upper layer software to communicate with hardware devices such as **Storage** (disk drives), **Input** devices (keyboard, mouse, touchscreen), **Output** devices (screen, printer) and **Networks** (Ethernet, Wi-Fi).

Hardware devices are made by many different manufacturers who build the hardware and provide the software to operate the device. Usually called a device driver, this software must be installed in the OS so that it can communicate with the hardware device. This communication is generally done by sending signals across a cable that connects the device to the main computer. You may be familiar with a USB cable, which is typically used for connecting to a low speed device such as a printer. Connections to a high speed device such as a disk drive use a different type of cable.

A well designed OS is a boon to the application software developer. The software required to read data from a file, for example, is very complicated. If every programmer needed to write code for this, it would add a great deal of complexity to the program. Instead a programmer can simply write a few simple lines of code to call the OS which then does all the hard work of locating the file on the disk drive and passing requests to the device driver to read the blocks of data.

Linking the Layers

Programming languages can “call” other program code, using the machine language jump instruction we saw in the sample machine code in Chapter 1. The jump can be to code in the same file or a different file, in a library or in the operating system.

In a large software project, there may be hundreds or thousands of files containing source code. Once the source code is written, the application code and the external code need to be compiled and “linked” together to produce the machine code.

The following schematic diagram shows how code from several sources is combined to produce the final program machine code.

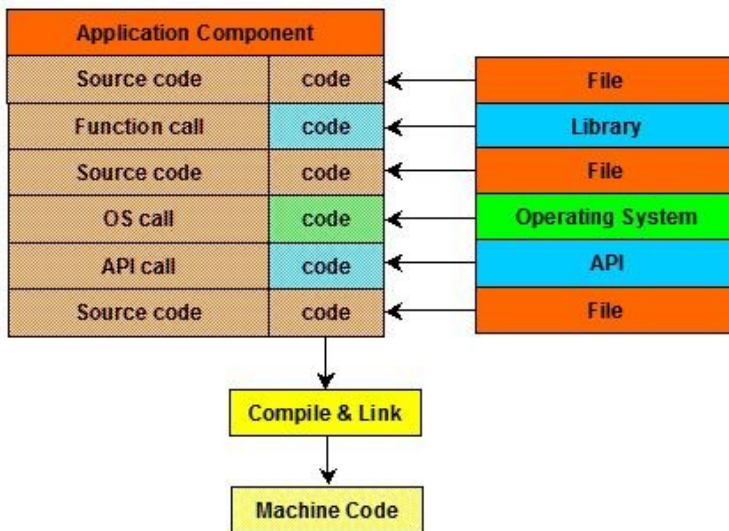


Figure 2-2. Producing the Machine Code

Software Development

Developing software is generally a large and expensive project that affects a number of departments. So organizations that engage in software development usually have a process in an effort to better control cost and delivery time. In spite of this, many software projects go over budget and are delivered late.

While a number of different processes have been proposed and used for software development, they generally incorporate the following key phases in a systems development life cycle.

1. Requirements

Marketing or product management defines the necessary capabilities and features of the software in a “Requirements” document.

2. Analysis

The software group provides a technical evaluation of the requirements in a “Functional Specifications” document. The analysis may also include evaluating third party software, prototyping and feasibility testing. Results of the Analysis may lead to changes in the Requirements.

3. Design

The software group expands on the Functional Specifications and provides the details in a “System Design” document. A work schedule (project plan) is developed to provide a timeline and the allocation of people and other resources needed for the project.

4. Coding

Finally the software development (coding) begins. Teams of programmers are assigned to different components of the project.

5. Testing

When the software has been completed, Development delivers the software to Quality Assurance (QA) for testing. Testing reveals “bugs”, things that do not work correctly, and are logged in a bug tracking system. When bugs are fixed, Development builds a new release of the software. QA must then repeat all the test cases, as bug fixes frequently break things that were working on earlier releases.

This bug fix/test cycle is repeated until no more serious bugs exist. In order to meet delivery schedules, software is often released with known bugs that are left on the bug list for correction in a later release. There may also be other bugs since QA testing usually cannot cover every possible case, particularly interactions that may occur with other software.

Consequently, most software has bugs. When using a computer, if you encounter something that does not seem to work, you may find a “workaround”, another way to do the same thing that does work.


6. Release

If the software is a product that is sold, the tested package is provided to a group within the sales department or to customer service to manage the sale and delivery to customers.

In many other cases, software is not distributed outside the organization. Financial institutions, manufacturing companies, government and many other organizations have large software development groups to develop and maintain systems needed for internal users. In this case, the software is delivered to the Operations or Information Technology (IT) group. They install the software on production servers in the data centers where it can be accessed by users.

A Software Development Example

As a development example, let's look at an online store. An early developer of this type of web application was Amazon.com, which opened in 1995 as an online bookstore. It has since expanded to carry thousands of different types of products. Many businesses now have a web site where people can view their products and select items to purchase. Following is a typical example.




Crock-Pot Countdown
Customize your cooking time

★★★★★ 25 Reviews

\$49⁹⁸

Add to cart




Exclusive

Rival Roaster Oven With Self-Basting
This roaster is large enough to cook a whole turkey

★★★★★ 60 Reviews

\$39⁹⁸

Add to cart



Rival 2 Quarts Slow Cooker
Capacity: 2qt, voltage: 120v

★★★★★ 7 Reviews

\$13⁷⁷

Add to cart

Requirements

Suppose marketing provided the following requirements.

1. Customer Shopping

The online store needs to be accessible by industry standard web browsers and mobile apps for iPhones and Android phones. The mobile apps should have the same functionality as the web store with layout and presentation optimized for the smaller mobile phone screens.

- There must be a products page that provides a picture and details of each product.
- It must be easy for a user to search for a product.
- Items can be selected from the products page and added to the user's shopping cart.
- It must be easy for a user to view their shopping cart and make changes — add and delete items.
- There must be a checkout process where the contents of the shopping cart are finalized, shipping and delivery details are provided and payment is made. There must be support for paying by major credit cards.
- When purchase of an item is confirmed, an email notice is sent to the purchaser and a shipping notice is sent to the product warehouse with the purchase and shipping details.

2. Administration

The system must provide management capabilities to allow an administrator with authorized access to:

- manage the product list — add and delete items, change prices.
- manage the shipping options.
- view status of outstanding orders.
- provide reports of sales for a selected time period.

Software Design

A software design that addresses these requirements is summarized in the following architecture diagram. The orange rectangles show the components that need to be developed.

The upper part of the diagram is the server side of the system that

runs in a data center. All server components except the Database are shown as running within one machine, the light orange rectangle.

The “Backend Server” provides a central point that handles all client requests and interfaces with the rest of the server components as necessary.

The components below the Internet are the client components. They communicate with the servers in the data center over the Internet.

The Firewalls are specialized network computers that provide security for the servers by blocking access to those machines from Internet hackers.

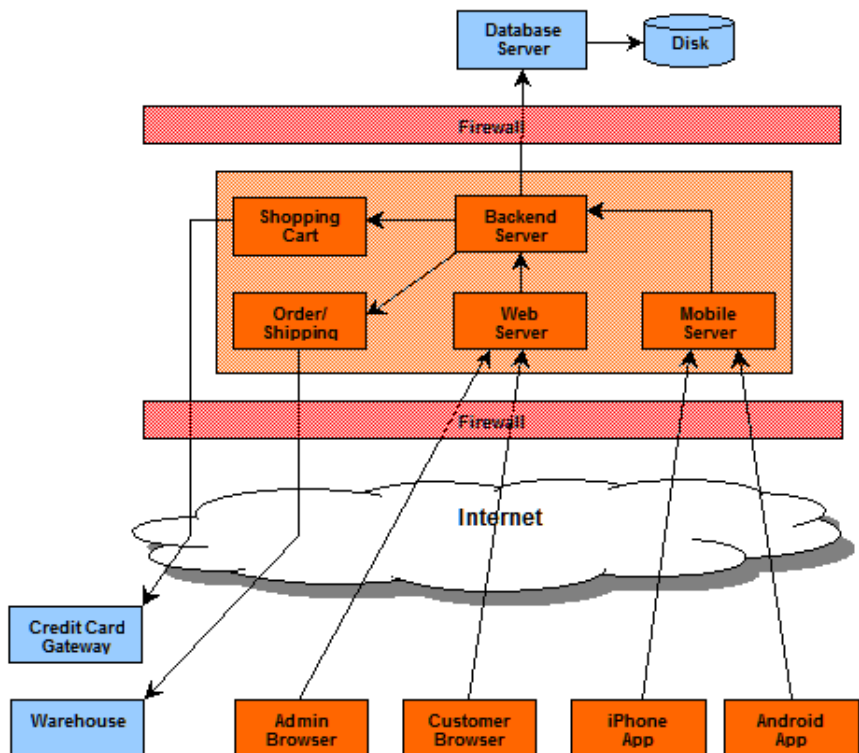


Figure 2-3. Software Architecture Online Store

As an example of how the different components communicate with each other, a search of the product list could be designed as a software call like

```
ProductSearch( searchstring )
```

When a user selects a search for a product from a mobile phone, the mobile app sends this request to the backend server via the mobile server. The backend server sends a database query for searchstring to the database server and returns the results to the user.

Different software technology and languages could be used for different components. While there are a number of different ways to build this system, the following would be a typical implementation.

- The servers are built on Linux computers. The Web Server is Apache, widely used open source code that just needs to be installed and configured. The software in the other server components is written in the Java language.
- The Admin and Customer web browsers load web pages from the Web Server. The web pages are written in HTML with embedded code written in PHP and JavaScript.
- The iPhone app is written in Swift, a language developed by Apple for developing apps for the iPhone and iPad.
- The Android app is written in Java.

In reality, you would not likely develop software from scratch for all of these components. That's because there is a lot of software available from third parties to build online store applications. It is usually more cost effective and faster to use existing, tested software than to develop it yourself. There are commercial packages available and also quite a lot of open source software. Open source software may be downloaded and used without cost, as long as the software is not being used as part of a package that is sold.

Software Applications

The power of software to do so many things has revolutionized many

industries by providing great cost reductions and/or new ways of doing things. Many new start-ups have been created based on their innovative software.

Some of the revolutionary applications of software have been:

Accounting - one of the first applications to be automated with computers, first large companies using mainframes and then small and medium companies when less expensive microcomputers became available.

Financial - automated teller machines (ATM), online banking, automated stock exchanges, online brokerages.

Spreadsheets - tabulating information in tables with simple calculations. Advanced users are able to automate various series of calculations, almost like writing programs, without having to become computer programmers.

Publishing/Desktop Publishing - word processing for books and articles; layout for newspapers, magazines and newsletters.

Communications - email, computer based instant messaging/chat.

Manufacturing

Computer-Aided Design (CAD) to create and analyze designs and prepare data for manufacturing.

Computer-Aided Manufacturing (CAM) to control machine tools in manufacturing.

Digital photography/video - software to create and manage digital media files made digital cameras and camcorders commercially feasible.

Computer graphics - has had a significant impact on many types of media. It has revolutionized film animation, movies, advertising and graphic design.

Games - are a huge market that appeals to many age groups. There are many game platforms: consoles that connect to a TV (Sony PlayStation, Microsoft Xbox, Nintendo), games you can play on your computer and your mobile phone, and online games you can play over

the Internet. Improvements in the video quality of games depends a lot on hardware advances. Leading edge game software has driven a lot of the computer graphics industry and has led to the development of special graphics chips.

Education - many universities and other educational organizations provide online courses and training over the Internet.

Science and Research - in physics, chemistry, astronomy, biology, medicine. For example, the Human Genome Project that identified and mapped all the human DNA in 2003 would not have been possible without software.

Social Networking - is based on providing a service where members can set up a network of friends and communicate with them by posting messages, photos and videos. When the Internet became widely available and made it easy for people to join such services, social networking grew very rapidly. Many people have found this a better way to stay in touch and communicate with friends than using older technologies like telephone and email. Although there are quite a number of different social network systems, Facebook has become the dominant force, with over 1.5 billion members. Twitter is a more public social network that allows people to “follow” anyone else or topics on Twitter. Twitter has become an important source for breaking news which often appear on Twitter before the traditional news media or other web sites.

Artificial Intelligence/Robotics - research is complicated and progress has been incremental. There is useable software for speech recognition and language translation and driverless cars are coming. Specialized robots have been successful in factory automation and there have been recent advances in more general purpose robots.

3. How Does the Internet Work?

In this chapter we look at the Internet to understand how it works and how computers use it to expand their basic capabilities.

What is a Network?

Computers in an office or a home are often connected together by a network, called a local area network (LAN) to make it easy to share files and printers.

One way to connect your computer to a network is with a cable. One end of the cable is plugged into a special socket in your computer and the other end is plugged into a Router. This is a wired connection called Ethernet.

You can also use a wireless connection if you have a wireless Router. This is a Wi-Fi connection that uses radio signals. Smartphones and tablets can use a Wi-Fi connection.

The Router is a specialized computer that receives data from any device on the network and routes it to the destination.

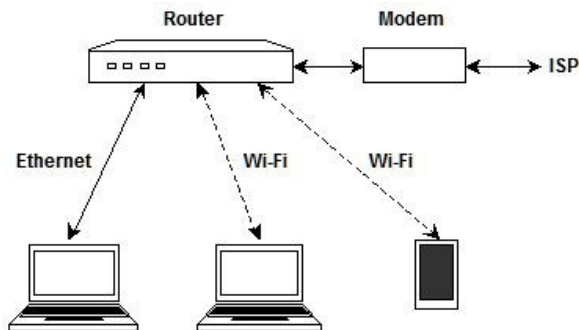


Figure 3-1. Local Area Network

In order to use the Internet, you must have an account with an Internet Service Provider (ISP). Your ISP will install a router with a modem that connects to their network. There are various connections

an ISP might provide such as fiber broadband (high speed), DSL or dial-up over a telephone line.

The modem (modulator-demodulator) does the conversion between the digital signals used in a computer network and the analog signals used on the telecommunications line.

What is the Internet?

The Internet (“inter net”) is a very large global network that interconnects other networks. It is built with many routers, each of which has high speed fiber optic connections to several other routers. The Internet allows computers in one network to communicate with computers in other networks.

For example, when you submit a search to Google, it is sent over the Internet to Google’s network, as shown by the **orange** path.

Note that the Internet is shown in diagrams as a cloud. This is a tradition that goes back many years predating the current use of “cloud” to refer to remote storage. When you use a cloud service such as Apple iCloud or Dropbox, your computer uses pretty much the same sort of Internet connection as shown in this diagram. But instead of connecting to the Google search server, it connects to a server that provides a lot of disk storage for you to save or backup your files.

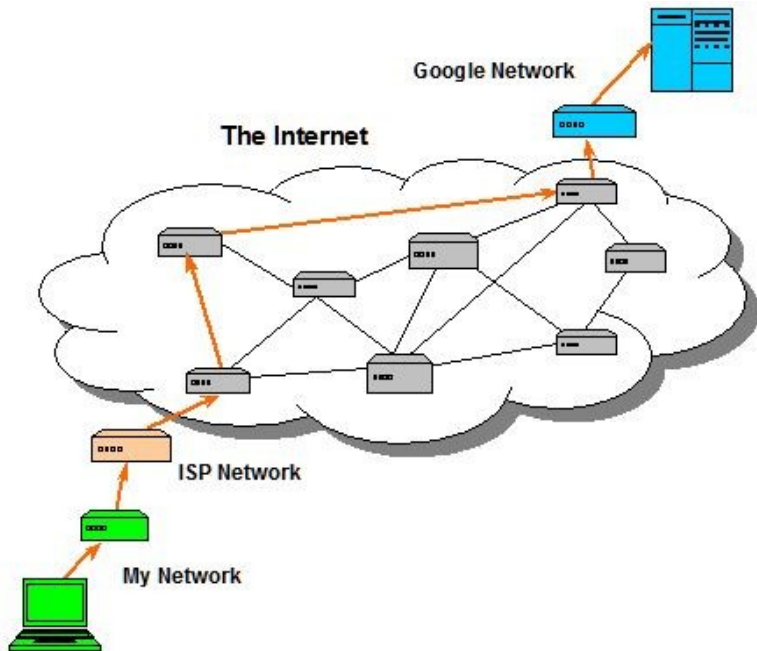
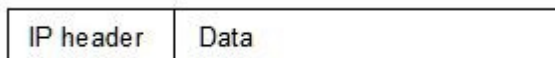


Figure 3-2. The Internet

How does the Internet work?

Data sent over the Internet is sent in “packets”. The maximum size of a packet depends on the characteristics of the network links but is usually 1500 bytes. Larger data must be split into multiple packets. So for example, if you download a 1.5 MB photo or music file (1,500,000 bytes), your computer or smartphone is receiving about 1000 packets over the Internet.

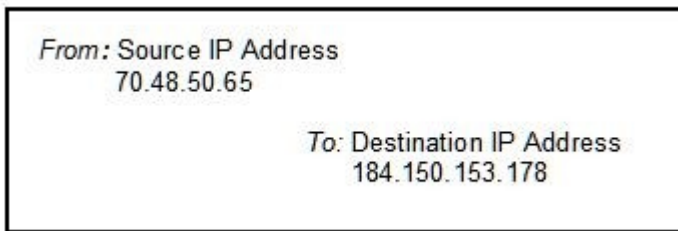
Each packet must be formatted to conform to the Internet Protocol (IP). This is done by adding IP information at the beginning of each packet, in an “IP header”.



Two important fields of information in the IP header are the “Destination IP Address” which identifies the computer to which the packet is being sent and the “Source IP Address” which is the address of the computer that sent the data. The source address is needed when the computer that received the data needs to send back a reply.

An IP address is a sequence of four numbers separated by dots, as in 76.250.32.149.

You can think of an IP packet as an envelope containing some data. Here is a picture.



Routers transmit IP packets by reading the destination IP address and forwarding the packet to the next router that is on the fastest route.

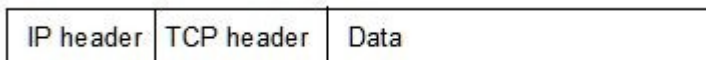
Note that the fastest route may not be the shortest distance. In *Figure 3-2 The Internet*, the selected path from your computer to Google is longer than either of the diagonal paths. This is analogous to picking a driving route in your car to use a highway instead of city streets even if the distance is longer.

Even in a simple network like this, you can see that there are many different paths between two points. In fact, the route a packet takes can be different from the route of the previous packet.

Here's why.

The Internet is subject to sudden changes in traffic patterns. A link between two routers may become congested or even go down. So the fastest route may be different at different times. This is similar to what you probably do when driving. If there is construction on your normal route or congestion in rush hour, you take an alternate route to avoid delays.

When packets don't get delivered due to congestion or to errors on a communication link, it is up to the sending machine to retransmit the missing data. This is done with a special Internet protocol called TCP (Transmission Control Protocol). TCP is used by most Internet applications and mobile apps, such as a web browser. A TCP header is added to each IP packet to provide control information. This is what an IP packet looks like when using TCP.



If a network link has a lot of interference, there will be packet errors that trigger retransmissions by TCP. This can make an Internet connection very slow. You may encounter this when using a public Wi-Fi network. You could try moving to another location where there might be a stronger Wi-Fi signal. However, public Wi-Fi can also be slow because a bandwidth limit has been set on each user so that the Wi-Fi capacity gets shared equally among all the people using the network.

For more advanced details of how packets are sent over the Internet, see [Appendix 5. Internet Routing](#).

What is the Web and how does it work?

The “web” is short for the “world wide web”, abbreviated as www. With a web browser (such as Microsoft Edge, Firefox, Chrome, Safari or others), you can enter the name of a website and get back a page of information. A web page is not simply text, it is “hypertext”. Hypertext is text that contains links to other text, which can be on another computer. By selecting a link, you can retrieve those pages. So with a web browser and the Internet, you have access to a world wide web of information.

As an example, let’s look for information about Steve Jobs.

If you enter “steve jobs” in the search line in your browser, you get back a list of search results. Each result is a summary with a link to a web page. Here is one of the search results.

Steve Jobs - Wikipedia

https://en.wikipedia.org/wiki/Steve_Jobs

Steven Paul Jobs was an American entrepreneur, ...

To look at the Wikipedia article, you just select the blue line and the browser uses the link shown in the green line.

Here is the first part of the web page that is returned and displayed on your screen.

Steve Jobs

From Wikipedia, the free encyclopedia

Steven Paul Jobs (February 24, 1955 – October 5, 2011) was an American entrepreneur, businessman, inventor, and industrial designer. He was the co-founder, chairman, and chief executive officer (CEO) of **Apple Inc.** Jobs and Apple co-founder **Steve Wozniak** are widely recognized as pioneers of the microcomputer revolution of the 1970s and 1980s.

So what happened here? Let's see how the web works by following what happens when a browser sends a request to a web server and gets back a web page.

Web browser request

First look at the link that you selected,

https://en.wikipedia.org/wiki/Steve_Jobs

This tells the browser to use the **https** protocol, connect to the web server **en.wikipedia.org** and get the web page **wiki/Steve_Jobs**

A protocol specifies the format and type of data. There are many different types of data that can be sent over the Internet. A web browser uses the protocol HTTP, an abbreviation for "Hypertext Transfer Protocol".

HTTPS is a Secure version of HTTP, which is just HTTP in which the data in the IP packet is encrypted. As IP packets travel over many network links, there are several opportunities for inspection or

recording of packets. Encrypting the data portion of the IP packet prevents anyone from easily reading data sent and received from a website, such as login account information.

To process your request for the web page **wiki/Steve_Jobs**, the browser creates an HTTP GET request, encrypts it if it was an HTTPS request, and puts it into a IP packet that looks like

IP header	TCP header	GET /wiki/Steve_Jobs HTTP/1.1 Host: en.wikipedia.org
-----------	------------	--

We saw earlier that the destination IP address needs to be put in the IP header. But we have the name of the web server, **en.wikipedia.org**, not its IP address. So a request is sent to a Domain Name Server (DNS) to lookup the name **en.wikipedia.org** and return its IP address, which can be put into the IP header.

Then the packet is sent to the network device driver via an OS call (review the overall sequence from Application to the Network in *Figure 2-1 Software Layers*).

The network driver transmits the packet over your local network to your router which forwards the packet to your ISP where it is sent over the Internet to the computer at **en.wikipedia.org** (as in *Figure 3-2 The Internet*).

Web server response

The Wikipedia web server extracts the HTTP data from the IP packet, decrypts it if it was an HTTPS request, and locates the web page **wiki/Steve_Jobs**. This is just the name of a file so it reads the file, puts it in the data part of an IP packet and returns the packet over the Internet back to your computer.

When your computer's network driver receives the IP packet containing the HTTP response, it passes it to the browser. The browser extracts the web page from the data part of the packet, formats it and displays it on your screen.

If the web page contains images, your browser will make additional

HTTP requests to get the image files. If there are a lot of images on the page, this can take some time, which is one reason why some web pages take longer to display than others.

HTML (Hyper Text Markup Language)

As mentioned earlier, web pages are hypertext because they contain links to other pages. Web pages also contain formatting instructions so that browsers will know how to display the page. This is accomplished by adding “tags” in the text to provide instructions. The rules for the tags are defined by HTML, the Hyper Text Markup Language.

You can actually see the whole web page with all the HTML tags by using a browser option to save the web page and then opening the HTML file in a text editor (this option may not be available in all browsers).

As a simple example, let’s look at the first sentence on the Wikipedia page for Steve Jobs as displayed by the browser.

Steven Paul Jobs (February 24, 1955 – October 5, 2011) was an American entrepreneur, businessman, inventor, and industrial designer. He was the co-founder, chairman, and chief executive officer (CEO) of [Apple Inc.](#)

Here is the portion of the actual web page returned by the web server with the HTML tags shown in **violet**.

Steven Paul Jobs (February 24, 1955 – October 5, 2011) was an American entrepreneur, businessman, inventor, and industrial designer. He was the co-founder, chairman, and chief executive officer (CEO) of [Apple Inc.](/wiki/Apple_Inc)

The angle brackets contain the HTML tags.

	means start making the text bold
	means stop making the text bold
<a>...	is an anchor tag containing a link

So at the beginning of the sentence, **Steven Paul Jobs**

tells the browser to display the name in bold letters.

The link to another web page when you select [Apple Inc.](#) in your browser is defined in the **href** (hypertext reference) part of the anchor tag

```
<a href="/wiki/Apple_Inc">Apple Inc.</a>
```

The link is to the file `/wiki/Apple_Inc` on the same website as the `Apple_Inc` page we are looking at. If the link is to another website, then a full link description would be required, as in

```
href="https://www.apple.com/wiki/Apple_Inc"
```

HTML, Hyper Text Markup Language, is not a programming language. It is strictly a markup language. You cannot do any computations with HTML or write a program. Web pages containing only HTML are quite static.

However you can use scripting (programming) languages in a web page using the HTML tag **<script>**. Browsers on the client machine run the script defined in the HTML file to add dynamic features to web pages.

The most widely used scripting language is **JavaScript**. JavaScript is not related to Java, it is a completely different language. However, both are similar to the generic code used in the Income Tax example in Chapter 1.

Many web sites need access to a database, for example when a form is used for you to input and edit your user account information (name, address, etc.). Instead of a script that runs in the browser on the client machine, what is needed is software that runs on the server to get information from the database. When the web server retrieves a page requested by a browser, it first runs the server-side code before returning the web page.

PHP is one of the most widely used server-side languages. PHP originally stood for “Personal Home Page” but was later changed to “PHP Hypertext Preprocessor”.

PHP is embedded in HTML with the tag **<?php** followed by the PHP

code and the closing tag `?>`

When a web server retrieves an HTML file, it processes the PHP code before sending the file to the browser.

Here is an example of an HTML file that uses PHP to display “Phone:” followed by a phone number that is read from a database.

The PHP code for reading the database is not shown but it returns the value from the database in `$phone`. The PHP `echo` command outputs this value which becomes part of the HTML file that is returned to the browser.

```
...  
Phone:  
<?php  
    (php code to read from database) ;  
    echo $phone;  
?>  
...
```

After server-side PHP processing, the HTML file that is actually returned to the browser includes only

```
...  
Phone: (855) 836-3987  
...
```

Web Summary: When you request a web page, your web browser constructs an HTTP request and puts it into an IP packet along with the destination IP address of the web server. Routers on your local network, your ISP and the Internet deliver the packet to the web server. The web server retrieves the file containing the web page and returns it as an HTTP response in one or more IP packets. Your browser formats the web page according to the embedded HTML tags and displays it on your screen.

Other Protocols

Many other protocols besides HTTP are used on the Internet. Some are legacy protocols that were used before HTTP existed.

FTP (File Transfer Protocol) is a protocol that is still widely used to transfer files between computers. In a browser you may have a link to an FTP server. This will be indicated when the link has **ftp://** instead of **http://**. For people who use FTP regularly, there are numerous free FTP client software packages that you can install that are more flexible than using a browser.

Email (Electronic mail) is a protocol that has been used for a long time. While many people now rely on social media and various messaging apps on their mobile phones to communicate with friends and contacts, email remains important. All organizations, from large to small, provide email accounts for their employees. Email provides strategic communications internally between staff and is equally important for external communications with business contacts such as customers, suppliers and partners. For an advanced explanation of the Email protocol, see [Appendix 6. How Does Email Work?](#)

Mobile Apps use a variety of protocols. Many of them use HTTP and HTTPS but transmit data that are not HTML web pages. HTTP can be used with many different types of data. This is done by specifying the type of data in the HTTP header.

Custom Protocols may also be developed for specialized client-server applications. However, it is generally preferable to use one of the many standard protocols that are available. A standard protocol has had input from many experts in the field, has been tested and there is usually open source code available. A custom protocol takes a lot more time to design, code and debug.

Protocol Summary: Whatever the protocol, the process of transmitting data over the Internet is fundamentally the same as was described for the web protocol HTTP. The data is formatted to conform to the particular protocol, packaged into the data part of an IP packet with the source and destination IP addresses in the IP header and sent out by your computer's network driver. Routers in your network, your ISP and the Internet route the packet to the destination computer.

What is the Domain Name System (DNS)?

We have seen how important IP addresses are to the operation of the Internet. It is the IP address that makes it possible for one computer to communicate with another computer over an IP network.

DNS is a service that provides names that can be used instead of IP addresses. When an IP address is needed, a request can be sent to a DNS server to translate a name to an IP address, as we saw in the example of the browser request.

A DNS server does not have a list of all the domain names defined on the Internet, which is quite large. DNS is a hierarchical distributed database. A DNS server generally needs to send queries to other DNS servers in order to lookup a name. There is a large global network of DNS servers that make the domain name system work.

If you wish to use your own domain name, for example for a web site, there is a two step process. First you have to register the name and then you have to make it available on the Internet by setting it up in a DNS server.

Step 1. Registering a domain name

To use a domain name, you first have to register it to make sure it is not already being used. There are numerous websites where you can register and buy a domain name for an annual fee, typically around \$15. You can pick any name that has not already been registered, except for the last part, the top-level domain (TLD). Most people are familiar with **.com**. This is the most widely used TLD.

Even if a name is registered there may be no website using that name. Many organizations register a name to protect their brand or in anticipation of a new product so that someone else cannot use the name. You can check to see if a name is registered by using a **Whois** service, easily found with a web search.

The Internet originally had only six general top-level domains.

.com commercial businesses

.org	non-profit organizations
.net	networking companies, now just general purpose
.int	international organizations
.edu	U.S. higher education
.gov	U.S. national and state government agencies
.mil	U.S. military

In addition, there is a TLD for each country in the world. Here are a few examples.

.aq	Antarctica
.cn	People's Republic of China
.de	Germany (Deutschland)
.eu	European Union
.me	Montenegro
.es	Spain (España)
.tv	Tuvalu
.uk	United Kingdom
.us	United States

Each TLD establishes its own policy as to who is able to register a domain. In most cases there are no restrictions. For example, you could register a domain like **top10.tv** even if you are not a resident of Tuvalu.

No new TLDs were added to DNS for many years. Starting in 2010, countries were given the opportunity to add TLDs in their own language. Here are a few examples.

.中国	China
.الاردن	Jordan

In 2013, a major expansion of DNS was begun. For the first time, an organization could apply for a top-level domain of its choice. This is an expensive process so it is only practical for large organizations. Some companies that have registered TLDs are

.apple
.bbc
.lexus

Most of the new TLDs represent special interest groups or industry organizations, a few examples being

.green	the environment
--------	-----------------

.hockey	the sport
.press	publishing and journalism
.pub	bars and pubs
.vegas	Las Vegas
みんな	“everyone” in Japanese (owned by Google)

Step 2. Adding a domain name to DNS

Once you have registered a domain name, it is your responsibility to set it up in a DNS server. Otherwise the name will not be available on the Internet.

A domain name actually represents more than a single name. It is a domain, under which additional names can be created. Since a registered domain name is unique, any additional names you create in that domain will also be unique.

In our web browser example, the name of the web server was **en.wikipedia.org**. Wikipedia is available in several languages and this is indicated by the first part of the name, in this case **en** for English. So Wikipedia registered the domain **wikipedia.org** and then set up domain names in a DNS server for each language they support.

The following shows the DNS entries for the main Wikipedia website and the English, French and Japanese websites. Remember that DNS basically translates names to IP addresses, so the IP address of the web server must also be provided.

wikipedia.org	208.80.154.224
en.wikipedia.org	208.80.154.224
fr.wikipedia.org	208.80.154.224
ja.wikipedia.org	208.80.154.224

Note that Wikipedia is using the same machine for all these names. A single web server can handle multiple websites. In other cases, the IP addresses would be different for different names within a domain.

Larger organizations often manage their own DNS servers. Others use third party services that have a large global network of DNS servers on the Internet optimized for fast response to DNS requests.

For an individual or a small business, setting up a web server on the Internet with a domain name may be difficult if you are not technically oriented. There are numerous “web hosting” services that can do this for you. All you have to do is upload your web page files to their web server. If in addition you do not want to write the HTML code for your own website, then there are many web designers and consultants who can do everything to set up your website.

How is the Internet managed?

There is no one body or organization that manages the Internet. The Internet backbone is actually made up of many interconnected networks that are owned and operated by large telecom companies. Here are a few:

- AT&T (USA)
- Cogent Communications (USA)
- Deutsche Telekom (Germany)
- NTT Communications (Japan)
- Tata Communications (India)

These large networks connect to each other and sell access to other smaller Internet network providers who in turn sell access to ISPs and others.

Technical operations and management are done by each company within their network portion of the Internet and between companies at the access points connecting the networks. On a day to day basis, this is just business between companies.

However, there are a number of bodies that play an important part in establishing technical standards and policies for the Internet as a whole. Following are two of the more important ones.

Internet Engineering Task Force (IETF)

The IETF develops technical standards for the network protocols that are transmitted over the Internet (all the protocols covered in this chapter, and many more) and the special protocols that operate

between routers. It does not cover languages that run over the protocols. So HTTP is an IETF standard, but HTML is a standard published by the World Wide Web Consortium (W3C).

Hardware standards are developed by other groups, for example Wi-Fi and Ethernet are standards published by the Institute of Electrical and Electronics Engineers (IEEE).

At any time there are several IETF working groups covering different areas and issues. Each working group publishes a charter with a list of the documents it plans to produce and target milestone dates. Some groups work on modifications to existing standards and others work on proposals for new protocols.

The IETF is a very open forum, oriented to individuals, not companies or any other organizations. Anyone can participate and there are no fees. All you need to do is join the mailing list for the working groups in which you have an interest.

The mailing list is where comments on the draft documents are posted and discussed along with any other issues that anyone can bring up. Member of the mailing list receive email of every comment sent to the list and anyone can reply. The working group strives to achieve a consensus among all the opinions expressed and the document authors incorporate the changes into the next drafts.

The IETF holds three meetings each year in locations around the world to continue the work on the mailing list and try and reach consensus on the most critical issues. If you are not able to attend meetings, there are facilities for remote participation.

After a document has addressed all concerns, there is a comment and review process through various levels of the IETF and other Internet bodies until it is approved for publication as an RFC. RFC stands for "Request for Comments". The name goes back to the very earliest days of the Internet when it included a small group of people at universities and other research organizations who had informal arrangements to exchange comments and agree when a document was complete.

If you are interested in looking up an RFC, you can search for it at

https://www.rfc-editor.org/search/rfc_search.php.

For example, a search for “HTTP” will return about 100 RFC documents. However, they are not all standards or proposed standards, as indicated by their Status. Many are “Informational” supporting documents.

Internet Corporation for Assigned Names and Numbers (ICANN)

The most important functions of ICANN are to establish policy and provide technical management of IP addresses and the Domain Names System.

On the technical side, IP addresses are managed by Regional Internet Registries that operate in five regions of the world. Any organization that needs IP addresses to operate on the Internet must apply to the appropriate registry to purchase a block of addresses for an annual fee.

One key technical function of ICANN is the management of the DNS “root” server. This is the master server where all the top-level domains (TLD) on the Internet are defined.

One of the mandates of ICANN is to protect the stability of the Internet. There are technical committees within ICANN who study proposed changes to make sure the existing IP address and DNS infrastructure on the Internet will not be adversely affected.

Management of Domain Names is a bit more complicated than management of IP addresses. While you can typically register a domain name on many different web sites, there is only one company that maintains the master database of all domains for a particular TLD. This company is known as a Registry Operator. Buying a domain from a web site such as GoDaddy.com is equivalent to a retail purchase while the Registry Operator is the wholesale supplier.

Each TLD is managed by a Registry Operator under contract with ICANN. Many Registry Operators have an ICANN contract for more than one TLD. ICANN contracts are an open bidding process and they

come up for renewal typically every five years.

Some policies about domain names have been controversial. For example, the expansion of domain names in 2013 to allow custom TLDs was implemented only after years of discussion, review and planning.

ICANN, like the IETF, is an open forum in which participation is encouraged from anyone who has an interest in Internet issues related to domain names and IP addresses. There are three meetings each year rotated through different regions of the world to encourage global participation. Between meetings there are various committees that investigate active issues and issue reports for comment. There are members on the ICANN executive and committees from many countries in the world.

4. How Do Smartphones Work?

What is a Smartphone?

A Smartphone is a mobile (cell) phone with a computer that allows you to do many of the things you can do on a computer, such as

- access the Internet with a web browser
- send and receive email
- manage appointments in a calendar
- play music, videos, games
- download software for other applications (apps) that did not come with the smartphone.

There are many smartphone apps that provide additional services, such as listening to streaming music, viewing public transit schedules or using the smartphone as a flashlight and a magnifying glass.

Many apps provide an alternative to using a web browser for a particular web service. The app is designed for the smaller screen mobile device and is easier to use than a web browser on a mobile phone. For example, all the major social networking services provide mobile apps.

Smartphones also have a number of additional capabilities beyond what is available on a computer. These are valuable on a mobile device and allow you to

- detect your location, show it on a map and locate nearby services such as taxis and restaurants.
- use a built-in camera to take pictures that can be stored on the phone and/or sent to other people or posted on social media.

Most smartphones are an Apple iPhone or an Android phone like Samsung.

Smartphone Hardware

A smartphone is basically a computer that has special input and output devices and corresponding software that together provide the smartphone capabilities.

The following figure shows the basic computer model introduced in *Figure 1-1* with the generic Input and Output (I/O) boxes replaced by specific input and output devices found in a laptop computer (gray) plus the additional devices found in a smartphone (green).

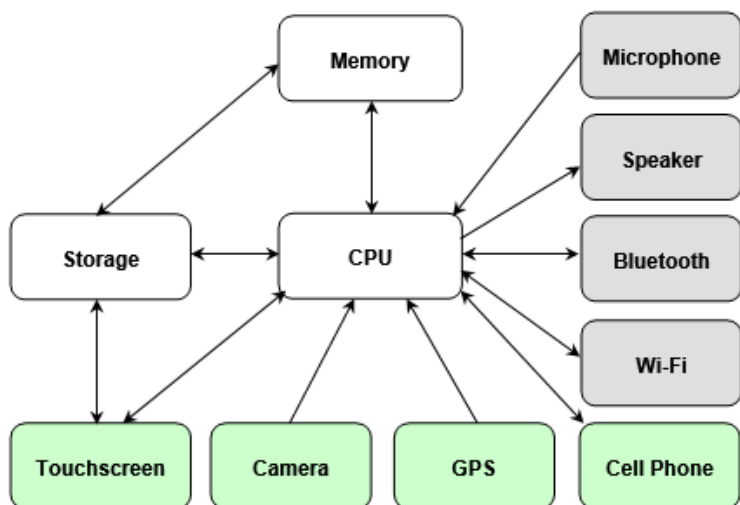


Figure 4-1. Smartphone Hardware

Smartphone manufacturers generally make only a small number of the hardware components that go into a phone. Third party manufacturers provide most of the hardware, including the CPU. Apple is one of the few companies that makes its own smartphone CPUs.

Microphone

The microphone component is required for the phone but is also used

by many other apps that use speech recognition so that you do not have enter information by typing. Most widely known such apps are Siri, Google Search, Mail and various text messaging apps.

Speaker

The speaker is required for the phone but can also be configured in the Settings to sound various alerts and notifications. It is also important for many apps such as music players, video, radio and TV.

Touchscreen

The touchscreen hardware is embedded in the glass screen with associated integrated circuits that recognize when you touch the screen and when you move your finger in various directions. The location and motion are translated into numbers and communicated to the software so that it knows which icon you have selected on the home screen or what key you have pressed on the keyboard.

Camera

Smartphone cameras are built with multiple plastic lenses that focus the image onto an image sensor. The sensor contains a photo detector for each pixel. The captured analog information about color and brightness are transformed by analog to digital hardware to numbers that can then be stored as jpeg (.jpg) files in the smartphone disk memory.

There are also apps that use the camera and its associated flash for purposes other than photos, such as a flashlight and apps to report your heart rate by measuring the reflected light from your finger.

The Network Components

The Cell Phone, Wi-Fi, Bluetooth and GPS components provide access to different data communication networks. They all use radio signals. There is no interference between them because they all use different

radio frequencies that are set by standards and regulations.

While shown as separate logical components, they generally share radio hardware. In fact, some more recent advanced hardware supports Cell, Wi-Fi, Bluetooth and GPS on a single chip that includes its own CPU.

Cell Phone

The main hardware in the Cell Phone component is an antenna, a radio transmitter and receiver to send and receive radio signals, an amplifier to increase the strength of those signals and a digital signal processor to convert the analog radio signals to digital binary data (0 and 1) that is needed for computer processing.

Wi-Fi

The Internet is a large inter-connected network of wired segments. In home and office installations, it is often inconvenient to run wires from computers to a router. So a wireless protocol was developed as an alternative to the wired Ethernet (see *Figure 3-1 Local Area Network*).

Wi-Fi uses radio frequency bands set by standards within different regions and countries. These standards are not consistent, so a Wi-Fi router purchased in one country may not work in another. A typical home router has a range of about 20 meters. Wi-Fi routers used in larger offices or public locations such as an airport have a greater range.

Wi-Fi routers need to be configured before use. It is important to configure the type of security that your home computers and smartphones support. Radio waves are not beamed directly at the router but spread out in all directions. Without configuring security, someone in a neighboring apartment or a car parked in front of your house can use your Wi-Fi router to connect to the Internet. Even more serious is that they can record all your Internet traffic and see private information you transmit, such as bank account numbers and

passwords. By configuring security, access to the Internet requires a Wi-Fi password and the transmissions are encrypted and cannot be deciphered by an eavesdropper.

Public Wi-Fi generally does not use encryption and has several other security risks, so data sent from your device may be readable by others. You should never use your smartphone, or laptop, over a public W-Fi connection with a web site or an app that transmits private information, such as an account login. However, with a smartphone, you can turn off Wi-Fi and transmit data over the cellular network, which will be encrypted.

Bluetooth

Bluetooth is a wireless networking standard that uses radio waves for exchanging data with nearby devices, typically less than 4 meters away. Each device has to have Bluetooth support.

One of the key uses of Bluetooth is for “hands-free” talking on smartphones. This can be done with a Bluetooth wireless headset or a Bluetooth “Car Kit” that connects the smartphone to the car’s audio system. Operating a mobile phone while driving is illegal in most states and countries, but hands-free use is generally legal and safe.

Another use of Bluetooth is when you need an Internet connection for your laptop computer in a location where there is no Wi-Fi. If you have a cellular data plan on your smartphone, you can share it with your laptop. You use the smartphone option to set up a hotspot Bluetooth connection between your smartphone and laptop (also called tethering). Internet connections from your laptop are sent over the Bluetooth connection to the smartphone where they are forwarded to the Internet over the cellular network. Using this feature often requires it to be enabled by your mobile provider who may charge an extra service fee.

GPS

GPS (Global Positioning Systems) is a radio receiver that reads signals

from GPS satellites that continually broadcast their position and time. The receiver's position is four unknown quantities — three position coordinates (x, y, z) and the satellite time. Your smartphone needs four equations to solve for these four unknowns. It does this by getting position readings from at least four satellites. The accuracy of a calculated GPS position is 4 meters or better.

GPS was developed by and restricted for use by the U.S. military during the cold war. In 1983, a Korean Air Lines plane on route from New York City to Seoul entered prohibited airspace because of navigational errors and was shot down by the Soviet Union. After this, the U.S. made GPS available for civilian use to try and avoid such disasters.

The Russian military developed a similar system, GLONASS (GLObal NAVigation Satellite System), which is also available for civilian use. Many smartphones support GLONASS as well as GPS for improved location detection and accuracy under some circumstances, such as when some GPS satellites are obscured by tall buildings in dense urban areas.

Smartphone Software

One of the major functions of smartphone operating systems is to manage all the I/O devices shown in *Figure 4-1 Smartphone Hardware*. In addition, smartphones come with a number of basic applications or Apps; for mail, web browser, contacts and calendar.

iOS is the operating system developed by Apple for the iPhone. Most other smartphones use Google's Android operating system, which is a more generic system that can run on different smartphone hardware.

In order to encourage development of iPhone Apps by independent developers, Apple provides an iOS Software Development Kit (SDK) that has custom software languages and tools to make development easier. Apple created the Objective-C language specifically for developing software in iOS. In 2014, a simpler language, Swift, was added as an alternative. Since iOS is also used for the iPad, most apps developed for the iPhone will work with little or no modification for

the iPad.

Apple also built an App Store to distribute apps. Users can search for applications and download them. This was a boon for small developers. Distributing software can be a difficult and an expensive process. With a centralized mechanism such as this, developers just need to register with Apple and upload their software to the App Store. Apple takes a 30% commission on each sale but this has proved to be no deterrent for developers to use this service. Independent apps have been extremely successful. As of 2015 there were 1.5 million apps in the App Store.

Google and other smartphone manufacturers have provided similar services. There are also about 1.5 million apps in Google Play, the app store for Android phones. Apps are usually developed with the Java programming language using the Android SDK (Software Development Kit).

The Cellular Phone Network

The telephone component in a smartphone is not the technology used in home or office telephones, which are landline phones. They must be plugged into a wired phone network.

Cellular phones are wireless and use radio. Fundamentally, a cellular phone communicates with the nearest cell phone tower using radio signals.



Cell Phone Tower

The radio equipment in the tower forwards the cell phone signals to a mobile switch, which has connections to the regular phone network for voice calls and to the Internet for data.

Phone networks are operated by different phone companies. *Figure 4-2* shows several networks: two cell phone networks (green and orange), a landline phone network (blue), the PSTN (gray) and the Internet cloud.

The PSTN is the “public switched telephone network”, the global telephone network that provides connections between phones in different networks. The mobile phones are shown as small green rectangles and their service provider is the green cell phone operator.

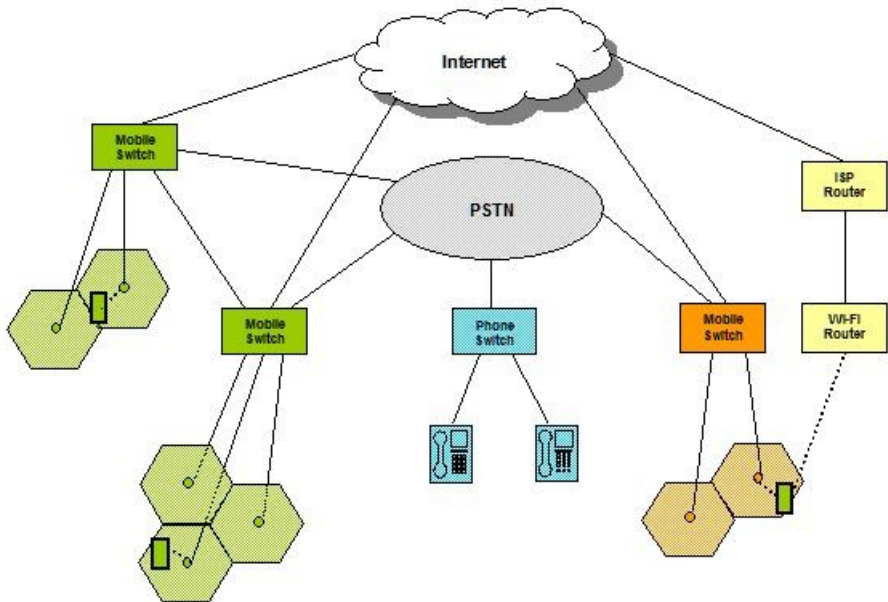


Figure 4-2. Phone Network

In a cellular network, mobile phones communicate with the nearest cell phone tower (the small circle in the middle of each cell) via wireless radio signals. The cell phone tower has radio receivers and transmitters of sufficient power to reach any point within its cell. Cell towers generally have a range from .5 miles in urban areas to 5 miles in rural areas. Cells are hexagonal areas that overlap at the boundaries so there can be a smooth handoff of a call from one cell tower to another as someone travels across cell boundaries.

The cell phone tower is connected to a mobile switching center. This connection is a high capacity phone line (fiber optic cable) in built up areas or microwave transmission in more remote areas.

The mobile switch is connected to other mobile switches within the operator's network (shown in the green network), to the PSTN for voice calls and to the Internet for data. Since these connections carry a lot of traffic, they are generally large capacity fiber optic cables. These

interconnections allow a mobile phone to call other mobile phones and landline phones pretty well anywhere in the world.

Mobile phones within the green network can generally make calls to each other without long distance charges. Calls to landline phones or to mobile phones in other cell phone networks generally have long distance charges.

If a mobile phone is not able to reach a cell tower operated by its service provider, it may be able to connect to a cell tower of a different service provider. This is shown by the green mobile phone in the orange cell phone network. The orange network will charge the green network for voice calls, text messages and data used by the green mobile phone. The green network will pass these charges on to the subscriber of the green mobile phone as “roaming” charges. Mobile phones generally show the name of the service provider to which it is connected so that you can quickly see when roaming charges are in effect. Since roaming charges can mount up quickly, many smartphones disable roaming by default and you have to enable it in the phone settings.

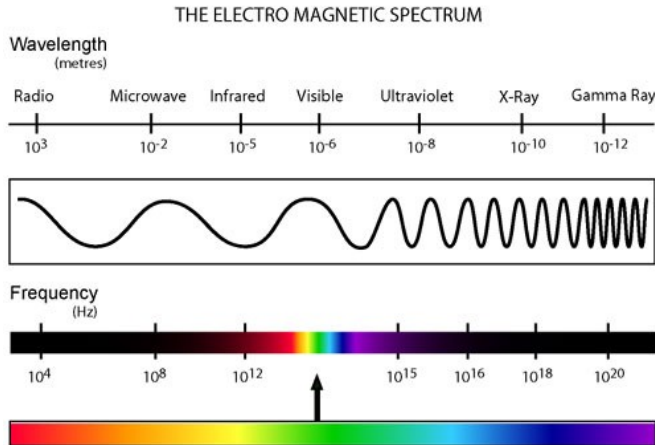
A smartphone may need to transmit data over the Internet from a web browser, mail, social networking or any of the many other apps that use the Internet. Smartphones have two ways to transmit data: by using the cell phone network or by using a Wi-Fi connection (yellow) to a Wi-Fi router that connects to an ISP.

Transmitting data over the cell phone network is generally an additional cost to the basic mobile phone voice and text message service, so smartphones will generally attempt to use a Wi-Fi connection first and only use the cell network if Wi-Fi is not available. Using cellular data may be disabled by default in the phone and you may have to turn it on before it can be used.

How does radio work?

Radio waves are one type of electromagnetic radiation, which is the radiant energy released by certain electromagnetic processes. The following diagram shows the complete electromagnetic spectrum with

the range of all of the frequencies and wavelengths of electromagnetic waves. Note that visible light that humans can see is only a small band of the electromagnetic spectrum and that the frequency determines the color.



Radio waves can be made to carry information, such as voice, by varying a combination of the amplitude, frequency and phase of the wave.

For example, if your favorite radio stations is 99.1 FM, then you are listening to radio frequency 99.1 MHz (million hertz) that is **F**requency **M**odulated.

But when you listen to 860 AM, then you are listening to radio frequency 860 kHz (thousand hertz) that is **A**mplitude **M**odulated.

There are a lot of uses for radio waves:

- public radio and television
- telegraph
- radar - vehicle speed measurement, air traffic, weather
- cellular telephones
- Global Positioning Systems (GPS)

- wireless computer networking (such as Wi-Fi, Bluetooth)
- cordless home phones
- garage door openers
- microwave ovens

To avoid interference from different devices, there are international standards and national regulatory agencies (such as the FCC in the U.S.) that set the allowable frequency band for each of these uses.

There have been several generations of cellular phone technology. To keep these systems from interfering with each other, the overall cellular frequency band has been divided so that each cellular technology has its own band. Smartphones often support more than one phone technology, but not all. That is why you have to get a phone that is approved by your network provider; not all mobile phones will operate on their network.

How does a smartphone connect to the cellular network?

The information in this section is based on a unified standard that removes many incompatibilities between earlier generation cell phone networks. It is a fourth generation technology (4G) called LTE (Long Term Evolution). Some key improvements in LTE over earlier generations are more optimal use of the radio frequency and increased speed for data, typically 4 times faster than 3G for accessing the Internet.

When you turn on your cell phone, it reads identity information from the SIM card (“Subscriber Identity Module”) inside your phone. This card has information that was stored by your cell phone provider and provides the identity code of your cellular network, your account number and your cell phone number.

Cell towers send out continuous broadcast messages on specific control channels (radio frequencies) with the identity of the cellular network and the hexagonal cell.

The phone scans the LTE broadcast channels for messages from your

cell phone network. There may be messages from more than one cell tower, so the phone will select the cell that transmitted the strongest signal. If no broadcasts from your cellular network are found, then the phone selects the strongest signal from other cellular networks and you are roaming. If no LTE broadcasts are received, then your phone will try to connect with an older cell phone technology and locate a 3G or 2G service. If all of this fails, then you are out of range of any cell tower and your phone will report “No Service”.

If your phone finds a cellular network, it sends a connection request containing your identity information to the cell tower. The cell tower forwards the request to the Mobile Switch where the identity information is validated against an administrative database. If your cell phone account is valid, the switch sends back an acceptance. Your cell location is also saved in a database so that when someone tries to call you, the cell network knows which cell tower to use.

When your phone receives the successful connection reply, your phone is in service. The signal strength is generally displayed on your phone along with the name of the cell phone network to which it is connected.

There are several more exchanges between the phone and the cellular network to complete the setup of network resources and security parameters. Finally the phone receives a message specifying a data channel — the radio frequency and power for transmitting and receiving. Your phone can now use the cellular network to send and receive voice calls, text messages and Internet data. Everything sent over the cellular channel is encrypted so that your voice conversation is heard as noise by any eavesdropper listening to the radio signals in the cell.

Popular film and television often misrepresent the security in the cell phone network by showing people listening in on cell phone calls. This can only be done if an investigator obtains a court order for a specific cell phone number and presents it to the cell phone operator. The operator then has to configure this in the Mobile Switch. The normal processing of the switch decrypts the traffic from the cell phone (voice and data) before forwarding it over the next stage in the phone

network. It is at this point that calls from that cell number are made available to the investigator.

Appendix 1. CalculateTax Program

This is the complete program for calculating the income tax for all of the rows in the Tax Worksheet in Chapter 1.

The lines of the program that get used for the case when Taxable Income is \$125,000 are highlighted.

The first line provides a name for the program and the input in parentheses.

```
1. CalculateTax( Income )
2.   if Income < 100,000
3.     Error("For income less than 100,000 use the Tax Table.")
4.   else if Income <= 190,150
5.     // row 1 of tax worksheet, tax rate 28%
6.     Tax = Income * .28 - 6,963.25
7.   else if Income <= 413,350
8.     // row 2 of tax worksheet, tax rate 33%
9.     Tax = Income * .33 - 16,470.75
10.  else if Income <= 415,050
11.    // row 3 of tax worksheet, tax rate 35%
12.    Tax = Income * .35 - 24,737.75
13.  else
14.    // row 4 of tax worksheet, tax rate 39.6%
15.    Tax = Income * .396 - 43,830.05
16.  return Tax
```

Note that this program will work for any Taxable Income, not just our particular value of \$125,000. General programs like this are more useful than a specific program, because it can be used for any tax return.

When you run (“execute”) the program, providing the Taxable Income in parentheses, the program returns the calculated tax.

```
CalculateTax(125,000)
28,036.75
```

Let’s explain in a little more detail how this program works. The computer follows the instructions in the program line by line.

Line 1 stores the input value 125,000 in Income, which is a location in memory. The computer then proceeds to line 2.

Line 2 is not true, since 125,000 is not less than 100,000, so the computer skips all lines until it reaches an "else" statement.

Line 4 is true since 125,000 is less than or equal to 190,150 so the computer continues on to line 5.

Line 5 is just a comment line, as indicated by the // at the beginning of the line. The computer just continues to line 6.

Line 6 calculates "Tax" according to the first row of the Worksheet. Tax is a location in memory. Then the computer skips to the end of all the rest of the else statements (Lines 7 to 15) and continues at Line 16.

Line 16 simply returns the calculated value of Tax.

Appendix 2. CalculateTax Machine Code

This is the machine code translated from the CalculateTax Program in Appendix 1 by the compiler. To follow step by step how this code works, read the comments to the right of the ; comment symbol on the highlighted code that shows the path of the program when Income is 125,000.

```
CALCULATETAX( INCOME )
    MOV EAX, INCOME      ; Copy the data from memory location INCOME
                          ; into the EAX register
    CMP EAX, 100000      ; Compare the number in EAX to 100,000
    JL ERROR             ; If less, then jump to line ERROR
    CMP EAX, 190150      ; Compare the number in EAX to 190,150
    JLE R1               ; If less than or equal, then jump to line R1
    CMP EAX, 413350      ; Else compare the number in EAX to 413,350
    JLE R2               ; If less or equal, then jump to line R2
    CMP EAX, 415050      ; Else compare the number in EAX to 415,050
    JLE R3               ; If less or equal, then jump to line R3
                          ; Else calculate tax using tax rate 39.6% (Row 4)

    MUL EAX, 396
    DIV EAX, 1000
    SUB EAX, 43830
    JMP END

ERROR:                   ; Taxable income must be 100,000 or greater
    MOV TAX, -1          ; Set error indicator in memory location TAX
    RET                  ; Return (exit) from the program

R1:                      ; Row 1, calculate tax using tax rate 28%
    MUL EAX, 28          ; Multiply the number in EAX by 28
    DIV EAX, 100         ; Divide EAX by 100 for 28%
    SUB EAX, 6824        ; Subtract 6824 from EAX
    JMP END              ; Jump to line END

R2:                      ; Row 2, calculate tax using tax rate 33%
    MUL EAX, 33
    DIV EAX, 100
    SUB EAX, 16470
    JMP END

R3:                      ; Row 3, calculate tax using tax rate 35%
    MUL EAX, 35
    DIV EAX, 100
    SUB EAX, 24737

END:
    MOV TAX, EAX         ; Copy EAX to memory location TAX
    RET                  ; Return (exit) from the program
```


Appendix 3. Binary Numbers, Bits and Bytes

In the decimal number system that we learned in grade school, numbers have ten digits, 0 through 9. Each column represents the powers of 10.

The number 238 is 2 hundreds + 3 tens + 8 ones:

column value	100	10	1
decimal number	2	3	8

$$= 200 + 30 + 8 = 238$$

In the binary number system, digits are called a bits and are just 0 and 1. Each column represents the powers of 2 instead of the powers of 10. The binary number 1001 is 9:

column value	8	4	2	1
binary number	1	0	0	1

$$= 8 + 1 = 9$$

Each computer memory location can store one byte. A byte contains 8 bits.

So what is the largest number that can fit in one byte? To find out, look at the largest binary number you can store in a byte, with a 1 in each of the 8 columns and convert to decimal.

binary number	1	1	1	1	1	1	1
column number	8	7	6	5	4	3	2
column value	128	64	32	16	8	4	2

$$= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$$
$$= 255$$

If you need a number larger than 255, you have to use more than one byte of storage. For example, four bytes (32 bits) can hold an integer number as large as 4,294,967,295.

For really large numbers and/or numbers with decimals, your program would need to use a different storage format from integers, called “floating point”. Floating point is based on scientific notation for numbers. A bank balance of say 7,382.64 is written as 7.38264×10^3 in scientific notation. It would be stored in memory in floating point format as

3	738264
---	--------

Appendix 4. Machine Instruction Sets

The number of machine instructions in an Intel CPU processor chip is more than 500. However, all these instructions can still be grouped into one of the four categories Copy, Calculate, Jump and Special instructions.

The reason there are so many instructions is that there are variations of each basic instruction to handle different data formats, such as integers and floating point, and different data sizes, such as 1, 2, 4, 8 or 16 bytes. So for example, there are about 35 different instructions for doing addition.

By comparison, the IBM System/360 mainframe computers introduced in 1964, had a pretty similar set of instructions. They had a total of 150 instructions that fit into the same four categories.

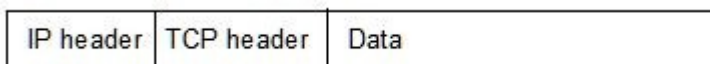
The only real functional difference is that the IBM instruction set did not have any of the advanced mathematical instructions for square roots, logarithms, trigonometry and encryption/decryption.

Most of the difference between the 150 instructions on the IBM and the 500 instructions on Intel is due to the larger number of variations of basic instructions on the Intel chip. For example, there are a total of 14 Add instructions on the IBM compared to 35 on Intel.

Appendix 5. Internet Routing

Chapter 3 covered the basics of how the Internet works. Refer to *Figure 3-2 The Internet*.

The Internet is a global “inter network” that connects other networks. It is built with large routers connected by high speed links. Data is sent over the Internet by splitting it into packets of 1500 bytes that conform to the Internet Protocol (IP). Each packet contains an IP header preceding the data that contains the IP addresses of the source and destination.



Routers forward each packet to the next router according to the fastest route. Most Internet applications and mobile apps add Transmission Control Protocol (TCP) information in each packet. TCP can detect if packets are discarded by the network due to congestion or to errors on a communication link and resend the data that was not received at the destination.

It is important that IP addresses on a network are unique, otherwise your request may go to the wrong computer. On the Internet it is necessary to apply for a “public” IP address from the official Regional Internet Registry which can make sure that duplicate addresses do not get assigned.

However, your computer is not on the Internet, it is on your local network, so you do not need to apply for a public Internet IP address. Instead your computer is assigned a unique “private” IP address on your network when it starts up. The ISP routers change the private source IP address in the IP header to one of their public IP address so that the packet can be sent over the Internet.

An IP address is a sequence of four numbers split into a network address and a machine address. Here is an example of an IP address showing the network address in green and the machine address in that network in yellow.

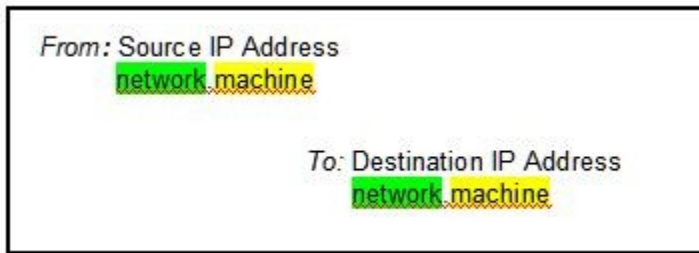
184.150.153.178

The network address is not always the first two numbers. The boundary may be anywhere after the first number. Each number is from 0 - 255 to fit into one byte, so the total size of an IP address is four bytes.

This is analogous to a mailing address, which contains a city address (network) and a building address (machine) in that city, as in

178 Spear Street
San Francisco, CA

Here is a picture of an IP packet envelope containing some data.



The Internet is similar to the postal system. Your letter is sent to the **destination city** by high speed transport (analogous to the Internet), such as plane or train, perhaps via other cities. When it arrives at the destination city, it is then delivered to the **destination building** by the local mail service (analogous to the local area network router).

It is difficult to remember numeric addresses, so there is a way to use a name instead of an IP address. This is the Domain Name System (DNS). If a **domain name** has been set up for a computer, then in most places you can use it instead of the IP address. For example, when using a web browser you can specify the web server by name, as in **google.com** as well as by its IP address 184.150.153.212.

However, the protocol for an IP packet only allows numeric addresses to keep the IP header as small as possible — IP addresses are four bytes which is smaller than a domain name. When an IP address is needed, a request can be sent to a DNS server to translate a name to

an IP address. DNS was covered in Chapter 3.

Now what is the fastest route on the Internet and how is it determined?

The links connecting routers have different speeds. A big router may have links that can send data at about 10 GB per second (GB is 1 billion bytes = 1000 MB), ten times faster than another router with 1 GB links. Selecting routes with the highest bandwidth results in the fastest route, even if it is not the shortest distance. This is analogous to picking a driving route in your car to use a multi-lane highway instead of slower city streets even if the distance is longer.

So how do routers find the fastest route?

Routers regularly transmit their routing information to all the routers to which they are directly connected. The routing information is a routing table with a list of all the networks that the router can reach and the speed of each route. Each router uses a standard Internet algorithm to compare the routes from all the adjoining routers, select the fastest path to each network and update its own routing table.

When a router receives a packet, it can quickly look up the destination network in its routing table and find out where to send the packet. Here is an example of part of a routing table.

Network	Link	Speed
184.109	2	10
184.150	3	5
207.164.34	3	1

So if the router receives a packet that has a destination IP address of **184.150.153.178**, it looks up the network **184.150** in the second row of the table and sends the packet out on link 3 to the next router.

The routing tables are updated frequently and can change. If a link between two routers becomes congested or goes down, the fastest path of a packet can be different from the route of the previous packet.

A frequent cause of packets being lost is transmission errors due to interference or noise on a communication line in a network. This can happen on wired Ethernet links but is much more common on radio

links like Wi-Fi.

When computers and routers receive a packet from a network link, they perform a mathematical check that determines if there are any errors in the packet. If an error is detected, the packet is discarded, which is the lesser of evils. The impact of forwarding a packet containing an error is unpredictable and could be disastrous. It could appear as an innocuous typographic error in a text document but it could also be a wrong number in an important financial or medical report.

When packets have been discarded in the network, it is up to the sending machine to retransmit the missing data. This is done with TCP (Transmission Control Protocol).

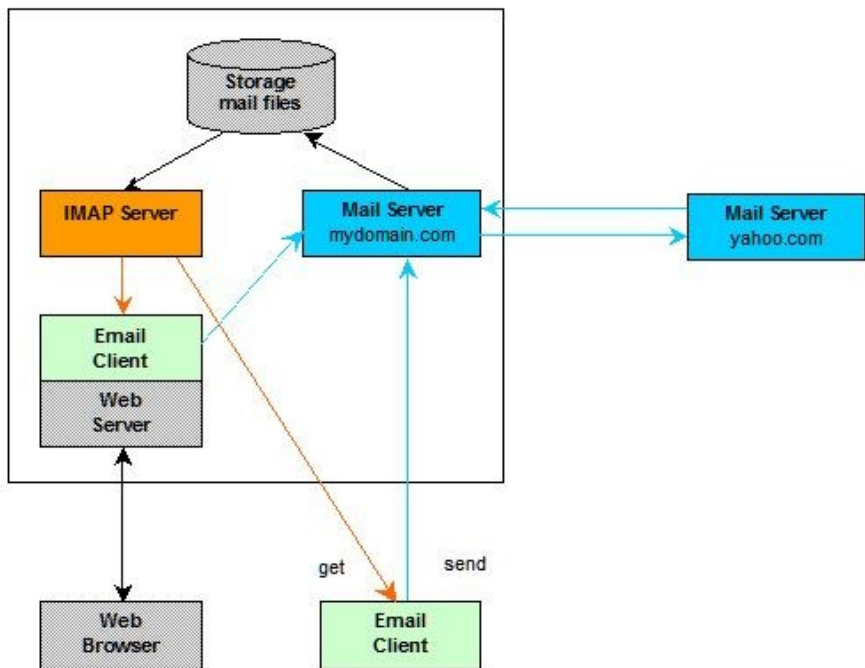
When using TCP, the first step is to make a connection to the destination machine, analogous to making a phone call and waiting for the other end to answer.

Once the connection is established, the sending computer puts the total number of bytes **sent** (sequence number) in the TCP header of every packet it sends. When the receiving computer sends back a reply, it puts the total number of bytes it has **received** (acknowledgment number) in the TCP header. If the number of bytes received is less than the number sent, the sending computer knows that data has been lost and it will resend the missing data by starting from the received number.

Appendix 6. How Does Email Work?

The diagram below shows an example of an email system based on Internet standards and open source code. There are also other proprietary systems that differ somewhat from this, such as Microsoft Exchange.

There are three main components that communicate with each other, an **Email Client (green)**, **Mail Server (blue)** and **IMAP Server (orange)**.



Email Clients provide the user interface for reading, composing and sending mail. An Email Client may be a web application that you access with your web browser (such as gmail, yahoo mail) or it may be software installed on your computer (such as Microsoft Outlook, Mozilla Thunderbird) or a mail app on your smartphone.

The **Mail Server** sends mail received from the Email Client to other Mail Servers. It also receives mail from other Mail Servers for delivery to your email account. These communications use the SMTP protocol (Simple Mail Transfer Protocol) shown in **blue** lines.

The **IMAP Server** (Internet Message Access Protocol) gets mail messages from Storage when requested by an Email Client. It uses the IMAP protocol shown in the **orange** lines.

The architecture of the email system is quite flexible. All three components may run on a single machine, as indicated by the block box outline. For email systems with high traffic, the components may be distributed on separate machines. The Internet is not shown in the diagram as connections may occur over an organization's internal network as well as over the Internet.

Most email services actually use the Secure version of these protocols, IMAPS and SMTPS. They operate in a similar manner as HTTPS and encrypt the data portion of the IP packet. This provides confidentiality of email over the network (**orange** and **blue** lines only). BUT email messages in Storage are generally not encrypted, so most email systems are not confidential end-to-end.

There are quite a number of third party email security solutions that encrypt the body of the email message so that it remains secure in storage. These systems require that sender and receiver all use the same software and users have to exchange security keys with each recipient before they can send or receive secure email. Secure email requires a lot of discipline and management by users and is generally not widely used except by organizations that require it for highly confidential communications.

Here is a step by step example of sending an email to your friend Joe at email address joe@yahoo.com

Open your Email Client (the Mail app in a smartphone) and login to your email account me@mydomain.com

Compose a new message and enter joe@yahoo.com on the **To** line.

Select **Send**. The Email Client makes a TCP connection to your Mail

Server and transmits your mail with a series of messages conforming to the SMTP protocol. Each SMTP message is put into the data portion of an IP packet. The details of creating a IP packet and routing it over a network to the destination are pretty much the same as was described for HTTP in the **Browser request** section.

Here's an actual SMTP exchange showing each message the Mail Client sends and the response from the Mail Server.

Mail Client

Mail Server

HELO mydomain.com

250 Hello mydomain.com

MAIL FROM: me@mydomain.com

250 Ok

RCPT TO: joe@yahoo.com

250 Ok

DATA

354 End data with <CR><LF>.<CR><LF>

Subject: Friday

Hey Joe,

Let's meet at Shoeless Joes at 6pm Friday.

Bring your shoes.

.

250 2.0.0 Message accepted for delivery

The Mail Server needs to determine where to send the mail. It does this by sending a query to a DNS Server to get the IP address of the Mail Server for the domain yahoo.com. It makes a TCP connection to the server at that IP address and sends your mail using a series of SMTP exchanges similar to the above.

After Joe uses his Mail Client to read your mail, he can send a reply in a manner similar to how you sent the mail. When his reply is received by your Mail Server, it is simply written to Storage.

It is up to your Email Client to connect to the IMAP Server to check for mail. Most Email Clients automatically check for new mail every 10 minutes or so, but you can change this or instruct the Email Client to refresh or get mail at any point in time. The IMAP Server reads your mail from Storage and returns it to the Email Client. This is done with an exchange of messages that conform to the IMAP protocol, where

again each message is put into an IP packet so it can be sent over a network.

As smartphones became more widely used for mail, continuously checking the IMAP Server for new mail was costly in terms of network traffic and battery usage. Alternatives to send new mail to the mobile phone as it arrived were developed by different mail services. Most use a separate connection that sends a notification to the Email Client when there is new mail. Upon receiving the notification, the Email Client connects to the IMAP Server to get the mail. Due to the many different methods for doing this, a particular smartphone may not support this feature with all mail services. Smartphone settings allow you to turn this feature (Push or Sync) on or off.

About the Author

After completing a B.Sc. in psychology and mathematics at McGill University in 1967, I became interested in seeing how computers might be used to model human thinking and learning. Based on a lunch time course in Fortran, I was hired into a training program at Univac in London, U.K. Univac was the company that had built one of the first general purpose commercial computers and was at that time the second largest computer company in the world, after IBM. After a stint as the operator of a mainframe computer in a data center, I joined the Univac support team at the National Engineering Laboratories near Glasgow. There I worked on configuring and building versions of the operating system to test the time-sharing support for Teletype terminals.

I returned to university to study “Artificial Intelligence” and completed a Masters degree in Computer Science at the University of Wisconsin. Following this, I worked on a large IBM mainframe computer at Simon Fraser University in Vancouver, B.C. where I provided consulting and programming services for faculty research from literature to geography to genetics, using a variety of computer languages and statistical analysis packages.

My interest in doing research shifted to an interest in building applied systems. I joined the Toronto branch of a service company that sold computer time to customers. It used Xerox Sigma computers, one of the best timesharing machines then available. I promoted the use of APL (“A Programming Language”) and used it to develop a number of financial programs that customers used. APL was a very interesting computer language, unlike any other. It operated on arrays of data and had a large set of special operators, many of which were Greek letters.

Shortly after the IBM PC was released in 1981, I joined a partner in a start-up company. We were literally “two guys working in an attic”. We developed an integrated financial planning package that achieved moderate success when combined with consulting services, but ultimately it could not compete with the widespread popularity of spreadsheets.

I moved on to work at a company that had developed a computer terminal that could connect to different mainframe computers using different network protocols. These terminals were a precursor to the Bloomberg terminal and were widely used by the Canadian stock brokerage industry to provide stock

quotes and financial information. One of the many products I worked on was an IP router that ran in an IBM PC. It forwarded IP packets from a local area network to another IP network over an X.25 connection. At the time, X.25 was a public global network that was more widely used than the Internet. X.25 was a standard that had been developed by the telephone industry and X.25 networks were provided by telephone companies in many countries.

When the Internet started to take off, I joined a start-up in 1966 that had built one of the first firewalls. A firewall is a computer positioned between the Internet and an internal network. It protects the network from hackers on the Internet by blocking unauthorized access to the organization's internal network. Firewalls were a new idea at the time but are now used by every organization with an Internet connection and are also embedded in home routers.

I then worked as a software engineer at a company that made network switches for Storage Area Networks (SAN). SANs are high speed networks that allow many servers to access large shared disk storage systems. SANs used a network protocol called Fibre Channel but there were new standards being developed at the IETF to support storage traffic over IP networks. I worked with hardware engineers on the research and design of a gateway that would transmit storage traffic between IP networks and Fibre Channel networks. This project included numerous trips to Silicon Valley to work with start-ups that were developing hardware chips for IP storage.

Following this I joined a start-up that was developing an enterprise software security system to protect confidential data on disks and mobile storage. Working in a small company I took on many roles, including drafting several patent applications.

My last position before retiring was research and development for a domain name registrar that managed .org and several other top level domains. I developed prototypes for several research projects using open source code enhanced with custom code using Amazon cloud servers. A "Big Data" project that analyzed DNS traffic was done on a cluster of Linux servers. Other projects were prototypes for new Internet standards being developed by the IETF. I participated in the Internationalized Email working group that extended the email standard to support email addresses in any language and another working group that developed a new protocol to replace the "Whois" domain lookup service.

Ernie Dainow
Toronto, February 2017