COMPUTER ARCHITECTURE AND SECURITY

COMPUTER ARCHITECTURE AND SECURITY

FUNDAMENTALS OF DESIGNING SECURE COMPUTER SYSTEMS

Shuangbao (Paul) Wang
George Mason University, USA

Robert S. Ledley
Georgetown University, USA

WILEY

高等教育出版社
HIGHER EDUCATION PRESS

This edition first published 2013

Library of Congress Cataloging-in-Publication Data

Computer architecture and security : fundamentals of designing secure computer
systems / Shuangbao (Paul) Wang, Robert S. Ledley.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-118-16881-3 (cloth)
1. Computer architecture. 2. Computer security. 3. System design. I.
Wang, Shuangbao Paul. II. Ledley, Robert Steven.
QA76.9.A73C6293 2012
005.8–dc23
2012027837

ISBN: 9781118168813

Set in 11/13 pt Times by Thomson Digital, Noida, India

To our parents who care and educate us throughout our journey.

In memory of Dr. Ledley, who pioneered Biomedical Computing.

Contents

* The star "*" here means the content is a little bit more advanced. For teaching purpose, this content may be omitted for entry level students.

About the Authors

Shuangbao (Paul) Wang is the inventor of a secure computer system. He is the recipient of Link Fellowship Award in advanced simulation and training. He holds four patents; three of them have been transferred into industry and put into production. One of his students appeared in *Time Magazine* for doing his class project which he commercialized and still pursues. In addition, one of his published papers ranked the first place in Science Direct's TOP 25 Hottest Articles. His research was awarded the Best Invention Award in Entrepreneurship Week USA at Mason. More recently, he received two university Technology Transfer Awards.

Dr. Wang has extensive experience in academia, industry, and public services. He has held many posts, including professor, director, CEO, CIO/CTO and ranking positions in public services. He is currently a professor at George Mason University. Dr. Wang served as the Chief Information and Technology Officer at National Biomedical Research Foundation/Georgetown University Medical Center. Earlier, he was the director of the Institute of Information Science and Technology at Qingdao (ISTIQ) where he oversaw more than 120 faculty and staff, acquired 12 grants, won 18 academic awards and was the PI for over 15 grants/projects.

Robert S. Ledley is the inventor of CT scanner and is a member of the National Academy of Science. He has numerous publications in *Science* and several books, and has hundreds of patents and grants. Dr. Ledley is the recipient of the National Medal of Technology that was awarded to him by President Clinton in 1997. He was admitted to the National Inventors Hall of Fame in 1990.

Dr. Ledley has been the president of the National Biomedical Research Foundation since 1960. He is also a professor (emeritus) at Georgetown University. Dr. Ledley is the editor-in-chief of four international journals. He has testified before the House and was interviewed by the Smithsonian Institution.

Preface

This book provides the fundamentals of computer architecture and security. It covers a wide range of computer hardware, system software and data concepts from a security perspective. It is essential for computer and information security professionals to understand both hardware and software security solutions to thrive in the workplace. It features a careful, in-depth, and innovative introduction to modern computer systems and patent-pending technologies in computer security.

In the past, computers were designed without security considerations. Later, firewalls were used to protect them from outside attacks. This textbook integrates security considerations into computer architecture in a way that it is immune from attacks. When necessary, the author creates simplified examples from patent-pending technologies that clearly explain architectural and implementation features.

This book is intended for graduate and undergraduate students, engineers, and researchers who are interested in secure computer architecture and systems. This book is essential for anyone who needs to understand, design or implement a secure computer system.

Studying computer architecture from a security perspective is a new area. There are many textbooks about computer architecture and many others about computer security. However, textbooks introducing computer architecture with security as the main theme are rare. This book introduces not only how to secure computer components (Memory, I/O, network interfaces and CPU) but also how to secure the entire computer system. The book proposes a new model that changes the Neumann architecture that has been the foundation of modern computers since 1945. The book includes the most recent patent-pending technology in computer architecture for security. It also incorporates experiences from the author's recent award-winning teaching and research.

This book also introduces the latest technologies, such as virtualization, cloud computing, Internet computing, ubiquitous computing, biocomputers and other advanced computer architectures, into the classroom in order to shorten the transition time from student to employee.

This book has a unique style of presentation. It uses diagrams to explain important concepts. For many key elements, the book illustrates the actual digital circuits so that interested readers can actually build such circuits for testing purposes. The book can also be used as experiment material.

The book also comes with a Wiley Companion Website (www.wiley.com/go/wang/comp_arch) that provides lecture notes, further readings and updates for students. It also provides resources for instructors as well. In addition, the website lists hundreds of security tools that can be used to test computers for security problems.

Students taking courses with this book can master security solutions in all aspects of designing modern computer systems. It introduces how to secure memory, buses, I/O and CPU. Moreover, the book explains how to secure computer architecture so that modern computers can be built on the new architecture free of data breaches.

The concept of computers as stand-alone machines is fading away. Computers are now interconnected and in many cases coordinated to accomplish one task. Most current computer architecture textbooks still focus on the single computer model without addressing any security issues. *Computer Architecture and Security* provides readers with all of the components the traditional textbooks have, but also the latest development of computer technology. As security is a concern for most people, this book addresses the security issues in depth in all aspects of computer systems.

Acknowledgements

The authors would like to thank Dr. and Mrs. McQuivey for the thorough reviews and editions. Dr. Kyle Letimar provided tremendous help in editing and revising the book proposal. The authors would also like to acknowledge Ms. Anna Chen for her incredible help in preparing this manuscript.

1

Introduction to Computer Architecture and Security

A Computer is composed of a number of different components:

Hardware: Computer hardware processes information by executing instructions, storing data, moving data among input and output devices, and transmitting and receiving information to and from remote network locations.

Software: Software consists of system software and application software or programs. Operating Systems such as Windows, UNIX/Linux and Snow Leopard are system software. Word, Firefox browser and iTunes are examples of application software.

Network: The network communication component is responsible for sending and receiving information and data through local area network or wireless connections.

Data is the fundamental representation of information and facts but usually formatted in a special way. All software is divided into two categories: data and programs. Programs are a collection of instructions for manipulating data.

Figure 1.1 shows a view of a computer system from a user perspective. Here a computer system no longer looks like an onion as traditional textbooks used to represent. Instead, a network component (including hardware and software) is added as a highway for data flowing in and out of the computer system.

Computer architecture is to study how to design computer systems. It includes all components: the central processing unit (CPU), computer memory and storage, input and output devices (I/O), and network components.

Since the invention of the Internet, computer systems are no longer standalone machines. The traditional "computing" concept of the single machine model is

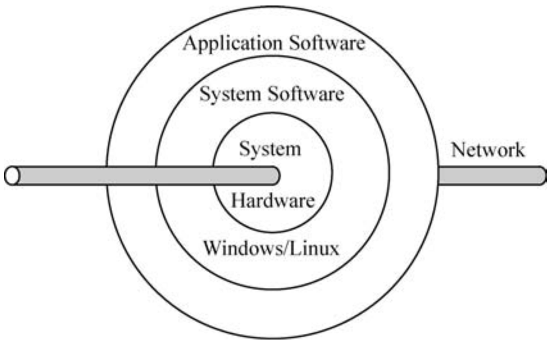


Figure 1.1 A conceptual diagram of a common computer system

fading away. For most users, information exchange has taken an important role in everyday computer uses.

As computer systems expose themselves over the Internet, the threat to computer systems has grown greater and greater. To protect a computer system (hardware, software, network, and data) from attacks, people have developed many counter-attack techniques such as firewalls, intrusion detection systems, user authentications, data encryptions and so on.

Despite the numerous efforts to prevent attacks, the threat to computer systems is far from over. Computer compromises and data bleach are still very common. If you look back to those counter-attack techniques, most of the detection systems are based on passive techniques. They only work after attacks have taken place.

A firewall by its name is a wall to prevent fire from spreading. On the other hand, it also likes a dam or levee to prevent flood. People can build a dam or levee high enough to protect against flood. However nobody can predict how high the water level will be. The 2005 New Orleans levee leak caused by Katrina is an example of this.

In medicine, people spent billions of dollars to develop new drugs to cure illness. However ancient Chinese people study how to eat well and exercise well to prevent illness. This is the same as now the so-called prevention medicine. If we apply the same mechanism to computer systems, we draw the conclusion that we not only need to build firewalls, more importantly we need to develop computer systems that are immune from attacks.

In early 2005, a US patent was filed to propose new technology that can prevent hackers from getting information stored in computer systems. The technology has drawn the attention of industry, academia, as well as government.

Figure 1.2 shows a conceptual diagram of the proposed secured computer system. Note that in addition to the traditional hardware and software, the system added an additional layer. It is like a sandbox that “separates” the computer system from the outside world. In this book, we call it a virtual semi-conductor or semi “network

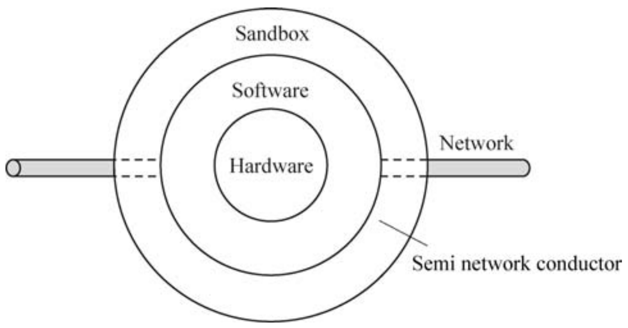


Figure 1.2 A conceptual diagram of a secured computer system

conductor." It allows the computer operator to control information and data access so that hackers are no longer able to steal data from the computer system. We will discuss this in more detail in the following chapters.

Computer Architecture and Security will teach you how to design secured computer systems. It includes information on how to secure central processing unit (CPU) memory, buses, input/output interfaces. Moreover, the book explains how to secure computer architecture as a whole so that modern computers can be built on the new architecture free of data breaches.

1.1 History of Computer Systems

Computers originally mean to compute or to calculate. The earliest computing devices date back more than two thousand years. The abacus (second century BC) which was introduced in China is one of them.

Blaise Pascal, a renowned French scientist and philosopher, invented a mechanical adding machine in 1645. Gottfried Leibniz invented the first calculator in 1694. The multiplication could be performed by repeated turns of a handle, and by shifting the position of the carriage relative to the accumulator. In December 26, 1837, Charles Babbage proposed a calculating engine that is capable of solving mathematical problems including addition, subtraction, multiplication, division, and finding the square root.

Herman Hollerith, a German-American statistician and the founder of the company that became IBM, developed a punched-card electric tabulating machine in 1889. The first program-controlled computing machine is the German machine Z3 which was developed in 1941. Mark-I, also known as IBM automatic sequence-controlled calculator, was developed by Howard Aiken at Harvard University in 1944. The Electronic Numerical Integrator and Calculator (ENIAC) was developed in May 1943. The machine was used to calculate bomb trajectories and to develop hydrogen bombs. It was not a stored-program machine, a key way to distinguish between earlier computing devices and modern computers.

The final step toward developing a modern computer was characterized as follows:

- General-purpose. The computer can be used by anybody in any domain.
- Electronic. The computer is controlled by electronic signals instead of mechanical devices.
- Stored-program. Programs are stored in its internal memory so they can run automatically without much human interaction.
- Computation. The computer can take numerical quantities to compute.

There are other features such as it has the ability for a program to read and modify itself during the course of a computation, using registers to store temporary data, indirect addressing and so on.

Professor John von Neumann, of the Institute for Advanced Study at Princeton University, one of the leading mathematicians of his time, developed a stored-program electronic computer in 1945. It is generally accepted that the first documented discussion of the advantages of using just one large internal memory, in which instructions as well as data could be held, was the draft report on EDVAC written by Neumann, dated June 30, 1945. (The full report is available on www.wiley.com/go/wang/comp_arch)

Since 1945, the Neumann computer architecture has been the foundation of modern computers, a CPU, memory and storage, input/output devices, a bus with address, data and control signals that connects the components.

Early computers were made of vacuum tubes. They are large and consume a great deal of energy. During the mid 1950s to early 1960s, solid-state transistors were used and in the mid 1960s to early 1970s, integrated circuits (IC) were used in computers. Minicomputer PDP-11 in 1970, supercomputer CDC (Cray) and mainframe IBM 360 are some examples of computers during that time. Intel 8080 and Zilog Z80 are 8-bit processors made of large-scale IC. Later, Intel's 8086 (16-bit), 80286 (16-bit) and Motorola's 68000 (16/32-bit) made of very large-scale IC (VLSI) opened the era of so-called microcomputers.

The uses of microcomputers were greatly increased by the software development. UNIX and MS-DOS later became Windows are still being used as operating systems (system software) today. Word processing, spreadsheets and databases, and many other application programs help people to carry out office works. Fortran, C, Java and many other computer languages assist software developers to program new software applications.

Now computers have grown from single-chip processors to multiple processors (cores) such as dual-cores, quad-cores and eight-cores in the near future. On the other hand, smaller devices or handheld devices such as pads and smart cell phones have the ability to handle information and data needs for many people.

With virtualization technology, a “guest” or virtual operating system may run as a process on a “host” or physical computer system. It is often considered as “computers on a computer.”

Now, network connections have become an essential part of a computer system. People have developed many ways to enhance the security of computer architecture from protecting CPU and memory to building “firewalls” to detect intrusions. The study of computer architecture with security as a whole was not started until recently. This book aims to provide readers with the latest developments in designing modern computer systems that are immune from attacks.

1.1.1 Timeline of Computer History

The timeline of computer history (Computer History, 2012) covers the most important advancements in computer research and development during 1939 to 1988.

1939: Hewlett-Packard is founded. David Packard and Bill Hewlett founded Hewlett-Packard in a Palo Alto, California garage. Their first product was the HP 200A Audio Oscillator, which rapidly became a popular piece of test equipment for engineers. Walt Disney Pictures ordered eight of the 200B models to use as sound effects generators for the 1940 movie “Fantasia.”

1940: The Complex Number Calculator (CNC) is completed. In 1939, Bell Telephone Laboratories completed this calculator, designed by researcher George Stibitz. In 1940, Stibitz demonstrated the CNC at an American Mathematical Society conference held at Dartmouth College. Stibitz stunned the group by performing calculations remotely on the CNC (located in New York City) using a Teletype connected via special telephone lines. This is considered to be the first demonstration of remote access computing.

1941: Konrad Zuse finishes the Z3 computer. The Z3 was an early computer built by German engineer Konrad Zuse working in complete isolation from developments elsewhere. Using 2,300 relays, the Z3 used floating point binary arithmetic and had a 22-bit word length. The original Z3 was destroyed in a bombing raid of Berlin in late 1943. However, Zuse later supervised a reconstruction of the Z3 in the 1960s which is currently on display at the Deutsches Museum in Munich.

1942: The Atanasoff-Berry Computer (ABC) is completed. After successfully demonstrating a proof-of-concept prototype in 1939, Atanasoff received funds to build the full-scale machine. Built at Iowa State College (now University), the ABC was designed and built by Professor John Vincent Atanasoff and graduate student Cliff Berry between 1939 and 1942. The ABC was at the center of a patent dispute relating to the invention of the computer, which was resolved in 1973 when it was shown that ENIAC co-designer John Mauchly had come to examine the ABC shortly after it became functional.

The legal result was a landmark: Atanasoff was declared the originator of several basic computer ideas, but the computer as a concept was declared un-patentable and thus was freely open to all. This result has been referred to as the "dis-invention of the computer." A full-scale reconstruction of the ABC was completed in 1997 and proved that the ABC machine functioned as Atanasoff had claimed.

1943: Project Whirlwind begins. During World War II, the US Navy approached the Massachusetts Institute of Technology (MIT) about building a flight simulator to train bomber crews. The team first built a large analog computer, but found it inaccurate and inflexible. After designers saw a demonstration of the ENIAC computer, they decided on building a digital computer. By the time the Whirlwind was completed in 1951, the Navy had lost interest in the project, though the US Air Force would eventually support the project which would influence the design of the SAGE program.

The Relay Interpolator is completed. The US Army asked Bell Labs to design a machine to assist in testing its M-9 Gun Director. Bell Labs mathematician George Stibitz recommended using a relay-based calculator for the project. The result was the Relay Interpolator, later called the Bell Labs Model II. The Relay Interpolator used 440 relays and since it was programmable by paper tape, it was used for other applications following the war.

1944: Harvard Mark-1 is completed. Conceived by Harvard professor Howard Aiken, and designed and built by IBM, the Harvard Mark-1 was a room-sized, relay-based calculator. The machine had a 50 ft long camshaft that synchronized the machine's thousands of component parts. The Mark-1 was used to produce mathematical tables but was soon superseded by stored program computers.

The first Colossus is operational at Bletchley Park. Designed by British engineer Tommy Flowers, the Colossus was designed to break the complex Lorenz ciphers used by the Nazis during WWII. A total of ten Colossi were delivered to Bletchley, each using 1,500 vacuum tubes and a series of pulleys transported continuous rolls of punched paper tape containing possible solutions to a particular code. Colossus reduced the time to break Lorenz messages from weeks to hours. The machine's existence was not made public until the 1970s.

1945: **John von Neumann** wrote "First Draft of a Report on the EDVAC" in which he outlined the architecture of a stored-program computer. Electronic storage of programming information and data eliminated the need for the more clumsy methods of programming, such as punched paper tape – a concept that has characterized mainstream computer development since 1945. Hungarian-born von Neumann demonstrated prodigious expertise in hydrodynamics, ballistics, meteorology, game theory, statistics, and the use of mechanical devices for computation. After the war, he concentrated on the development of Princeton's Institute for Advanced Studies computer and its copies around the world.

1946: In February, the public got its first glimpse of the ENIAC, a machine built by John Mauchly and J. Presper Eckert that improved by 1,000 times on the speed of its contemporaries.

- *Start of project*: 1943
- *Completed*: 1946
- *Programmed*: plug board and switches
- *Speed*: 5,000 operations per second
- *Input/output*: cards, lights, switches, plugs
- *Floor space*: 1,000 square feet
- *Project leaders*: John Mauchly and J. Presper Eckert.

An inspiring summer school on computing at the University of Pennsylvania's Moore School of Electrical Engineering stimulated construction of stored-program computers at universities and research institutions. This free, public set of lectures inspired the EDSAC, BINAC, and, later, IAS machine clones like the AVIDAC. Here, Warren Kelleher completes the wiring of the arithmetic unit components of the AVIDAC at Argonne National Laboratory. Robert Dennis installs the inter-unit wiring as James Woody Jr. adjusts the deflection control circuits of the memory unit.

1948: IBM's Selective Sequence Electronic Calculator computed scientific data in public display near the company's Manhattan headquarters. Before its decommissioning in 1952, the SSEC produced the moon-position tables used for plotting the course of the 1969 Apollo flight to the moon.

- *Speed:* 50 multiplications per second
- *Input/output:* cards, punched tape
- *Memory type:* punched tape, vacuum tubes, relays
- *Technology:* 20,000 relays, 12,500 vacuum tubes
- *Floor space:* 25 feet by 40 feet
- *Project leader:* Wallace Eckert.

1949: Maurice Wilkes assembled the EDSAC, the first practical stored-program computer, at Cambridge University. His ideas grew out of the Moore School lectures he had attended three years earlier.

For programming the EDSAC, Wilkes established a library of short programs called subroutines stored on punched paper tapes.

- *Technology:* vacuum tubes
- *Memory:* 1 K words, 17 bits, mercury delay line
- *Speed:* 714 operations per second.

The Manchester Mark I computer functioned as a complete system using the Williams tube for memory. This university machine became the prototype for Ferranti Corp.'s first computer.

- *Start of project:* 1947
- *Completed:* 1949
- *Add time:* 1.8 microseconds
- *Input/output:* paper tape, teleprinter, switches
- *Memory size:* 128 + 1024 40-digit words
- *Memory type:* cathode ray tube, magnetic drum
- *Technology:* 1,300 vacuum tubes
- *Floor space:* medium room
- *Project leaders:* Frederick Williams and Tom Kilburn.

1950: Engineering Research Associates of Minneapolis built the ERA 1101, the first commercially produced computer; the company's first customer was the US Navy. It held 1 million bits on its magnetic drum, the earliest magnetic storage devices. Drums registered information as magnetic pulses in tracks around a metal cylinder. Read/write heads both recorded and recovered the data. Drums eventually stored as many as 4,000 words and retrieved any one of them in as little as five-thousandths of a second.

The National Bureau of Standards constructed the Standards Eastern Automatic Computer (SEAC) in Washington as a laboratory for testing components and systems for setting computer standards. The SEAC was the first computer to use all-diode logic, a technology more reliable than vacuum tubes, and the first stored-program computer completed in the United States. Magnetic tape in the external storage units (shown on the right of this photo) stored programming information, coded subroutines, numerical data, and output.

The National Bureau of Standards completed its SWAC (Standards Western Automatic Computer) at the Institute for Numerical Analysis in Los Angeles. Rather than testing components like its companion, the SEAC, the SWAC had an objective of computing using already-developed technology.

1951: MIT's Whirlwind debuted on Edward R. Murrow's "See It Now" television series. Project director Jay Forrester described the computer as a "reliable operating system," running 35 hours a week at 90% utility using an electrostatic tube memory.

- *Start of project:* 1945
- *Completed:* 1951
- *Add time:* 0.05 microseconds
- *Input/output:* cathode ray tube, paper tape, magnetic tape
- *Memory size:* 2048 16-digit words
- *Memory type:* cathode ray tube, magnetic drum, tape (1953 – core memory)
- *Technology:* 4,500 vacuum tubes, 14,800 diodes
- *Floor space:* 3,100 square feet
- *Project leaders:* Jay Forrester and Robert Everett.

1952: John von Neumann's IAS computer became operational at the Institute for Advanced Studies in Princeton, N.J. Contract obliged the builders to share their designs with other research institutes. This resulted in a number of clones: the MANIAC at Los Alamos Scientific Laboratory, the ILLIAC at the University of Illinois, the Johnniac at Rand Corp., the SILLIAC in Australia, and others.

1953: IBM shipped its first electronic computer, the 701. During three years of production, IBM sold 19 machines to research laboratories, aircraft companies, and the federal government.

1954: The IBM 650 magnetic drum calculator established itself as the first mass-produced computer, with the company selling 450 in one year. Spinning at 12,500 rpm, the 650s magnetic data-storage drum allowed much faster access to stored material than drum memory machines.

1956: MIT researchers built the TX-0, the first general-purpose, programmable computer built with transistors. For easy replacement, designers placed each transistor circuit inside a "bottle," similar to a vacuum tube. Constructed at MIT's Lincoln Laboratory, the TX-0 moved to the MIT Research Laboratory of Electronics, where it hosted some early imaginative tests of programming, including a Western movie shown on TV, 3-D tic-tac-toe, and a maze in which mice found martinis and became increasingly inebriated.

1958: SAGE – Semi-Automatic Ground Environment – linked hundreds of radar stations in the United States and Canada in the first large-scale computer communications network. An operator directed actions by touching a light gun to the screen.

The air defense system operated on the AN/FSQ-7 computer (known as Whirlwind II during its development at MIT) as its central computer. Each computer used a full megawatt of power to drive its 55,000 vacuum tubes, 175,000 diodes and 13,000 transistors.

1959: IBM's 7000 series mainframes were the company's first transistorized computers. At the top of the line of computers – all of which emerged significantly faster and more dependable than vacuum tube machines – sat the 7030, also known as the "Stretch." Nine of the computers, which featured a 64-bit word and other innovations, were sold to national laboratories and other scientific users. L. R. Johnson first used the term "architecture" in describing the Stretch.

1960: The precursor to the minicomputer, DEC's PDP-1 sold for $120,000. One of 50 built, the average PDP-1 included with a cathode ray tube graphic display, needed no air conditioning and required only one operator. It's large scope intrigued early hackers at MIT, who wrote the first computerized video game, SpaceWar!, for it. The SpaceWar! creators then used the game as a standard demonstration on all 50 computers.

1961: According to Datamation magazine, IBM had an 81.2% share of the computer market in 1961, the year in which it introduced the 1400 Series. The 1401 mainframe, the first in the series, replaced the vacuum tube with smaller, more reliable transistors and used a magnetic core memory.

Demand called for more than 12,000 of the 1401 computers, and the machine's success made a strong case for using general-purpose computers rather than specialized systems.

1962: The LINC (Laboratory Instrumentation Computer) offered the first real time laboratory data processing. Designed by Wesley Clark at Lincoln Laboratories, Digital Equipment Corp. later commercialized it as the LINC-8.

Research faculty came to a workshop at MIT to build their own machines, most of which they used in biomedical studies. DEC supplied components.

1964: IBM announced the System/360, a family of six mutually compatible computers and 40 peripherals that could work together. The initial investment of $5 billion was quickly returned as orders for the system climbed to 1,000 per month within two years. At the time IBM released the System/360, the company was making a transition from discrete transistors to integrated circuits, and its major source of revenue moved from punched-card equipment to electronic computer systems.

CDC's 6600 supercomputer, designed by Seymour Cray, performed up to 3 million instructions per second – a processing speed three times faster than that of its closest competitor, the IBM Stretch. The 6600 retained the distinction of being the fastest computer in the world until surpassed by its successor, the CDC 7600, in 1968. Part of the speed came from the computer's design, which had 10 small computers, known as peripheral processors, funneling data to a large central processing unit.

1965: Digital Equipment Corp. introduced the PDP-8, the first commercially successful minicomputer. The PDP-8 sold for $18,000, one-fifth the price of a small IBM 360 mainframe. The speed, small size, and reasonable cost enabled the PDP-8 to go into thousands of manufacturing plants, small businesses, and scientific laboratories.

1966: The Department of Defense Advanced Research Projects Agency contracted with the University of Illinois to build a large parallel processing computer, the ILLIAC IV, which did not operate until 1972 at NASA's Ames Research Center. The first large-scale array computer, the ILLIAC IV achieved a computation speed of 200 million instructions per second, about 300 million operations per second, and 1 billion bits per second of I/O transfer via a unique combination of parallel architecture and the overlapping or "pipe-lining" structure of its 64 processing elements.

This photograph shows one of the ILLIAC's 13 Burroughs disks, the debugging computer, the central unit, and the processing unit cabinet with a processing element.

Hewlett-Packard entered the general purpose computer business with its HP-2115 for computation, offering a computational power formerly found only in much larger computers. It supported a wide variety of languages, among them Basic, ALGOL, and Fortran.

1968: Data General Corp., started by a group of engineers that had left Digital Equipment Corp., introduced the Nova, with 32 kilobytes of memory, for $8,000.

The simple architecture of the Nova instruction set inspired Steve Wozniak's Apple I board eight years later.

The Apollo Guidance Computer made its debut orbiting the Earth on Apollo 7. A year later, it steered Apollo 11 to the lunar surface. Astronauts communicated with the computer by punching two-digit codes and the appropriate syntactic category into the display and keyboard unit.

1971: The Kenbak-1, the first personal computer, advertised for $750 in *Scientific American*. Designed by John V. Blankenbaker using standard medium-scale and small-scale integrated circuits, the Kenbak-1 relied on switches for input and lights for output from its 256-byte memory. In 1973, after selling only 40 machines, Kenbak Corp. closed its doors.

1972: Hewlett-Packard announced the HP-35 as "a fast, extremely accurate electronic slide rule" with a solid-state memory similar to that of a computer. The HP-35 distinguished itself from its competitors by its ability to perform a broad variety of logarithmic and trigonometric functions, to store more intermediate solutions for later use, and to accept and display entries in a form similar to standard scientific notation.

1973: The TV Typewriter, designed by Don Lancaster, provided the first display of alphanumeric information on an ordinary television set. It used $120 worth of electronics components, as outlined in the September 1973 issue of *Radio Electronics*. The original design included two memory boards and could generate and store 512 characters as 16 lines of 32 characters. A 90-minute cassette tape provided supplementary storage for about 100 pages of text.

The Micral was the earliest commercial, non-kit personal computer based on a micro-processor, the Intel 8008. Thi Truong developed the computer and Philippe Kahn the software. Truong, founder and president of the French company R2E, created the Micral as a replacement for minicomputers in situations that didn't require high performance. Selling for $1,750, the Micral never penetrated the US market. In 1979, Truong sold Micral to Bull.

1974: Researchers at the Xerox Palo Alto Research Center designed the Alto – the first work station with a built-in mouse for input. The Alto stored several files simultaneously in windows, offered menus and icons, and could link to a local area network. Although Xerox never sold the Alto commercially, it gave a number of them to universities. Engineers later incorporated its features into work stations and personal computers.

1975: The January edition of *Popular Electronics* featured the Altair 8800 computer kit, based on Intel's 8080 microprocessor, on its cover. Within weeks of the computer's debut, customers inundated the manufacturing company, MITS, with orders. Bill Gates and Paul Allen licensed Basic as the software language for the Altair. Ed Roberts invented the 8800 – which sold for $297, or $395 with a case – and coined the term "personal computer." The machine came with 256 bytes of memory (expandable to 64 K) and an open 100-line bus structure that evolved into the S-100 standard. In 1977, MITS sold out to Pertec, which continued producing Altairs through 1978.

1976: Steve Wozniak designed the Apple I, a single-board computer. With specifications in hand and an order for 100 machines at $500 each from the Byte Shop, he and Steve Jobs got their start in business. In this photograph of the Apple I board, the upper two rows are a video terminal and the lower two rows are the computer. The 6502 microprocessor in the white package sits on the lower right. About 200 of the machines sold before the company announced the Apple II as a complete computer.

The Cray I made its name as the first commercially successful vector processor. The fastest machine of its day, its speed came partly from its shape, a C, which reduced the length of wires and thus the time signals needed to travel across them.

- *Project started:* 1972
- *Project completed:* 1976
- *Speed:* 166 million floating-point operations per second
- *Size:* 58 cubic feet
- *Weight:* 5,300 lbs.
- *Technology:* Integrated circuit
- *Clock rate:* 83 million cycles per second
- *Word length:* 64-bit words
- *Instruction set:* 128 instructions.

1977: The Commodore Personal Electronic Transactor (PET) – the first of several personal computers released in 1977 – came fully assembled and was straightforward to operate, with either 4 or 8 kilobytes of memory, two built-in cassette drives, and a membrane "chiclet" keyboard.

The Apple II became an instant success when released in 1977 with its printed circuit motherboard, switching power supply, keyboard, case assembly, manual, game paddles, A/C powercord, and cassette tape with the computer game "Breakout." When hooked up to a color television set, the Apple II produced brilliant color graphics.

In the first month after its release, Tandy Radio Shack's first desktop computer – the TRS-80 – sold 10,000 units, well more than the company's projected sales of 3,000 units for one year. Priced at $599.95, the machine included a Z80 based microprocessor, a video display, 4 kilobytes of memory, Basic, cassette storage, and easy-to-understand manuals that assumed no prior knowledge on the part of the consumer.

1978: The VAX 11/780 from Digital Equipment Corp. featured the ability to address up to 4.3 gigabytes of virtual memory, providing hundreds of times the capacity of most minicomputers.

1979: Atari introduces the Model 400 and 800 Computer. Shortly after delivery of the Atari VCS game console, Atari designed two microcomputers with game capabilities: the Model 400 and Model 800. The two machines were built with the idea that the 400 would serve primarily as a game console while the 800 would be more of a home computer. Both sold well, though they had technical and marketing problems, and faced strong competition from the Apple II, Commodore PET, and TRS-80 computers.

1981: IBM introduced its PC, igniting a fast growth of the personal computer market. The first PC ran on a 4.77 MHz Intel 8088 microprocessor and used Microsoft's MS-DOS operating system.

Adam Osborne completed the first portable computer, the Osborne I, which weighed 24 pounds and cost $1,795. The price made the machine especially attractive, as it included software worth about $1,500. The machine featured a 5-inch display, 64 kilobytes of memory, a modem, and two 5 1/4-inch floppy disk drives.

Apollo Computer unveiled the first work station, its DN100, offering more power than some minicomputers at a fraction of the price. Apollo Computer and Sun Microsystems, another early entrant in the work station market, optimized their machines to run the computer-intensive graphics programs common in engineering.

1982: The Cray XMP, first produced in this year, almost doubled the operating speed of competing machines with a parallel processing system that ran at 420 million floating-point operations per second, or megaflops. Arranging two Crays to work together on different parts of the same problem achieved the faster speed. Defense and scientific research institutes also heavily used Crays.

Commodore introduces the Commodore 64. The C64, as it was better known, sold for $595, came with 64KB of RAM and featured impressive graphics. Thousands of software titles were released over the lifespan of the C64. By the time the C64 was discontinued in 1993, it had sold more than 22 million units and is recognized by the 2006 Guinness Book of World Records as the greatest selling single computer model of all time.

1983: Apple introduced its Lisa. The first personal computer with a graphical user interface, its development was central in the move to such systems for personal computers. The Lisa's sloth and high price ($10,000) led to its ultimate failure.

The Lisa ran on a Motorola 68000 microprocessor and came equipped with 1 megabyte of RAM, a 12-inch black-and-white monitor, dual 5 1/4-inch floppy disk drives and a 5 megabyte Profile hard drive. The Xerox Star – which included a system called Smalltalk that involved a mouse, windows, and pop-up menus – inspired the Lisa's designers.

Compaq Computer Corp. introduced the first PC clone that used the same software as the IBM PC. With the success of the clone, Compaq recorded first-year sales of $111 million, the most ever by an American business in a single year.

With the introduction of its PC clone, Compaq launched a market for IBM-compatible computers that by 1996 had achieved an 83% share of the personal computer market. Designers reverse-engineered the Compaq clone, giving it nearly 100% compatibility with the IBM.

1984: Apple Computer launched the Macintosh, the first successful mouse-driven computer with a graphic user interface, with a single $1.5 million commercial during the 1984 Super Bowl. Based on the Motorola 68000 microprocessor, the Macintosh included many of the Lisa's features at a much more affordable price: $2,500.

Apple's commercial played on the theme of George Orwell's "1984" and featured the destruction of Big Brother with the power of personal computing found in a Macintosh. Applications that came as part of the package included MacPaint, which made use of the mouse, and MacWrite, which demonstrated WYSIWYG (What You See Is What You Get) word processing.

IBM released its PC Jr. and PC-AT. The PC Jr. failed, but the PC-AT, several times faster than original PC and based on the Intel 80286 chip, claimed success with its notable increases in performance and storage capacity, all for about $4,000. It also included more RAM and accommodated high-density 1.2-megabyte 5 1/4-inch floppy disks.

1985: The Amiga 1000 is released. Commodore's Amiga 1000 sold for $1,295 dollars (without monitor) and had audio and video capabilities beyond those found in most other personal computers. It developed a very loyal following and add-on components allowed it to be upgraded easily. The inside of the case is engraved with the signatures of the Amiga designers, including Jay Miner as well as the paw print of his dog Mitchy.

1986: Daniel Hillis of Thinking Machines Corp. moved artificial intelligence a step forward when he developed the controversial concept of massive parallelism in the Connection Machine. The machine used up to 65,536 processors and could complete several billion operations per second. Each processor had its own small memory linked with others through a flexible network that users could alter by reprogramming rather than rewiring.

The machine's system of connections and switches let processors broadcast information and requests for help to other processors in a simulation of brainlike associative recall. Using this system, the machine could work faster than any other at the time on a problem that could be parceled out among the many processors.

IBM and MIPS released the first RISC-based workstations, the PC/RT and R2000-based systems. Reduced instruction set computers grew out of the observation that the simplest 20% of a computer's instruction set does 80% of the work, including most base operations such as add, load from memory, and store in memory.

The IBM PC-RT had 1 megabyte of RAM, a 1.2-megabyte floppy disk drive, and a 40-megabyte hard drive. It performed 2 million instructions per second, but other RISC-based computers worked significantly faster.

1987: IBM introduced its PS/2 machines, which made the 3 1/2-inch floppy disk drive and video graphics array standard for IBM computers. The first IBMs to include Intel's 80386 chip, the company had shipped more than 1 million units by the end of the year. IBM released a new operating system, OS/2, at the same time, allowing the use of a mouse with IBMs for the first time.

1988: Apple cofounder Steve Jobs, who left Apple to form his own company, unveiled the NeXT. The computer he created failed but was recognized as an important innovation. At a base price of $6,500, the NeXT ran too slowly to be popular.

The significance of the NeXT rested in its place as the first personal computer to incorporate a drive for an optical storage disk, a built-in digital signal processor that allowed voice recognition, and object-oriented languages to simplify programming. The NeXT offered Motorola 68030 microprocessors, 8 megabytes of RAM, and a 256-megabyte read/write optical disk storage.

The milestones during this period are: the stored program computer architecture proposed by John von Neumann in 1945; the first transistorized computer IBM 7000 series in 1958; IBM 360 in 1964; the first vector processor Cray I in 1976; Apple II in 1977; IBM-PC in 1981; Apple Macintosh in 1984; the first RISC-based workstation IBM PC/RT in 1986.

Innovation and commercialization are the main characteristics during this 50 year period.

1.1.2 Timeline of Internet History

The timeline of Internet history covers most important advancements in Internet research and development from year 1962 to 1992.

1962: At MIT, a wide variety of computer experiments are going on. Ivan Sutherland uses the TX-2 to write Sketchpad, the origin of graphical programs for computer-aided design.

J.C.R. Licklider writes memos about his Intergalactic Network concept, where everyone on the globe is interconnected and can access programs and data at any site from anywhere. He is talking to his own “Intergalactic Network” of researchers across the country. In October, “Lick” becomes the first head of the computer research program at ARPA, which he calls the Information Processing Techniques Office (IPTO).

Leonard Kleinrock completes his doctoral dissertation at MIT on queuing theory in communication networks, and becomes an assistant professor at UCLA.

The SAGE (Semi Automatic Ground Environment), based on earlier work at MIT and IBM, is fully deployed as the North American early warning system. Operators of “weapons directing consoles” use a light gun to identify moving objects that show up on their radar screens. SAGE sites are used to direct air defense. This project provides experience in the development of the SABRE air travel reservation system and later air traffic control systems.

1963: Licklider starts to talk with Larry Roberts of Lincoln Labs, director of the TX-2 project, Ivan Sutherland, a computer graphics expert whom he has hired to work at ARPA and Bob Taylor, who joins ARPA in 1965. Lick contracts with MIT, UCLA, and BBN to start work on his vision.

Syncom, the first synchronous communication satellite, is launched. NASA's satellite is assembled in the Hughes Aircraft Company's facility in Culver City, California. Total payload is 55 pounds.

A joint industry-government committee develops American Standard Code for Information Interchange (ASCII), the first universal standard for computers. It permits machines from different manufacturers to exchange data. 128 unique 7-bit strings stand for either a letter of the English alphabet, one of the Arabic numerals, one of an assortment of punctuation marks and symbols, or a special function, such as the carriage return.

1964: Simultaneous work on secure packet switching networks is taking place at MIT, the RAND Corporation, and the National Physical Laboratory in Great Britain. Paul Baran, Donald Davies, Leonard Kleinrock, and others proceed in parallel research. Baran is one of the first to publish, *On Data Communications Networks*. Kleinrock's thesis is also published as a seminal text on queuing theory.

IBM's new System 360 computers come onto the market and set the de facto worldwide standard of the 8-bit byte, making the 12-bit and 36-bit word machines almost instantly obsolete. The $5 billion investment by IBM into this family of six mutually compatible computers pays off, and within two years orders for the System 360 reach 1,000 per month.

On-line transaction processing debuts with IBM's SABRE air travel reservation system for American Airlines. SABRE (Semi-Automatic Business Research Environment) links 2,000 terminals in sixty cities via telephone lines.

Licklider leaves ARPA to return to MIT, and Ivan Sutherland moves to IPTO. With IPTO funding, MIT's Project MAC acquires a GE-635 computer and begins the development of the Multics timesharing operating system.

1965: DEC unveils the PDP-8, the first commercially successful minicomputer. Small enough to sit on a desktop, it sells for $18,000 – one-fifth the cost of a low-end IBM/360 mainframe. The combination of speed, size, and cost enables the establishment of the minicomputer in thousands of manufacturing plants, offices, and scientific laboratories.

With ARPA funding, Larry Roberts and Thomas Marill create the first wide-area network connection. They connect the TX-2 at MIT to the Q-32 in Santa Monica via a dedicated telephone line with acoustic couplers. The system confirms the suspicions of the Intergalactic Network researchers that telephone lines work for data, but are inefficient, wasteful of bandwidth, and expensive. As Kleinrock predicts, packet switching offers the most promising model for communication between computers.

Late in the year, Ivan Sutherland hires Bob Taylor from NASA. Taylor pulls together the ideas about networking that are gaining momentum among IPTO's computer-scientist contractors.

The ARPA-funded JOSS (Johnniac Open Shop System) at the RAND Corporation goes on line. The JOSS system permits online computational problem solving at a number of remote electric typewriter consoles. The standard IBM Model 868 electric typewriters are modified with a small box with indicator lights and activating switches. The user input appears in green, and JOSS responds with the output in black.

1966: Taylor succeeds Sutherland to become the third director of IPTO. In his own office, he has three different terminals, which he can connect by telephone to three different computer systems research sites around the nation. Why can't they all talk together? His problem is a metaphor for that facing the ARPA computer research community.

Taylor meets with Charles Herzfeld, the head of ARPA, to outline his issues. Twenty minutes later he has a million dollars to spend on networking. The idea is to link all the IPTO contractors. After several months of discussion, Taylor persuades Larry Roberts to leave MIT to start the ARPA network program.

Simultaneously, the English inventor of packet switching, Donald Davies, is theorizing at the British National Physical Laboratory (NPL) about building a network of computers to test his packet switching concepts.

Honeywell introduces the DDP-516 minicomputer and demonstrates its ruggedness with a sledgehammer. This catches Roberts' eye.

1967: Larry Roberts convenes a conference in Ann Arbor, Michigan, to bring the ARPA researchers together. At the conclusion, Wesley Clark suggests that the network be managed by interconnected "Interface Message Processors" in front of the major computers. Called IMPs, they evolve into today's routers.

Roberts puts together his plan for the ARPANET. The separate strands of investigation begin to converge. Donald Davies, Paul Baran, and Larry Roberts become aware of each other's work at an ACM conference where they all meet. From Davies, the word "packet" is adopted and the proposed line speed in ARPANET is increased from 2.4 Kbps to 50 Kbps.

The acoustically coupled modem, invented in the early 1960s, is vastly improved by John van Geen of the Stanford Research Institute (SRI). He introduces a receiver that can reliably detect bits of data amid the hiss heard over long-distance telephone connections.

1968: Roberts and the ARPA team refine the overall structure and specifications for the ARPANET. They issue an RFQ for the development of the IMPs.

At Bolt, Beranek and Newman (BBN), Frank Heart leads a team to bid on the project. Bob Kahn plays a major role in shaping the overall BBN designs. BBN wins the project in December.

Roberts works with Howard Frank and his team at Network Analysis Corporation designing the network topology and economics. Kleinrock's team prepares the network measurement system at UCLA, which is to become the site of the first node.

The ILLIAC IV, the largest supercomputer of its time, is being built at Burroughs under a NASA contract. More than 1,000 transistors are squeezed onto its RAM chip, manufactured by the Fairchild Semiconductor Corporation, yielding 10 times the speed at one-hundredth the size of equivalent core memory.

ILLIAC-IV will be hooked to the ARPANET so that remote scientists can have access to its unique capabilities.

1969: Frank Heart puts a team together to write the software that will run the IMPs and to specify changes in the Honeywell DDP-516 they have chosen. The team includes Ben Barker, Bernie Cosell, Will Crowther, Bob Kahn, Severo Ornstein, and Dave Walden.

Four sites are selected. At each, a team gets to work on producing the software to enable its computers and the IMP to communicate. At UCLA, the first site, Vint Cerf, Steve Crocker, and Jon Postel work with Kleinrock to get ready. On April 7, Crocker sends around a memo entitled "Request for Comments." This is the first of thousands of RFCs that document the design of the ARPANET and the Internet.

The team calls itself the Network Working Group (RFC 10), and comes to see its job as the development of a "protocol," the collection of programs that comes to be known as NCP (Network Control Protocol).

The second site is the Stanford Research Institute (SRI), where Doug Engelbart saw the ARPA experiment as an opportunity to explore wide-area distributed collaboration, using his NLS system, a prototype "digital library." SRI supported the Network Information Center, led by Elizabeth (Jake) Feinler and Don Nielson.

At the University of California, Santa Barbara (UCSB) Glen Culler and Burton Fried investigate methods for display of mathematical functions using storage displays to deal with the problem of screen refresh over the net. Their investigation of computer graphics supplies essential capabilities for the representation of scientific information.

After installation in September, handwritten logs from UCLA show the first host-to-host connection, from UCLA to SRI, is made on October 29, 1969. The first "Log-In" crashes the SRI host, but the next attempt works!

1970: Nodes are added to the ARPANET at the rate of one per month.

Programmers Dennis Ritchie and Kenneth Thompson at Bell Labs complete the UNIX operating system on a spare DEC minicomputer. UNIX combines many of the time-sharing and file-management features offered by Multics and wins a wide following, particularly among scientists.

Bob Metcalfe builds a high-speed (100 Kbps) network interface between the MIT IMP and a PDP-6 to the ARPANET. It runs for 13 years without human intervention. Metcalfe goes on to build another ARPANET interface for Xerox PARC's PDP-10 clone (MAXC).

DEC announces the Unibus for its PDP-11 minicomputers to allow the addition and integration of myriad computer-cards for instrumentation and communications.

In December, the Network Working Group (NWG) led by Steve Crocker finishes the initial ARPANET Host-to-Host protocol, called the Network Control Protocol (NCP).

1971: The ARPANET begins the year with 14 nodes in operation. BBN modifies and streamlines the IMP design so it can be moved to a less cumbersome platform than the DDP-516. BBN also develops a new platform, called a Terminal Interface Processor (TIP) which is capable of supporting input from multiple hosts or terminals.

The Network Working Group completes the Telnet protocol and makes progress on the file transfer protocol (FTP) standard. At the end of the year, the ARPANET contains 19 nodes as planned.

Intel's release of the 4004, the first "computer on a chip," ushers in the epoch of the microprocessor. The combination of memory and processor on a single chip reduces size and cost, and increases speed, continuing the evolution from vacuum tube to transistor to integrated circuit.

Many small projects are carried out across the new network, including the demonstration of an aircraft-carrier landing simulator. However, the overall traffic is far lighter than the network's capacity. Something needs to stimulate the kind of collaborative and interactive atmosphere consistent with the original vision. Larry Roberts and Bob Kahn decide that it is time for a public demonstration of the ARPANET. They choose to hold this demonstration at the International Conference on Computer Communication (ICCC) to be held in Washington, DC, in October 1972.

1972: The ARPANET grows by ten more nodes in the first 10 months of 1972. The year is spent finishing, testing, and releasing all the network protocols, and developing network demonstrations for the ICCC.

At BBN, Ray Tomlinson writes a program to enable electronic mail to be sent over the ARPANET. It is Tomlinson who develops the "user@host" convention, choosing the @ sign arbitrarily from the non-alphabetic symbols on the keyboard. Unbeknownst to him, @ is already in use as an escape character, prompt, or command indicator on many other systems. Other networks will choose other conventions, inaugurating a long period known as the e-mail "header wars." Not until the late 1980s will "@" finally become a worldwide standard.

Following the lead of Intel's 4004 chip, hand-held calculators ranging from the simple Texas Instruments four-function adding machines to the elaborate Hewlett-Packard scientific calculators immediately consign ordinary slide rules to oblivion.

Xerox PARC develops a program called Smalltalk, and Bell Labs develops a language called "C."

Steve Wozniak begins his career by building one of the best-known "blue boxes;" tone generators that enable long-distance dialing while bypassing the phone company's billing equipment.

The ICCC demonstrations are a tremendous success. One of the best known demos features a conversation between ELIZA, Joseph Weizenbaum's artificially-intelligent psychiatrist located at MIT, and PARRY, a paranoid computer developed by Kenneth Colby at Stanford. Other demos feature interactive chess games, geography quizzes, and an elaborate air traffic control simulation. An AT&T delegation visits ICCC but leaves in puzzlement.

1973: Thirty institutions are connected to the ARPANET. The network users range from industrial installations and consulting firms like BBN, Xerox PARC and the MITRE Corporation, to government sites like NASA's Ames Research Laboratories, the National Bureau of Standards, and Air Force research facilities.

The ICCC demonstrations prove packet-switching a viable technology, and ARPA (now DARPA, where the "D" stands for "Defense") looks for ways to extend its reach. Two new programs begin: Packet Radio sites are modeled on the ALOHA experiment at the University of Hawaii designed by Norm Abramson, connecting seven computers on four islands; and a satellite connection enables linking to two foreign sites in Norway and the UK.

Bob Kahn moves from BBN to DARPA to work for Larry Roberts, and his first self-assigned task is the interconnection of the ARPANET with other networks. He enlists Vint Cerf, who has been teaching at Stanford. The problem is that ARPANET, radio-based PRnet, and SATNET all have different interfaces, packet sizes, labeling, conventions and transmission rates. Linking them together is very difficult.

Kahn and Cerf set about designing a net-to-net connection protocol. Cerf leads the newly formed International Network Working Group. In September 1973, the two give their first paper on the new Transmission Control Protocol (TCP) at an INWG meeting at the University of Sussex in England.

Meanwhile, at Xerox PARC, Bob Metcalfe is working on a wire-based system modeled on ALOHA protocols for Local Area Networks (LANs). It will become Ethernet.

1974: Ethernet is demonstrated by networking Xerox PARC's new Alto computers.

BBN recruits Larry Roberts to direct a new venture, called Telenet, which is the first public packet-switched service. Roberts' departure creates a crisis in the DARPA IPTO office.

DARPA has fulfilled its initial mission. Discussions about divesting DARPA of operational responsibility for the network are held. Because it is DARPA-funded, BBN has no exclusive right to the source code for the IMPs. Telenet and other new networking enterprises want BBN to release the source code. BBN argues that it is always changing the code and that it has recently undergone a complete rewrite at the hands of John McQuillan. Their approach makes Roberts' task of finding a new director for IPTO difficult. J.C.R. Licklider agrees to return to IPTO from MIT on a temporary basis.

In addition to DARPA, The National Science Foundation (NSF) is actively supporting computing and networking at almost 120 universities. The largest NSF installation is at the National Center for Atmospheric Research (NCAR) in Boulder, Colorado. There, scientists use a home-built "remote job entry" system to connect to NCAR's CDC 7600 from major universities.

Bob Kahn and Vint Cerf publish “A Protocol for Packet Network Interconnection” in the May 1974 issue of IEEE Transactions on Communications Technology. Shortly thereafter, DARPA funds three contracts to develop and implement the Kahn-Cerf Transmission Control Protocol (TCP) protocol described in their paper, one at Stanford (Cerf and his students), one at BBN (Ray Tomlinson), and one at University College London (directed by Peter Kirstein and his students).

Daily traffic on the ARPANET exceeds 3 million packets.

1975: The ARPANET geographical map now shows 61 nodes. Licklider arranges its administration to be turned over to the Defense Communications Agency (DCA). BBN remains the contractor responsible for network operations. BBN agrees to release the source code for IMPs and TIPs.

The Network Working Group maintains its open system of discussion via RFCs and e-mail lists. Discomfort grows with the bureaucratic style of DCA.

The Department of Energy creates its own net to support its own research. This net operates over dedicated lines connecting each site to the computer centers at the National Laboratories.

NASA begins planning its own space physics network, SPAN. These networks have connections to the ARPANET so the newly developed TCP protocol begins to get a workout. Internally, however, the new networks use such a variety of protocols that true interoperability is still an issue.

1976: DARPA supports computer scientists at UC Berkeley who are revising a Unix system to incorporate TCP/IP protocols. Berkeley Unix also incorporates a second set of Bell Labs protocols, called UUCP, for systems to use dial-up connections.

Seymour Cray demonstrates the first vector-processor supercomputer, the CRAY-1. The first customers include Lawrence Livermore National Laboratory, Los Alamos National Laboratory, and NCAR. The CRAY-1 hardware is more compact and faster than previous supercomputers. No wire is more than 4 feet long, and the clock period is 12.5 nanoseconds (billionths of a second). The machine is cooled by freon circulated through stainless steel tubing bonded within vertical wedges of aluminum between the stacks of circuit boards (Cray patents the bonding process). The CRAY-1’s speed and power attract researchers who want access to it over networks.

Vint Cerf moves from Stanford to DARPA to work with Bob Kahn on networking and the TCP/IP protocols.

1977: Steve Wozniak and Steve Jobs announce the Apple II computer. Also introduced are the Tandy TRS-80 and the Commodore Pet. These three off-the-shelf machines create the consumer and small business markets for computers.

Cerf and Kahn mount a major demonstration, “internetting” among the Packet Radio net, SATNET, and the ARPANET. Messages go from a van in the Bay Area across the US on ARPANET, then to University College London and back via satellite to Virginia, and back through the ARPANET to the University of Southern California’s Information Sciences Institute. This shows its applicability to international deployment.

Larry Landweber of the University of Wisconsin creates THEORYNET providing e-mail between over 100 researchers and linking elements of the University of Wisconsin in different cities via a commercial packet service like Telenet.

1978: The appearance of the first very small computers and their potential for communication via modem to dial up services starts a boom in a new set of niche industries, like software and modems.

Vint Cerf at DARPA continues the vision of the Internet, forming an International Cooperation Board chaired by Peter Kirstein of University College London, and an Internet Configuration Control Board, chaired by Dave Clark of MIT.

The ARPANET experiment formally is complete. This leaves an array of boards and task forces over the next few years trying to sustain the vision of a free and open Internet that can keep up with the growth of computing.

1979: Larry Landweber at Wisconsin holds a meeting with six other universities to discuss the possibility of building a Computer Science Research Network to be called CSNET. Bob Kahn attends as an advisor from DARPA, and Kent Curtis attends from NSF's computer research programs. The idea evolves over the summer between Landweber, Peter Denning (Purdue), Dave Farber (Delaware), and Tony Hearn (Utah).

In November, the group submits a proposal to NSF to fund a consortium of eleven universities at an estimated cost of $3 million over five years. This is viewed as too costly by the NSF.

USENET starts a series of shell scripts written by Steve Bellovin at UNC to help communicate with Duke. Newsgroups start with a name that gives an idea of its content. USENET is an early example of a client server where users dial in to a server with requests to forward certain newsgroup postings. The server then "serves" the request.

1980: Landweber's proposal has many enthusiastic reviewers. At an NSF-sponsored workshop, the idea is revised in a way that both wins approval and opens up a new epoch for NSF itself. The revised proposal includes many more universities. It proposes a three-tiered structure involving ARPANET, a TELENET-based system, and an e-mail only service called PhoneNet. Gateways connect the tiers into a seamless whole. This brings the cost of a site within the reach of the smallest universities. Moreover, NSF agrees to manage CSNET for two years, after which it will turn it over to the University Corporation for Atmospheric Research (UCAR), which is made up of more than fifty academic institutions.

The National Science Board approves the new plan and funds it for five years at a cost of $5 million. Since the protocols for interconnecting the subnets of CSNET include TCP/IP, NSF becomes an early supporter of the Internet.

NASA has ARPANET nodes, as do many Department of Energy (DOE) sites. Now several Federal agencies support the Internet, and the number is growing.

Research by David Patterson at Berkeley and John Hennessy at Stanford promotes "reduced instruction set" computing. IBM selects the disk operating system DOS, developed by Microsoft, to operate its planned PC.

1981: By the beginning of the year, more than 200 computers in dozens of institutions have been connected in CSNET. BITNET, another startup network, is based on protocols that include file transfer via e-mail rather than by the FTP procedure of the ARPA protocols.

The Internet Working Group of DARPA publishes a plan for the transition of the entire network from the Network Control Protocol to the Transmission Control Protocol/ Internet Protocol (TCT/IP) protocols developed since 1974 and already in wide use (RFC 801).

At Berkeley, Bill Joy incorporates the new TCP/IP suite into the next release of the Unix operating system. The first "portable" computer is launched in the form of the Osborne, a 24-pound suitcase-sized device.

The IBM PC is launched in August 1981.

Meanwhile, Japan mounts a successful challenge to US chip makers by producing 64-kbit chips so inexpensively that US competitors charge the chips are being "dumped" on the US market.

1982: *Time Magazine* names "the computer" its "Man of the Year." Cray Research announces plans to market the Cray X-MP system in place of the Cray-1. At the other end of the scale, the IBM PC "clones" begin appearing.

An NSF panel chaired by the Courant Institute's Peter Lax reports that US scientists lack access to supercomputers. It contains the testimony of University of Illinois astrophysicist Larry Smarr that members of his discipline have been forced to travel to Germany to use American-made supercomputers.

The period during which *ad hoc* networking systems have flourished has left TCP/IP as only one contender for the title of "standard." Indeed, the International Organization for Standards (ISO) has written and is pushing ahead with a "reference" model of an interconnection standard called Open Systems Interconnection (OSI) – already adopted in preliminary form for interconnecting DEC equipment. But while OSI is a standard existing for the most part on paper, the combination of TCP/IP and the local area networks created with Ethernet technology are driving the expansion of the living Internet.

Drew Major and Kyle Powell write Snipes, an action game to be played on PCs over the network. They package the game as a "demo" for a PC software product from SuperSet Software, Inc. This is the beginning of Novell.

Digital Communications Associates introduces the first coaxial cable interface for micro-to-mainframe communications.

1983: In January, the ARPANET standardizes on the TCP/IP protocols adopted by the Department of Defense (DOD). The Defense Communications Agency decides to split the network into a public "ARPANET" and a classified "MILNET," with only 45 hosts remaining on the ARPANET. Jon Postel issues an RFC assigning numbers to the various interconnected nets. Barry Leiner takes Vint Cerf's place at DARPA, managing the Internet.

Numbering the Internet hosts and keeping tabs on the host names simply fails to scale with the growth of the Internet. In November, Jon Postel and Paul Mockapetris of USC/ISI and Craig Partridge of BBN develop the Domain Name System (DNS) and recommend the use of the now familiar user@host.domain addressing system.

The number of computers connected via these hosts is much larger, and the growth is accelerating with the commercialization of Ethernet.

Having incorporated TCP/IP into Berkeley Unix, Bill Joy is key to the formation of Sun Microsystems. Sun develops workstations that ship with Berkeley Unix and feature built-in networking. At the same time, the Apollo workstations ship with a special version of a token ring network.

In July 1983, an NSF working group, chaired by Kent Curtis, issues a plan for "A National Computing Environment for Academic Research" to remedy the problems noted in the Lax report. Congressional hearings result in advice to NSF to undertake an even more ambitious plan to make supercomputers available to US scientists.

1984: In January, Apple announces the Macintosh. Its user-friendly interface swells the ranks of new computer users.

Novelist William Gibson coins the term cyberspace in *Neuromancer*, a book that adds a new genre to science fiction and fantasy.

The newly developed DNS is introduced across the Internet, with the now familiar domains of .gov, .mil, .edu, .org, .net, and .com. A domain called .int, for international entities, is not much used. Instead, hosts in other countries take a two-letter domain indicating the country. The British JANET explicitly announces its intention to serve the nation's higher education community, regardless of discipline.

Most important for the Internet, NSF issues a request for proposals to establish supercomputer centers that will provide access to the entire US research community, regardless of discipline and location. A new division of Advanced Scientific Computing is created with a budget of $200 million over five years.

Datapoint, the first company to offer networked computers, continues in the marketplace, but fails to achieve critical mass.

1985: NSF announces the award of five supercomputing center contracts:

- Cornell Theory Center (CTC), directed by Nobel laureate Ken Wilson.
- The John Von Neumann Center (JVNC) at Princeton, directed by computational fluid dynamicist Steven Orszag.
- The National Center for Supercomputing Applications (NCSA), directed at the University of Illinois by astrophysicist Larry Smarr.
- The Pittsburgh Supercomputing Center (PSC), sharing locations at Westinghouse, the University of Pittsburgh, and Carnegie Mellon University, directed by Michael Levine and Ralph Roskies.
- The San Diego Supercomputer Center (SDSC), on the campus of the University of California, San Diego, and administered by the General Atomics Company under the direction of nuclear engineer Sid Karin.

By the end of 1985, the number of hosts on the Internet (all TCP/IP interconnected networks) has reached 2,000.

MIT translates and publishes *Computers and Communication* by Dr. Koji Kobayashi, the Chairman of NEC. Dr. Kobayashi, who joined NEC in 1929, articulates his clear vision of “C & C,” the integration of computing and communication.

1986: The 56 Kbps backbone between the NSF centers leads to the creation of a number of regional feeder networks – JVNCNET, NYSERNET, SURANET, SDSCNET and BARRNET – among others. With the backbone, these regionals start to build a hub and spoke infrastructure. This growth in the number of interconnected networks drives a major expansion in the community including the DOE, DOD and NASA.

Between the beginning of 1986 and the end of 1987 the number of networks grows from 2,000 to nearly 30,000.

TCP/IP is available on workstations and PCs such as the newly introduced Compaq portable computer. Ethernet is becoming accepted for wiring inside buildings and across campuses. Each of these developments drives the introduction of terms such as bridging and routing and the need for readily available information on TCP/IP in workshops and manuals. Companies such as Proteon, Synoptics, Banyan, Cabletron, Wellfleet, and Cisco emerge with products to feed this explosion.

At the same time, other parts of the US Government and many of the traditional computer vendors mount an attempt to validate their products being built to the OSI theoretical specifications, in the form of the Corporation for Open Systems.

USENET starts a major shakeup which becomes known as the “Great Renaming.” A driving force is that, as many messages are traveling over ARPANET, desirable new news groups such as “alt.sex” and “alt.drugs” are not allowed.

1987: The NSF, realizing the rate and commercial significance of the growth of the Internet, signs a cooperative agreement with Merit Networks which is assisted by IBM and MCI. Rick Adams co-founds UUNET to provide commercial access to UUCP and the USENET newsgroups, which are now available for the PC. BITNET and CSNET also merge to form CREN.

The NSF starts to implement its T1 backbone between the supercomputing centers with 24 RT-PCs in parallel implemented by IBM as “parallel routers.” The T1 idea is so successful that proposals for T3 speeds in the backbone begin.

In early 1987 the number of hosts passes 10,000 and by year-end there have been over 1,000 RFCs issued.

Network management starts to become a major issue and it becomes clear that a protocol is needed between routers to allow remote management. SNMP is chosen as a simple, quick, near term solution.

1988: The upgrade of the NSFNET backbone to T1 completes and the Internet starts to become more international with the connection of Canada, Denmark, Finland, France, Iceland, Norway and Sweden.

In the US more regionals spring up – Los Nettos and CERFnet both in California. In addition, Fidonet, a popular traditional bulletin board system (BBS) joins the net.

Dan Lynch organizes the first Interop commercial conference in San Jose for vendors whose TCP/IP products interoperate reliably. Fifty companies make the cut and 5,000 networkers come to see it all running, to see what works, and to learn what doesn't work.

The US Government pronounces its OSI Profile (GOSIP) is to be supported in all products purchased for government use, and states that TCP/IP is an interim solution!

The Morris WORM burrows on the Internet into 6,000 of the 60,000 hosts now on the network. This is the first worm experience and DARPA forms the Computer Emergency Response Team (CERT) to deal with future incidents.

CNRI obtains permission from the Federal Networking Council and from MCI to interconnect the commercial MCI Mail service to the Internet. This broke the barrier to carrying commercial traffic on the Internet backbone. By 1989 MCI Mail, OnTyme, Telemail and CompuServe had all interconnected their commercial e-mail systems to the Internet and, in so doing, interconnected with each other for the first time. This was the start of commercial Internet services in the United States (and possibly the world).

1989: The number of hosts increases from 80,000 in January to 130,000 in July to over 160,000 in November!

Australia, Germany, Israel, Italy, Japan, Mexico, Netherlands, New Zealand and the United Kingdom join the Internet.

Commercial e-mail relays start between MCIMail through CNRI and Compuserve through Ohio State. The Internet Architecture Board reorganizes again reforming the IETF and the IRTF.

Networks speed up. NSFNET T3 (45 Mbps) nodes operate. At Interop 100 Mbps LAN technology, known as FDDI, interoperates among several vendors. The telephone companies start to work on their own wide area packet switching service at higher speeds – calling it SMDS.

Bob Kahn and Vint Cerf at CNRI hold the first Gigabit (1,000 Mbps) Testbed workshops with funding from ARPA and NSF. Over 600 people from a wide range of industry, government, and academia attend to discuss the formation of 6 gigabit testbeds across the country.

The Cray 3, a direct descendant of the Cray line, starting from the CDC 6600, is produced.

In Switzerland at CERN Tim Berners-Lee addresses the issue of the constant change in the currency of information and the turn-over of people on projects. Instead of an hierarchical or keyword organization, Berners-Lee proposes a hypertext system that will run across the Internet on different operating systems. This was the World Wide Web (WWW).

1990: ARPANET formally shuts down. In 20 years, “the net” has grown from four to over 300,000 hosts. Countries connecting in 1990 include Argentina, Austria, Belgium, Brazil, Chile, Greece, India, Ireland, South Korea, Spain, and Switzerland.

Several search tools, such as ARCHIE, Gopher, and WAIS start to appear. Institutions like the National Library of Medicine, Dow Jones, and Dialog are now on line.

More “**worms**” burrow on the net, with as many as 130 reports leading to 12 real ones! This is a further indication of the transition to a wider audience.

1991: The net’s dramatic growth continues with NSF lifting any restrictions on commercial use. Interchanges form with popular providers such as UUNET and PSInet. Congress passes the Gore Bill to create the National Research and Education Network, or NREN initiative. In another sign of popularity, privacy becomes an “issue,” with proposed solutions such as Pretty Good Privacy (PGP).

The NSFNET backbone upgrades to T3, or 44 Mbps. Total traffic exceeds 1 trillion bytes, or 10 billion packets per month! Over 100 countries are now connected with over 600,000 hosts and nearly 5,000 separate networks.

WAIS’s and Gophers help meet the challenge of searching for information throughout this exploding infrastructure of computers.

1992: The Internet becomes such a part of the computing establishment that a professional society forms to guide it on its way. The Internet Society (ISOC), with Vint Cerf and Bob Kahn among its founders, validates the coming of age of internetworking and its pervasive role in the lives of professionals in developed countries. The IAB and its supporting committees become part of ISOC.

The number of networks exceeds 7,500 and the number of computers connected passes 1,000,000. The MBONE for the first time carries audio and video. The challenge to the telephone network’s dominance as the basis for communicating between people is seen for the first time; the Internet is no longer just for machines to talk to each other.

During the summer, students at NCSA in University of Illinois at Urbana-Champaign modify Tim Berners-Lee’s hypertext proposal. In a few weeks MOSAIC is born within the campus. Larry Smarr shows it to Jim Clark, who founds Netscape as a result.

The WWW bursts into the world and the growth of the Internet explodes like a supernova. What had been doubling each year now doubles in three months. What began as an ARPA experiment has, in the span of just 30 years, become a part of the world’s popular culture.

The milestones during this period are: APRANET connects four nodes in 1969; Telnet proposed in 1974; incorporating TCP/IP protocol in Unix system in 1976; client/server model on USENET in 1979; DNS is introduced with its common domains such as .com, .edu, .gov in 1984; TCP/IP interconnected hosts reached 2,000 on the Internet in 1985; NSFNET upgraded to T1 connecting eight countries in 1988; the Morris worm spread the Internet in 1988; more worms burrow on Internet in 1990; the first browser Mosaic came to earth in 1992.

In 30 years, Internet has become a vital part in our daily life. Technology innovation speeded fast, on the other hand, computer and network security emerged as the result of worms and viruses.

1.1.3 Timeline of Computer Security History

The broad concept of computer security happened long before computers were invented. However attacks on computers and networks had not become common until the early 1990s. Now, it becomes not only a serious problem to regular computer users, organizations and government agencies but also a threat to national security.

1965: William D. Mathews from MIT found a vulnerability in a Multics CTSS running on a IBM 7094. This flaw discloses the contents of the password file. The issue occurred when multiple instances of the system text editor were invoked, causing the editor to create temporary files with a constant name. This would inexplicably cause the contents of the system CTSS password file to display to any user logging into the system.

1971: John T. Draper (later nicknamed Captain Crunch), his friend Joe Engressia, and blue box phone phreaking hit the news with an *Esquire* feature story.

1981: The Warelords forms in the United States, founded by Black Bart (cracker of Dung Beetles in 1982) in St. Louis, Missouri, and was composed of many teenage hackers, phreakers, coders, and largely black hat-style underground computer geeks. One of the more notable group members was Tennessee Tuxedo, a young man who was instrumental in developing conference calls via the use of trunk line phreaking via the use of the Novation Apple Cat II that allowed them to share their current hacks, phreaking codes, and new software releases. Other notable members were The Apple Bandit, Krakowicz, and Krac-man. Black Bart was clever at using his nationally known and very popular BBS system in order to promote the latest gaming software. He used that relationship to his advantage, often shipping the original pre-released software to his most trusted code crackers during the beta-testing phase, weeks prior to their public release. The Warelords often collaborated with other piracy groups at the time, such as The Syndicate and The Midwest Pirates Guild and developed an international ring of involved piracy groups that reached as far away as Japan. Long before the movie War Games went into pre-production, The Warelords had successfully infiltrated such corporations and institutions as The White House, Southwestern Bell "Ma Bell" Mainframe Systems, and large corporate providers of voice mail systems.

1990: Operation Sundevil introduced. After a prolonged sting investigation, Secret Service agents swoop down on organizers and prominent members of BBSs in 14 US cities including the Legion of Doom, conducting early-morning raids and arrests. The arrests involve and are aimed at cracking down on credit-card theft and telephone and wire fraud. The result is a breakdown in the hacking community, with members informing on each other in exchange for immunity. The offices of Steve Jackson Games are also raided, and the role-playing sourcebook GURPS Cyberpunk is confiscated, possibly because the government fears it is a "handbook for computer crime." Legal battles arise that prompt the formation of the Electronic Frontier Foundation, including the trial of Knight Lightning.

Australian federal police tracking Realm members Phoenix, Electron, and Nom are the first in the world to use a remote data intercept to gain evidence for a computer crime prosecution.

1994: Summer: Russian crackers siphon $10 million from Citibank and transfer the money to bank accounts around the world. Vladimir Levin, the 30-year-old ringleader, uses his work laptop after hours to transfer the funds to accounts in Finland and Israel. Levin stands trial in the United States and is sentenced to three years in prison. Authorities recover all but $400,000 of the stolen money.

Hackers adapt to emergence of the World Wide Web quickly, moving all their how-to information and hacking programs from the old BBSs to new hacker Web sites.

AOHell is released, a freeware application that allows a burgeoning community of unskilled script kiddies to wreak havoc on America Online. For days, hundreds of thousands of AOL users find their mailboxes flooded with multi-megabyte e-mail bombs and their chat rooms disrupted with spam messages.

1996: Hackers alter Web sites of the United States Department of Justice (August), the CIA (October), and the US Air Force (December).

Canadian hacker group, Brotherhood, breaks into the Canadian Broadcasting Corporation.

The US General Accounting Office reports that hackers attempted to break into Defense Department computer files some 250,000 times in 1995 alone. About 65% of the attempts were successful, according to the report.

The MP3 format gains popularity in the hacker world. Many hackers begin setting up sharing sites via FTP, Hotline, IRC, and Usenet.

1997: A 15-year-old Croatian youth penetrates computers at a US Air Force base in Guam.

June: Eligible Receiver 97 tests the American government's readiness against cyberattacks.

December: Information Security publishes first issue.

First high-profile attacks on Microsoft's Windows NT operating system.

In response to the MP3 popularity, the Recording Industry Association of America begins cracking down on FTPs. The RIAA begins a campaign of lawsuits shutting down many of the owners of these sites including the more popular ripper/distributors The Maxx (Germany, age 14), Chapel976 (USA, age 15), Bulletboy (UK, age 16), Sn4rf (Canada, age 14) and others in their young teens via their ISPs. Their houses are raided and their computers and modems are taken. The RIAA fails to cut off the head of the MP3 beast and within a year and a half, Napster is released.

1998: January: Yahoo! notifies Internet users that anyone visiting its site in recent weeks might have downloaded a logic bomb and worm planted by hackers claiming a “logic bomb” will go off if Kevin Mitnick is not released from prison.

January: Anti-hacker runs during Super Bowl XXXII.

February: The Internet Software Consortium proposes the use of DNSSEC (domain-name system security extensions) to secure DNS servers.

May 19: The seven members of the hacker think tank known as L0pht testifies in front of the US congressional Government Affairs committee on “Weak Computer Security in Government.”

June: Information Security publishes its first annual Industry Survey, finding that nearly three-quarters of organizations suffered a security incident in the previous year.

October: “US Attorney General Janet Reno announces National Infrastructure Protection Center.”

1999: Software security goes mainstream in the wake of Microsoft’s Windows 98 release, 1999 becomes a banner year for security (and hacking). Hundreds of advisories and patches are released in response to newfound (and widely publicized) bugs in Windows and other commercial software products. A host of security software vendors release anti-hacking products for use on home computers.

The Electronic Civil Disobedience project, an online political performance-art group, attacks the Pentagon calling it conceptual art and claiming it to be a protest against the US support of the suppression of rebels in southern Mexico by the Mexican government. ECD uses the FloodNet software to bombard its opponents with access requests.

US President Bill Clinton announces a $1.46 billion initiative to improve government computer security. The plan would establish a network of intrusion detection monitors for certain federal agencies and encourage the private sector to do the same.

January 7: an international coalition of hackers (including CULT OF THE DEAD COW, 2600’s staff, Phrack’s staff, L0pht, and the Chaos Computer Club) issued a joint statement condemning the LoU’s declaration of war. The LoU responded by withdrawing its declaration.

A hacker interviewed by Hilly Rose during the Art Bell Coast-to-Coast Radio Show exposes a plot by Al-Qaida to derail Amtrak trains. This results in ALL trains being forcibly stopped over Y2K as a safety measure.

March: The Melissa worm is released and quickly becomes the most costly malware outbreak to date.

July: CULT OF THE DEAD COW releases Back Orifice 2000 at DEF CON.

August: Kevin Mitnick, “the most wanted man in cyberspace,” sentenced to five years, of which over four years had already been spent pre-trial including eight months solitary confinement.

September: Level Seven hacks the US Embassy in China's Web site and places racist, anti-government slogans on embassy site in regards to 1998 US embassy bombings.

September 16: The United States Department of Justice sentences the "Phone Masters."

October: American Express introduces the "Blue" smart card, the industry's first chip-based credit card in the US.

2000: May: The ILOVEYOU worm, also known as VBS/Loveletter and Love Bug worm, is a computer worm written in VBScript. It infected millions of computers worldwide within a few hours of its release. It is considered to be one of the most damaging worms ever. It originated in the Philippines; made by an AMA Computer College student for his thesis.

September: teenage hacker Jonathan James becomes first juvenile to serve jail time for hacking.

2001: Microsoft becomes the prominent victim of a new type of hack that attacks the domain name server. In these denial-of-service attacks, the DNS paths that take users to Microsoft's Web sites are corrupted.

February: A Dutch cracker releases the Anna Kournikova virus, initiating a wave of viruses that tempts users to open the infected attachment by promising a sexy picture of the Russian tennis star.

April: FBI agents trick two into coming to the US and revealing how they were hacking US banks.

May: Spurred by elevated tensions in Sino-American diplomatic relations, US and Chinese hackers engage in skirmishes of Web defacements that many dub "The Sixth Cyberwar."

July: Russian programmer Dmitry Sklyarov is arrested at the annual Def Con hacker convention. He is the first person criminally charged with violating the Digital Millennium Copyright Act (DMCA).

August: Code Red worm, infects ts.

2002: January: Bill Gates decrees that Microsoft will secure its products and services, and kicks off a massive internal training and quality control campaign.

May: Klez.H, a variant of the worm discovered in November 2001, becomes the biggest malware outbreak in terms of machines infected, but causes little monetary damage.

June: The Bush administration files a bill to create the Department of Homeland Security, which, among other things, will be responsible for protecting the nation's critical IT infrastructure.

August: Researcher Chris Paget publishes a paper describing "shatter attacks," detailing how Windows' unauthenticated messaging system can be used to take over a machine. The paper raises questions about how securable Windows could ever be.

October: The International Information Systems Security Certification Consortium – (ISC)2 – confers its 10,000th CISSP certification.

2003: The hacker group Anonymous was formed.

March: CULT OF THE DEAD COW and Hacktivismo are given permission by the United States Department of Commerce to export software utilizing strong encryption.

December 18: Milford Man pleas guilty to hacking.

2004: March: Myron Tereshchuk is arrested for attempting to extort $17 million from Micropatent.

July: North Korea claims to have trained 500 hackers who successfully crack South Korean, Japanese, and their allies' computer systems.

2005: April 2: Rafael Núñez aka RaFa, a notorious member of the hacking group World of Hell is arrested following his arrival at Miami International Airport for breaking into the Defense Information Systems Agency computer system on June 2001.

September 13: Cameron Lacroix is sentenced to 11 months for gaining access to T-Mobile USA's network and exploiting Paris Hilton's Sidekick.

November 3: Jeanson James Ancheta, whom prosecutors say was a member of the "Botmaster Underground," a group of script kiddies mostly noted for their excessive use of bot attacks and propagating vast amounts of spam, was taken into custody after being lured to FBI offices in Los Angeles.

2006: January: One of the few worms to take after the old form of malware, destruction of data rather than the accumulation of zombie networks to launch attacks from, is discovered. It had various names, including Kama Sutra (used by most media reports), Black Worm, Mywife, Blackmal, Nyxem version D, Kapser, KillAV, Grew and CME-24. The worm would spread through e-mail client address books, and would search for documents and fill them with garbage, instead of deleting them to confuse the user. It would also hit a Web page counter when it took control, allowing the programmer who created it as well as the world to track the progress of the worm. It would replace documents with random garbage on the third of every month. It was hyped by the media but actually affected relatively few computers, and was not a real threat for most users.

February: Direct-to-video film The Net 2.0 is released, as a sequel to The Net, following the same plotline, but with updated technology used in the film, using different characters, and different complications. The director of The Net 2.0, Charles Winkler, is son of Irwin Winkler, the director of The Net.

May: Jeanson James Ancheta receives a 57-month prison sentence, and is ordered to pay damages amounting to $15,000.00 to the Naval Air Warfare Center in China Lake and the Defense Information Systems Agency, for damage done due to distributed denial of service (DDoS) attacks and hacking. Ancheta also had to forfeit his gains to the government, which include $60,000 in cash, a BMW, and computer equipment.

May: Largest Defacement in Web history is performed by the Turkish hacker iSKORPiTX who successfully hacked 21,549 websites in one shot.

July: Robert Moore and Edwin Pena featured on Americas Most Wanted with Kevin Mitnick presenting their case commit the first VOIP crime ever seen in the

USA. Robert Moore served two years in federal prison with a $152,000.00 restitution while Edwin Pena was sentenced to 10 years and a $1 million restitution.

September: Viodentia releases FairUse4WM tool which would remove DRM information off WMA music downloaded from music services such as Yahoo Unlimited, Napster, Rhapsody Music and Urge.

2007: May 17: Estonia recovers from massive denial-of-service attack.

June 13: FBI Operation Bot Roast finds over 1 million botnet victims.

June 21: A spear phishing incident at the Office of the Secretary of Defense steals sensitive US defense information, leading to significant changes in identity and message-source verification at OSD.

August 11: United Nations Web site hacked by Turkish Hacker Kerem125.

October 7: Trend Micro Web site successfully hacked by Turkish hacker Janizary (aka Utku).

November 29: FBI Operation Bot Roast II: 1 million infected PCs, $20 million in losses and eight indictments.

2008: January 17: Project Chanology; Anonymous attacks Scientology Web site servers around the world. Private documents are stolen from Scientology computers and distributed over the Internet.

March 7: Around 20 Chinese hackers claim to have gained access to the world's most sensitive sites, including The Pentagon. They operate from a bare apartment on a Chinese island.

2009: April 4: Conficker worm infiltrated millions of PCs worldwide including many government-level top-security computer networks.

2010: March 24: UN department of safety and security hacked by Turkish hacker DigitALL (1923Turk) Mirror Link.

January 12: Operation Aurora Google publicly reveals that it has been on the receiving end of a "highly sophisticated and targeted attack on our corporate infrastructure originating from China that resulted in the theft of intellectual property from Google."

June: Stuxnet The Stuxnet worm is found by VirusBlokAda. Stuxnet was unusual in that while it spread via Windows computers, its payload targeted just one specific model and type of SCADA system. It slowly became clear that it was a cyber attack on Iran's nuclear facilities – with most experts believing that Israel was behind it – perhaps with US help.

December 3: The first Malware Conference, MALCON takes place in India. Malware coders are invited to showcase their skills at this annual event and an advanced malware for Symbian OS is released.

2011: The Hacker group Lulz security is formed.

April 17: An "external intrusion" sends the PlayStation Network offline, and compromises personally identifying information (possibly including credit card details) of its 77 million accounts, in what is claimed to be one of the five largest data breaches ever.

The hacker group LulzRaft is formed.

September: Bangladeshi hacker TiGER-M@TE made world record in defacement history by hacking 700,000 Web sites in one shot.

October 16: The YouTube channel of Sesame Street was hacked, streaming pornographic content for about 22 minutes.

November 1: The main phone and Internet networks of the Palestinian territories sustained a hacker attack from multiple locations worldwide.

December 14: Five members of the Norwegian hacker group Noria were arrested, allegedly suspected for hacking into the e-mail account of the terrorist Anders Behring Breivik.

2012: Saudi hacker, 0xOmar, published over 400,000 credit cards online, and threatened Israel with the release 1 million credit cards in the future.

In response to that incident, an Israeli hacker published over 200 Saudi's credit cards online.

January 6: Hacker group The Hacker Encrypters found and reported an open SQLi exploit on Facebook. The results of the exploit have been posted on Pastebin.

January 7: Team Appunity, a group of Norwegians who knew how to use sqli tools developed by others, got arrested for breaking into and publishing the user database of Norway's largest prostitution Web site.

February 8: Foxconn is hacked by rising hacker group, Swagg Security, releasing a massive amount of data including e-mail logins, server logins, and even more alarming – bank account credentials of large companies like Apple and Microsoft. Swagg Security stages the attack just as a Foxconn protest ignites against terrible working conditions.

After 50 years' of computer architecture innovation and 30 years of Internet revolution, we are now in the situation of finding ways to protect the computer systems. Firewalls, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) and many other types of security equipment are common practice but still no one can guarantee that information stored on computers cannot be stolen. This book hopes to address this problem and to discuss solutions to prevent data bleaches on computers.

1.2 John von Neumann Computer Architecture

The Neumann architecture was the first modern design for computers based on the stored program concept (Dumas, 2006). Figure 1.3 shows a block diagram of Neumann architecture. The arithmetic and logic unit is the place where calculations take place. The control unit interprets the instructions and coordinates the operations. Memory is used to store instructions and data as well as intermediate results. Input and output interfaces are used to read data and write results.

Neumann defines a computer as a "very high speed automatic digital computing system, and in particular with its logical control." (Foster and Iberall, 1985)

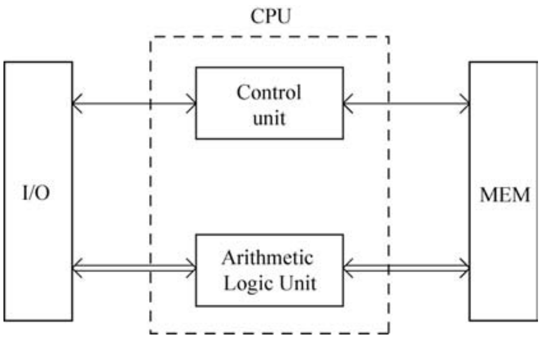


Figure 1.3 Neumann computer architecture

As a computing device, it will have to perform the elementary arithmetic operations most frequently including addition, subtraction, multiplication, and division. It may be extended to include such operations as square roots, logarithmic, and triangulation (sin, cos, arcsin, arccos). This part is the central arithmetical (CA) component which could be considered as central arithmetic and logic unit (ALU).

The logic control device is used to proper sequence its operations more efficiently by a central control (CC) component. A considerable memory is needed to carry out a multiplication or a division and store a series of intermediate results. Various parts of this memory have to perform functions which differ somewhat in their nature and considerably in their purpose. At any rate the total memory constitutes the specific part of a computer system: M.

The three specific parts CA, CC (double C which is now named the central processing unit) and M correspond to the associate neurons in the human nervous system. There are input and output components that are equivalent to the motor or efferent neurons.

In summary, Neumann architecture depicts a computer as a machine with a central processing unit (CPU), internal memory (M) and external storage (S), input/output devices (I/O) and wires that link them together called a bus.

Nowadays, computers still stick to this architecture. When purchasing a computer, people usually consider how fast the CPU or multi-core CPU is, how big the (internal) memory and hard drive are, and how well the I/O is such as the video card, screen size and multimedia features, and so on.

RISC, or reduced instruction set computer, is another type of computer architecture that utilizes a small, highly-optimized set of instructions, rather than a more specialized set of instructions. RISC processors have a CPI (clock per instruction) of one cycle (Shiva, 2000). This is due to the optimization of each instruction on the CPU and a technique called pipelining. Pipelining allows for simultaneous execution of instructions to speed up the execution speed. RISC also uses a large number of registers to prevent large amounts of interaction with memory.

Parallel processing architecture uses multiple processors to form a super computer to solve large problems. The task is often broken into small pieces and distributed across multiple processors or machines, and solved concurrently.

From a security point of view, both RISC architecture and parallel processing architecture still fall into the Neumann architecture category. RISC uses one cycle execution time to improve the efficiency in pipelining. The change is mostly within the processors. The other system components (memory, I/O, storage) are still the same as Neumann's. Parallel processing, on the other hand, contains multiple processors or computers. Each one is a Neumann architecture computer.

1.3 Memory and Storage

As Neumann has described, memory is used to carry out arithmetic operations and store intermediate results. There are two factors to the memory that affect a computer speed the most: the size and the speed. More memory can result in less intermediate results storage and less operation therefore improved computer speed. High speed memory can reduce the delays during each operation. A complex problem requires a tremendous amount of computation. Thus the delay would be significant if a low speed memory is used.

The ideal memory can be characterized as zero or less delay in read and write access, large scale or high density in size, reliable and durable, less expensive and low power. That is why memory is now divided into different categories. Memory closer to CPU is usually faster and more expensive. So it is usually small in capacity. Memory far away from CPU is usually slower and less expensive. So it is usually large in capacity. We often refer to it as storage. USB flash drives and hard drives are examples of external memories or storages.

The read and write property for most memories make them convenient to store data. On the other hand, it also easily altered if a computer is compromised. So memory needs to be defined into regions where program and data are separated. In addition some policies and security measures need to be implemented.

Computer memory is organized into hierarchies centered with the CPU. Registers are the fastest memory. They are specially wired to guarantee the maximum speed. Cache memory is the next fastest memory. The idea behind using cache is to reduce the time for CPU to read/write data directly from the main memory, as this may cause longer delay. Cache memory usually uses static RAM or SRAM, is accessed by blocks and is often divided into two to three levels. Main memory has much larger capacity to store application and data. It usually uses dynamic RAM or DRAM, a type of memory that is cheaper and can be highly integrated, but with a little bit less speed than SRAM.

Hard drives are external memory or storage that has a vast amount of capacity. Solid state drive (SSD) is a new type of storage that acts as a normal hard drive but without mechanical movements. So they are more reliable and faster. With the advancement of technology, SSDs are becoming cheaper and more affordable.

Virtual memory emerges as a result of applications which need more and more memory to improve performance. Virtual memory by its name is virtual. There is no physical memory associated with it. So virtual memory uses a region of external memory to store programs and data.

Main memory is usually divided into different segments: program, data, data buffers and so on. If a section, for example the data buffer, is extended beyond the limit reserved for that section, then overflow may affect the other segment. If data overflow to the program segment, then programs may execute with unexpected results. If the overflow is set by an attacker, then it may direct the program execution to some preset areas where malicious programs reside, the attacker may take control of the computer system and do even more damage to the computer system. This is called the buffer overflow attack.

1.4 Input/Output and Network Interface

Computer systems typically have several I/O devices. The I/O interface is used to receive data in or write data out with devices such as a keyboard, a mouse, displays, and printers.

Compared with the speed of CPU and memory, a keyboard is a very slow device so an input interface can store data temporally typed from a keyboard and send to memory when reaching the capacity (or buffer size).

A display with animation, however, requires high speed to refresh the screen data. So the interface (video card) needs more speed and more buffer size (or video memory) to accommodate this.

Sometimes, people need to send volume data from input devices to the computer memory without requiring any computation or vice versa. This can be done by using an interface called direct memory access (DMA). The benefit of using a DMA is that data will not go through CPU so a computer can handle other tasks while block data are read in or written out (Randell, 1982).

Computers now are interconnected through network or Internet. Most people check emails every day if not every hour. Vast information can be found on the Internet. So a network interface card (NIC) has become an integral part of a computer system to handle data communication needs.

Computer networks did not exist in Neumann's age so there was no network component on an old computer. People usually consider a network component as essentially belonging to the input and output devices. When we look carefully at the differences between a general I/O and a network component, we can find out that a network by no mean can be considered simply as a general I/O device.

With the widespread use of the Internet in the 1990s, data security has become a problem. Hackers are able to break into computers and servers to steal data, in many cases sensitive data. The loss caused by the data bleach has reached multiple billion dollars every year and is still on the increase. The damage is beyond money and threatens national security. As a result, network security has become a big challenge in everyday server and computer operations.

I/O devices have slower speed compared to processors. Synchronous communication between them may cause problems. I/O devices will lose data if CPU sends/receives data using processor cycles. On the other hand, according to the I/O device speed, the CPU would have to wait during each I/O operation, thus downgrading the performance. Interrupt is a signal that is used when collaborating communications between high speed processors and low speed I/O devices. When an I/O device has finished handling data it received from the processor, it issues an interrupt request. When CPU receives the request, it saves the current environment, sends/receives data and then resumes the original tasks.

There are different interfaces to connect the processor and the I/O devices. Many recent I/O devices are equipped with the universal series bus (USB). USB unifies the interfaces with its easy to use, plug-and-play connectivity, and hot plug characteristics. In addition, the speed of USB is considerable high so that it can support most I/O devices. The new Super Speed USB 3.0 has the transfer speed of 5 Gb/s.

I/O security is also related to data security. Data stored on hard drives may be exposed to attackers especially insider attackers. Encrypting all data on hard drives is one solution but it may result in performance issues. USB storage devices can be used as bootable devices to start a computer. Unauthorized uses can let attackers steal data on computer systems. In places where data security is a concern, the USB ports should be disabled.

1.5 Single CPU and Multiple CPU Systems

Traditional computers contain one central processing unit. It is like a human brain that controls all activities and remembers all things around it. To solve a problem, an algorithm is programmed and executed as a serial stream in instructions. Only one instruction may execute at a time.

A multiple CPU system on the other hand can process a computation task simultaneously by breaking the problem into independent parts (Hwang, 1993). Each CPU then executes a portion of the problem. Such a procedure is called parallel processing. Parallel processing systems contain multiple CPUs that are interconnected or manufactured together as multi-core processors.

Here is an example of a multiple CPU system compared with a single CPU system. Suppose we want to process a color picture. As we know a color picture consists of three basic colors: red, green and blue (RGB). If we process this picture using a computer with three CPUs, then each CPU can process one color. For a single CPU system, theoretically it would have to take triple time to process this picture, first red, then green, and finally blue.

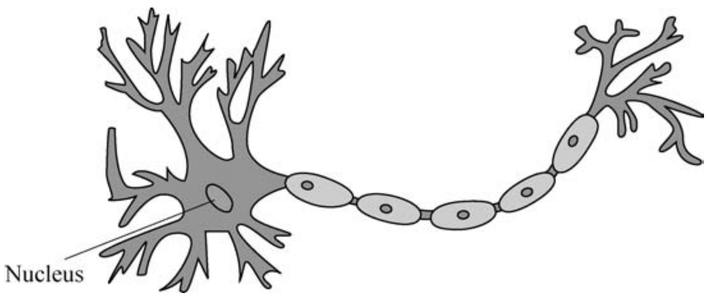


Figure 1.4 Structure of a typical neuron

People often refer to CPU as the "brain" of a computer system. People in some parts of the world think of a computer as an "electronic brain." For a multiple CPU computer system, how does such a match correspond to this image?

If you look inside the brain, there are about 100 billion (10^{11}) neurons. A neuron is a special type of cell that has a nucleus which contains genes and acts like the brain of the cell. It has electrical excitability and the presence of synapses used to transmit signals to other cells. Figure 1.4 shows the structure of a typical neuron.

There is a trend to add more CPUs into a computer. It by no means indicates there will be more "brains" in an identity (here is a computer). Instead some people consider CPUs in parallel processing or multicore computers as "neurons." A parallel processing computer has many CPUs, local memory, and interconnection networks. So it does function like a neuron that contains synapses used to transmit and receive signals between neurons. This may be another explanation for why no matter how many CPUs a computer has, its speed is no match with the human brain because no computer will contain a million CPUs, not to mention the 100 billion neurons in a human brain.

Processors were originally developed with only one core. More and more computers use multi-core processors where each core is acting like an independent processor. The architecture of multi-core processors can be organized as each core contains a CPU and level 1 cache. A bus interface connects all cores and also provides a level 2 cache. All cores and interfaces are integrated on one circuit (one die).

More advanced parallel processing computer systems use multiple processors or an array of processors to form a super computer. The processors can be at a centralized location or they can be at distributed locations.

In corporate businesses, more companies and organizations use services provided for them instead of buying a number of powerful computers (servers). This leads to the cloud computing concept. In a cloud computing model, some companies build the cloud which contains applications, platforms, and infrastructure. Other companies or organizations acting as users use these infrastructures, software and platforms as services. Since different organizations may share the same cloud, data separation and security are a common concern for running critical applications that contain sensitive data on a cloud.

On the other hand, thin clients also have the momentum as the client is cheaper, consumes much less energy therefore is more “green,” and is easy to manage. Thin clients use computation power of a remote computer. The technology is perfect for situations where high speed network is available and animation is not the main concern.

Distributed computing has tremendous power to accomplish jobs which otherwise would not be performed with traditional computers. Pharmaceutical companies use hundreds of thousands of computers distributed across the country to simulate clinical tests. It can greatly shorten the time for a new drugs test. Sophisticated computer hackers can also benefit from using the distributed computing platform. They are able to shut down VISA/Mastercard company’s websites and other government websites. They launched so called a distributed denial of service attack, or DDoS attack.

In a typical DDoS attack, the army of the attacker consists of master zombies and slave zombies. The hosts of both categories are compromised machines that have arisen during the scanning process and are infected by malicious code. The attacker coordinates and orders master zombies and they, in turn, coordinate and trigger slave zombies. More specifically, the attacker sends an attack command to master zombies and activates all attack processes on those machines, which are in hibernation, waiting for the appropriate command to wake up and start attacking. Then, master zombies, through those processes, send attack commands to slave zombies, ordering them to mount a DDoS attack against the victim. In that way, the agent machines (slave zombies) begin to send a large volume of packets to the victim, flooding its system with useless load and exhausting its resources. Figure 1.5 shows this kind of DDoS attack.

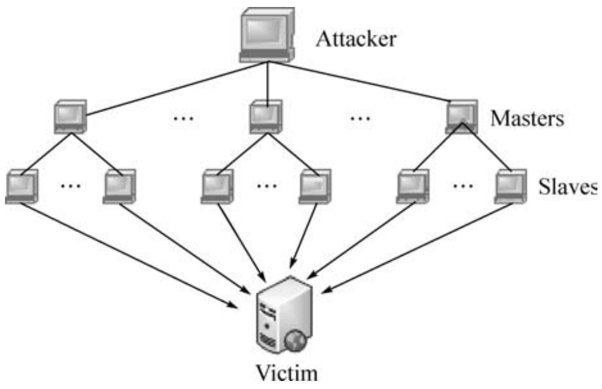


Figure 1.5 A DDoS attack

1.6 Overview of Computer Security

Computer security relies on confidentiality, integrity, and availability. A more comprehensive description includes:

- Authentication
- Authorization
- Confidentiality
- Integrity
- Availability
- Accountability
- Non-repudiation.

Certified information systems security professional (CISSP) classifies information security into ten domains:

- Access Control
- Application Development Security
- Business Continuity and Disaster Recovery Planning
- Cryptography
- Information Security Governance and Risk Management
- Legal, Regulations, Investigations and Compliance
- Operations Security
- Physical (Environmental) Security
- Security Architecture and Design
- Telecommunications and Network Security.

1.6.1 Confidentiality

From an information security point of view, confidentiality is the concealment of information or resources. From a computer architecture perspective, confidentiality is to design a computer system that is capable of protecting data and preventing intruders from stealing data in computer systems.

Access control mechanisms support confidentiality. A software solution to enforce access control mechanism is cryptography, which encrypts data to make it incomprehensible. A hardware solution is physical separation. Total physical separation is impractical so in many cases we use a hybrid method which is partial physical separation plus (software) access control polices.

Below are two examples that further describe the concept of confidentiality:

Example 1: A company named GEE does not want competitors to be able to see exactly how many or which equipment its clients happen to be ordering from the company, because that may give them competitive information. So GEE use the secure protocol to encrypt all the communication between its clients and the Web site. (More examples on this case will follow.)

Example 2: A newly proposed computer model enables a computer operator (owner) to control what data can go in and go out. Nobody can get data from this computer system without the permission of the computer operator. In this way, the confidentiality is guaranteed. Interested readers can read Chapter 10 for a detailed technical description about how to design and implement such a secure computer system.

Information hiding is another important aspect of confidentiality. For example, a program can encrypt a secret message into a picture. Only people with the program (or key) can reveal the secret. Without the key, people can only view it as a normal image. An example of such an encryption application programmed by the book author can be downloaded from www.wiley.com/go/wang/comp_arch.

1.6.2 Integrity

Integrity refers to the trustworthiness of data. There are two aspects: data in motion and data at rest. Data in motion usually refers to the origin of the data. Authentication is a way to assure data integrity of this kind. Data at rest means the content of the information itself. When data are stored on a computer, they need to be protected and non-altered.

Here is the example that follows the earlier example described in Section 1.6.1. Suppose that a client named Alice wants to order ten equipments from GEE, but an attacker wants to alter her order to zero equipment. If the attacker succeeds then the client may eventually get frustrated with GEE and might decide to go to a competitor.

1.6.3 Availability

Availability refers to the ability to use the information or resources desired. For enterprise applications running on a server, availability is critical to the success of the business. Attempts to block availability, called denial of service (DoS) attacks, can be difficult to detect. A distributed DoS (DDoS) attack may be even harder to identify.

When designing a secured computer system, there is a tradeoff between availability and confidentiality. We understand that both are important but for a server, availability cannot be sacrificed. While for a personal computer, we may consider sacrificing the availability a little bit to enhance the confidentiality and integrity. For example many people would agree to sacrifice a millisecond (1/1,000 second) to ensure their data will never be stolen.

1.6.4 Threats

A threat is a potential violation of security. The actions that might occur are called attacks. Those who execute such actions are called attackers or hackers. Here are some common threats:

- Snooping: The unauthorized interception of data, such as wiretapping.
- Alteration: An unauthorized change of data, such as man-in-the-middle attack.
- Spoofing: An impersonation of an entity by another.
- Phishing: An e-mail scam that attempts to acquire sensitive information.
- Delay or denial of service: An inhibition of a service.

Computer security is to study, to detect, to prevent threats and to recover from threats. For detection, applications include intrusion detection systems (IDS) and firewalls on the network or installed on a computer. Detection is like building a dam or levee to prevent flood. It is passive and has limited protection.

"Computer architecture and security" aims to study how to design a secure computer system that can **prevent** threats and attacks. In other words, the goal of a secured computer system should be to become immune from attacks.

For recovery, most organizations have a disaster recovery plan, a part of the security policy. In the event of security being bleached, the plan is to help reduce the loss of data and shorten the mean time between failures (MTBF).

1.6.5 Firewalls

Firewalls are software or hardware implementations of a series of algorithms that detect and filter the transmission of information and data (usually divided into small packets). A request to access local resources from outside the firewall can be either granted or denied depending on the policies preset. On the other hand, a request to access local or remote resources can also be granted or denied. Correctly configured firewalls can protect computers or local area networks while still allowing legitimate packets to go through. Sometimes, it is hard to tell whether the communication is an attack string or real data. So tightly configured firewall may block legitimate traffic and is called **false positive**. Loosely configured firewalls, on the other hand, may let the attack string pass through without being detected, and is called **false negative**. This may cause severe damage to the computer system. In our daily life these principles are also very useful. Remember the DC Metro train crash in June 22, 2009. If the metrorail system had a policy to stop any trains in the event of any false alarms, the accident would not have happened.

Firewalls are used to block attacks just like levees are used to block flooded rivers. No matter how robust the levee system is, failure may still happen. Hurricane Katrina in New Orleans is a lesson in this. Similarly, firewalls can only have limited protection for computer systems. Attackers may find vulnerabilities in firewall algorithms and bypass the firewall to gain access to the computer systems. Computer security should not only rely on passive detecting techniques, it should expand its scope to work actively to prevent attacks. The invention discussed in Chapter 10 is such a preventing technique.

1.6.6 Hacking and Attacks

Hacking means finding out weaknesses in an established system and exploiting them. A computer hacker is a person who finds out weaknesses in a computer and exploits it. Hackers may be motivated by a multitude of reasons, such as profit, protest, or challenge (Hacker, 2012). The subculture that has evolved around hackers is often referred to as the computer underground but it is now an open community.

A **white hat** hacker breaks security for non-malicious reasons, for instance testing their own security system. This classification also includes individuals who perform penetration tests and vulnerability assessments within a contractual agreement. Often, this type of "white hat" hacker is called an ethical hacker.

A **black hat** hacker is a hacker who violates computer security for little reason beyond maliciousness or for personal gain. Black hat hackers are the epitome of all that the public fears in a computer criminal. They break into secure networks to destroy data or make the network unusable for those who are authorized to use it. The way black hat hackers choose the networks that they are going to break into is by a process that can be broken down into two parts: targeting and research and information gathering.

Bots are automated software tools, some freeware, available for use by any type of hacker.

An attack is to compromise a computer system. A typical approach in an attack on an Internet-connected system is:

1. *Network enumeration.* Discovering information about the intended target.
2. *Vulnerability analysis.* Identifying potential ways of attack.
3. *Exploitation.* Attempting to compromise the system by employing the vulnerabilities found through the vulnerability analysis.

In order to do so, there are several recurring tools of the trade and techniques used by computer criminals and security experts.

Here is a list of attacking techniques often used by both black hat attackers and white hat attackers:

- Vulnerability scanner: A vulnerability scanner is a tool used to quickly check computers on a network for known weaknesses. Hackers also commonly use port scanners. These check to see which ports on a specified computer are "open" or available to access the computer, and sometimes will detect what program or service is listening on that port, and its version number.

- Password cracking: Password cracking is the process of recovering passwords from data that has been stored in or transmitted by a computer system. A common approach is to repeatedly try guesses for the password.
- Packet sniffer: A packet sniffer is an application that captures data packets, which can be used to capture passwords and other data in transit over the network.
- Spoofing attack (Phishing): A spoofing attack involves one program, system, or Web site successfully masquerading as another by falsifying data and thereby being treated as a trusted system by a user or another program. The purpose of this is usually to fool programs, systems, or users into revealing confidential information, such as user names and passwords, to the attacker.
- Rootkit: A rootkit is designed to conceal the compromise of a computer's security, and can represent any set of programs which work to subvert control of an operating system from its legitimate operators. Usually, a rootkit will obscure its installation and attempt to prevent its removal through a subversion of standard system security. Rootkits may include replacements for system binaries so that it becomes impossible for the legitimate user to detect the presence of the intruder on the system by looking at process tables.
- Social engineering: When a hacker, typically a black hat, is in the second stage of the targeting process, he or she will typically use some social engineering tactics to get enough information to access the network. A common practice for hackers who use this technique is to contact the system administrator and play the role of a user who cannot get access to his or her system. Hackers who use this technique have to be quite savvy and choose the words they use carefully, in order to trick the system administrator into giving them information. In some cases only an employed help desk user will answer the phone and they are generally easy to trick. Another typical hacker approach is for the hacker to act like a very angry supervisor and when his/her authority is questioned they will threaten the help desk user with their job. Social Engineering is so effective because users are the most vulnerable part of an organization. All the security devices and programs in the world won't keep an organization safe if an employee gives away a password. Black Hat Hackers take advantage of this fact. Social Engineering can also be broken down into four sub-groups. These are intimidation, helpfulness, technical, and name-dropping.
- Trojan horses: A Trojan horse is a program which seems to be doing one thing, but is actually doing another. A trojan horse can be used to set up a back door in a computer system such that the intruder can gain access later. (The name refers to the horse from the Trojan War, with the conceptually similar function of deceiving defenders into bringing an intruder inside.)
- Viruses: A virus is a self-replicating program that spreads by inserting copies of itself into other executable code or documents. Therefore, a computer virus

behaves in a way similar to a biological virus, which spreads by inserting itself into living cells.

- Worms: Like a virus, a worm is also a self-replicating program. A worm differs from a virus in that it propagates through computer networks without user intervention. Unlike a virus, it does not need to attach itself to an existing program. Many people conflate the terms "virus" and "worm," using them both to describe any self-propagating program.
- Key loggers: A key logger is a tool designed to record ("log") every keystroke on an affected machine for later retrieval. Its purpose is usually to allow the user of this tool to gain access to confidential information typed on the affected machine, such as a user's password or other private data. Some key loggers uses virus-, trojan-, and rootkit-like methods to remain active and hidden. However, some key loggers are used in legitimate ways and sometimes to even enhance computer security. As an example, a business might have a key logger on a computer used at a point of sale and data collected by the key logger could be used for catching employee fraud.

1.7 Security Problems in Neumann Architecture

In Neumann architecture, memory, storage and I/O are connected directly to CPU. People consider the connections as a path or highway for the data to go through. On the other hand, it functions like a vehicle to carry information between the main components of a computer system. Technically we name it a bus.

A "system bus" representation of the Neumann model is shown in Figure 1.6. This is just another view of the Neumann model, with the introduction of the concept of DMA. DMA enables data exchange directly between memory and I/O that otherwise cannot be done with the initial Neumann model.

Since the 1990s, computer networks especially the Internet has been widespread around the world. Computers are no longer only being used to compute as a stand alone machine. The feature of information exchange through a network becomes a

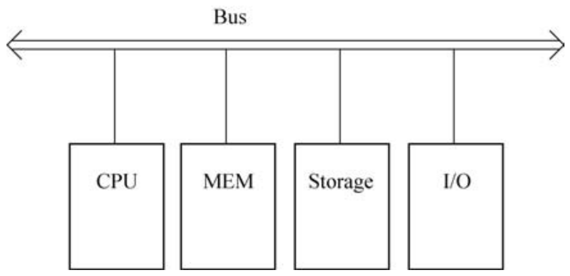


Figure 1.6 A "system bus" representation of the Neumann model. It is equivalent to Figure 1.3 with the introduction of DMA

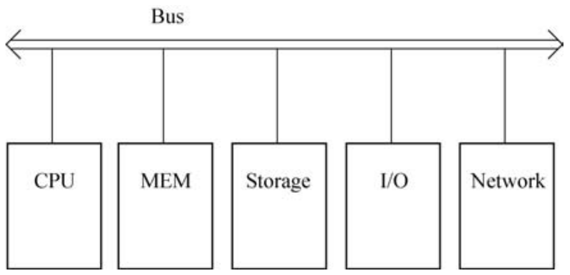


Figure 1.7 Updated Neumann computer architecture model with network interface added and separated from the general I/O devices

vital component in today's computers. Unfortunately John Neumann was not able to foresee this change.

One can argue that we can consider that a network is part of input/output devices which are already included in the Neumann model. However, the network interface is so important that it is not appropriate to classify it as a general I/O device. Furthermore, an I/O device in the Neumann model refers to those devices such as a keyboard, a display, and a printer, and so on, which are used for direct interacting with the computers. Now, the way people use a computer is considerably different to that of 60 years ago. So an update of Neumann's computer architecture model is necessary to reflect this change.

Figure 1.6 shows the modified Neumann model. In Figure 1.7, a network unit (interface) is added to the bus of a computer system. The I/O unit only deals with input and output devices such as keyboard, mouse, and display, and so on. The network interface is responsible for communicating with other computers and devices over the Internet. The separation of network unit from the general I/O devices offers great advantages.

The Neumann model is so dominant that no one has ever challenged it since its birth in 1945. However, if we look into the Neumann model from a security perspective, we could find out that it does have some potential problems.

In the Neumann model, CPUs, Memory, I/O, external storage are all connected to one single bus that includes control bus, data bus, and address bus. Once intruders break into the system from any network location, they can totally take over the computer system and do whatever they want.

For the Neumann model, the concept of CPU is a centralized control and arithmetic unit. Even though nowadays a computer with multiprocessors is very common, those processors are merely coordinated together by software to perform one task or a series of tasks. In other words, they share the same system bus. Intruders can still take over the whole system once they break into the system from any network port.

To solve this problem, numerous people have proposed different methods from securing CPU and memory to securing network interfaces by applying firewalls and intrusion detection systems. However, those solutions have not solved the information security problems completely due to the limitation of the computer architecture they used. The authors of this book have found out that there is a problem that exists in John von Neumann's computer architecture model – the foundation of modern computer architecture. If this problem is not solved, information stored on a computer will hardly be secure. As a result, a modified Neumann architecture that is capable of securing data stored on computer systems is proposed. More information about the new secured architecture will be discussed in Chapter 10.

1.8 Summary

A computer is composed of hardware, software, network, and data. People have different views about a computer system. The level of abstraction from a computer designer and a general user is different.

Computer architecture is to study how to design computer systems. *Computer Architecture and Security* will teach you how to design secured computer systems. Moreover, this book explains how to secure computer architecture as a whole so that modern computers can be built on new architecture free of data breaches.

The Neumann architecture was the first modern design for computers based on the stored program concept. It is composed of an arithmetic and logic unit (ALU), a control unit (CU), memory (M), storage (S), and input/output (I/O) devices. With the introduction of Internet, a network interface card (NIC) has become an integral part of a computer system.

Computer history reveals invention and innovation from the early days followed by rapid network and Internet development. As technologies for both computers and networks are still growing, concerns about computer and network security are also increasing.

Computer security is to study confidentiality, integrity, and availability of a computer system. The Neumann system uses single bus architecture therefore it is vulnerable to intruder attacks.

Modified Neumann architecture separates network from the other computer components. Experiments and tests have proved that it is capable of securing data stored on a computer system.

Exercises

1.1 Describe the main components of a modern computer system.

1.2 Why is a network interface different to general I/O devices?

1.3 Traditionally, a user's view of a computer system is like an onion. Explain why this concept is outdated and draw a new diagram based on the concept proposed in this book.

1.4 Computer firewalls are used to protect computer from data loss. Why firewalls can not guarantee data security?

1.5 What is a computer bus? Which types of data are carried on a computer bus?

1.6 What are the main advantages of Neumann architecture compared with those of earlier computing devices?

1.7 Draw a diagram of Neumann architecture and a single-bus diagram of Neumann architecture.

1.8 Why is assembly language called low-level language?

1.9 Virtualization sometimes is called "computers on a computer." Describe how this works?

1.10 What are the differences between memory and storage?

1.11 Discover the NIC(s) on your computer? Is it wired or wireless?

1.12 Why is Neumann architecture still considered the architecture for multiple CPU computers or multicore computers.

1.13 Provide an example for each of the following security terms: authentication, authorization, confidentially, integrity, and availability.

1.14 What is cryptography? If you want to send a secret message to your friend using a computer, what is the best way to do it?

1.15 What does the term DoS attack stand for?

1.16 List some common threats to a computer system and network.

1.17 What are the problems that exist in Neumann architecture? How can those problems be addressed?

1.18 Suppose Alice is an employee of a hospital called FX Hospital, and her job responsibility is to order medical equipment for the hospital from a company called GEE. GEE has a Web site that Alice to procure medical equipment for her hospital. Answer the following questions.

(1) Alice has to log into GEE and give that Web site her username and password through SSL protocol so that GEE knows that it is Alice. This is called _________.

Authorization, authentication, confidentiality, integrity

(2) If Alice tries to order equipment from GEE, the website will allow it, but if Bob attempts to order equipment, the order will be rejected. The website does this by conducting _________ check.

Security, accountability, authentication, authorization

(3) The SSL protocol encrypts all the communication between Alice and the GEE Web site with an algorithm such as Triple DEC to make sure of the ______ of the information.

Integrity, availability, confidentiality, security

(4) The SSL protocol uses message authentication codes in the messages that are sent between Alice and the Web site to make sure that no competitor or other malicious party can tamper with the data. This is to ensure the data's __________.

Confidentiality, integrity, accountability, non-repudiation

(5) GEE have a competitor that launches a DoS attack against the site in order that Alice will stop buying from GEE and instead come to their competing site. This affects the GEE website's _________.

Confidentiality, deny of service, integrity, availability

(6) Every time Alice places an order from the GEE website, it produces a log entry so that Alice cannot later claim to have not ordered the equipment. This is to ensure the ___________.

Integrity, authentication, authorization, accountability

(7) If the web browser and website run a ________ protocol, it is then possible for Alice to prove to a third party that she only ordered, say, 10 equipments, and not the 12 that GEE may claim she ordered.

TCP/IP, non-repudiation, repudiation, multicast

References

Computer history (2012) Computer history museum. Retrieved March 8 2010 at www.computerhistory.org

Dumas, J.D. (2006) *Computer Architecture: Fundamentals and Principles of Computer Design*, Taylor & Francis.

Foster, C.C. and Iberall, T. (1985) *Computer Architecture*, 3rd edn, Wan Nostrand Reinhold Company, New York.

Hacker (computer security) (2012) Wikipedia. Retrieved February 2 2012 at http://en.wikipedia.org/wiki/Hacker_(computer_security).

Hwang, K. (1993) *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, Inc.

Randell, B. (ed.) (1982) *The Designs of Digital Computers*, Springer-Verlag.

Shiva, S.G. (2000) *Computer Design and Architecture*, 3rd edn, Marcel Dekker, Inc., New York.

2

Digital Logic Design

To design a computer, we need to not only understand the basic architecture but also know the building blocks and know what those building blocks are made of. This chapter will answer such questions. The chapter also provides some experiments for interested readers to get the hands on experience in digital logic design.

2.1 Concept of Logic Unit

A computer is composed of different components. Every hardware component is built of many logic circuits. A logic circuit is an interconnection of several primitive logic components. It is like a “black box” that has one or more inputs and one or more outputs.

Input and output signals are restricted to one of the two states: on-off, one-zero, or represented as true-false. We know that a simple representation is sufficient to express all the numbers we use in everyday life. We also understand that it is sufficient to communicate a great amount of information to a computer system. Moreover, this binary system is the foundation upon which all basic logic components for computer systems are built.

Consider the logic circuit shown in Figure 2.1. As a binary system can only have a value of 0 or 1, the four possible combinations of those two inputs are listed in Figure 2.2.

We can easily expand the pattern for three inputs. The combinations of three inputs are 8. In general, for N inputs there will be 2^N possible combinations.

Logic circuits are divided into two categories: the one where output is a function of inputs at that particular time is called a combinational circuit; a circuit with memory is called a sequential circuit. The output of a sequential circuit is a function of not only inputs but also the state of the circuit at that time. The state of the circuit is dependent on what has happened to the circuit prior to that time. In other words, the state is a function of previous inputs and states. So we say it has memory.

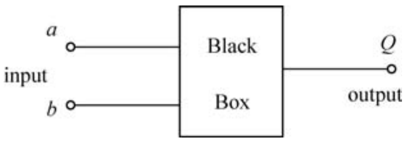


Figure 2.1 A logic circuit with two inputs and one output

For small-scale logic, designers now use prefabricated logic gates from families of devices such as the TTL 7400 series by Texas Instruments and the CMOS 4000 series by RCA, and their more recent descendants. Increasingly, these fixed-function logic gates are being replaced by programmable logic devices, which allow designers to pack a large number of mixed logic gates into a single integrated circuit. The field-programmable nature of programmable logic devices such as Field Programmable Gate Arrays (FPGA) has removed the "hard" property of hardware; it is now possible to change the logic design of a hardware system by reprogramming some of its components, thus allowing the features or function of a hardware implementation of a logic system to be changed.

2.2 Logic Functions and Truth Tables

We have looked at the input side of a black box (logic circuit) shown in Figure 2.1. We also know the output can only be 0 or 1. The question is how to design the logic circuit.

Example 2.1

Consider a black box with its inputs and output as shown in Figure 2.3, what is the logic circuit inside the box?

Figure 2.3 clearly shows that the output is set too high if and only if (iff) both inputs are high. It is an "AND" relations. Such a circuit is represented using a symbol shown in Figure 2.4. It is called an AND gate. The table (see Figure 2.3) is called a truth table (Foster, 1985).

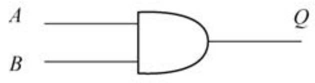
A NOT gate will reverse the input. It has only one input and one output. A NAND gate can be considered as a NOT gate with an AND gate. Figure 2.5 shows a NOT gate and Figure 2.6 shows a NAND gate.

a	*b*
0	0
0	1
1	0
1	1

Figure 2.2 Possible combinations of two binary inputs

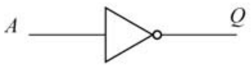
a	*b*	*Q*
0	0	0
0	1	0
1	0	0
1	1	1

Figure 2.3 An example of logic circuit inputs and output



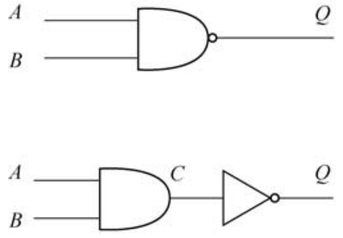
A	*B*	*Q*
0	0	0
0	1	0
1	0	0
1	1	1

Figure 2.4 An AND gate with its truth table



A	*Q*
0	1
1	0

Figure 2.5 A Not gate with its truth table



A	*B*	*C*	*Q*
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Figure 2.6 A NAND gate is equivalent to a NOT gate and an AND gate

A NAND gate functions in this way: if any of the two inputs are low, then the output is set to high. On the other hand, the output is set to low iff both inputs are high.

A truth table is a very effective way of understanding logic circuits. It is also very useful in designing logic circuit and optimizes designs. A CMOS 4001 integrated circuit (IC) contains four 2-input NAND gates. You can get it online or at stores such as RadioShack.

2.3 Boolean Algebra

The Boolean algebra is a symbolic notation to logic circuits with binary values of either true or false (Shiva, 2000). We often refer to true as 1 or high (voltage) and false as 0 or low (ground). Like algebra we learned in elementary school, Boolean algebra has a theory to govern and theorems to define. This book will not study that part. Interested readers can find it in many computer science books. Unlike algebra, Boolean algebra only studies number 1 and 0 (true and false).

Let us look at the primitives, the building blocks of logic circuits: AND, OR, and NOT. For an AND gate with two input *A, B* and one output *Q*, its algebra expression is

$$Q(A, B) = A \cdot B \tag{2.1}$$

Figure 2.7 lists some common gates and truth tables. The exclusive OR (XOR) is very useful in cryptography. It has a special property that can be used for both encryption and decryption.

Suppose we want to reverse engineer (RE) a general combinational circuit (like a black box) with M inputs and N outputs. We can apply all the 2^M possible combinations to the inputs and measure the outputs. The results can be put in a truth table for further analysis based on the Boolean algebra. Once we have the

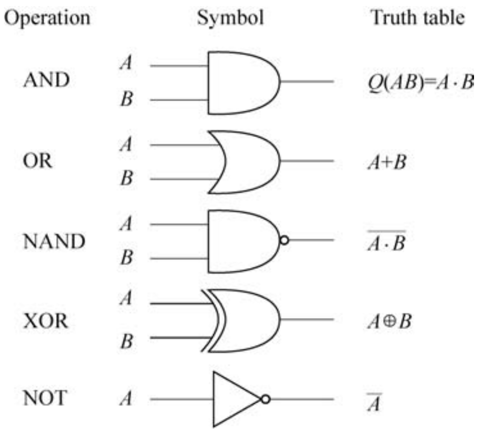


Figure 2.7 Gates and Boolean algebra expressions

function of inputs corresponding to the outputs, we can then design a logic circuit to replace the black box.

On the other hand, when we design a logic circuit, we can use Boolean algebra to calculate the outputs and put the results in a truth table. Compared with building the circuit directly and testing the truthfulness, the mathematical method saves time and money. It also avoids building the circuit again should the first one fail to operate. Such a device is very common nowadays. We call it a simulator. It can be a real device such as hardware or software running on the computer, or both.

2.4 Logic Circuit Design Process

A computer can be built with standard ICs, in many cases VLSIs. It seems that there is no need to build a circuit from the ground up. If we look at a new Sony play station or even a USB key that enables your computer or server to run certain programs, we can easily draw the conclusion that circuit design is very important. Without circuit design personnel and their skills, an IC (not to mention VLSIs) can not be manufactured.

Designing a logic circuit usually involves the following steps: feasibility study, circuit design, circuit simulation, and optimization. A well-designed circuit is then manufactured either using printed circuit board (PCB) with ICs on it or is packed into a special purpose IC.

Example 2.2

Design a home alarm system. All six windows and two doors should be monitored. In the event that a person is trying to break in, the alarm should sound.

We are now following the design steps to build the circuit. There is no doubt about the feasibility that we can build such a system. A standard CMOS IC 4068, a NOR gate with eight inputs and one output, would serve the purpose.

We can get eight electric switches, similar to light switches but much smaller in size. We then connect each switch to one input pin of the CD 4078 and wire the switches in the windows and doors. After that, we adjust the switches such that they are in “stable on” position when the windows and doors are closed. Then, we connect a buzzer and a battery.

At a normal state, all inputs to the CD4078 are high and the output is low. The buzzer does not alarm. When one or more of the windows or doors are opened while the alarm system is on, the low input(s) of the CD4078 will set the output to high. The buzzer will alarm.

Figure 2.8 provides the CD 4078 circuit and the home alarm system circuit. If we replace the mechanical switches with Hall switches, small contact-less switches rely on magnetic field, there will be no mechanical contact no matter whether the windows or doors are closed or open.

Next, the components need to be connected. This is usually done using a bread board or printed circuit board (PCB). All connections need to be checked several times making sure they are correct before connecting the power. This working process is generally for very small and new hardware designers only. More complex projects usually start with designing the circuit and then performing the simulation.

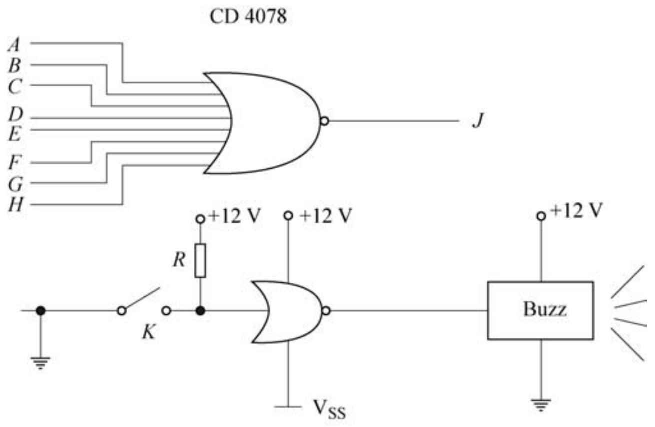


Figure 2.8 CD 4078 pin diagram, logic circuit and circuit of a simple home alarm system

Experienced hardware engineers may add other circuits to assist the main circuit such as anti-vibrating circuit and LED coupler. Mechanical switches may vibrate during the switching process. This may cause malfunction for the circuit as a string of 01010101 may occur instead of a clean 1 or 0. A LED coupler can separate the digital circuit with high voltage or current circuits to reduce the interference.

2.5 Gates and Flip-Flops

From the previous section we know that logic gates process signals which represent true (1, high) or false (0, low). Normally the positive supply voltage $+V_{cc}$ represents true and 0 V (GND) represents false. Gates are identified by their function: NOT, AND, NAND, OR, NOR, XOR. Capital letters are normally used in text to make it clear that the term refers to a logic gate.

Logic gates are available on special ICs which usually contain several gates of the same type. For example, a CD4001 IC contains four 2-input NOR gates. There are several families of logic ICs and they can be categorized into two groups: 4000 series and 74 series. The 4000 and 74HC families are the best for battery powered projects because they work with a wide range of supply voltages (usually 3–15 V) and they consume little power.

The logic circuits we have examined so far are all **combinational** logic, meaning any past states will not affect the current output. The output is only determined by the current inputs. There is another group of logic circuit where an output is determined by not only the current input states but also the previous states. We call these circuits sequential circuits. Since the past states of the circuit affect the current and future output, the sequential circuit has memory.

Flip-flops are a very common type of **sequential** circuit (Dumas, 2006). It can be constructed easily by using two NOR gates and by cross-couple connecting one

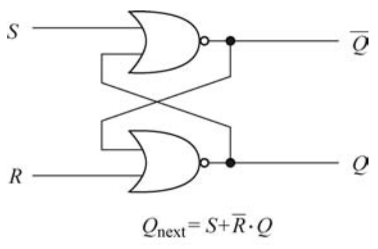


Figure 2.9 A flip-flop constructed with two NOR circuits

input of the two circuits as shown in Figure 2.9. There are two inputs names *S* (set) and *R* (reset), the output is marked *Q*.

The initial state is *S* and *R* both being low. When *S* changes from low to high and *R* remains low, then *Q* is set to high. When *S* changes back from high to low, then the output ($Q = 1$) will not be affected, which means the circuit state remains or the circuit is in memory mode. When *R* changes from low to high while *S* keeps low, then *Q* is set to low. When *R* changes back from high to low, then the output ($Q = 0$) will not be affected, which once again means the circuit is in memory mode. Figure 2.10 shows a partial truth table we call a state table (Randell, 1982) for the SR flip-flop.

In Boolean algebra expression, a SR flip-flop can be represented as:

$$Q_{\text{next}} = S + \bar{R} \cdot Q \tag{2.2}$$

Where *Q* is the current state and Q_{next} is the next state where change usually takes place by a synchronized clock. Formula (2.2) tell us that *S* can set the flip-flop to high and it will maintain the state (data) even when *S* goes off, on condition that *R* remains low. When *S* is low and the output *Q* is high, input *R* can reset the flip-flop and it will remain low even when input *R* is cleared, on condition that *S* remains low. By observing the formula (2.2), we can come to the conclusion that a flip-flop's output is determined by both inputs and the previous output. Most importantly, a flip-flop can store either a 1 or a 0 and it remembers the data unless the condition changes. The property makes it a perfect candidate to build computer memory.

Chapter 3 provides more in-depth discussion about computer memory.

S	*R*	State
0	0	Keep state (memory mode)
0	1	$Q = 0$
1	0	$Q = 1$
1	1	Unstable / race condition

Figure 2.10 State table of a SR flip-flop

2.6 Hardware Security

Generally speaking, hardware security is to protect hardware applications from being attacked, misused or reverse engineered by unauthorized users. For a computer system, hardware security involves storing passwords on a chip, restricting unauthorized remote access, and replacing vulnerable software with secure hardware and so on. It also means securing computer components such as secure memory and storage, secure CPU, secure I/O devices and secure network devices on a computer system.

It is easy to understand why we need to secure computer hardware but the reason for securing a logic circuit is not obvious. Suppose you build a new computer with standard ICs (this was how a computer used to be built), it is not difficult for other people to duplicate one by measuring the pins and making a PCB with the same circuit and plug on the standard ICs that can be purchased anywhere in the world.

TV manufacturers original equipment manufacturers (OEMs) their products in developing countries where most countries have copyright issues. How can they get the cheap labor while still protecting their intellectual property?

Previously, Japanese companies set up production lines in China and they used a counter to calculate the total TV sets the production line had manufactured. The counter was claimed unbreakable by adding a hardware security mechanism to their products. However, it is not difficult to build a TV set outside their product line or play tricks on the counters.

Since 1990s, TV companies have enhanced hardware security by putting the core technology and algorithms into an IC. They control the production by selling the special IC to the original equipment manufacturers (OEMs). Without this custom designed IC, the TV set cannot be built.

To counter reverse engineering, TV industries now use FPGA, a microprocessor with memory and interfaces, to replace the custom designed IC. One benefit of using FPGA is that hardware security is enhanced. It prevents reverse engineering and destroys itself should the number of attempts exceed the present value (for example three).

More hardware security related topics regarding computer components will be discussed in the following chapters including memory access security, buses security, I/O devices security, CPU security, architecture security, network security and so on.

2.7 FPGA and VLSI

FPGA stands for field-programmable gate array. It usually is a very large scale integrated circuit (VLSI). A modern FPGA functions like a microcomputer even though technically we call it a microprocessor. Theoretically it can be used to implement any logic circuits. So it can replace most of the logic circuits designed with standard IC like gates, flip-flops and registers and so on.

The fundamental blocks of FPGA architecture are hierarchical, integration of different building blocks such as combinational and sequential logic circuits, arithmetic

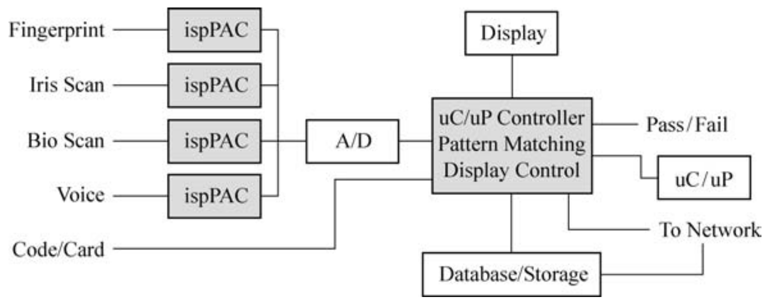


Figure 2.11 A biometric security system with Lattice FPGAs

logic unit, clocks, input/output, delay locked loop (DLL), memory (Hwang, 1993). It is also integrated with a routing feature that can be used to interconnect with other FPGAs or computers.

2.7.1 Design of an FPGA Biometric Security System

Figure 2.11 is a security system with fingerprint and other biometric inputs. The use of two series of FPGAs has made the system simpler and easy for quick development. The biometric-based security system had a configurable interface and processing power, and the ability to communicate with larger security networks.

uC: Micro-Controller
uP: Micro-Processor
A/D: Analog to Digital converter
ispPAC and uC/uP: Lattice FPGAs.

Here we can see how easily a security system can be built with the FPGAs. FPGAs can not only help us to build a security system, they also have the advantages of securing themselves. Many FPGAs have control bits that can be programmed. Once the control bits are set, it will prevent the flash and SRAM memory from being read. An optional 64-bit flash lock key is required for erasure and programming prevents accidental or unauthorized reprogramming. The one time programmable (OTP) mode blocks any further erasure or reprogramming therefore provides the ultimate assurance that the FPGA content has not been tampered with.

2.7.2 A RIFD Student Attendance System

Radio-frequency identification (RFID) is the use of a wireless non-contact system that uses radio-frequency electromagnetic fields to transfer data from a tag attached to an object, for the purposes of automatic identification and tracking. Some tags require no battery and are powered by the electromagnetic fields used to read them. Others use a local power source and emit radio waves (electromagnetic radiation at radio frequencies). The tag contains electronically stored information which can be read from up to several meters (yards) away. Unlike a bar code, the tag does not need to be within line of sight of the reader and may be embedded in the tracked object.

A basic RFID system consists of three components:

- An antenna or coil. The antenna emits radio signals to activate the tag and to read and write data to it.
- A transceiver (with decoder) and a transponder (RF tag) electronically programmed with unique information.
- The reader emits radio waves in ranges of anywhere from one inch to 100 feet or more, depending upon its power output and the radio frequency used. When an RFID tag passes through the electromagnetic zone, it detects the reader's activation signal.

The reader decodes the data encoded in the tag's integrated circuit (silicon chip) and the data is passed to the host computer for processing. Figure 2.12 shows such a system.

The purpose of an RFID system is to enable data to be transmitted by a portable device, called a tag, which is read by an RFID reader and processed according to the needs of a particular application. The data transmitted by the tag may provide identification or location information, or specifics about the product tagged, such as price, color, date of purchase, and so on. RFID technology has been used by thousands of companies for a decade or more. RFID quickly gained attention because of its ability to track moving objects. As the technology is refined, more pervasive – and invasive – uses for RFID tags are in the pipeline.

A typical RFID tag consists of a microchip attached to a radio antenna mounted on a substrate. The chip can store as much as 2 kilobytes of data.

To retrieve the data stored on an RFID tag, you need a reader. A typical reader is a device that has one or more antennas that emit radio waves and receive signals back from the tag. The reader then passes the information in digital form to a computer system.

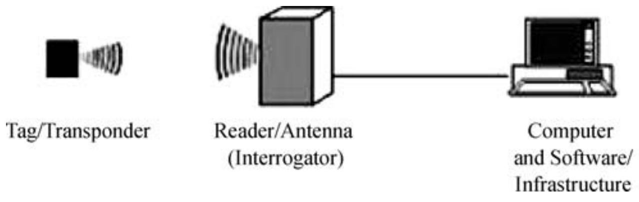

Tag/Transponder
Reader/Antenna (Interrogator)
Computer and Software/ Infrastructure

Figure 2.12 A basic RFID system

RFID tags are used in many industries. An RFID attached to an automobile during production can be used to track its progress through the assembly line. Pharmaceuticals can be tracked through warehouses. Livestock and pets may have tags injected, allowing positive identification of the animal. RFID identity cards can give employees access to locked areas of a building, and RF transponders mounted on automobiles can be used to bill motorists for access to toll roads or parking.

A radio-frequency identification system uses tags, or labels attached to the objects to be identified. Two-way radio transmitter-receivers called interrogators or readers send a signal to the tag and read its response. The readers generally transmit their observations to a computer system running RFID software or RFID middleware.

The tag's information is stored electronically in a non-volatile memory. The RFID tag includes a small RF transmitter and receiver. An RFID reader transmits an encoded radio signal to interrogate the tag. The tag receives the message and responds with its identification information. This may be only a unique tag serial number, or may be product-related information such as a stock number, lot or batch number, production date, or other specific information.

RFID tags can be either passive, active or battery assisted passive. An active tag has an on-board battery that periodically transmits its ID signal. A battery assisted passive (BAP) has a small battery on board that is activated when in the presence of a RFID reader. A **passive** tag is cheaper and smaller because it has no battery. Instead, the tag uses the radio energy transmitted by the reader as its energy source. The interrogator must be close to a RF field to be strong enough to transfer sufficient power to the tag. Since tags have individual serial numbers, the RFID system design can discriminate several tags that might be within the range of the RFID reader and read them simultaneously.

Tags may either be read-only, having a factory-assigned serial number that is used as a key into a database, or may be read/write, where object-specific data can be written into the tag by the system user. Field programmable tags may be write-once, read-multiple; "blank" tags may be written with an electronic product code by the user.

RFID tags contain at least two parts: an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency (RF) signal, collecting DC power from the incident reader signal, and other specialized functions; and an antenna for receiving and transmitting the signal. Figure 2.13 shows such an RFID tag.

Fixed readers are set up to create a specific interrogation zone which can be tightly controlled. This allows a highly defined reading area for when tags go in and out of the interrogation zone. Figure 2.14 shows such a reader. Mobile readers may be hand-held or mounted on carts or vehicles.

The student attendance system we are trying to build is a biometrics and RFID based attendance management system for schools. It provides a robust, secure and automatic attendance management system for students. The system has an inbuilt

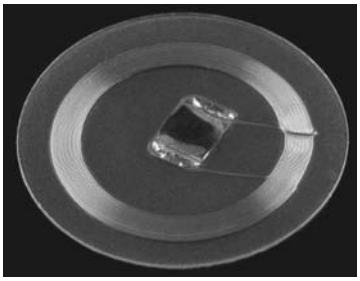


Figure 2.13 A RFID tag

facility of sending automatic SMS and e-mail alerts to the parents/guardians of the students.

The advantages of the student attendance system are:

- More efficient student attendance – system automates the student attendance hence reducing irregularities in the attendance process arising due to human error.
- Saving time – important administrative and educational resources could be freed up by utilizing the system.
- Environment friendly – reduces paper and other resource requirements. This system with its extremely small carbon footprint is a greener option.
- Better parent/guardian information system – parents/guardians are better informed about their childrens' whereabouts with automatic SMS, and e-mail facilities in the system.
- Improved parent/guardian–school relationship – the system brings the two principal stakeholders in a student's education closer, considerably improving the parent/guardian – school rapport, leading to an atmosphere of comfort and trust.

Figure 2.14 A solid-state RFID reader

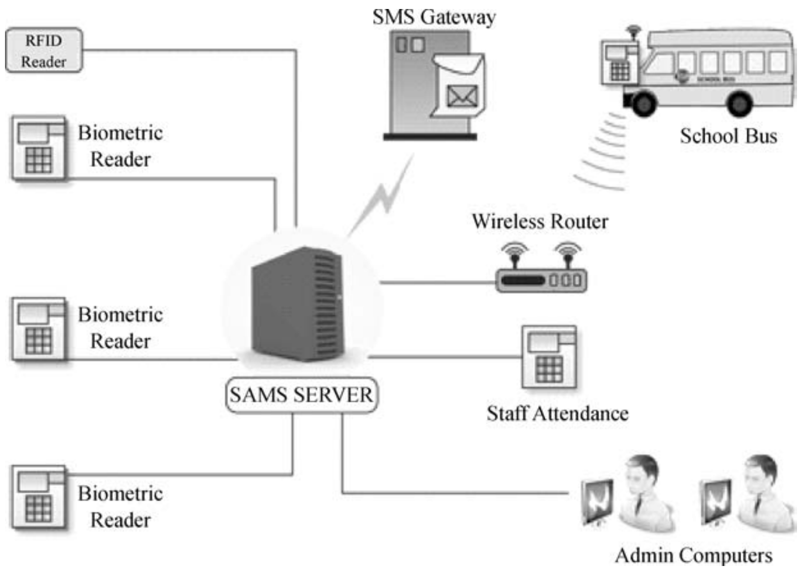


Figure 2.15 Student attendance system diagram

The system consists of the following technology elements as shown in Figure 2.15.

- Biometric attendance devices
- Biometric mobile devices
- Radio Frequency ID (RFID) Tags
- A server
- Client interface
- Management software
- Bulk SMS facility
- Automatic e-mail alerts.

In a typical system setup there is a RFID reader installed at each entrance of the school. These readers are connected to the server housed in the school. A typical RFID reader that can be connected to the computer system via USB is shown in Figure 2.16.

Each day the students will register their attendance with the RFID device. The server will download the attendance data at a pre-set time. It will process the attendance and send a set SMS message to the parents/guardians of the absentee student via SAMS SMS gateway server. The message template can be set by the school authorities to their liking.

The system also allows school authorities to send SMS alerts to parents/guardians regarding special events and emergencies.

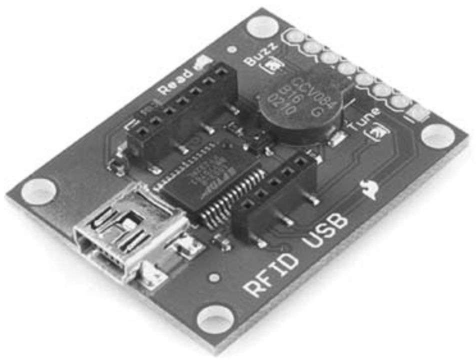


Figure 2.16 A RFID USB Reader that can be connected to a computer system via the USB interface

The system also has a module which can monitor the students who use the school bus. This data can be transferred to the schools server via WIFI or GPRS.

The system is built on robust client-server architecture and supports multiple simultaneous clients which enables admin staff to perform their function with utmost ease.

The student attendance system also has fully integrated Time Manager Software which provides various employee attendance and HR related functionalities for example, shift scheduling, attendance summary, leave management and so on. The time manager module can also be integrated with leading payroll processing software.

The features of the attendance system include:

- Accurate student attendance
- Automatic attendance collection
- Daily absentee report
- Automatic SMS alert to parent/guardian of absentee student
- Daily attendance register
- Monthly attendance register
- Yearly attendance report
- Bulk SMS facility for special events and announcements
- RFID options for young kids
- Mobile attendance data collection and reporting
- Robust employee attendance system.

Figure 2.17 shows the student attendance system working flow. When a student walks through a school entrance, the RFID reader sends a signal to the RFID tag and power the tag. The RFID tag then sends out its unique ID number (stored in built-in memory). The reader receives the data and sends the data to a computer.

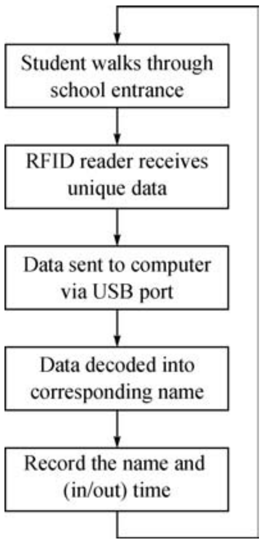

Student walks through school entrance
RFID reader receives unique data
Data sent to computer via USB port
Data decoded into corresponding name
Record the name and (in/out) time

Figure 2.17 RFID student attendance system works flow

Data received by the computer is decoded to the corresponding student name. Final information such as student name and current time are stored in the database.

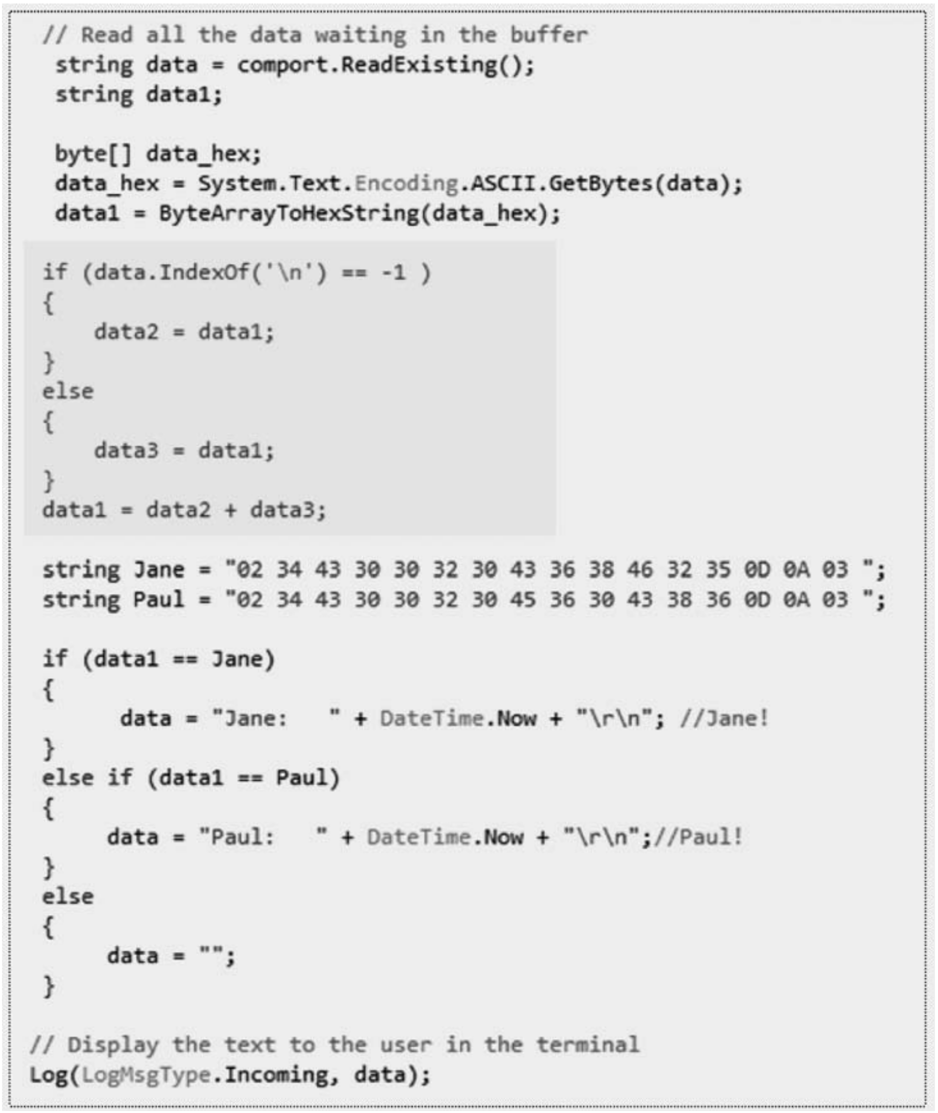
Figure 2.18 shows the algorithm in C# for a computer to receive data from a USB port. In the program, the RFID tag data is received in ASCII format and also converted into hexadecimal (HEX) format. For testing purposes, two names are defined as string. If the data match to the person, then the name and the date/time information is displayed.

There are a few lines in the beginning (shaded in light gray) to handle the data correctly. Due to the buffer limitation, each data string is received in two parts. The program is putting them together before sending them out to display. Otherwise there will be a duplicate data entry every time a student walks in.

In the system, all data are stored into a database for archiving and further analysis. When a parent wants to check the child, she/he can either call the school or check online herself/himself. The system also has a Web system that interacts with school administrators and parents. The mobile app makes it possible for parents to check their children anytime at anywhere.

2.8 Summary

Logic circuits are the building blocks for modern computers. Computers use binary number systems as it is easy to be implemented with logic gates that have two stable states: on/off, true/false or 1/0. A truth table provides all possible values of a combinational circuit's output corresponding to all inputs. A flip-flop's output is



```
// Read all the data waiting in the buffer
 string data = comport.ReadExisting();
 string data1;

 byte[] data_hex;
 data_hex = System.Text.Encoding.ASCII.GetBytes(data);
 data1 = ByteArrayToHexString(data_hex);

if (data.IndexOf('\n') == -1 )
{
    data2 = data1;
}
else
{
    data3 = data1;
}
data1 = data2 + data3;

string Jane = "02 34 43 30 30 32 30 43 36 38 46 32 35 0D 0A 03 ";
string Paul = "02 34 43 30 30 32 30 45 36 30 43 38 36 0D 0A 03 ";

if (data1 == Jane)
{
      data = "Jane:   " + DateTime.Now + "\r\n"; //Jane!
}
else if (data1 == Paul)
{
     data = "Paul:   " + DateTime.Now + "\r\n";//Paul!
}
else
{
     data = "";
}

// Display the text to the user in the terminal
Log(LogMsgType.Incoming, data);
```

Figure 2.18 Data receiving algorithm

determined by not only the inputs of that moment but also the previous states of the circuit. So it can remember, it is the prototype of one-bit memory.

Computer system security includes three aspects: hardware, software, and management. Circuit security enables designers to prevent the circuit being reverse engineered. It has many applications in our everyday life. FPGAs can be implemented to replace a complex digital circuit. In addition, it has built-in security features and processing powers.

RFID tags have the advantage of using power provided by the readers wirelessly. They are commonly used in daily life such as EZPass, department stores, and animal monitoring. The student attendance system discussed here is another example.

Exercises

2.1 A D flip-flop assumes the state of the D input. $Q(t+1)=1$ if $D(t)=1$ and $Q(t+1)=0$ if $D(t)=0$. Using a RS flip-flop construct a D flip-flop.

2.2 Given the characteristic table shown in Figure 2.10 determine which type of flip-flop it is?

2.3 Provide the truth table for the NAND gate.

2.4 A CD 4001 IC contains four 2-input NAND. How to use it to make two AND gates?

2.5 Build an experimental home alarm system with standard IC that monitors 10 windows and two doors. Draw the logic circuit.

2.6 Logic circuits built with standard ICs are easy to duplicate. How can people secure the circuits such that they cannot be reverse engineered?

2.7 If we want to monitor temperature inside and outside the home, what kind of circuit do we need?

2.8 What are the purposes of using a flash lock key and one time programmable mode in an FPGA?

2.9 Provide the truth table for an XOR gate.

2.10 Some RFID tags contain no battery or any other power connector. How do the tags send data out to the readers?

References

Dumas, J. D. (2006) *Computer Architecture: Fundamentals and Principles of Computer Design*, Taylor & Francis.

Foster, C. C. and Iberall, T. (1985) *Computer Architecture*, 3rd edn, Wan Nostrand Reinhold Company, New York.

Hwang, K. (1993) *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, Inc.

Randell, B. (ed.) (1982) *The Designs of Digital Computers*, Springer-Verlag.

Shiva, S.G. (2000) *Computer Design and Architecture*, 3rd edn, Marcel Dekker, Inc., New York.

3

Computer Memory and Storage

In this chapter, we will progress from how to design a one bit memory to how a large memory module is organized and connected through the memory interface. We will also study different memories including external memory and what are the main advantages for each of them. This chapter ends with some security issues about memory and how to secure data stored in memory.

Memory is the key component in computer systems. It determines greatly the speed of a computer system. Two of the most important parameters of memory are the speed and the cost. The "access time" of computer memory includes memory read time and memory write time. In order to make memory fast, we usually use the fastest memory circuits and add a considerable number of circuits to the main memory. This makes the memory expensive. To reduce the memory cost, data and programs not immediately needed by the system are normally stored in less expensive secondary memory such as hard drives. Secondary memory is also called storage. Data stored in the storage are brought into the main memory as the CPU needs them.

3.1 A One Bit Memory Circuit

People use flash memory to store photos and documents. However fewer people have asked how data on the flash memory are stored, what a single memory unit looks like and why it can hold data without need for a battery?

First let us look at the computer memory. A memory is a device that data can be written in and read out. Moreover, data need to be kept (stored) even after the writing/reading action is completed.

Figure 3.1 shows one of Edison's great inventions, a light bulb. When we turn on the switch K, the light is on and will stay on. When we turn off the switch, the light is off and will stay off until it is turned on again next time. Now we see that the bulb and switch device can remember.

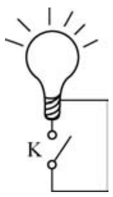


Figure 3.1 A light bulb and a switch

We would not use this device to build a computer since every action (write) needs human interaction. A transistor can be used as an electronic switch. Figure 3.2 shows a circuit that uses a transistor as an electronic switch. When the input is low (zero volts), there will be no electric current from b to e. The "switch" c to e is shut off (disconnected) and the output is high (+5 V). When a high (+5 V volts) voltage is applied to the input, there will be current from b to e. The switch is on (c and e are connected) therefore output is low (0 V).

The traditional transistor-based electronic switch uses electric current to drive the component. So it is called current-based component. Current-based component consumes a great deal of power, has heat problems and therefore cannot be made on a large scale. The next step is to use metal–oxide–semiconductor (MOS), later replaced by complementary metal–oxide–semiconductor (CMOS) component. Figure 3.3 shows a simple MOS switch.

Among other advantages, a CMOS component theoretically consumes no current. It is a voltage-based component. With CMOS technology, a large number of circuits can be integrated into one circuit thus making large scale integrated circuit (LSI) and very large scale integrated circuit (VLSI) possible. Now most processors are manufactured with VLSI technology.

Figure 3.4 shows an individual memory cell. It is one bit memory corresponding to the light bulb and switch. It is a circuit that connects two NOR gates together. This circuit is called a flip-flop or a *RS* trigger. The set (*S*) and reset (*R*) are inputs, the *Q* is the output and the $\bar{Q}$ is the complement output.

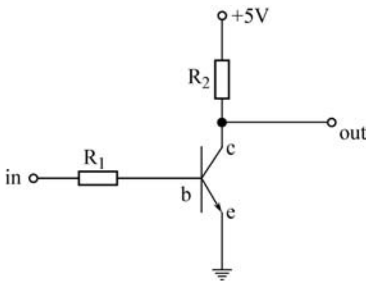


Figure 3.2 An electronic switch made of a transistor and resistors

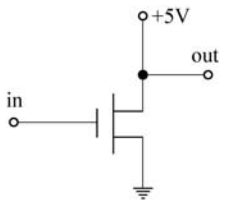


Figure 3.3 An electronic switch made of MOS component

Suppose that *S* is 1 and *R* is 0, since *S* is 1 then $\bar{Q}$ must be 0. *R* and $\bar{Q}$ makes the *Q* equals to 1. If *S* now goes to 1 so that both *S* and *R* are 0, the circuit remains in the state described above.

If *R* now becomes a 1, that will force *Q* to 0, but now both input, *S* and *Q* to the upper NOR gate are zero so that $\bar{Q}$ becomes a 1. This state will also persist.

Here we see this flip-flop "remembers" each of these two input signals, *R* and *S*. It functions like the light bulb and switch circuit but all its components are electronic including the control signal. This is the brick of a computer memory system. Computer memory composed of billions of this type of memory unit with other circuits signaling with the processor.

3.2 Register, MAR, MDR and Main Memory

There is a special type of memory that is used to hold a binary value temporarily for basic calculations and storage. It is wired within the CPU to perform specific role. We call this special memory registers (Dumas, 2006). Unlike memory, each register services a particular purpose based on the way it is wired. In terms of speed, register is the fastest memory possible in a computer system working directly with the CPU.

Registers are used in many different ways in a computer. One can be used to hold the address of current instruction being executed. It is called program counter register (PC). The instruction register (IR) holds the actual instruction being executed currently by the computer. The status register holds several 1-bit registers and each one keeps track of special conditions of the CPU. The flags contain information such as arithmetic carry, negative sign, overflow and other critical information that is monitored by the control unit.

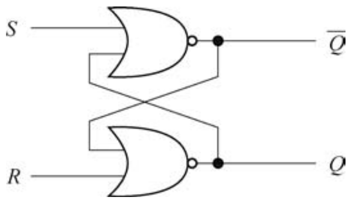


Figure 3.4 A 1-bit memory cell constructed by a flip-flop

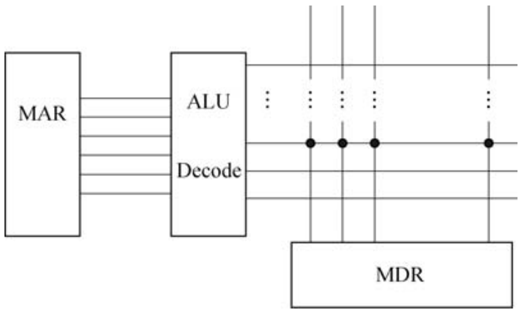


Figure 3.5 The relationship between the MAR, MDR, and memory

Memory, or random access memory (RAM) can be individually accessed in a random manner. A computer memory system usually contains billions of single 1-bit memory units. In order to access those memory units, a processor uses two registers: memory address register (MAR) and memory data register (MDR) (Foster and Iberall, 1985). Sometimes a memory data register is referred to as a memory buffer register as it holds the data that is being stored or retrieved from the memory location currently addressed by the memory address register.

Figure 3.5 shows the relationship between the MAR, MDR and memory. An address decoder is used to reduce the number of addresses required for the large number of memory units. For example, a 32-bit address bus can address $2^{32} = 4G$ memory.

In MAR, the highest bit called most significant bit (MSB) and the lowest bit called the least significant bit (LSB) (Randell, 1982). In MDR, the number of bits that can be accessed at the same time is determined by the width of the register that is connected to the processor. For a 32-bit data bus, one instruction can process 32-bit data. While 64-bit data can be processed within one instruction for a 64-bit data bus. So generally speaking a 64-bit computer is faster.

The interaction between the CPU and the memory registers takes place as follows: to store data at a particular memory location, the CPU loads the address to the memory address register and data to the memory data register. The address is interpreted in the decoder such that only one output is activated according to the address in MAR. The CPU then sends a signal to the memory enabling the write line. Data is then transferred from MDR to the memory being selected.

To read data from a particular location, the CPU sends the address to MAR. The address is decoded and that address is selected. CPU then issues a read command, data is then read from selected memory location to MDR.

Here we see that the MDR is a two-way register with both in and out while MAR is a one-way register that is always sending the signal out.

We know that a 32-bit computer can have as much as 4 gigabytes memory. In general, for a computer with k bits address in width, the total number of memory addresses is

$$M = 2^k$$

M here is also the total possible memory in a computer system. For an 8-bit processor such as Intel 8080 or Z80, the total possible memory is $2^8 = 256$. A 16-bit computer may have $2^{16} = 65{,}536$ (64 K) memory. Again a 32-bit computer can have

$$2^{32} = 2^2 \times 2^{10} \times 2^{10} \times 2^{10}$$

unique memory address, which is roughly

$$4 \times 10^3 \times 10^3 \times 10^3 = 4 \times 10^9$$

As one gigabyte (G) approximately equals to 10^9 bytes, the total memory amounts to 4G for a 32-bit computer.

Modern computers usually use random access memory (RAM) as the main memory. Dynamic RAM (DRAM) is most often seen on personal computers as it is cheaper and can be highly integrated due to its lower power consumption. DRAM need to be refreshed periodically to avoid data loss. Static RAM (SRAM) is faster than DRAM but with less integration rate. It is also more expensive so it is commonly seen on servers or special, fast computers.

3.3 Cache Memory

We know registers are a special type of memory that offer the fastest speed but with very limited numbers. On the other hand, memory (RAM) is much cheaper compared with registers and can be integrated in large quantities with easy access. But its speed is slower. To fill the gap, there is another type of memory called cache memory. A memory hierarchy of computer systems is shown in Figure 3.6.

The cache is a small amount of fast memory that sits between the processor and memory to bridge the speed gap between the CPU and main memory (Hwang, 1993). Cache is much smaller than the main memory. The working mechanism for the cache memory is to prefetch the data from the main memory and make them handy when the processor needs them. If the prediction is accurate then the processor can get the data directly from the fast cache memory without requiring the main memory to be accessed.

It is not surprising people would ask why cache memory works and how to predict the data needed before executing the program. Let us look at the "block" idea. Suppose we want to add two matrices with M row and N column together, we need to add $M \times N$ times. If the data are all in cache, we call it a read **hit**, then it would save

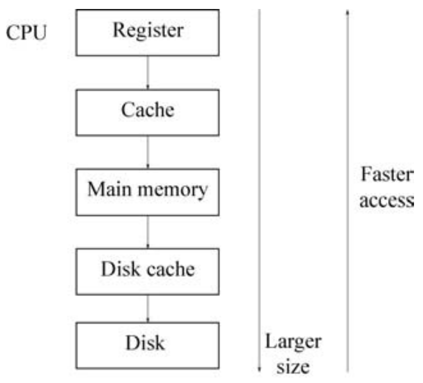


Figure 3.6 A memory hierarchy of computer systems

more time getting data with the cache than sending the address to the address buffer (MAR) and waiting for data from the data buffers (MDR) with every add operation.

If the data that the processor is requesting are not in the cache, we call a read **miss**, then the address and data buffers are enabled and data are read directly from the main memory. The predicting algorithm will move more adjacent locations into cache foreseeing that there might be an immediate need for those data as well. Statistics shows that the block data moving from main memory to cache whenever there is a read miss reduces the overall access time therefore improves the machine speed.

With the “block” concept, now we see the data transfer between cache memory and main memory is block based, while data transfer between cache memory and register is word based depending on the actual length of the instruction and data.

Some processors have two level caches, primary and secondary. A few newer processors have three level caches labeled L1, L2, and L3. The processor first attempts to access L1 cache for the requested data, if a miss occurs, then it goes to level 2 cache. If an L2 cache miss occurs, then the data must be retrieved from main memory. The selected block from the main memory is also written into both caches. A two-level cache system is shown in Figure 3.7.

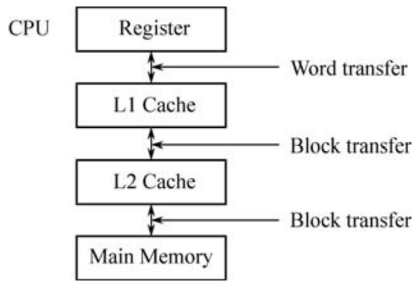


Figure 3.7 Two-level cache memory and data transfer

Cache memory improves the performance of computer systems. By using blocks and predicting and moving those blocks from main memory to cache memory, the memory access time is greatly improved.

3.4 Virtual Memory

We understand the importance of the amount of main memory corresponding to the computer speed. It is very common for many programs running simultaneously. That is why we add a large amount of memory. No matter how large the main memory is, it may still not satisfy the amount of memory needed for the running programs. Virtual memory is to give programs a larger memory space than its physical memory. It occupies part of the much slower disk storage. Like cache memory, virtual memory is a transition between main memory and the disk storage. Unlike cache memory, virtual memory does not exist on its own. It directly uses part of the disk storage. Figure 3.8 shows the concept of virtual memory in computer memory systems.

The key here is how to map a much larger virtual memory address space to the physical memory address place? To implement virtual memory, the virtual memory address is divided into segments or pages. The operating system translates a virtual page number to a physical page number (Shiva, 2000).

For example, to map a 32-bit virtual address space to a 24-bit physical address space, we can reserve the 12 least significant bits (LSB) for one page. So the page size is $2^{12} = 4$ KB. The remaining bits can be used as a pointer (page number) pointing to the page. In this case we need to map 2^{20} virtual pages to 2^{12} physical pages.

If the main memory is full, then the system will replace the current page with another one based on the mapping function in the operating system. The operating system is responsible for managing translation of virtual memory address to physical memory address. If a virtual page is to be replaced, the system always replaces the oldest one with the current one. Such a mechanism is called a first-in-first-out (FIFO) policy.

Virtual memory shares similarities with cache memory for increasing the accessing speed. In addition, it virtually increases the main memory so that programs are able to run on a computer concurrently without worrying about memory limitations.

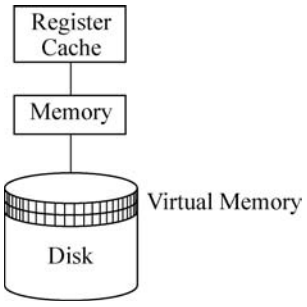


Figure 3.8 Computer memory, virtual memory, and disk storage

*3.4.1 Paged Virtual Memory**

Nearly all implementations of virtual memory divide a virtual address space into pages, blocks of contiguous virtual memory addresses. Pages are usually at least 4 kilobytes in size; systems with large virtual address ranges or amounts of real memory generally use larger page sizes.

Page tables are used to translate the virtual addresses seen by the application into physical addresses used by the hardware to process instructions; such hardware that handles this specific translation is often known as the memory management unit. Each entry in the page table holds a flag indicating whether the corresponding page is in real memory or not. If it is in real memory, the page table entry will contain the real memory address at which the page is stored. When a reference is made to a page by the hardware, if the page table entry for the page indicates that it is not currently in real memory, the hardware raises a page fault exception, invoking the paging supervisor component of the operating system.

Systems can have one page table for the whole system, separate page tables for each application and segment, a tree of page tables for large segments or some combination of these. If there is only one page table, different applications running at the same time use different parts of a single range of virtual addresses. If there are multiple page or segment tables, there are multiple virtual address spaces and concurrent applications with separate page tables redirected to different real addresses.

Paging supervisor is a part of the operating system which creates and manages page tables. If the hardware raises a page fault exception, the paging supervisor accesses secondary storage, returns the page that has the virtual address that resulted in the page fault, updates the page tables to reflect the physical location of the virtual address and tells the translation mechanism to restart the request.

When all physical memory is already in use, the paging supervisor must free a page in primary storage to hold the swapped-in page. The supervisor uses one of a variety of page replacement algorithms such as the one least recently used to determine which page to free.

*3.4.2 Segmented Virtual Memory**

Some systems use segmentation instead of paging, dividing virtual address spaces into variable-length segments. A virtual address here consists of a segment number and an offset within the segment. The Intel 80286 supports a similar segmentation scheme as an option, but it is rarely used. Segmentation and paging can be used together by dividing each segment into pages; systems with this memory structure, such as Multics and IBM System/38, are usually paging-predominant, segmentation providing memory protection.

In the Intel 80386 and later IA-32 processors, the segments reside in a 32-bit linear, paged address space. Segments can be moved in and out of that space; pages there can "page" in and out of main memory, providing two levels of virtual memory; few if any operating systems do so, instead using only paging. Early non-hardware-assisted x86 virtualization solutions combined paging and segmentation because x86 paging offers only two protection domains whereas a VMM/guest OS/guest applications stack needs three. The difference between paging and segmentation systems is not only about memory division; segmentation is visible to user processes, as part of memory model semantics. Hence, instead of memory that looks like a single large vector, it is structured into multiple spaces.

This difference has important consequences; a segment is not a page with variable length or a simple way to lengthen the address space. Segmentation that can provide a single-level memory model in which there is no differentiation between process memory and file system consists of only a list of segments (files) mapped into the process's potential address space.

This is not the same as the mechanisms provided by calls such as mmap and Win32's MapViewOfFile, because inter-file pointers do not work when mapping files into semi-arbitrary places. In Multics, a file (or a segment from a multi-segment file) is mapped into a segment in the address space, so files are always mapped at a segment boundary. A file's linkage section can contain pointers for which an attempt to load the pointer into a register or make an indirect reference through it causes a trap. The unresolved pointer contains an indication of the name of the segment to which the pointer refers and an offset within the segment; the handler for the trap maps the segment into the address space, puts the segment number into the pointer, changes the tag field in the pointer so that it no longer causes a trap, and returns to the code where the trap occurred, re-executing the instruction that caused the trap. This eliminates the need for a linker completely and works when different processes map the same file into different places in their private address spaces.

3.5 Non-Volatile Memory

Most computers use random access memory (RAM) as the main memory. RAM is a type of read-write memory that has the fast access speed and can be easily integrated on a large scale. On the other hand, RAM is volatile which means it will lose data once power is removed.

Non-volatile memory keeps data even when power is removed. Flash memory and erasable electrically programmable ROM (EEPROM) are non-volatile memories.

The author of this book, an engineer at a research institution, invented a flash memory card in 1985, three years earlier than the first commercial flash chip was manufactured by Intel Corporation in 1988 (Non-volatile memory, 1988). The flash memory card was integrated on a standard bus (STD bus) board with digital circuits designed by the author. The technology was tested at a well-known computer corporation and applied to the anti-skip/anti-slide system (now called ABS) for rail trains, a key national science and technology project from 1985 to 1989. A photo of the non-volatile memory card is shown in Figure 3.9.

Figure 3.9 A non-volatile memory card invented in 1985, three years earlier than the first commercial flash chip was manufactured

Read-only memory (ROM) is non-volatile but it cannot be re-written. Programmable ROM (PROM) can only be written once with a special device. Erasable PROM (EPROM) has a window on the chip. Data can be erased by ultraviolet light lasting 20–30 minutes. So it is not considered to be random read/write. Electrically erasable PROM (EEPROM or E^{2}PROM) is the father of modern flash memory. In earlier EEPROMs, data can be electrically erased but after erasing, the entire chip is set to blank.

The USB flash drive and memory cards for digital cameras are a new type of non-volatile memory based on EEPROM technology with the speed between main memory and disk drives.

3.6 External Memory

External memory, or secondary memory, is high capacity data storage. The external memory is usually much larger than main memory. The most commonly seen external memory is computer hard drives. USB flash drives, backup drives, and network storage are all external memory. In many cases, we refer to external memory as storage.

Some external memory such as hard drives is connected directly to CPU via the system bus. Some others such as USB flash drives are connected via serial interface. It takes time to put together the address and data that is required to communicate with CPU. So it is usually slower than direct connection.

With the technology advances, SSD are emerging in the market. SSD has the advantages of being more reliable due to not containing any mechanical parts, and is faster and more compact. It is projected that in a not too distant future, the price of SSD will break even with the hard disk drive (HDD).

Cloud computing brings another storage model. That is storage in the cloud. As network has become one of the four key components (hardware, software, network, and data) in a computer system, storing data across the network is very common. Cloud storage adds another dimension to the computer memory system and has become more and more popular.

3.6.1 Hard Disk Drives

In modern computers, hard disk drives are usually used as secondary storage. The time taken to access a given byte of information stored on a hard disk is typically a few thousandths of a second, or milliseconds. By contrast, the time taken to access a given byte of information stored in random access memory is measured in billionths of a second, or nanoseconds. This illustrates the significant access-time difference which distinguishes solid-state memory from rotating magnetic storage devices: hard disks are typically about a million times slower than memory. Rotating optical storage devices, such as CD and DVD drives, have even longer access times. With disk drives, once the disk read/write head reaches the proper placement and the data of interest rotates under it, subsequent data on the track are very fast to access. As a result, in order to hide the initial seek time and rotational latency, data are transferred to and from disks in large contiguous blocks.

When data reside on disk, block access to hide latency offers a ray of hope in designing efficient external memory algorithms. Sequential or block access on disks is orders of magnitude faster than random access, and many sophisticated paradigms have been developed to design efficient algorithms based upon sequential and block access. Another way to reduce the I/O bottleneck is to use multiple disks in parallel in order to increase the bandwidth between primary and secondary memory.

Some other examples of secondary storage technologies are: flash memory (e.g., USB flash drives or keys), floppy disks, magnetic tape, paper tape, punched cards, standalone RAM disks, and Iomega Zip drives.

The secondary storage is often formatted according to a file system format, which provides the abstraction necessary to organize data into files and directories, providing additional information (called metadata) describing the owner of a certain file, the access time, the access permissions, and other information.

Most computer operating systems use the concept of virtual memory, allowing utilization of more primary storage capacity than is physically available in the system. As the primary memory fills up, the system moves the least-used chunks (pages) to secondary storage devices (to a swap file or page file), retrieving them later when they are needed. As more of these retrievals from slower secondary storage are necessary, the more the overall system performance is degraded.

*3.6.2 Tertiary Storage and Off-Line Storage**

Tertiary storage or tertiary memory provides a third level of storage. Typically it involves a robotic mechanism which will mount (insert) and dismount removable mass storage media into a storage device according to the system's demands; these data are often copied to secondary storage before use. It is primarily used for archiving rarely accessed information since it is much slower than secondary storage (e.g., 5–60 seconds vs. 1–10 milliseconds). This is primarily useful for extraordinarily large data stores, accessed without human operators. Typical examples include tape libraries and optical jukeboxes.

When a computer needs to read information from the tertiary storage, it will first consult a catalog database to determine which tape or disc contains the information. Next, the computer will instruct a robotic arm to fetch the medium and place it in a drive. When the computer has finished reading the information, the robotic arm will return the medium to its place in the library.

Off-line storage is a computer data storage on a medium or a device that is not under the control of a processing unit. The medium is recorded, usually in a secondary or tertiary storage device, and then physically removed or disconnected. It must be inserted or connected by a human operator before a computer can access it again. Unlike tertiary storage, it cannot be accessed without human interaction.

Off-line storage is used to transfer information, since the detached medium can be easily physically transported. Additionally, in case a disaster, for example a fire, destroys the original data, a medium in a remote location will probably be unaffected, enabling disaster recovery. Off-line storage increases general information security, since it is physically inaccessible from a computer, and data confidentiality or integrity cannot be affected by computer-based attack techniques. Also, if the information stored for archival purposes is rarely accessed, off-line storage is less expensive than tertiary storage.

In modern personal computers, most secondary and tertiary storage media are also used for off-line storage. Optical discs and flash memory devices are most popular, and to a much lesser extent removable hard disk drives. In enterprise uses, magnetic tape is predominant. Older examples are floppy disks, Zip disks, or punched cards.

3.6.3 Serial Advanced Technology Attachment (SATA)

Serial ATA (SATA or Serial Advanced Technology Attachment) is a computer bus interface for connecting host bus adapters to mass storage devices such as hard disk drives and optical drives. Serial ATA was designed to replace the older parallel ATA (PATA) standard (often called by the old name IDE), offering several advantages over the older interface: reduced cable size and cost (7 conductors instead of 40), native hot swapping, faster data transfer through higher signaling rates, and more efficient transfer through an (optional) I/O queuing protocol.

SATA host-adapters and devices communicate via a high-speed serial cable over two pairs of conductors. In contrast, parallel ATA (the redesign for the legacy ATA specifications) used a 16-bit wide data bus with many additional support and control signals, all operating at much lower frequency. To ensure backward compatibility with legacy ATA software and applications, SATA uses the same basic ATA and ATAPI command-set as legacy ATA devices.

As of 2009, SATA has replaced parallel ATA in most shipping consumer desktop and laptop computers, and is expected to eventually replace PATA in embedded applications where space and cost are important factors. SATAs market share in the desktop PC market was 99% in 2008. PATA remains widely used in industrial and embedded applications that use CompactFlash storage, though even there, the next CFast storage standard will be based on SATA.

Serial ATA industry compatibility specifications originate from the Serial ATA International Organization (aka. SATA-IO, serialata.org). The SATA-IO group collaboratively creates, reviews, ratifies, and publishes the interoperability specifications, the test cases, and plug-fests. As with many other industry compatibility standards, the SATA content ownership is transferred to other industry bodies: primarily the INCITS T13subcommittee ATA, the INCITS T10 subcommittee (SCSI); a subgroup of T10 responsible for Serial Attached SCSI (SAS). The complete specification can be sourced from SATA-IO. The remainder of this article will try to use the terminology and specifications of SATA-IO.

3.6.4 Small Computer System Interface (SCSI)

SCSI is a set of standards for physically connecting and transferring data between computers and peripheral devices. The SCSI standards define commands, protocols, and electrical and optical interfaces. SCSI is most commonly used for hard disks and tape drives, but it can connect a wide range of other devices, including scanners and CD drives, although not all controllers can handle all devices. The SCSI standard defines command sets for specific peripheral device types; the presence of "unknown" as one of these types means that in theory it can be used as an interface to almost any device, but the standard is highly pragmatic and addressed toward commercial requirements.

SCSI is an intelligent, peripheral, buffered, peer to peer interface. It hides the complexity of physical format. Every device attaches to the SCSI bus in a similar manner. Up to 8 or 16 devices can be attached to a single bus. There can be any number of hosts and peripheral devices but there should be at least one host. SCSI uses handshake signals between devices, SCSI-1, SCSI-2 have the option of parity error checking. Starting with SCSI-U160 (part of SCSI-3) all commands and data are error checked by a CRC32 checksum. The SCSI protocol defines communication from host to host, host to a peripheral device, peripheral device to a peripheral device. However, most peripheral devices are exclusively SCSI targets, incapable of acting as SCSI initiators – unable to initiate SCSI transactions themselves. Therefore peripheral-to-peripheral communications are uncommon, but possible in most SCSI applications. The Symbios Logic 53C810 chip is an example of a PCI host interface that can act as a SCSI target.

SCSI is available in a variety of interfaces. The first, still very common, was parallel SCSI (now also called SPI), which uses a parallel electrical bus design. As of 2008, SPI is being replaced by Serial Attached SCSI (SAS), which uses a serial design but retains other aspects of the technology. Many other interfaces which do not rely on complete SCSI standards still implement the SCSI command protocol; others (such as iSCSI) drop physical implementation entirely while retaining the SCSI architectural model. iSCSI, for example, uses TCP/IP as a transport mechanism.

SCSI interfaces have often been included on computers from various manufacturers for use under Microsoft Windows, Mac OS, Unix, Commodore Amiga and Linux operating systems, either implemented on the motherboard or by the means of plug-in adaptors. With the advent of SAS and SATA drives, provision for SCSI on motherboards is being discontinued. A few companies still market SCSI interfaces for motherboards supporting PCIe and PCI-X.

3.6.5 Serial Attached SCSI (SAS)

SAS is a communication protocol used to move data to and from computer storage devices such as hard drives and tape drives. SAS is a point-to-point serial protocol that replaces the parallel SCSI bus technology that first appeared in the mid 1980s in data centers and workstations, and it uses the standard SCSI command set. SAS offers backwards-compatibility with second-generation SATA drives. SATA 3 Gbit/s drives may be connected to SAS backplanes, but SAS drives may not be connected to SATA backplanes.

The T10 technical committee of the International Committee for Information Technology Standards (INCITS) develops and maintains the SAS protocol; the SCSI Trade Association (SCSITA) promotes the technology.

A typical Serial Attached SCSI system consists of the following basic components:

- An Initiator: a device that originates device-service and task-management requests for processing by a target device and receives responses for the same requests from other target devices. Initiators may be provided as an on-board component on the motherboard (as is the case with many server-oriented motherboards) or as an add-on host bus adapter.
- A Target: a device containing logical units and target ports that receives device service and task management requests for processing and sends responses for the same requests to initiator devices. A target device could be a hard disk or a disk array system.
- A Service Delivery Subsystem: the part of an I/O system that transmits information between an initiator and a target. Typically cables connecting an initiator and target with or without expanders and backplanes constitute a service delivery subsystem.
- Expanders: devices that form part of a service delivery subsystem and facilitate communication between SAS devices. Expanders facilitate the connection of multiple SAS End devices to a single initiator port.

A SAS Domain is the SAS version of a SCSI domain – it consists of a set of SAS devices that communicate with one another by means of a service delivery subsystem. Each SAS port in a SAS domain has a SCSI port identifier that identifies the port uniquely within the SAS domain. It is assigned by the device manufacturer, like an Ethernet device's MAC address, and is typically world-wide unique as well. SAS devices use these port identifiers to address communications to each other.

In addition, every SAS device has a SCSI device name, which identifies the SAS device uniquely in the world. One doesn't often see these device names because the port identifiers tend to identify the device sufficiently.

For comparison, in parallel SCSI, the SCSI ID is the port identifier and device name. In fiber channel, the port identifier is a WWPN and the device name is a WWNN.

In SAS, both SCSI port identifiers and SCSI device names take the form of a SAS address, which is a 64-bit value, normally in the NAA IEEE Registered format. People sometimes call a SAS address a World Wide Name or WWN, because it is essentially the same thing as a WWN in fiber channel.

Below are the comparisons between SAS with parallel SCSI:

- The SAS bus operates point-to-point while the SCSI bus is multidrop. Each SAS device is connected by a dedicated link to the initiator, unless an expander is used. If one initiator is connected to one target, there is no opportunity for contention; with parallel SCSI, even this situation could cause contention.
- SAS has no termination issues and does not require terminator packs like parallel SCSI.
- SAS eliminates clock skew.
- SAS allows up to 65,535 devices through the use of expanders, while Parallel SCSI has a limit of 8 or 16 devices on a single channel.
- SAS allows a higher transfer speed (3 or 6 Gbit/s) than most parallel SCSI standards. SAS achieves these speeds on each initiator-target connection, hence getting higher throughput, whereas parallel SCSI shares the speed across the entire multidrop bus.
- SAS controllers may connect to SATA devices, either directly connected using native SATA protocol or through SAS expanders using SATA Tunneled Protocol (STP).
- Both SAS and parallel SCSI use the SCSI command-set.

*3.6.6 Network-Attached Storage (NAS)**

Network-attached storage (NAS) is file-level computer data storage connected to a computer network providing data access to heterogeneous clients. NAS not only operates as a file server, but is specialized for this task either by its hardware, software, or configuration of those elements. NAS is often made as a computer appliance – a specialized computer built from the ground up for storing and serving files – rather than simply a general purpose computer being used for the role.

As of 2010 NAS devices are gaining popularity, as a convenient method of sharing files among multiple computers. Potential benefits of network-attached storage, compared to file servers, include faster data access, easier administration, and simple configuration.

NAS systems are networked appliances which contain one or more hard drives, often arranged into logical, redundant storage containers or RAID arrays. Network-attached storage removes the responsibility of file serving from other servers on the network. They typically provide access to files using network file sharing protocols such as NFS, SMB/CIFS, or AFP.

A Network Attached Storage (NAS) unit is a computer connected to a network that only provides file-based data storage services to other devices on the network. Although it may technically be possible to run other software on a NAS unit, it is not designed to be a general purpose server. For example, NAS units usually do not have a keyboard or display, and are controlled and configured over the network, often using a browser.

A fully featured operating system is not needed on a NAS device, so often a stripped-down operating system is used. For example, FreeNAS, an open source NAS solution designed for commodity PC hardware, is implemented as a stripped-down version of FreeBSD.

NAS systems contain one or more hard disks, often arranged into logical, redundant storage containers or RAID arrays.

NAS uses file-based protocols such as NFS (popular on UNIX systems), SMB/CIFS (Server Message Block/Common Internet File System) (used with MS Windows systems), or AFP (used with Apple Macintosh computers). NAS units rarely limit clients to a single protocol.

3.6.7 Storage Area Network (SAN)[*]

A storage area network (SAN) is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system. A SAN typically has its own network of storage devices that are generally not accessible through the local area network by other devices. The cost and complexity of SANs dropped in the early 2000s to levels allowing wider adoption across both enterprise and small to medium sized business environments.

A SAN does not provide file abstraction, only block-level operations. However, file systems built on top of SANs do provide file-level access, and are known as SAN filesystems or shared disk file systems.

Historically, data centers first created “islands” of SCSI disk arrays as direct-attached storage (DAS), each dedicated to an application, and visible as a number of “virtual hard drives” (i.e., LUNs). Essentially, a SAN consolidates such storage islands together using a high-speed network.

Operating systems maintain their own file systems on their own dedicated, non-shared LUNs, as though they were local to themselves. If multiple systems were simply to attempt to share a LUN, these would interfere with each other and quickly corrupt the data. Any planned sharing of data on different computers within a LUN requires advanced solutions, such as SAN file systems or clustered computing.

Despite such issues, SANs help to increase storage capacity utilization, since multiple servers consolidate their private storage space onto the disk arrays.

Common uses of a SAN include provision of transactionally accessed data that require high-speed block-level access to the hard drives such as e-mail servers, databases, and high usage file servers.

NAS provides both storage and a file system. This is often contrasted with SAN (Storage Area Network), which provides only block-based storage and leaves file system concerns on the “client” side. SAN protocols are SCSI, Fiber Channel, iSCSI, ATA over Ethernet (AoE), or HyperSCSI.

One way to loosely conceptualize the difference between a NAS and a SAN is that a NAS appears to the client OS (operating system) as a file server (the client can map network drives to shares on that server) whereas a disk available through a SAN still appears to the client OS as a disk, visible in disk and volume management utilities (along with client’s local disks), and available to be formatted with a file system and mounted.

Despite their differences, SAN and NAS are not mutually exclusive, and may be combined as a SAN-NAS hybrid, offering both file-level protocols (NAS) and block-level protocols (SAN) from the same system. An example of this is Openfiler, a free software product running on Linux-based systems.

Sharing storage usually simplifies storage administration and adds flexibility since cables and storage devices do not have to be physically moved to shift storage from one server to another.

Other benefits include the ability to allow servers to boot from the SAN itself. This allows for a quick and easy replacement of faulty servers since the SAN can be reconfigured so that a replacement server can use the LUN of the faulty server. While this area of technology is still new many view it as being the future of the enterprise datacenter.

SANs also tend to enable more effective disaster recovery processes. A SAN could span a distant location containing a secondary storage array. This enables storage replication either implemented by disk array controllers, by server software, or by specialized SAN devices. Since IP WANs are often the least costly method of long-distance transport, the Fiber Channel over IP (FCIP) and iSCSI protocols have been developed to allow SAN extension over IP networks. The traditional physical

SCSI layer could only support a few meters of distance – not nearly enough to ensure business continuance in a disaster.

3.6.8 Cloud Storage

Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Hosting companies operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers.

Cloud storage services may be accessed through a Web service application programming interface (API), or through a Web-based user interface.

Cloud storage has the same characteristics as cloud computing in terms of agility, scalability, elasticity, and multi-tenancy. It is believed to have been invented by Joseph Carl Robnett Licklider in the 1960s. Since the 1960s, cloud computing has developed along a number of lines, with Web 2.0 being the most recent evolution. However, since the Internet only started to offer significant bandwidth in the 1990s, cloud computing for the masses has been something of a late developer.

One of the first milestones for cloud computing was the arrival of Salesforce.com in 1999, which pioneered the concept of delivering enterprise applications via a simple Web site. The services firm paved the way for both specialist and mainstream software firms to deliver applications over the Internet. Files Anywhere also helped pioneer cloud based storage services that also enable users to securely share files online. Both of these companies continue to offer those services today.

It is difficult to pin down a canonical definition of cloud storage architecture, but object storage is reasonably analogous. Cloud storage services like Amazon S3, cloud storage products like EMC Atmos, and distributed storage research projects like OceanStore are all examples of object storage and infer the following guidelines.

Cloud storage is:

- made up of many distributed resources, but still acts as one
- highly fault tolerant through redundancy and distribution of data
- highly durable through the creation of versioned copies
- typically eventually consistent with regard to data replicas.

The advantages of cloud storage include:

- Companies need only pay for the storage they actually use as it is also possible for companies by utilizing actual virtual storage features like thin provisioning.

- Companies do not need to install physical storage devices in their own datacenter or offices, but the fact that storage has to be placed anywhere stays the same (maybe localization costs are lower in offshore locations).
- Storage maintenance tasks, such as backup, data replication, and purchasing additional storage devices are offloaded to the responsibility of a service provider, allowing organizations to focus on their core business, but the fact stays the same that someone has to pay for the administrative effort for these tasks.
- Cloud storage provides users with immediate access to a broad range of resources and applications hosted in the infrastructure of another organization via a Web service interface.

There is a number of security issues associated with the cloud storage for using public cloud services. Three general areas that concern people the most: security and privacy, compliance, and legal or contractual issues.

3.7 Memory Access Security

Executable space protection is the making of memory regions as non-executable, such that an attempt to execute machine code in these regions will cause an exception. It makes use of hardware features such as the NX (non-executable) bit.

If an operating system can mark some or all writable regions of memory as non-executable, it may be able to prevent the stack and heap memory areas from being executed. This helps to prevent certain buffer overflow exploits from succeeding, particularly those that inject and execute code, such as the Sasser and Blaster worms. There attacks rely on some part of memory, usually the stack, being both writable and executable; if it is not, the attack fails.

As we know, dynamic radon access memory (DRAM) stores data in separate capacitors within an integrated circuit. It needs to be refreshed, say every 64ms, to avoid data loss. In reality the memory cell capacitors will often retain their values for significantly longer, particularly at low temperatures. In many cases, most of the data in DRAM can be recovered even if the DRAM has not been refreshed for several minutes.

This long-lasting capacitor property can be used by hackers to recover data kept in memory by quickly rebooting the computer and dumping the contents of the RAM or by cooling the chips and transferring them to a different computer.

USB flash drives are convenient external memory to store data or even sensitive data. The main disadvantage is the physical security. It is easily lost or stolen. To ensure the data security, people usually use hardware solutions or software solutions, or both, to protect data.

A software solution to protect data is to use encryption. There are programs that can encrypt data on a USB drive automatically and transparently. A hardware solution is to embed hardware encryption. Hardware solution may offer additional feature by overwriting the data on the drive if a wrong password is entered more than a certain number of times.

Segmentation is another technique to not only provide more virtual address but also provide protection of the data stored in the segments. Stacks and some arrays use dynamic data structure. A growing data structure might bump into the next one. If operating systems allocate the addresses linearly or in other words use single address space it may overwrite each other.

The memory being overwritten may cause the system to crash or a breach of system security. A carefully planned memory that is overwritten may let hackers take control of a computer system, stop services and steal data. It is commonly seen as buffer overflow attacks (Buffer overflow, 2012).

A buffer is an area of memory that can be used to store user input, for example. Buffers often have some fixed maximum size. If the user provides more input than it can fit into the buffer, the extra input might end up in unexpected places in memory, and the buffer is said to overflow.

An execution stack is to keep track of what function programs are running at any particular time, and what function they need to return to after finishing the current function.

Figure 3.10a shows single memory address space. As the stack grows up, it may overwrite the program. In Figure 3.10b, program, data and stack are stored in separate segments. We can make the program segment execute only and make the data segment non-executable. This will address the memory overwritten and some buffer overflow problems.

Let us look at another example as shown in Figure 3.11. The main program checks the password, if the password is correct then open ATM. Here the password is supposed to be 16 characters long.

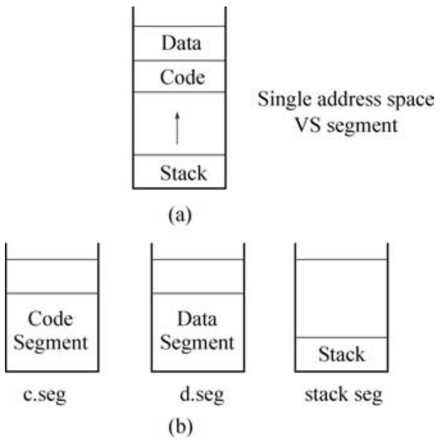


Figure 3.10 Memory segmentation and memory access security

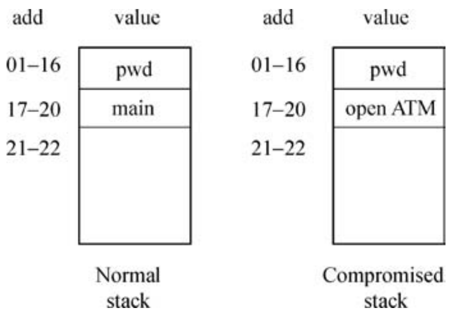


Figure 3.11 Buffer overflow attack

If the user enters a password that is more than 16 characters, the extra characters will overwrite the return address (back to the main program). A hacker can construct the 17th to 20th characters in a way that they are exactly the same as the address of open ATM subroutine. Now even the password is entered incorrect, the ATM may still be opened! The crafted 17th to 20th characters are called attack string.

It is common for corporations to use cloud services for e-mail and some non-critical applications. Storing sensitive data such as financial and marketing data in the cloud is not common. The main skeptical issue is the data security. Cloud data security is an important and ongoing topic that needs to be solved. Until then, we will not see more companies put the critical data in the cloud.

3.8 Summary

We have discussed different memories and how a basic memory unit is constructed using a flip-flop.

A computer memory system contains a vast amount of memory units. To access each memory unit, we use the matrix style method by decoding the address using MAR and store a word of data with MDR.

DRAM and SRAM are common seen memory in a computer system. They provide fast speed but are volatile.

Cache memory is faster memory than main memory but small in quantity. It provides handy data access to CPU by predicting and pre-fetching the data blocks.

Virtual memory shares some similarities with the cache memory to increase memory address space and increase the accessing speed. With the help of virtual memory, programs are able to run concurrently without worrying about the memory limitations.

Non-volatile memory includes ROM, PROM, EPRPM, and EEPROM. USB flash drives are built based on EEPROM technology.

Hard drives belong to external memory. They provide vast amounts of memory but the speed is usually slower than flash drives and much slower than main memory.

Many computer security breaches result from memory violations. Protecting memory is important to ensure computer security. Buffer overflow is a common problem in computer security. Executable space protection can protect memory from being executed by malicious programs.

Data encryption can protect unauthorized data access stored on computers and flash drives. Hardware encryption and protection can remove data in the event a hacker is trying to break the password at any given time. Public clouds provide infrastructure, platforms, and software for customers. Before moving sensitive data in a cloud, studies should be conducted to guarantee the data security.

Exercises

3.1 Design a 1-bit memory circuit and provide a truth table for the circuit.

3.2 MDR is used to decode address signals from CPU. For a 16-bit address bus, how much memory can be attached to the system directly?

3.3 RAM is volatile, why not use non-volatile memory in main memory?

3.4 We know registers are faster than cache memory and cache memory is faster than main memory. Why not replace all main memory with cache memory?

3.5 Now some computers use flash memory-based solid disk to replace hard drives. In addition to be noise-less, what are other advantages?

3.6 A 32-bit processor can address 4G memory. How do you build a 32-bit computer with more than 4G memory?

3.7 What are the similarities between cache memory and virtual memory?

3.8 What are the differences between cache memory and virtual memory?

3.9 A computer uses virtual memory to increase the memory space. How do you translate a virtual address to a physical address?

3.10 What are the benefits of using memory segmentation technique?

3.11 How can "executable space protection" enhance memory access security?

3.12 How memory be secured to prevent buffer overflow attacks?

3.13 Visit a computer store such as Microcenter or online, discover USB flash drives that provide the best security features to protect data.

References

Buffer overflow (2012) Wikipedia. Retrieved February 29, 2012 at http://en.wikipedia.org/wiki/buffer_overflow.

Dumas, J. D. (2006) *Computer Architecture: Fundamentals and Principles of Computer Design*, Taylor & Francis.

Foster, C. C. and Iberall, T. (1985) *Computer Architecture*, 3rd edn, Wan Nostrand Reinhold Company, New York.

Hwang, K. (1993) *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, Inc.

Non-volatile memory (1988) Qingday Daily. Dec. 19, 1988.

Randell, B. (ed.) (1982) *The Designs of Digital Computers*, Springer-Verlag.

Shiva, S. G. (2000) *Computer Design and Architecture*, 3rd edn, Marcel Dekker, Inc., New York.

4

Bus and Interconnection

In Neumann architecture, CPU, memory and I/O are connected through lines consisting of address, data, and control signals. The combination of address, data, and control signals are called a bus. In single bus architecture, all bus signals are usually integrated together according to a certain standard which can be easily connected to CPU, memory, a video card, a network card, and other I/O devices.

Earlier computers put CPU and memory together with some general bus interfaces on one circuit board named a motherboard. Other I/O cards such as video and network can be inserted later if they comply with the bus standard on the motherboard.

Laptop computers and all-in-one computers are usually designed to integrate all circuits including CPU, memory, video, sound and network cards into one circuit board. In these systems, the system bus is not visibly obvious, but it still exists as in traditional computers.

4.1 System Bus

A system bus, or in other words, an internal bus, is usually referred to as all signals that connect to a processor. A system bus consists of three sub bus systems, an address bus, a data bus and a control bus. Figure 4.1 shows the Neumann architecture and the three buses.

The address bus is used for addressing a specific memory unit or a specific I/O device.

The width (the number of address lines) determines the CPU's addressing ability. For an 8-bit processor, the addressing space is 256. For a 32-bit processor, the addressing space is 4G.

The data bus is used for accessing data with the main memory. The wider it is (more data bus lines), the more data the processor will be able to access at one time. For an 8-bit data bus, data can be read/write 8 bits at one time. If the data contains

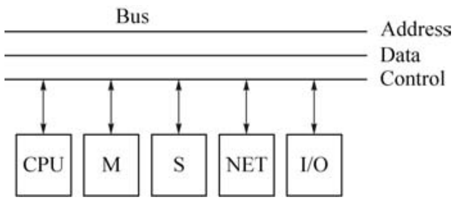


Figure 4.1 A system bus consists of an address bus, a data bus and a control bus

30 bits, then it requires four reading times to get the data. While for a 32-bit data bus, it only requires one read instruction to get all the 30 bits data.

The control bus is a collection of control signals from the CPU. Here are some control signals in the control bus:

- *Clock.* The clock signal (usually represented as φ) is used to synchronize memory and other devices with the processor. Some instructions only need one cycle (such as shift, logical bit operations), others may require more cycles to perform the instruction. For a processor with the clock frequency f, the cycle $T = 1/f$.
- *Reset.* The reset signal will initialize the processor. It is usually referred to as warm reboot. Many computers come with a reset button. In the event the system is frozen, a warm start usually can solve the problem.
- *Read/write.* The R/W signal is used to communicate with the memory or I/O devices to send or receive data to or from the CPU.
- *Interrupt.* The interrupt signal is used to indicate if there is a device requesting data exchange with the CPU. There is also an acknowledgment (ACK) signal that tells the device if the request has been granted.

4.1.1 Address Bus

An address bus is a computer bus (a series of lines connecting two or more devices) that is used to specify a physical address. When a processor or DMA-enabled device needs to read or write to a memory location, it specifies that memory location on the address bus (the value to be read or written is sent on the data bus). The width of the address bus determines the amount of memory a system can address (Dumas, 2006). For example, a system with a 32-bit address bus can address 2^{32} (4,294,967,296) memory locations. If each memory address holds one byte, the addressable memory space is 4 GB.

Early processors used one wire for each bit of the address. For example, a 16-bit address bus had 16 physical wires making up the bus. As the buses became wider, this approach became expensive in terms of the number of chip pins and board traces. Beginning with the Mostek 4096 DRAM, multiplexed addressing became common. In a multiplexed address scheme, the address is sent in two equal parts. This halves the number of address bus signals required to connect to the memory. For example a

32-bit address bus can be implemented by using 16 wires and sending the first half of the memory address, immediately followed by the second half.

A 64-bit register can store $2^{64} = 18{,}446{,}744{,}073{,}709{,}551{,}616$ different values, a number in excess of 18 quintillion. Hence, a processor with 64-bit memory addresses can theoretically directly access 2^{64} bytes of byte-addressable memory.

Without further qualification, a 64-bit computer architecture generally has integer and addressing registers that are 64 bits wide, allowing direct support for 64-bit data types and addresses. However, a CPU might have external data buses or address buses with different sizes from the registers, even larger or smaller (the 32-bit Pentium had a 64-bit data bus, for instance). The term may also refer to the size of low-level data types, such as 64-bit floating-point numbers.

Some computers have direct connections from the address bus of the processor and other system devices to the main memory. Many peripheral controllers can share system memory with the processor using a technique called DMA. A network, hard disk or graphics controller may be a DMA-enabled device (Foster and Iberall, 1985). This allows the controller to transfer data to and from the system faster than sending it through the processor one piece at a time.

Regardless of whether the physical address comes from the processor or a DMA device, it is latched onto the address bus. This action alerts the memory that a read or write request for that memory address is about to be made. If a write operation is pending, the data to be written is latched onto the data bus and a memory write signal is triggered. A read operation can be performed by triggering the memory read signal and reading the data bus.

Most personal computer (PC) compatible servers and desktops use a memory controller chip which is separate from the main processor. This controller communicates with the main system memory over the memory bus. This bus includes the address bus, data bus, and many control signals. The memory controller is located in the northbridge device and interfaces with the main processor using the front-side bus (FSB).

The northbridge memory controller and the FSB can create a bottleneck in some systems, slowing the processor's memory access. For this reason, a system's high-speed cache memory uses an entirely separate and wider cache bus. The cache is directly connected to the processor through this bus, bypassing the FSB and the northbridge completely. The cache bus, also known as the back-side bus (BSB), functions as an address bus, data bus, and control bus for the cache memory exclusively.

Some PC-compatible processors include a memory controller in the main processor itself. This controller accesses the main system memory directly, without using the FSB or the northbridge device. With these bottlenecks removed, the processor spends less time waiting on main system memory accesses. Cache memory is often included in these processors as well, and any external cache is accessed through the cache bus.

4.1.2 Data Bus

A data bus is a computer subsystem that allows for the transferring of data from one component to another on a motherboard or system board, or between two computers. This can include transferring data to and from the memory, or from the central processing unit (CPU) to other components. Each one is designed to handle so many bits of data at a time. The amount of data a data bus can handle is called bandwidth.

A typical data bus is 32-bits or 64-bits wide. This means that up to 32 bits or 64 bits of data can travel through a data bus every cycle.

In the early days of the personal computer, manufacturers created motherboards with data buses that were directly connected to the computer's memory and peripherals. These electrical buses were designed to run parallel to each other and had multiple connections. This direct connection was problematic for a number of reasons, but especially because all devices were forced to run at the same speed.

To eliminate this problem, developers used a bus controller to separate the CPU and memory from the peripheral devices, allowing CPU speed to be increased without requiring the same increase in peripheral speeds. This system also allowed expansion cards to speak to each other without going through the CPU, leading to quicker data transfer. All devices still must speak to each other at the same speed, however, so low bus speeds may slow an entire computer system.

Modern computers use both parallel and serial data buses. Parallel data buses carry data on many wires simultaneously. Each wire, or path, as they are sometimes called, carries one bit of data. The most common parallel buses found in computers today are the ATA, which stands for Advanced Technology Attachment; the PC card, which stands for personal computer and is used in laptops, and the SCSI, or Small Computer System Interface. A serial data bus has one wire or path, and carries all the bits, one after the other. The most common serial data buses include the USB, also known as the Universal Serial Bus; FireWire; Serial ATA; and Serial Attached SCSI.

Nearly every computer contains internal and external data buses. The internal data bus, also known as a local bus, connects all components that are on the motherboard, like the CPU and memory. The external data bus connects all peripheral devices to the motherboard. A variety of different external data buses are available; the appropriate type of data bus depends on the peripheral being attached to the computer.

4.1.3 Control Bus

A control bus is what a computer's central processing unit (CPU) uses to communicate with other devices inside the machine over a set of physical connections like cables or printed circuits. It is a diverse collection of signals, including read, write, and interrupt, that allow the CPU to direct and monitor what the different parts of the computer are doing. This is one of three types of buses that make up the system or computer bus. Its exact composition varies among processors.

In general, the purpose of any bus is to decrease the number of pathways necessary for communication between computer components. A bus allows communication between components over one data channel and is characterized by how much information it can transmit at once. The amount of data is expressed in bits and corresponds to the number of physical lines over which the information is sent. For example, a ribbon cable with 32 wires can send 32 bits in parallel.

Each computer usually has an internal and an expansion bus. The internal or front-side bus facilitates communication between the CPU and the central memory, while the expansion or input/output bus links the motherboard components like hard drives and ports. Most system buses are typically composed of between 50 and 100 separate physical lines for communication. These lines are subdivided into three subassemblies or types of buses: the address or memory bus, the data bus, and the command or control bus.

The control bus is bidirectional; it transmits command signals from the CPU and response signals from the hardware. It helps the CPU synchronize its command signals to the computer's components and slower external devices. As a result, the control bus consists of control lines that each send a specific signal, like read, write, and interrupt. The control lines that make up a control bus differ between processors, but most include system clock lines, status lines, and byte enable lines.

For example, a computer's CPU will use the data bus to transmit information to and from the central memory. The control bus allows the CPU to determine whether and when the system is sending or receiving this data. This is because a control bus has a control line for read and one for write that determine the direction the information flows (memory to CPU or CPU to memory). If the CPU needs to write some data to the central memory, it will send a signal on (assert) the control bus's write control line. Sending a signal on the read control line allows the CPU to receive data from memory.

The other types of buses that make up a system bus are the data and address buses. The data bus moves instructions and information between all the functional computer components. It is bidirectional and can transmit in only one direction at a time. The data bus transmits information between the CPU and memory and also between memory and the input/output section.

The address bus is unidirectional and functions like a map for the memory. When the computer system needs to access a particular memory location or input/output device, it asserts the appropriate address on the address bus. This address is recognized by the appropriate circuitry that then instructs the corresponding memory or device to read or send data on the data bus. Only the device or memory location that corresponds to the address on the address bus will respond.

4.2 Parallel Bus and Serial Bus

Buses on the motherboard are mostly parallel buses. A word of data can be read from memory at one time with one instruction. The industry standard bus (ISA) bus and peripheral component interconnect (PCI), mini PCI and STD-bus are parallel buses.

PCI express is a high speed serial system bus that has 1–32 bits bus width. It is commonly used for disk array controllers, gigabit Ethernet and Wi-Fi to supersede Accelerated Graphics Post (AGP) and PCI buses.

Universal Serial Bus (USB) is the most common serial bus we are using right now. It can connect with most computer peripherals such as keyboard, mouse, printers, and external hard drives.

A standard USB connector has four pins, two reserved for data and two for the power (Shiva, 2000). Figure 4.2 shows the cable wiring for USB 1.0/2.0. To reduce the interferences, the two data lines are pairs and twisted.

USB 2.0 is able to transmit at a speed up to 480 Mbps. It is generally considered sufficient for a variety of devices. However, with today's ever increasing demands on high-definition video content and terabyte storage devices, this speed is not really fast enough.

USB 3.0, or super speed USB, offers higher transfer rate up to 5 Gbps (Universal serial bus, 2012). It is backward compatible with USB 2.0. In addition, it provides better power management features.

4.2.1 Parallel Buses and Parallel Communication

In telecommunication and computer science, parallel communication is a method of sending several data signals simultaneously over several parallel channels. It contrasts with serial communication; this distinction is one way of characterizing a communications link.

The basic difference between a parallel and a serial communication channel is the number of distinct wires or strands at the physical layer used for simultaneous transmission from a device. Parallel communication implies more than one such wire-/strand, in addition to a ground connection. An 8-bit parallel channel transmits eight

PIN	Name	Des	Color
1	V_{CC}	+5V	Red
2	D–	data–	White
3	D+	data+	Green
4	GND	Ground	Black

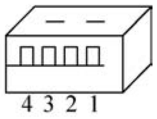


Figure 4.2 USB pinout and cable wiring diagram

bits (or a byte) simultaneously. A serial channel would transmit those bits one at a time. If both operated at the same clock speed, the parallel channel would be eight times faster. A parallel channel will generally have additional control signals such as a clock, to indicate that the data is valid, and possibly other signals for handshaking and directional control of data transmission.

Examples of parallel communication include peripheral buses such as ISA, ATA, SCSI PCI, and IEEE-1284 (printer port).

Before the development of high-speed serial technologies, the choice of parallel links over serial links was driven by these factors:

- *Speed.* Superficially, the speed of a parallel data link is equal to the number of bits sent at one time times the bit rate of each individual path; doubling the number of bits sent at once doubles the data rate. In practice, clock skew reduces the speed of every link to the slowest of all of the links.
- *Cable length.* Crosstalk creates interference between the parallel lines, and the effect worsens with the length of the communication link. This places an upper limit on the length of a parallel data connection that is usually shorter than a serial connection.
- *Complexity.* Parallel data links are easily implemented in hardware, making them a logical choice. Creating a parallel port in a computer system is relatively simple, requiring only a latch to copy data onto a data bus. In contrast, most serial communication must first be converted back into parallel form by a universal asynchronous receiver/transmitter (UART) before they may be directly connected to a data bus.

The decreasing cost of integrated circuits, combined with greater consumer demand for speed and cable length, has led to parallel communication links becoming deprecated in favor of serial links; for example, IEEE 1284 printer ports vs. USB, Parallel ATA vs. Serial ATA, and SCSI vs. FireWire.

On the other hand, there has been a resurgence of parallel data links in RF communication. Rather than transmitting one bit at a time (as in Morse code and BPSK), well-known techniques such as PSM, PAM, and Multiple-input multiple-output communication send a few bits in parallel. (Each such group of bits is called a "symbol"). Such techniques can be extended to send an entire byte at once (256-QAM). More recently techniques such as OFDM have been used in Asymmetric Digital Subscriber Line to transmit over 224 bits in parallel, and in DVB-T to transmit over 6048 bits in parallel.

4.2.2 Serial Bus and Serial Communication

In telecommunication and computer science, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels. Serial communication is used for all long-haul communication and most computer networks, where the cost of cable and synchronization difficulties makes parallel communication impractical. Serial computer buses are becoming more common even at shorter distances, as improved signal integrity and transmission speeds in newer serial technologies have begun to outweigh the parallel bus's advantage of simplicity (no need for serializer and deserializer, or SerDes) and to outstrip its disadvantages (clock skew, interconnect density). The migration from PCI to PCI Express is an example.

Integrated circuits are more expensive when they have more pins. To reduce the number of pins in a package, many ICs use a serial bus to transfer data when speed is not important. Some examples of such low-cost serial buses include SPI, I^2C, UNI/O, and 1-Wire.

The communication links across which computers – or parts of computers – talk to one another may be either serial or parallel. A parallel link transmits several streams of data simultaneously along multiple channels (e.g., wires, printed circuit tracks, or optical fibers); a serial link transmits a single stream of data.

Although a serial link may seem inferior to a parallel one, since it can transmit less data per clock cycle, it is often the case that serial links can be clocked considerably faster than parallel links in order to achieve a higher data rate. A number of factors allow serial to be clocked at a higher rate:

- Clock skew between different channels is not an issue (for unclocked asynchronous serial communication links).
- A serial connection requires fewer interconnecting cables (e.g., wires/fibers) and hence occupies less space. The extra space allows for better isolation of the channel from its surroundings.
- Crosstalk is less of an issue, because there are fewer conductors in proximity.

In many cases, serial is a better option because it is cheaper to implement. Many ICs have serial interfaces, as opposed to parallel ones, so that they have fewer pins and are therefore less expensive.

Examples of serial communication architecture include: Morse code, RS-232, Ethernet, Musical Instrument Digital Interface (MIDI), USB, SAS, Serial ATA (SATA), PCI Express, SONET, and so on.

4.2.2.1 Morse Code

Morse code is a method of transmitting textual information as a series of on-off tones, lights, or clicks that can be directly understood by a skilled listener or observer without special equipment. The International Morse Code encodes the ISO basic Latin alphabet, some extra Latin letters, the Arabic numerals and a small set of punctuation and procedural signals as standardized sequences of short and long signals called “dots” and “dashes” respectively, or “dits” and “dahs.” Because many non-English natural languages use more than the 26 Roman letters, extensions to the Morse alphabet exist for those languages.

Each character (letter or numeral) is represented by a unique sequence of dots and dashes. The duration of a dash is three times the duration of a dot. Each dot or dash is followed by a short silence, equal to the dot duration. The letters of a word are separated by a space equal to three dots (one dash), and two words are separated by a space equal to seven dots. The dot duration is the basic unit of time measurement in code transmission.

Morse code speed is measured in words per minute (wpm) or characters per minute (cpm). Characters have differing lengths because they contain differing numbers of dots and dashes. Consequently words also have different lengths in terms of dot duration, even when they contain the same number of characters. For this reason, a standard word is helpful to measure operator transmission speed. “PARIS” and “CODEX” are two such standard words.

One important feature of Morse code is coding efficiency. The length of each character in Morse is approximately inversely proportional to its frequency of occurrence in English. Thus, the most common letter in English, the letter “E,” has the shortest code, a single dot.

A related but different code was originally created for Samuel F. B. Morse’s electric telegraph by Alfred Vail in the early 1840s. This code was the forerunner on which modern International Morse code is based. In the 1890s it began to be extensively used for early radio communication before it was possible to transmit voice. In the late nineteenth and early twentieth century, most high-speed international communication used Morse code on telegraph lines, undersea cables and radio circuits.

Morse code is most popular among amateur radio operators although it is no longer required for licensing in most countries, including the US. Pilots and air traffic controllers are usually familiar with Morse code and require a basic understanding. Aeronautical navigational aids, such as VORs and NDBs, constantly identify in Morse code. An advantage of Morse code for transmitting over radio waves is that it is able to be received over poor signal conditions that would make voice communications impossible.

Because it can be read by humans without a decoding device, Morse is sometimes a useful alternative to synthesized speech for sending automated digital data to skilled listeners on voice channels. Many amateur radio repeaters, for example, identify with Morse even though they are used for voice communications.

For emergency signals, Morse code can be sent by way of improvised sources that can be easily “keyed” on and off, making it one of the simplest and most versatile methods of telecommunication. The most common distress signal is SOS or three dots, three dashes and three dots, internationally recognized by treaty.

4.2.2.2 RS-232

In telecommunications, RS-232 is the traditional name for a series of standards for serial binary single-ended data and control signals connecting between a DTE (Data Terminal Equipment) and a DCE (Data Circuit-terminating Equipment) (Randell, 1982). It is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin out of connectors. The current version of the standard is TIA-232-F Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange, issued in 1997.

A RS-232 port (as shown in Figure 4.3) was once a standard feature of a personal computer for connections to modems, printers, mice, data storage, un-interruptible power supplies, and other peripheral devices. However, the limited transmission speed, relatively large voltage swing, and large standard connectors motivated development of the universal serial bus which has displaced RS-232 from most of its peripheral interface roles. Many modern personal computers have no RS-232 ports and must use an external converter to connect to older peripherals. Some RS-232 devices are still found especially in industrial machines or scientific instruments.

In the book *PC 97 Hardware Design Guide*, Microsoft deprecated support for the RS-232 compatible serial port of the original IBM PC design. Today, RS-232 has mostly been replaced in personal computers by USB for local communications. Compared with RS-232, USB is faster, uses lower voltages, and has connectors that are simpler to connect and use. However, USB is limited by standard to no more than 5 meters of cable, thus favoring RS-232 when longer distances are needed. Both standards have software support in popular operating systems. USB is designed to make it easy for device drivers to communicate with hardware. However, there is no direct analog to the terminal programs used to let users communicate directly with serial ports. USB is more complex than the RS-232 standard because it includes a protocol for transferring data to devices. This requires more software to support the protocol used. RS-232 only standardizes the voltage of signals and the functions of the physical interface pins. Serial ports of personal computers are also sometimes used to directly control various hardware devices, such as relays or lamps, since the control lines of the interface can be easily manipulated by software. This is not feasible with USB, which requires some form of receiver to decode the serial data.

As an alternative, USB docking ports are available which can provide connectors for a keyboard, mouse, one or more serial ports, and one or more parallel ports.

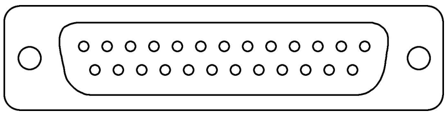


Figure 4.3 25pin RS-232 connector

Corresponding device drivers are required for each USB-connected device to allow programs to access these USB-connected devices as if they were the original directly connected peripherals. Devices that convert USB to RS-232 may not work with all software on all personal computers and may cause a reduction in bandwidth along with higher latency.

Personal computers may use a serial port to interface to devices such as uninterruptible power supplies. In some cases, serial data is not exchanged, but the control lines are used to signal conditions such as loss of power or low battery alarms.

Many fields (for example, laboratory automation, surveying) provide a continued demand for RS-232 I/O due to sustained use of very expensive but aging equipment. It is often far cheaper to continue to use RS-232 than it is to replace the equipment. Additionally, modern industrial automation equipment, such as PLCs, VFDs, servo drives, and CNC equipment are programmable via RS-232. Some manufacturers have responded to this demand: Toshiba re-introduced the DE-9M connector on the Tecra laptop.

Serial ports with RS-232 are also commonly used to communicate to headless systems such as servers, where no monitor or keyboard is installed, during boot when the operating system is not running yet and therefore no network connection is possible. A RS-232 serial port can communicate to some embedded systems such as routers as an alternative to network mode of monitoring.

4.2.2.3 Ethernet

Ethernet is a family of computer networking technologies for local area networks (LANs) commercially introduced in 1980. Standardized in IEEE 802.3, Ethernet has largely replaced competing wired LAN technologies.

Systems communicating over Ethernet divide a stream of data into individual packets called frames. Each frame contains source and destination addresses and error-checking data so that damaged data can be detected and re-transmitted.

The standards define several wiring and signaling variants. The original 10BASE5 Ethernet used coaxial cable as a shared medium. Later the coaxial cables were replaced by twisted pair and fiber optic links in conjunction with hubs or switches. Data rates were periodically increased from the original 10 megabits per second, to 100 gigabits per second.

Since its commercial release, Ethernet has retained a good degree of compatibility. Features such as the 48-bit MAC address and Ethernet frame format have influenced other networking protocols. Figure 4.4 shows a RJ-45 jack.

Ethernet evolved to include higher bandwidth, improved media access control methods, and different physical media. The coaxial cable was replaced with point-to-point links connected by Ethernet repeaters or switches to reduce installation costs, increase reliability, and improve management and troubleshooting. Many variants of Ethernet remain in common use.

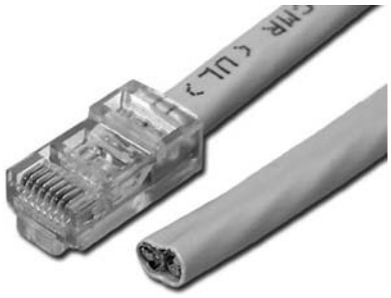


Figure 4.4 RJ-45 jack

Ethernet stations communicate by sending each other data packets: blocks of data individually sent and delivered. As with other IEEE 802 LANs, each Ethernet station is given a 48-bit MAC address. The MAC addresses are used to specify both the destination and the source of each data packet. Ethernet establishes link level connections, which can be defined using both the destination and source addresses. On reception of a transmission, the receiver uses the destination address to determine whether the transmission is relevant to the station or should be ignored. Network interfaces normally do not accept packets addressed to other Ethernet stations. Adapters come programmed with a globally unique address. An Ethertype field in each frame is used by the operating system on the receiving station to select the appropriate protocol module (i.e., the Internet protocol module). Ethernet frames are said to be self-identifying, because of the frame type. Self-identifying frames make it possible to intermix multiple protocols on the same physical network and allow a single computer to use multiple protocols together. Despite the evolution of Ethernet technology, all generations of Ethernet (excluding early experimental versions) use the same frame formats (and hence the same interface for higher layers), and can be readily interconnected through bridging.

Due to the ubiquity of Ethernet, the ever-decreasing cost of the hardware needed to support it, and the reduced panel space needed by twisted pair Ethernet, most manufacturers now build Ethernet interfaces directly into PC motherboards, eliminating the need for installation of a separate network card.

For signal degradation and timing reasons, coaxial Ethernet segments had a restricted size. Somewhat larger networks could be built by using an Ethernet repeater. Early repeaters had only two ports, allowing, at most, a doubling of network size. Once repeaters with 4, 6, 8, and more ports became available, it was possible to wire the network in a star topology. Early experiments with star topologies (called "Fibernet") using optical fiber were published by 1978.

Shared cable Ethernet was always hard to install in offices because its bus topology was in conflict with the star topology cable plans designed into buildings for telephony. Modifying Ethernet to conform to twisted pair telephone wiring already installed in commercial buildings provided another opportunity to lower costs, expand the installed base, and leverage building design, and, thus, twisted-pair Ethernet was the next logical development in the mid-1980s.

Ethernet on unshielded twisted-pair cables (UTP) began with StarLAN at 1 Mbit/s in the mid-1980s. In 1987 SynOptics introduced the first twisted-pair Ethernet at 10 Mbit/s in a star-wired cabling topology with a central hub, later called LattisNet. These evolved into 10BASE-T, which was designed for point-to-point links only, and all termination was built into the device. This changed repeaters from a specialist device used at the center of large networks to a device that every twisted pair-based network with more than two machines had to use. The tree structure that resulted from this made Ethernet networks easier to maintain by preventing most faults with one peer or its associated cable from affecting other devices on the network.

Despite the physical star topology and the presence of separate transmit and receive channels in the twisted pair and fiber media, repeater based Ethernet networks still use half-duplex and CSMA/CD, with only minimal activity by the repeater, primarily the Collision Enforcement signal, in dealing with packet collisions. Every packet is sent to every port on the repeater, so bandwidth and security problems are not addressed. The total throughput of the repeater is limited to that of a single link, and all links must operate at the same speed.

While repeaters could isolate some aspects of Ethernet segments, such as cable breakages, they still forwarded all traffic to all Ethernet devices. This created practical limits on how many machines could communicate on an Ethernet network. The entire network was one collision domain, and all hosts had to be able to detect collisions anywhere on the network. This limited the number of repeaters between the farthest nodes. Segments joined by repeaters had to all operate at the same speed, making phased-in upgrades impossible.

To alleviate these problems, bridging was created to communicate at the data link layer while isolating the physical layer. With bridging, only well-formed Ethernet packets are forwarded from one Ethernet segment to another; collisions and packet errors are isolated. Prior to discovery of network devices on the different segments, Ethernet bridges (and switches) work somewhat like Ethernet repeaters, passing all traffic between segments. However, as the bridge discovers the addresses associated with each port, it forwards network traffic only to the necessary segments, improving overall performance. Broadcast traffic is still forwarded to all network segments. Bridges also overcame the limits on total segments between two hosts and allowed the mixing of speeds, both of which became very important with the introduction of Fast Ethernet.

In 1989, the networking company Kalpana introduced their EtherSwitch, the first Ethernet switch. This worked somewhat differently from an Ethernet bridge, in that only the header of the incoming packet would be examined before it was either dropped or forwarded to another segment. This greatly reduced the forwarding latency and the processing load on the network device. One drawback of this cut-through switching method was that packets that had been corrupted would still be propagated through the network, so a jabbering station could continue to disrupt the entire network. The eventual remedy for this was a return to the original store and forward approach of bridging, where the packet would be read into a buffer on the switch in its entirety, verified against its checksum and then forwarded, but using more powerful application-specific integrated circuits. Hence, the bridging is then done in hardware, allowing packets to be forwarded at full wire speed.

When a twisted pair or fiber link segment is used and neither end is connected to a repeater, full-duplex Ethernet becomes possible over that segment. In full-duplex mode, both devices can transmit and receive to and from each other at the same time, and there is no collision domain. This doubles the aggregate bandwidth of the link and is sometimes advertised as double the link speed (e.g., 200 Mbit/s). The elimination of the collision domain for these connections also means that all the link's bandwidth can be used by the two devices on that segment and that segment length is not limited by the need for correct collision detection.

Since packets are typically delivered only to the port they are intended for, traffic on a switched Ethernet is less public than on shared-medium Ethernet. Despite this, switched Ethernet should still be regarded as an insecure network technology, because it is easy to subvert switched Ethernet systems by means such as ARP spoofing and MAC flooding.

The bandwidth advantages, the slightly better isolation of devices from each other, the ability to easily mix different speeds of devices and the elimination of the chaining limits inherent in non-switched Ethernet have made switched Ethernet the dominant network technology.

The Ethernet physical layer evolved over a considerable time span and encompasses quite a few physical media interfaces and several magnitudes of speed. The most common forms used are 10BASE-T, 100BASE-TX, and 1000BASE-T. All three utilize twisted pair cables and 8P8C modular connectors. They run at 10 Mbit/s, 100 Mbit/s, and 1 Gbit/s, respectively. Fiber optic variants of Ethernet offer high performance, electrical isolation and distance (tens of kilometers with some versions). In general, network protocol stack software will work similarly on all varieties.

4.2.2.4 MIDI*

MIDI is an electronic musical instrument industry specification that enables a wide variety of digital musical instruments, computers and other related devices to connect and communicate seamlessly with one another.

The primary functions of MIDI include communicating event messages about musical notation, pitch, velocity, control signals for parameters (such as volume, vibrato, audio panning, cues, and clock signals (to set and synchronize tempo) between multiple devices; these complete a signal chain and produce audible sound from a sound source. For users, MIDI enables a single player to sound as though they are playing two or more instruments simultaneously. As an electronic protocol, it is notable for its widespread adoption throughout the music industry.

Establishment of the MIDI standard occurred in the early 1980s, yielding a number of significant benefits to musicians, recording artists, and hobbyists.

- Common language and syntax – Electronic keyboards, drum machines, computers, music sequencers and other specialty instruments designed to work with MIDI indiscriminately communicate with one another.
- Simplified connectivity – Decreased the complexity (and volume) of connection cables required between devices.
- Fewer contributors required – Beginning in the 1980s, musical acts can perform live with as few as one or two members who operate multiple MIDI-enabled devices simultaneously and successfully deliver a performance which sounds similar to that of a much larger group of live musicians.
- Increased accessibility – Enabled users to create, edit, layer and build high-quality digital music recordings with less expense; professional musicians can now do this using a home recording space (or any other environment) without the need for renting a professional recording studio and staff. It has also enabled hobbyists, with little or no musical training, to produce high-quality recordings with the powerful capabilities of MIDI music editing software.
- Portability of electronic music gear – Greatly reduced the amount and variety of equipment (as well as wired connections) that performing musicians needed to travel around with, haul, pack and unpack, set up and connect, in order to produce a variety of sounds.

Standard applications that use MIDI:

- Electronic keyboards – Synthesizers and samplers which feature a built-in keyboard, MIDI keyboards (also referred to as MIDI controllers), or hardware music workstations.
- Personal computers – Equipped with an internal MIDI-capable sound card.
- MIDI interfaces – Used to connect MIDI devices to devices without built-in MIDI capability. A common usage scenario is enabling MIDI support on a computer via an external sound card, which connects via USB or FireWire. Other applications include inter-connectivity with analog (non-digital) audio outputs, microphone inputs, optical audio cables.
- Audio control surfaces – Often resembling a mixing console in appearance, enable a level of hands-on control for changing parameters such as sound levels and effects applied to individual tracks of a multitrack recording or live performance output.

- Digital effects units – Apply audio effects such as reverb, delay, and chorus to simulate the sound of the music being played in a large hall, or in a canyon, or with multiple voices all playing at once, respectively.
- Digital percussion devices – Triggers percussive or other relatively short sounds, usually via specifying a pattern or order in which the sounds should be played, on a drum machine or rhythm machine (also referred to as simply "beat boxes").
- Other musical instruments – Non-traditional and DIY devices custom-built to accept MIDI input, or devices adapted with additional hardware to provide MIDI-compatible signals. The MIDI guitar and MIDI violin are two such instruments.

In popular parlance, piano-style musical keyboards are called "keyboards," regardless of their functions or type. Among MIDI enthusiasts, however, keyboards and other devices used to trigger musical sounds are called "controllers," because with most MIDI set-ups, the keyboard or other device does not make any sounds by itself. MIDI controllers need to be connected to a voice bank or sound module in order to produce musical tones or sounds; the keyboard or other device is "controlling" the voice bank or sound module by acting as a trigger. The most common MIDI controller is the piano-style keyboard, either with weighted or semi-weighted keys, or with unweighted synth-style keys. Keyboard-style MIDI controllers are sold with as few as 25 keys (2 octaves), with larger models such as 49 keys, 61 keys, or even the full 88 keys being available. Different models have different feature sets, the simplest being only keys, while the more extravagant have sliders, knobs, and wheels to provide more controlling options. These include a variety of parameters that can be programmed within the controller, or sent to a computer to control software.

MIDI controllers are also available in a range of other forms, such as electronic drum triggers; pedal keyboards that are played with the feet (e.g., with an organ); wind controllers for performing saxophone-style music; and MIDI guitar synthesizer controllers. A wind controller is designed for performers who want to play saxophone, clarinet, oboe, bassoon, and other wind instrument sounds with a synthesizer module. When wind instruments are played using a MIDI keyboard, it is hard to reproduce the expressive control found on wind instruments that can be generated with the wind pressure and embouchure. A typical wind controller has an air-pressure level sensor and (usually) a bite sensor in the mouthpiece and touch sensors or keys (commonly approximating saxophone key arrangement) arrayed along the body. Additionally, controls such as buttons, touch sensors, and pitch wheels for generating additional midi messages or changing the way the controller behaves (for example, note sustain or octave shifts) are typically located in positions where they can, more or less easily, be accessed while playing. A less common type of wind controller mimics the mechanics of valved brass instruments.

Pad controllers are used by musicians and DJs who make music through use of sampled sounds or short samples of music. Pad controllers often have banks of assignable pads and assignable faders and knobs for transmitting MIDI data or changes; the better-quality models are velocity-sensitive. More rarely, some performers use more specialized MIDI controllers, such as triggers that are affixed to their clothing or stage items (e.g., magicians Penn and Teller's stage show).

A MIDI foot-controller is a pedalboard-style device with rows of switches that control banks of presets, MIDI program change commands and send MIDI note numbers (some also do MIDI merges). Another specialized type of controller is the drawbar controller; it is designed for Hammond organ players who have MIDI-equipped organ voice modules. The drawbar controller provides the keyboard player with many of the controls which are found on a vintage 1940s or 1950s Hammond organ, including harmonic drawbars, a rotating speaker speed control switch, vibrato and chorus knobs, and percussion and overdrive controls. As with all controllers, the drawbar controller does not produce any sounds by itself; it only controls a voice module or software sound device.

While most controllers do not produce sounds, there are some exceptions. Some controller keyboards called "performance controllers" have MIDI-assignable keys, sliders, and knobs, which allow the controller to be used with a range of software synthesizers or voice modules; yet at the same time, the controller also has an internal voice module which supplies keyboard instrument sounds (piano, electric piano, clavichord), sampled or synthesized voices (strings, woodwinds), and Digital Signal Processing (distortion, compression, flanging, etc.). These controller keyboards are designed to allow the performer to choose between the internal voices or external modules.

MIDI composition and arrangement typically takes place using either MIDI sequencing/editing software on PC-type computers, or using specialized hardware music workstations. Some composers may take advantage of MIDI 1.0 and General MIDI (GM) technology to allow musical data files to be shared among various electronic instruments by using a standard, portable set of commands and parameters. On the other hand, composers of complex, detailed works to be distributed as produced audio typically use MIDI to control the performance of high-quality digital audio samples and/or external hardware or software synthesizers.

Digital Audio Workstations (DAW) are becoming one of the most centric and common tools in the studio, and many are specifically designed to work with MIDI as an integral component. Through the use of MIDI mapping, various MIDI controllers can be used to command the program. MIDI piano rolls have been developed in many DAWs so that the recorded MIDI messages can be extensively modified. Virtual Instruments created by third party companies in one of a number of commonly used formats (for example, VST or RTAS) may be loaded as plug-ins thus providing a virtually limitless supply of sounds for a musician, and are designed to be commanded by MIDI controllers, especially in the DAW environment.

MIDI data files are much smaller than recorded audio waveforms. Many computer-sequencing programs allow manipulation of the musical data such that composing for an entire orchestra of sounds is possible. This ability to manipulate musical data has also introduced the concept of surrogate orchestras, providing a combination of half sequenced MIDI recordings and half musicians to make up an entire orchestral arrangement; however, scholars believe surrogate orchestras have the possibility of affecting future live musical performances in which the use of live musicians in orchestral arrangements may cease entirely because the composition of music via MIDI recordings proves to be more efficient and less expensive. Further, the data composed via the sequenced MIDI recordings can then be saved as a Standard MIDI File (SMF), digitally distributed, and reproduced by any computer or electronic instrument that also adheres to the same MIDI, GM, and SMF standards.

Although not a wave audio file format, the Standard MIDI File was, due to its much smaller file size, attractive to computer users as a substitute before broadband Internet became widespread. Later, the advent of high quality audio compression such as the MP3 format has decreased the size advantages of MIDI-encoded music to some degree, though MP3 is still much larger than SMF.

A standard for MIDI over USB was developed in 1999 as a joint effort between IBM, Microsoft, Altec Lansing, Roland Corporation, and Philips. To transmit MIDI over USB a Cable Number and Cable Index are added to the message, and the result is encapsulated in a USB packet. The resulting USB message can be double the size of the native MIDI message. Since USB is over 15,000 times faster than MIDI (480 000 kbit/s vs 31.25 kbit/s,) USB has the potential to be much faster. However, due to the nature of USB there is more latency and jitter introduced that is usually in the range of 2 to 10 ms, or about 2 to 10 MIDI commands. Some comparisons done in the early part of the 2000s showed USB to be slightly slower with higher latency, and this is still the case today. Despite the latency and jitter disadvantages, MIDI over USB is increasingly common on musical instruments.

4.3 Synchronous Bus and Asynchronous Bus

Synchronous buses use system clock to provide time information for all signals on the buses. A signal is considered invalid unless it is synchronized by the falling edge (or raising edge) of the clock. On the other hand, signals should be in a stable state before the clock edge to avoid competition that may cause the circuit or system to be in an unstable state.

When CPU wants to read data from memory, the address signals is first enabled and then the read commend is issued. The data will not be read until the next clock edge is reached. So at the time of the “read” action has taken place, all data are already stored in MDR and in stable condition. The whole procedure is shown in Figure 4.5.

DMA uses synchronous method to transfer data. Once the initial negotiation process is succeeded, a block of data will be transferred between the device and memory synchronized by the clock.

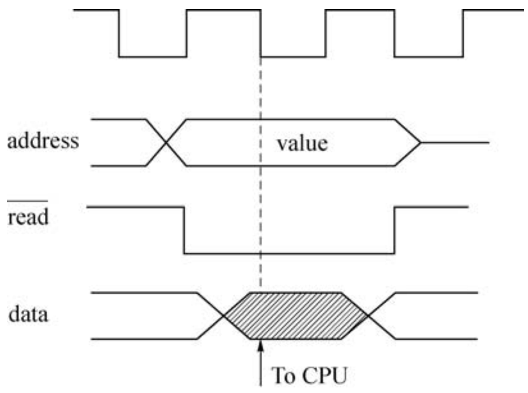


Figure 4.5 A process of synchronous read

Unlike synchronous bus uses clock to coordinate transactions, asynchronous bus uses handshakes to conduct bus communication. Handshaking enables low speed devices to communicate with high speed CPU easily.

As technology advances, most computers use synchronous bus. The clock frequency is an important factor for a computer bus. If the devices have different speed, we have to set the bus speed to the speed of the slowest devices in the system. For example, the conventional PCI bus uses 33 MHz or 66 MHz. The capacity is up to 266 MB/s for a 32-bit computer. The capacity of PCI Express bus can reach up to 8 GB/s for a version 2.0 with 16 lane slots.

To connect I/O devices to the processor, asynchronous operations are more efficient. Asynchronous buses have the advantage of matching devices communicating with the processor with various speeds.

Synchronous systems negotiate the communication parameters at the data link layer before communication begins. Basic synchronous systems will synchronize the signal clocks on both sides before transmission begins, reset their numeric counters and take other steps. More advanced systems may negotiate things like error correction and compression.

It is possible to have both sides try to synchronize the connection at the same time. Usually, there is a process to decide which end should be in control. Both sides in synchronous communication can go through a lengthy negotiation cycle where they exchange communications parameters and status information. With a lengthy connection establishment process, a synchronous system using an unreliable physical connection will spend a great deal of time in negotiating, but not in actual data transfer. Once a connection is established, the transmitter sends out a signal, and the receiver sends back data regarding that transmission, and what it received. This connection negotiation process takes longer on low error-rate lines, but is highly efficient in systems where the transmission medium itself (an electric wire, radio signal or laser beam) is not particularly reliable.

Asynchronous communication utilizes a transmitter, a receiver and a wire without coordination about the timing of individual bits. There is no coordination between the two end points on just how long the transmitter leaves the signal at a certain level to represent a single digital bit. Each device uses a clock to measure out the "length" of a bit. The transmitting device simply transmits. The receiving device has to look at the incoming signal and figure out what it is receiving and coordinate and retime its clock to match the incoming signal.

Sending data encoded into your signal requires that the sender and receiver are both using the same encoding/decoding method, and know where to look in the signal to find data. Asynchronous systems do not send separate information to indicate the encoding or clocking information. The receiver must decide the clocking of the signal on its own. This means that the receiver must decide where to look in the signal stream to find ones and zeroes, and decide for itself where each individual bit stops and starts. This information is not in the data in the signal sent from transmitting unit.

When the receiver of a signal carrying information has to derive how that signal is organized without consulting the transmitting device, it is called asynchronous communication. In short, the two ends do not always negotiate or work out the connection parameters before communicating. Asynchronous communication is more efficient when there is low loss and low error rates over the transmission medium because data is not retransmitted and no time is spent negotiating the connection parameters at the beginning of transmission. Asynchronous systems just transmit and let the far end station figure it out. Asynchronous is sometimes called "best effort" transmission because one side simply transmits, and the other does its best to receive and any lost data is recovered by a higher level protocol.

4.4 Single Bus and Multiple Buses

A bus is usually referred to as address, data and control signals that connect to a processor. An actual computer may have more buses than just only one single bus. For example a PCI bus that is used to connect SCSI, LAN, and on board graphics; an AGP bus can be used to connect newer graphics cards; a memory bus that is used to connect memory; an ISA bus that connects keyboard, mouse, and other ISA cards. There may be an IDE bus that connects old IDE hard drives and CD-ROMs. Figure 4.6 is a diagram of a computer with abovementioned buses.

Moreover, many recent computers come with two to four cores (processors) that were manufactured together on one chip. Some others use multiple independent processors as we often see on server computers. For those computers, exchange data and coordinate operations between processors are important. The interconnection buses are designed for this purpose. We will discuss interconnection buses in detail in the next section.

Now let us look into the single process and multi-core (dual or quad) computers. Figure 4.4 shows that there are many buses in a computer system. If we look into it,

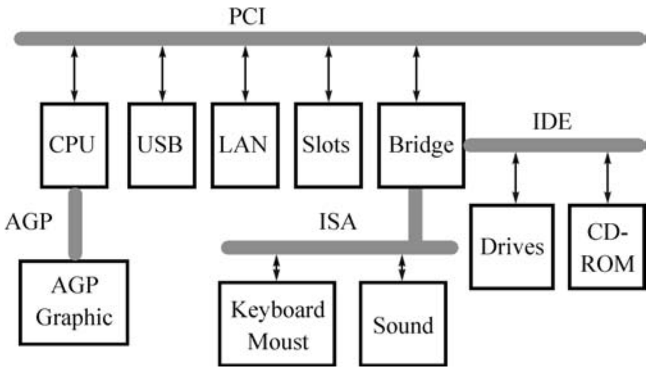


Figure 4.6 A computer with PCI, AGP, IDE and IDA buses

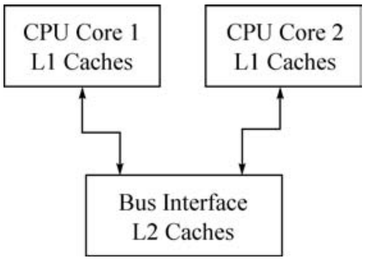


Figure 4.7 Diagram of a dual-core processor

we can see that it is essentially a single bus system. In addition to the PCI bus, the AGP is merely a graphics accelerator. It goes through a bridge that bridges AGP and PCI together. Similarly, ISA and IDE also go through a bridge that exchange signals between ISA and IDE with PCI. So it is indeed a single bus system with some variations.

As a multi-core (dual-core or quad-core) computer contains two or more processors integrated on one chip, people would think this is a multi-bus system. If we look at the architecture (diagram shown in Figure 4.7), we can see that both processors are connected to a bus interface. This tells us that it uses one bus to connect to the outside "world." Two or more processors (cores) just add up the computation power. It should still be classified as a single bus system.

For a computer with multiple independent CPUs (not multiple cores), they can each has its own memory or share the common memory modules. To do this we need an interconnection network (bus) to connect each CPU with the resources.

4.5 Interconnection Buses

Computers with multiple processors located in a central location or distributed locations are generally considered as a parallel processing system (Hwang, 1993). Here

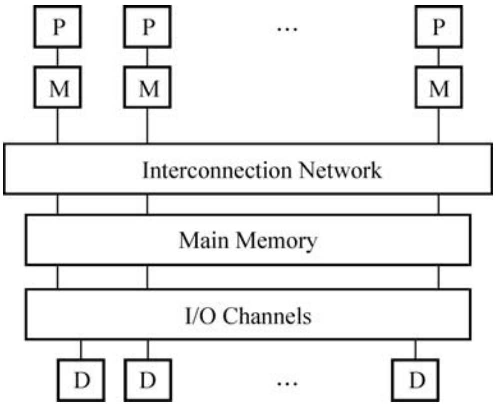


Figure 4.8 A typical MIMD architecture

multiple processors mean multiple processing units. It is different than multi-core computes we have just discussed.

In parallel processing, instructions and data have four combinations to connect them together:

- Single Instruction stream, Single Data stream (SISD)
- Single Instruction stream, Multiple Data stream (SIMD)
- Multiple Instruction stream, Single Data stream (MISD)
- Multiple Instruction stream, Multiple Data stream (MIMD).

A SISD computer is a Neumann machine built with a single processor.

For a MIMD system, each processor has its own cache memory. They also share the main memory, storage and I/O devices in order to share data among the processors.

Figure 4.8 shows a typical MIMD architecture where P stands for processors; M stands for main memory and D stands for disks (storage).

In Figure 4.6, processors are interconnected through the interconnection bus that connects not only the processors but also the main memory and I/O devices.

A distributed MIMD system is illustrated in Figure 4.9.

4.6 Security Considerations for Computer Buses

Fewer people have studied the vulnerabilities of computer buses to attackers. A computer system can be taken over from devices that are attached to the buses. The data stored on the computer can also be exposed to the programs or virtual machines that were booted from a USB memory attached to a computer USB port.

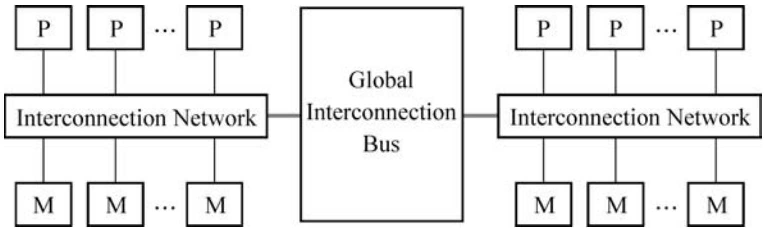


Figure 4.9 A diagram of a distributed MIMD system

In single bus system, CPU, memory and I/O devices are all connected to the system bus. A compromised device can issue attacks on this computer system through interruptions to stop processors from executing normal tasks.

On the other hand, attackers may divert the program handling the interruptions to a pre-defined location where malicious program is stored. The malicious program then can steal data or stop the services.

USB-booting attack is another type of attack that is launched from a universal serial bus. Most recent computers are able to be booted from a flash drive that is connected to the USB port. An attacker can insert a USB flash drive and boot a virtual service or virtual machine on top of the current operating system. The virtual service or virtual machine then may be able to access the local hard drives to get the data.

The challenge for distributed interconnection buses is even bigger. MIMD computers share common memory. If one local area is compromised, the problem could spread out to the entire system.

To stop USB-booting attack, people can either physically block the access to the USB ports or disable the ports by default. On the other hand, people can disable the USB boot feature so that no program can be executed from USB devices.

4.7 A Dual-Bus Interface Design

A dual-bus interface is a circuit that:

- connects one device to two buses; and
- enables the device to be accessed simultaneously from those two buses.

Being able to access from both buses at the same time is key for a dual bus interface.

Here is an example, an external DVD may have two or more buses (USB and 1394) that can be connected. This does not necessary mean it is dual-bus enabled. We can choose to use either USB or 1394 to connect to a computer system. Since normally we cannot connect both buses to a computer system, therefore it is not a dual-bus device.

*4.7.1 Dual-Channel Architecture**

Dual-channel-enabled memory controllers in a PC system architecture utilize two 64-bit data channels. Dual channel should not be confused with double data rate (DDR), in which data exchange happens twice per DRAM clock. The two technologies are independent of each other and many motherboards use both, by using DDR memory in a dual-channel configuration.

Dual-channel architecture requires a dual-channel-capable motherboard and two or more DDR, DDR2 SDRAM, or DDR3 SDRAM memory modules. The memory modules are installed into matching banks, which are usually color coded on the motherboard. These separate channels allow each memory module access to the memory controller, increasing throughput bandwidth. It is not required that identical modules be used (if motherboard supports it), but this is often recommended for best dual-channel operation. It is possible to use a single-sided module of 512 MB and a double-sided module of 512 MB in dual-channel configuration, but how fast and stable it is depends on the memory controller.

If the motherboard has two pairs of differently colored DIMM sockets (the colors indicate which bank they belong to, bank 0 or bank 1), then one can place a matched pair of memory modules in bank 0, but a different-capacity pair of modules in bank 1, as long as they are of the same speed. Using this scheme, a pair of 1 GB memory modules in bank 0 and a pair of matched 512 MB modules in bank 1 would be acceptable for dual-channel operation.

Modules rated at different speeds can be run in dual-channel mode, although the motherboard will then run all memory modules at the speed of the slowest module. Some motherboards, however, have compatibility issues with certain brands or models of memory when attempting to use them in dual-channel mode. For this reason, it is generally advised to use identical pairs of memory modules, which is why most memory manufacturers now sell “kits” of matched-pair DIMMs. Several motherboard manufacturers only support configurations where a “matched pair” of modules is used. A matching pair needs to match in:

- Capacity (e.g., 1024 MB). Certain Intel chipsets support different capacity chips in what they call Flex Mode: the capacity that can be matched is run in dual-channel, while the remainder runs in single-channel.
- Speed (e.g., PC5300). If speed is not the same, the lower speed of the two modules will be used. Likewise, the higher latency of the two modules will be used.
- Number of chips and sides (e.g., two sides with four chips on each side).
- Matching size of rows and columns.

Dual-channel architecture is a technology implemented on motherboards by the motherboard manufacturer and does not apply to memory modules. Theoretically any matched pair of memory modules may be used in either single- or dual-channel operation, provided the motherboard supports this architecture.

Dual-channel technology was created to address the issue of bottlenecks. Increased processor speed and performance requires other, less prominent components to keep pace. In the case of dual-channel design, the intended target is the memory controller, which regulates data flow between the CPU and system memory (RAM). The memory controller determines the types and speeds of RAM as well as the maximum size of each individual memory module and the overall memory capacity of the system. However, when the memory is unable to keep up with the processor, a bottleneck occurs, leaving the CPU with nothing to process. Under the single-channel architecture, any CPU with a bus speed greater than the memory speed would be susceptible to this bottleneck effect.

The dual-channel configuration alleviates the problem by doubling the amount of available memory bandwidth. Instead of a single memory channel, a second parallel channel is added. With two channels working simultaneously, the bottleneck is reduced. Rather than wait for memory technology to improve, dual-channel architecture simply takes the existing RAM technology and improves the method in which it is handled. While the actual implementation differs between Intel and AMD motherboards, the basic theory stands.

There have been varying reports as to the performance increase of dual-channel configurations, with some tests citing significant performance gains while others suggest almost no gain.

Tom's Hardware found little significant difference between single-channel and dual-channel configurations in synthetic and gaming benchmarks (using a "modern" system setup). In its tests, dual channel gave at best a 5% speed increase in memory-intensive tasks. Another comparison by laptoplogic.com resulted in a similar conclusion for integrated graphics. The test results published by Tom's Hardware had a discrete graphics comparison.

The difference can be far more significant in applications that manipulate large amounts of data in memory. A comparison by *TechConnect Magazine* demonstrated considerable gains for dual-channel in tasks using block sizes greater than 4 MB, and during stream processing by the CPU.

4.7.2 Triple-Channel Architecture[*]

DDR3 triple-channel architecture is used in the Intel Core i7-900 series (the Intel Core i7-800 series only support up to dual-channel), and the LGA 1366 platform (e.g., Intel X58). AMD Socket AM3 processors do not use the DDR3 triple-channel architecture but instead use dual-channel DDR3 memory. The same applies to the Intel Core i3, Core i5 and Core i7-800 series, which are used on the LGA 1156 platforms (e.g., Intel P55). According to Intel, a Core i7 with DDR3 operating at 1,066 MHz will offer peak data transfer rates of 25.6 GB/s when operating in triple-channel interleaved mode. This, Intel claims, leads to faster system performance as well as higher performance per watt.

When operating in triple-channel mode, memory latency is reduced due to interleaving, meaning that each module is accessed sequentially for smaller bits of data rather than completely filling up one module before accessing the next one. Data is spread among the modules in an alternating pattern, potentially tripling available memory bandwidth for the same amount of data, as opposed to storing it all on one module.

The architecture can only be used when all three, or a multiple of three, memory modules are identical in capacity and speed, and are placed in three-channel slots. When two memory modules are installed, the architecture will operate in dual-channel mode.

DDR3 Quadruple-channel architecture is used in the AMD G34 platform and the Intel LGA 2011 platform (e.g., Intel X79). AMD processors which are used on the C32 platform instead use dual-channel DDR3 memory. Intel processors which are used on the LGA 1155 platform (e.g., Intel Z68) instead use dual-channel DDR3 memory.

The architecture can only be used when all four, or a multiple of four, memory modules are identical in capacity and speed, and are placed in quad-channel slots. When two memory modules are installed, the architecture will operate in dual-channel mode. When three memory modules are installed, the architecture will operate in triple-channel mode.

4.7.3 A Dual-Bus Memory Interface

Figure 4.10 shows a diagram of a dual-bus interface. Here control signal EN is used to enable or disable the communication between a bus and the storage S.

Motorola's MPC8260 is a chip that contains a 64-bit PowerPC microprocessor and a versatile communications processor module (CPM). The MPC 8260 is used in a wide array of applications, especially those in the communications and networking markets. Examples include remote access servers, regional office routers, cellular base stations, and SONET transmission controllers.

A Lattice's ispGDX2$^{\text{TM}}$ Generic Digital Crosspoint Switch is used as a multiport interface. The ispGDX2 device can interface the MPC 8260 with an external master and a number of slaves including SDRAM and FLASH. The control logic for the SDRAM and FLASH is built in a CPLD which is used to interface the MPC 8260 to the ispGDX2 device and to control the read/write to the memory. This function can be implemented in Lattice CPLDs. Figure 4.11 shows the diagram using MPC 8260 with the multiport interface.

4.8 Summary

In this chapter, we studied different kinds of computer buses and how different components are interconnected. A system bus is composed of an address bus, a data bus and a control bus. The address bus determines the addressing space of a computer system. In other words, it specifies how much memory can be attached onto a computer system. The data bus determines how much data can be read or written between CPU and memory or I/O devices. A wider data bus will reduce the read/-write times therefore will increase the system speed.

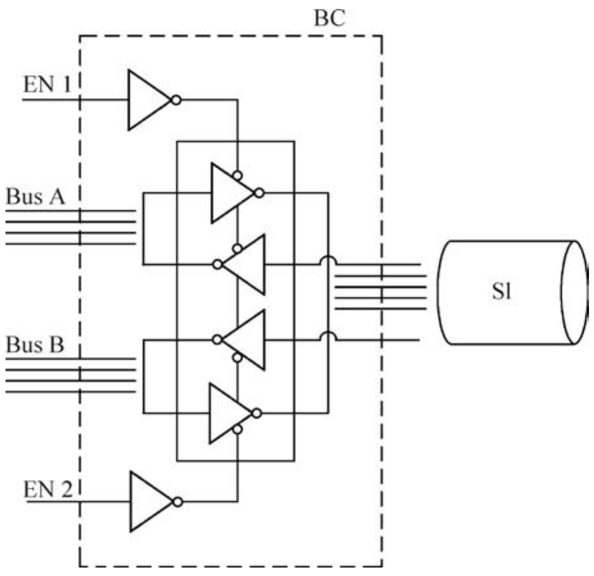


Figure 4.10 A dual-bus interface diagram

Computer buses can be categorized as parallel buses and serial buses. A parallel bus enables high throughput so it is most commonly seen in the motherboard and interfaces that require best speed performances. A serial bus, on the other hand, is small in size and easy to build. With the technology advancement, the speed carried on by serial buses is considerable high. So serial buses have wide applications. USB 3.0 is able to transfer data at a speed up to 4.8 Gbps with only two data signals. Parallel computers usually have many processors each with its own memory and also share the common memory. They also have shared I/O devices that are connected through the interconnection buses.

In a similar way to other computer components, computer buses are also vulnerable to different kinds of attacks. USB-booting attacks can be prevented by physically

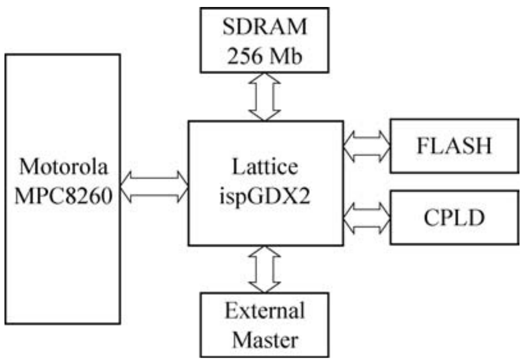


Figure 4.11 Diagram of ispGDX2 Multiport Interface

blocking the USB ports or disabling the booting feature from the operating system. Attacks launched from compromised devices can be addressed in the following chapters.

Exercises

4.1 What is a system bus? What is a system bus composed of?

4.2 A processor has 8-bit data bus width. How many times does it needed to read a 28-bit data into the processor? What if the processor has a 32-bit data bus?

4.3 What is a parallel bus? What is a serial bus?

4.4 USB is the most common serial bus. It can connect many computer peripherals. Is it possible to use it as the video interface, why?

4.5 Why DMA transfers data efficiently than handshakes mode?

4.6 If we need to contact low speed devices to a processor, what considerations should we take?

4.7 A computer has an AGP bus that connects to the graphic card. It has a PCI bus that connects SCSI. It also has an ISA that connects keyboard and mouse. In addition, it has an IDE bus that connects a hard drive and CD-ROM. Why is this multiple-bus computer still considered as single bus Neumann architecture?

4.8 People consider a dual-core or quad-core computer as single bus Neumann architecture. Do you agree or disagree, why?

4.9 What does SISD, SIMD, MISD, MIMD mean? What are used to connect the processors in those systems?

4.10 How does a USB-booting attack work? How to prevent such attacks?

4.11 An external hard drive has two interfaces: USB 2.0 and IEEE 1393. Do you think it is a dual-port interface device or not?

References

Dumas, J. D. (2006) *Computer Architecture: Fundamentals and Principles of Computer Design*, Taylor & Francis.

Foster, C. C. and Iberall, T. (1985) *Computer Architecture*, 3rd edn, Wan Nostrand Reinhold Company, New York.

Hwang, K. (1993) *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, Inc.

Randell, B. (ed.) (1982) *The Designs of Digital Computers*, Springer-Verlag.

Shiva, S. G. (2000) *Computer Design and Architecture*, 3rd edn, Marcel Dekker, Inc., New York.

Universal serial bus (2012) Wikipedia. Retrieved February 29, 2012 at http://en.wikipedia.org/wiki/Universal_serial_bus

5

I/O and Network Interface

Input and output devices are as important as the CPU. No matter how fast a CPU is, it needs to take instructions and send data out. In many cases, the destinations are remote. In this case, data will be transmitted in and out through a network interface which handles the communication task.

I/O devices contain several internal registers. The registers are used to receive command and data from the processor. On the other hand, I/O devices carry data differently. Some data are in high speed, some others in very low speed. Input from a keyboard is very slow in speed. In designing keyboard interface, we would not want CPU to spend time waiting for input from a keyboard. A general I/O diagram is shown in Figure 5.1.

Animated video images on the other hand are quite different. For every second, a CPU needs to send out at least 60 frames. Each frame may contain million pixels and each pixel contains 32-bit color depth. If we add double buffer and stereo feature, for example 3D movie, the overall bit rate would exceed 10^9 bps or even higher.

Data transmitted through I/O devices could be very few, or it could be very large. For a large data transfer, like a hard drive, in order to save CPU time, we usually use a specialized interface named DMA.

5.1 Direct Memory Access

If we move large trunks of data from I/O devices to main memory, the overhead for each move has to be considered. Traditionally, data are read into registers and then written to the memory from the CPU. It requires two steps.

With a DMA controller, data can be read/written directly from/to memory without requiring interactions with CPU during transmission (Dumas, 2006).

The DMA controller is controlled by CPU. It receives data transfer instructions from the processor. Usually the CPU sends out I/O device numbers, memory start address and number of bytes to the DMA controller. Once this is done, the DMA

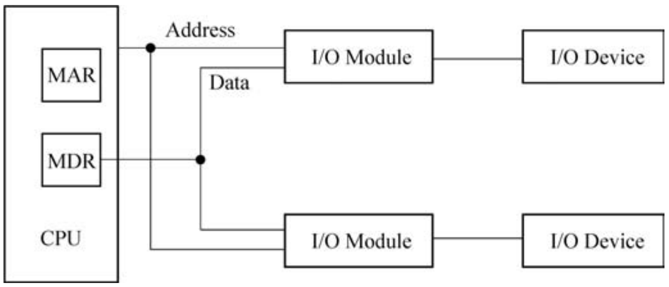


Figure 5.1 A general I/O interface diagram

controller will do the rest. It will initiate the transmission until the data are all read. The writing process from memory to I/O is the same except data are moved in reverse direction.

Once the DMA controller completes the transmission, it will send an interrupt signal to notify CPU.

A DMA device usually contains a word-count register (WCR), an address register (AR), and a data register/buffer (DR). Figure 5.2 shows different I/O diagrams:

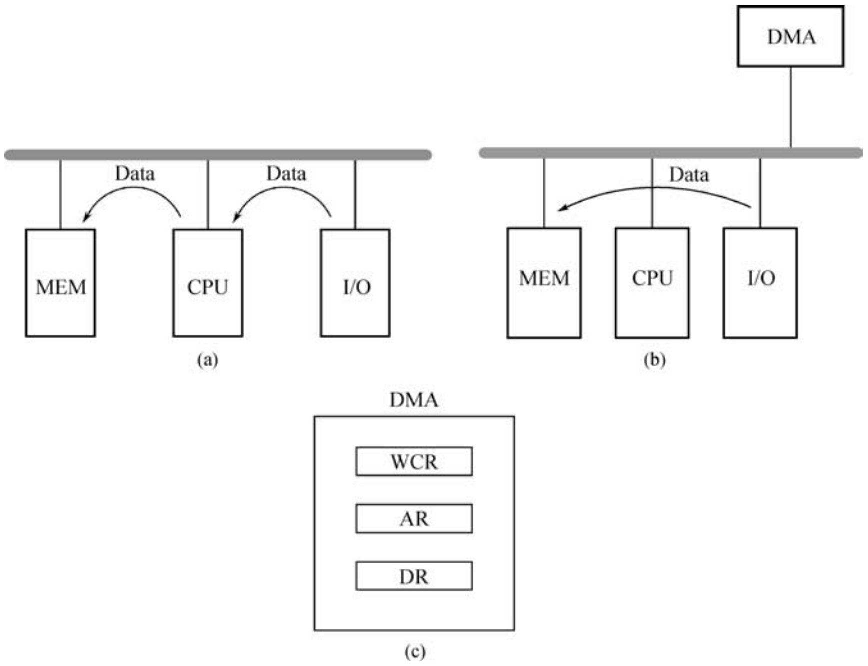


Figure 5.2 I/O diagrams (a) I/O data exchange without DMA, (b) I/O data exchange with DMA and (c) architecture diagram of a DMA

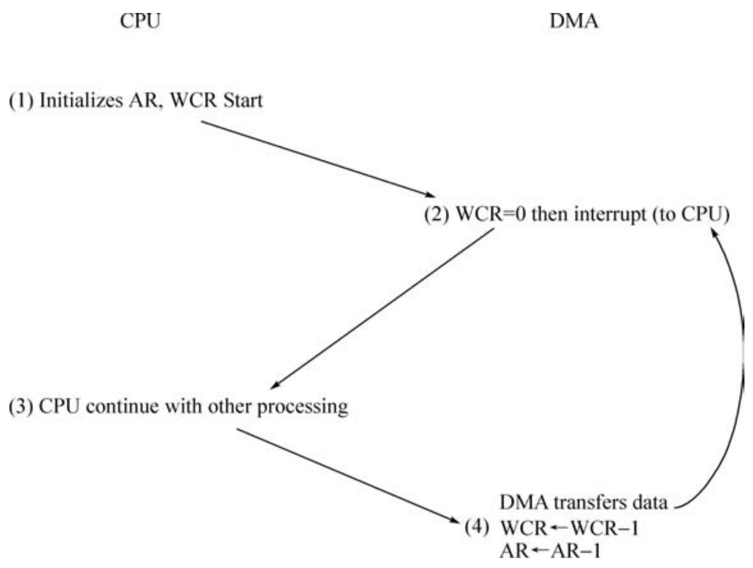

CPU
DMA
(1) Initializes AR, WCR Start
(2) WCR=0 then interrupt (to CPU)
(3) CPU continue with other processing
DMA transfers data
(4) WCR←WCR−1
AR←AR−1

Figure 5.3 DMA handshake process

(a) I/O data exchange without DMA, (b) I/O data exchange with DMA and (c) architecture of a DMA.

DMA data transfers include a handshaking process. First the CPU initializes the AR and WCR on the DMA side, it first checks whether $\text{WCR} = 0$. If yes, it means the transfer is complete, DMA will send an interrupt signal to CPU.

If $\text{WCR} \neq 0$, then the DMA starts data transfer. At each iteration, the work-count register reduces by one and the address register increases by one. $\text{WCR} \leftarrow \text{WCR} - 1$, $\text{AR} \leftarrow \text{AR} + 1$. The data continue transferring until $\text{WCR} = 0$. Figure 5.3 shows a DMA handshake process.

5.2 Interrupts

A processor normally executes programs with no interrupts. In some situations, the processor may be interrupted by external condition to handle some emergency tasks (Foster and Iberall, 1985). Here are some special conditions:

- A system reset
- Power failure
- Buffer overflow or underflow
- Illegal instruction code
- Completion of DMA data transfer
- Software requests.

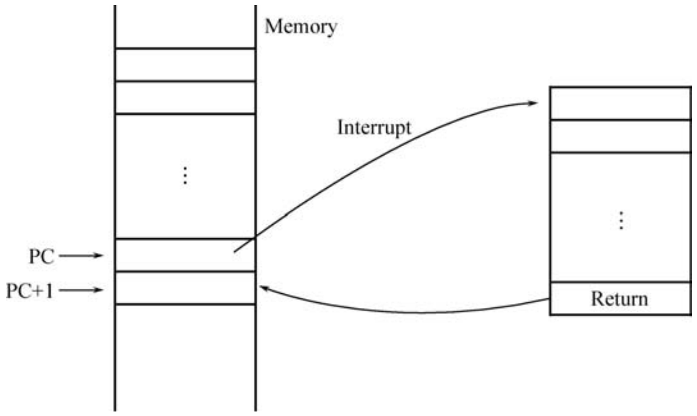


Figure 5.4 Interrupt handling

In each of these conditions the processor must stop the current process to handle the event. After handling the interruptions, the processor will resume the current operation. Figure 5.4 shows a diagram of how CPU handles an interrupt.

The interrupts handling process first saves the current program counter +1 to a stack and then jumps to the interruption service procedure.

The interruption service routine usually performs the following functions:

- Save the processor status
- Handle the interrupt
- Return from the interrupt
- Pop the stack and get the returning address: PC + 1
- Jump to PC + 1.

During the process of saving the processor status, the processor may disable further interrupts in order to prevent status change (Randell, 1982).

Interrupts can be prioritized in program or in hardware. With flip-flops, interrupt requests can be grouped into multi-priority. At any time, a higher priority device can always interrupt a lower priority device, while a lower priority device can not interrupt a higher priority device. Some CPUs use a mask register to enable/disable certain devices to interrupt.

5.3 Programmed I/O

A keyboard, mouse, display, and printer are all standard I/O devices. The interfaces that connect to the registers in CPU are designed for a single purpose, just like a keyboard and a mouse cannot be misplaced on PS/2 interfaces.

In many cases, people want to have a general I/O module that can either be set to input or at some other time be set to output depending on the applications. The CPU

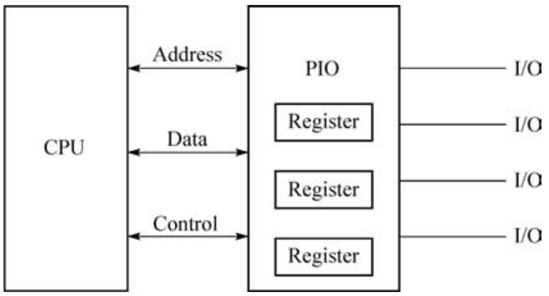


Figure 5.5 Programmed I/O

first tells the I/O module what function CPU wants it to be, and then sends the data. I/O modules where their functions can be set by instructions are called programmed I/O or in short, PIO. Figure 5.5 shows a diagram of PIO connected to a CPU.

The input/output data and address registers work similarly to the MAR and MDR. They share the same bus as the memory does.

There are two ways of mapping I/O ports: memory-mapped I/O and isolated I/O. Memory-mapped I/O maps I/O ports to reserved memory address space. In this method, processors writing to an I/O port are similar to writing to memory.

Isolated I/O mapping does not use any memory space. It used separate I/O address space. Most Intel processors support isolated mapping. In these systems, special I/O instructions are needed to access I/O devices.

Systems designed with processors that support the isolated I/O have the flexibility of using memory mapped I/O or isolated I/O. For example, a keyboard, a mouse and a monitor (KVM) are usually mapped to I/O space while a video card could be mapped to a block of memory address using the memory mapping interface.

5.4 USB and IEEE 1394

The universal Serial Bus was developed in 1995. The major goal of the USB was to define an external expansion bus that makes attaching peripherals to a computer as easy as hooking up a telephone to a jack.

Before USB, computer users had many problems with connecting peripherals. We have seen a normal PC may contain PS/2 for keyboard and mouse, parallel for printer, VGA for display, serial and modem for gaming and communication, and so on. If people want to connect two printers, then they need a splitter to switch manually.

With the introduction of USB, now people can connect a keyboard, a mouse, printers, external memory and hard drives, camera up to 127 devices through USB ports.

If we connect a keyboard and a printer through USB ports, the biggest question is how a processor identifies whether it is a keyboard or a printer? USB supports true plug-and-play connectivity. It avoids setting jumpers and configuring the new devices. In addition, people can hot plug the USB devices and the system will automatically detach the devices and configure it for immediate use.

5.4.1 USB Advantages

Here are some main advantages of USB:

- *Power distribution.* The USB cable can provide +5 V power to the device. For small devices (100–500 mA), it helps avoid using an external power supply.
- *Expandable.* USB provides expandability through USB hubs that are as small as an eraser and are very cheap. A four-port hub costs less than $10.
- *Power conservation.* USB devices enter a suspend state if there is no activity on the bus for 3 mS. In the suspended state, devices draw only about 2.5 mA current.

5.4.2 USB Architecture

The host hardware consists of a USB host controller and a root hub. The host controller initiates transmissions over the USB, and the root hub provides the connection points. Figure 5.6 shows the diagram of USB architecture.

A host controller interface (HCI) is a register level interface which allows host controller hardware to communicate with the operating system of a host computer. There are three types of USB host controller interfaces:

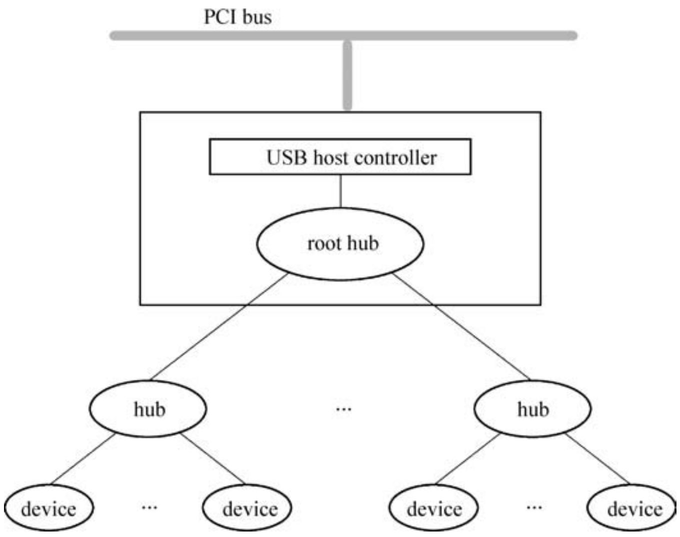


Figure 5.6 USB host controller, root hub, and hubs

- Open host Controller Interface (OHCI)
- Universal HCI
- Enhanced HCI.

With USB revision 1.0 and 1.1, there were two HCI specifications OHCI and UHCI. With the introduction of USB revision 2.0, EHCI was developed. The single integrated EHCI specification eliminates many of the problems that existed because of competing OHCI and UHCI standards. EHCI is used by all USB 2.0 devices and is the only HCI that supports high speed transfers.

USB encapsulates application data in several transactions. Each transaction consists of several packets. USB packets consist of the following fields:

- *Sync.* All packets must start with a sync field. The sync field is 8 bits long at low and full speed or 32 bits long for high speed and is used to synchronize the clock of the receiver with that of the transmitter. The last two bits indicate where the PID fields starts.
- *Pid.* PID stands for Packet ID. This field is used to identify the type of packet that is being sent. There are 4 bits to the PID, however to insure it is received correctly, the 4 bits are complemented and repeated, making an 8 bit PID in total. The resulting format is shown below.
- *Addr.* The address field specifies which device the packet is designated for. Being 7 bits in length allows for 127 devices to be supported. Address 0 is not valid, as any device which is not yet assigned an address must respond to packets sent to address zero.
- *Endp.* The endpoint field is made up of 4 bits, allowing 16 possible endpoints.
- *Crc.* Cyclic Redundancy Checks are performed on the data within the packet payload. All token packets have a 5 bit CRC while data packets have a 16 bit CRC.
- *Eop.* End of packet. Signaled by a Single Ended Zero (SE0) for approximately 2 bit times followed by a J for 1 bit time.

The packet format is shown in Figure 5.7.

A synchronization sequence is 00000001. Each packet consists of a packet ID, packet data, and a CRC field and so on.

5.4.3 USB Version History

USB 1.0 Released in January 1996 with specified data rates of 1.5 Mbit/s and 12 Mbit/s.

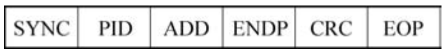


Figure 5.7 USB transmission packet format

USB 2.0 Released in April 2000 with the maximum bandwidth of 480 Mbit/s (60 MB/s). It is also call Hi-Speed USB.

USB 3.0 was released in November 2008 with a maximum transmission speed of up to 5 Gbit/s (625 MB/s), which is more than 10 times as fast as USB 2.0 (480 Mbit/s, or 60 MB/s), although this speed is typically only achieved using powerful professional grade or developmental equipment. USB 3.0 reduces the time required for data transmission, reduces power consumption, and is backward compatible with USB 2.0.

The USB 3.0 Promoter Group announced on 17 November 2008 that the specification of version 3.0 had been completed and had made the transition to the USB Implementers Forum (USB-IF), the managing body of USB specifications. This move effectively opened the specification to hardware developers for implementation in future products. A new feature is the "SuperSpeed" bus, which provides a fourth transfer mode at 5.0 Gbit/s. The raw throughput is 4 Gbit/s (using 8b/10b encoding), and the specification considers it reasonable to achieve around 3.2 Gbit/s (0.4 GB/s or 400 MB/s), increasing as hardware advances in the future take hold. Two-way communication is also possible.

In USB 3.0, full-duplex communications are done when using SuperSpeed (USB 3.0) transfer. In previous USB versions (i.e., 1.x or 2.0), all communication is half-duplex and directionally controlled by the host.

*5.4.4 USB Design and Architecture**

The design architecture of USB is asymmetrical in its topology, consisting of a host, a multitude of downstream USB ports, and multiple peripheral devices connected in a tiered-star topology. Additional USB hubs may be included in the tiers, allowing branching into a tree structure with up to five tier levels. A USB host may implement multiple host controllers and each host controller may provide one or more USB ports. Up to 127 devices, including hub devices if present, may be connected to a single host controller.

USB devices are linked in series through hubs. One hub is known as the root hub which is built into the host controller.

A physical USB device may consist of several logical sub-devices that are referred to as device functions. A single device may provide several functions, for example, a webcam (video device function) with a built-in microphone (audio device function). This kind of device is called composite device. An alternative for this is compound device in which each logical device is assigned a distinctive address by the host and all logical devices are connected to a built-in hub to which the physical USB wire is connected.

USB device communication is based on pipes (logical channels). A pipe is a connection from the host controller to a logical entity, found on a device, and named an endpoint. Because pipes correspond 1-to-1 to endpoints, the terms are sometimes used interchangeably. A USB device can have up to 32 endpoints: 16 into the host controller and 16 out of the host controller. The USB standard reserves one endpoint of each type, leaving a theoretical maximum of 30 for normal use. USB devices seldom have this many endpoints.

There are two types of pipes: stream and message pipes depending on the type of data transfer.

- *Isochronous transfers:* at some guaranteed data rate (often, but not necessarily, as fast as possible) but with possible data loss (e.g., realtime audio or video).
- *Interrupt transfers:* devices that need guaranteed quick responses (bounded latency) (e.g., pointing devices and keyboards).
- *Bulk transfers:* large sporadic transfers using all remaining available bandwidth, but with no guarantees on bandwidth or latency (e.g., file transfers).
- *Control transfers:* typically used for short, simple commands to the device, and a status response, used, for example, by the bus control pipe number 0.

A stream pipe is a uni-directional pipe connected to a uni-directional endpoint that transfers data using an isochronous, interrupt, or bulk transfer. A message pipe is a bi-directional pipe connected to a bi-directional endpoint that is exclusively used for control data flow. An endpoint is built into the USB device by the manufacturer and therefore exists permanently. An endpoint of a pipe is addressable with a tuple (device_address, endpoint_number) as specified in a TOKEN packet that the host sends when it wants to start a data transfer session. If the direction of the data transfer is from the host to the endpoint, an OUT packet (a specialization of a TOKEN packet) having the desired device address and endpoint number is sent by the host. If the direction of the data transfer is from the device to the host, the host sends an IN packet instead. If the destination endpoint is a uni-directional endpoint whose manufacturer's designated direction does not match the TOKEN packet (e.g., the manufacturer's designated direction is IN while the TOKEN packet is an OUT packet), the TOKEN packet will be ignored. Otherwise, it will be accepted and the data transaction can start. A bi-directional endpoint, on the other hand, accepts both IN and OUT packets.

Endpoints are grouped into interfaces and each interface is associated with a single device function. An exception to this is endpoint zero, which is used for device configuration and which is not associated with any interface. A single device function composed of independently controlled interfaces is called a composite device. A composite device only has a single device address because the host only assigns a device address to a function.

When a USB device is first connected to a USB host, the USB device enumeration process is started. The enumeration starts by sending a reset signal to the USB device. The data rate of the USB device is determined during the reset signaling. After reset, the USB device's information is read by the host and the device is assigned a unique 7-bit address. If the device is supported by the host, the device drivers needed for communicating with the device are loaded and the device is set to a configured state. If the USB host is restarted, the enumeration process is repeated for all connected devices.

The host controller directs traffic flow to devices, so no USB device can transfer any data on the bus without an explicit request from the host controller. In USB 2.0, the host controller polls the bus for traffic, usually in a round-robin fashion. The throughput of each USB port is determined by the slower speed of either the USB port or the USB device connected to the port.

High-speed USB 2.0 hubs contain devices called transaction translators that convert between high-speed USB 2.0 buses and full and low speed buses. When a high-speed USB 2.0 hub is plugged into a high-speed USB host or hub, it will operate in high-speed mode. The USB hub will then either use one transaction translator per hub to create a full/low-speed bus that is routed to all full and low speed devices on the hub, or will use one transaction translator per port to create an isolated full/low-speed bus per port on the hub.

Because there are two separate controllers in each USB 3.0 host, USB 3.0 devices will transmit and receive at USB 3.0 data rates regardless of USB 2.0 or earlier devices connected to that host. Operating data rates for them will be set in the legacy manner.

5.4.5 USB Mass Storage

USB implements connections to storage devices using a set of standards called the USB mass storage device class (MSC or UMS). This was at first intended for traditional magnetic and optical drives, but has been extended to support a wide variety of devices, particularly flash drives, because many systems can be controlled with the familiar metaphor of file manipulation within directories. The process of making a novel device look like a familiar device is also known as extension. The ability to boot a write-locked SD card with a USB adapter is particularly advantageous for maintaining the integrity and non-corruptible, pristine state of the booting medium.

Though most post-2005 computers are capable of booting from USB mass storage devices, USB is not intended to be a primary bus for a computer's internal storage: buses such as Parallel ATA (PATA or IDE), Serial ATA (SATA), or SCSI fulfill that role in PC class computers. However, USB has one important advantage in that it is possible to install and remove devices without rebooting the computer (hot-swapping), making it useful for mobile peripherals, including drives of various kinds. Originally conceived and still used today for optical storage devices (CD-RW drives, DVD drives and so on), several manufacturers offer external portable USB hard disk drives, or empty enclosures for disk drives, which offer performance comparable to internal drives, limited by the current number and type of attached USB devices and by the upper limit of the USB interface (in practice about 30 MB/s for USB 2.0 and potentially 400 MB/s or more or USB 3.0). These external drives have typically included a "translating device" that bridges between a drive's interface to a USB interface port. Functionally, the drive appears to the user much like an internal drive. Other competing standards for external drive connectivity include eSATA, Express-Card (now at version 2.0), and FireWire (IEEE 1394).

Another use for USB mass storage devices is the portable execution of software applications (such as Web browsers and VoIP clients) with no need to install them on the host computer.

Joysticks, keypads, tablets and other human-interface devices (HID) are also progressively migrating from MIDI, and PC game port connectors to USB. USB mice and keyboards can usually be used with older computers that have PS/2 connectors with the aid of a small USB-to-PS/2 adapter. Such adaptors contain no logic circuitry: the hardware in the USB keyboard or mouse is designed to detect whether it is connected to a USB or PS/2 port, and communicate using the appropriate protocol. Converters also exist to allow PS/2 keyboards and mice (usually one of each) to be connected to a USB port. These devices present two HID endpoints to the system and use a microcontroller to perform bidirectional translation of data between the two standards.

5.4.6 USB Interface Connectors

The connectors specified by the USB committee were designed to support a number of USBs underlying goals, and to reflect lessons learned from the menagerie of connectors which have been used in the computer industry. The connector mounted on the host or device is called the receptacle, and the connector attached to the cable is called the plug. In the case of an extension cable, the connector on one end is a receptacle. The official USB specification documents periodically define the term male to represent the plug, and female to represent the receptacle. Figure 5.8 shows some common USB connectors.

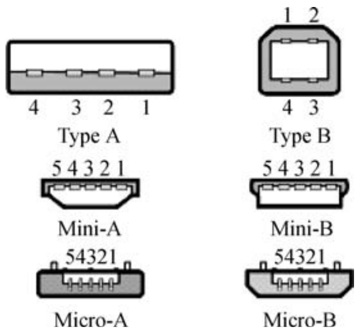


Figure 5.8 USB connectors

By design, it is difficult to attach a USB connector incorrectly. Connectors cannot be plugged in upside down and it is clear from the physical act of making a connection, when the plug and receptacle are correctly mated. The USB specification states that the required USB Icon is to be “embossed” on the “topside of the USB plug, which “provides easy user recognition and facilitates alignment during the mating process.” The specification also shows that the “recommended” (optional) “Manufacturer’s logo” (“engraved” on the diagram but not specified in the text) is on the opposite side of the USB Icon. The specification further states “the USB Icon is also located adjacent to each receptacle. Receptacles should be oriented to allow the icon on the plug to be visible during the mating process.” However, the specification does not consider the height of the device compared to the eye level height of the user, so the side of the cable that is “visible” when mated to a computer on a desk can depend on whether the user is standing or kneeling.

Only moderate insertion/removal force is needed. USB cables and small USB devices are held in place by the gripping force from the receptacle (without need of the screws, clips, or thumb-turns other connectors have required). The force needed to make or break a connection is modest, allowing connections to be made in awkward circumstances (i.e., behind a floor-mounted chassis, or from below) or by those with motor disabilities.

The standard connectors were deliberately intended to enforce the directed topology of a USB network: type A connectors on host devices that supply power and type B connectors on target devices that receive power. This prevents users from accidentally connecting two USB power supplies to each other, which could lead to dangerously high currents, circuit failures, or even fire. USB does not support cyclical networks and the standard connectors from incompatible USB devices are themselves incompatible. Unlike other communications systems (e.g., network cabling) gender changers make little sense with USB and are almost never used.

The standard connectors were designed to be robust. Many previous connector designs were fragile, specifying embedded component pins or other delicate parts which proved vulnerable to bending or breakage, even with the application of modest force. The electrical contacts in a USB connector are protected by an adjacent plastic tongue, and the entire connecting assembly is usually protected by an enclosing metal sheath.

The connector construction always ensures that the external sheath on the plug makes contact with its counterpart in the receptacle before any of the four connectors within make electrical contact. The external metallic sheath is typically connected to system ground, thus dissipating damaging static charges. This enclosure design also provides a degree of protection from electromagnetic interference to the USB signal while it travels through the mated connector pair (the only location when the otherwise twisted data pair travels in parallel). In addition, because of the required sizes of the power and common connections, they are made after the system ground but before the data connections. This type of staged make-break timing allows for electrically safe hot-swapping.

The newer Micro-USB receptacles are designed for up to 10,000 cycles of insertion and removal between the receptacle and plug, compared to 1,500 for the standard USB and 5,000 for the Mini-USB receptacle. This is accomplished by adding a locking device and by moving the leaf-spring connector from the jack to the plug, so that the most-stressed part is on the cable side of the connection. This change was made so that the connector on the less expensive cable would bear the most wear instead of the more expensive micro-USB device.

The USB standard specifies relatively loose tolerances for compliant USB connectors to minimize physical incompatibilities in connectors from different vendors. To address a weakness present in some other connector standards, the USB specification also defines limits to the size of a connecting device in the area around its plug. This was done to prevent a device from blocking adjacent ports due to the size of the cable strain relief mechanism (usually molding integral with the cable outer insulation) at the connector. Compliant devices must either fit within the size restrictions or support a compliant extension cable which does.

In general, cables have only plugs (very few have a receptacle on one end, although extension cables with a standard A plug and jack are sold), and hosts and devices have only receptacles. Hosts almost universally have type-A receptacles, and devices of one or another type-B variety. Type-A plugs mate only with type-A receptacles, and type-B with type-B; they are deliberately physically incompatible. However, an extension to USB standard specification called USB On-The-Go allows a single port to act as either a host or a device – chosen by which end of the cable plugs into the receptacle on the unit. Even after the cable is hooked up and the units are communicating, the two units may "swap" ends under program control. This capability is meant for units such as PDAs in which the USB link might connect to a PCs host port as a device in one instance, yet connect as a host itself to a keyboard and mouse device in another instance.

USB 3.0 receptacles are electrically compatible with USB Standard 2.0 device plugs if they physically match. USB 3.0 type-A plugs and receptacles are completely backward compatible, and USB 3.0 type-B receptacles will accept USB 2.0 and earlier plugs. However, USB 3.0 type-B plugs will not fit into USB 2.0 and earlier receptacles. eSATAp (eSATA/USB) port is also compatible with USB 2.0 devices.

5.4.7 USB Connector Types

There are several types of USB connectors, including some that have been added while the specification progressed. The original USB specification detailed Standard-A and Standard-B plugs and receptacles. The first engineering change notice to the USB 2.0 specification added Mini-B plugs and receptacles.

The data connectors in the Standard-A plug are actually recessed in the plug as compared to the outside power connectors. This permits the power to connect first which prevents data errors by allowing the device to power up first and then transfer the data. Some devices will operate in different modes depending on whether the data connection is made. This difference in connection can be exploited by inserting the connector only partially. For example, some battery-powered MP3 players switch into file transfer mode and cannot play MP3 files while a USB plug is fully inserted, but can be operated in MP3 playback mode using USB power by inserting the plug only part way so that the power slots make contact while the data slots do not. This enables those devices to be operated in MP3 playback mode while getting power from the cable.

To reliably enable a charge-only feature, modern USB accessory peripherals now include charging cables that provide power connections to the host port but no data connections, and both home and vehicle charging docks are available that supply power from a converter device and do not include a host device and data pins, allowing any capable USB device to be charged or operated from a standard USB cable.

The USB 2.0 Standard-A type of USB plug is a flattened rectangle which inserts into a “downstream-port” receptacle on the USB host, or a hub, and carries both power and data. This plug is frequently seen on cables that are permanently attached to a device, such as one connecting a keyboard or mouse to the computer via USB connection.

USB connections eventually wear out as the connection loosens through repeated plugging and unplugging. The lifetime of a USB-A male connector is approximately 1,500 connect/disconnect cycles.

A Standard-B plug – which has a square shape with beveled exterior corners – typically plugs into an “upstream receptacle” on a device that uses a removable cable, for example, a printer. A Type B plug delivers power in addition to carrying data. On some devices, the Type B receptacle has no data connections, being used solely for accepting power from the upstream device. This two-connector-type scheme (A/B) prevents a user from accidentally creating an electrical loop. Figure 5.9 shows standard A/B connector.

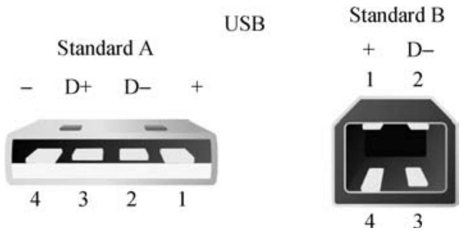


Figure 5.9 USB standard A/B connector

Various mini and micro connectors have been used for smaller devices such as PDAs, mobile phones or digital cameras. These include the now-deprecated (but standardized) Mini-A and the currently standard Mini-B, Micro-A, and Micro-B connectors.

The Mini-A and Mini-B plugs are approximately 3 by 7 mm. The micro-USB plugs have a similar width and approximately half the thickness, enabling their integration into thinner portable devices. The micro-A connector is 6.85 by 1.8 mm with a maximum overmold size of 11.7 by 8.5 mm. The micro-B connector is 6.85 by 1.8 mm with a maximum overmold size of 10.6 by 8.5 mm.

The Micro-USB connector was announced by the USB-IF in January 2007. The Mini-A connector and the Mini-AB receptacle connector were deprecated in May 2007. While many currently available devices and cables still use Mini plugs, the newer Micro connectors are being widely adopted and as of December 2010, they are the most widely used. The thinner micro connectors are intended to replace the Mini plugs in new devices including smartphones and personal digital assistants. The Micro plug design is rated for at least 10,000 connect-disconnect cycles which is significantly more than the Mini plug design. The Universal Serial Bus Micro-USB Cables and Connectors Specification details the mechanical characteristics of Micro-A plugs, Micro-AB receptacles, and Micro-B plugs and receptacles, along with a Standard-A receptacle to Micro-A plug adapter.

The cellular phone carrier group, Open Mobile Terminal Platform (OMTP) in 2007 have endorsed Micro-USB as the standard connector for data and power on mobile devices.

These include various types of battery chargers, allowing Micro-USB to be the single external cable link needed by some devices.

As of January 2009 Micro-USB has been accepted and is being used by almost all cell phone manufacturers as the standard charging port (including Hewlett-Packard, HTC, LG, Motorola, Nokia, Research In Motion, Samsung, Sony Ericsson) in most of the world.

On 29 June 2009, following a request from the European Commission and in close co-operation with the Commission services, major producers of mobile phones have agreed in a Memorandum of Understanding ("MoU") to harmonize chargers for data-enabled mobile phones sold in the European Union. Industry commits to provide charger compatibility on the basis of the Micro-USB connector. Consumers will be able to purchase mobile phones without a charger, thus logically reducing their cost. Following a mandate from the European Commission, the European Standardization Bodies CEN-CENELEC and ETSI have now made available the harmonized standards needed for the manufacture of data-enabled mobile phones compatible with the new common External Power Supply (EPS) based on micro-USB.

In addition, in October 2009 the International Telecommunication Union (ITU) has also announced that it had embraced micro-USB as the Universal Charger Solution its "energy-efficient one-charger-fits-all new mobile phone solution," and added: “Based on the Micro-USB interface, UCS chargers will also include a 4-star or higher efficiency rating – up to three times more energy-efficient than an unrated charger.”

A USB On-The-Go device is required to have one, and only one USB connector: a Mini-AB or Micro-AB receptacle. This receptacle is capable of accepting both Mini-A and Mini-B plugs, and alternatively, Micro-A and Micro-B plugs, attached to any of the legal cables and adapters as defined in Micro-USB1.01.

The OTG device with the A-plug inserted is called the A-device and is responsible for powering the USB interface when required and by default assumes the role of host. The OTG device with the B-plug inserted is called the B-device and by default assumes the role of peripheral. An OTG device with no plug inserted defaults to acting as a B-device. If an application on the B-device requires the role of host, then the HNP protocol is used to temporarily transfer the host role to the B-device.

OTG devices attached either to a peripheral-only B-device or a standard/embedded host will have their role fixed by the cable since in these scenarios it is only possible to attach the cable one way around.

5.4.8 USB Power and Charging

The USB 1.x and 2.0 specifications provide a 5 V supply on a single wire from which connected USB devices may draw power. The specification provides for no more than 5.25 V and no less than 4.75 V (5 V $\pm$ 5%) between the positive and negative bus power lines. For USB 3.0, the voltage supplied by low-powered hub ports is 4.45–5.25 V.

A unit load is defined as 100 mA in USB 2.0, and 150 mA in USB 3.0. A device may draw a maximum of 5 unit loads (500 mA) from a port in USB 2.0; 6 (900 mA) in USB 3.0. There are two types of devices: low-power and high-power. A low-power device draws at most 1 unit load, with minimum operating voltage of 4.4 V in USB 2.0, and 4 V in USB 3.0. A high-power device draws the maximum number of unit loads permitted by the standard. Every device functions initially as low-power but the device may request high-power and will get it if the power is available on the providing bus.

Some devices, such as high-speed external disk drives, require more than 500 mA of current and therefore cannot be powered from one USB 2.0 port. Such devices usually come with Y-shaped cable that has two USB connectors to be plugged into a computer. With such a cable, a device can draw power from two USB ports simultaneously.

A bus-powered hub initializes itself at 1 unit load and transitions to maximum unit loads after it completes hub configuration. Any device connected to the hub will draw 1 unit load regardless of the current draw of devices connected to other ports of the hub (i.e., one device connected on a four-port hub will draw only 1 unit load despite the fact that more unit loads are being supplied to the hub).

A self-powered hub will supply maximum supported unit loads to any device connected to it. In addition, the VBUS will present 1 unit load upstream for communication if parts of the Hub are powered down.

The Battery Charging Specification of 2007 defines new types of USB ports, for example, charging ports. As compared to standard downstream ports, where a portable device can only draw more than 100 mA current after digital negotiation with the host or hub, charging ports can supply currents above 0.5 A without digital negotiation. A charging port supplies up to 500 mA at 5 V, up to the rated current at 3.6 V or more, and drops its output voltage if the portable device attempts to draw more than the rated current. The charger port may shut down if the load is too high.

Charging ports exist in two flavors: charging downstream ports (CDP), supporting data transfers as well, and dedicated charging ports (DCP), without data support. A portable device can recognize the type of USB port from the way the D+ and D− pins are connected. For example, on a dedicated charging port, the D+ and D− pins are shorted. With charging downstream ports, current passing through the thin ground wire may interfere with high-speed data signals. Therefore, current draw may not exceed 900 mA during high-speed data transfer. A dedicated charge port may have a rated current between 0.5 and 1.5 A. There is no upper limit for the rated current of a charging downstream port, as long as the connector can handle the current (standard USB 2.0 A-connectors are rated at 1.5 A).

Before the battery charging specification was defined, there was no standardized way for the portable device to inquire how much current was available. For example, Apple's iPod and iPhone chargers indicate the available current by voltages on the D− and D+ lines. When $D+ = D- = 2$ V, the device may pull up to 500 mA. When $D+ = 2.0$ V and $D - = 2.8$ V, the device may pull up to 1,000 mA of current.

Dedicated charging ports can be found on USB power adapters that convert utility power or another power source – for example, a car's electrical system – to run attached devices and battery packs. On a host (such as a laptop computer) with both standard and charging USB ports, the charging ports should be labeled as such.

To support simultaneous charge and sync, even if the communication port doesn't support charging a demanding device, so called accessory charging adapters are introduced, where a charging port and a communication port can be combined into a single port.

The Battery Charging Specification 1.2 of 2010 makes clear that there are safety limits to the rated current at 5 A coming from USB 2.0. On the other hand several changes are made and limits are increasing including allowing 1.5 A on charging ports for unconfigured devices, allowing high speed communication while having a current up to 1.5 A and allowing a maximum current of 5 A.

Sleep-and-charge USB ports can be used to charge electronic devices even when the computer is switched off. Normally when a computer is powered off the USB ports are powered down. This prevents phones and other devices from being able to charge unless the computer is powered on. Sleep-and-charge USB ports remain powered even when the computer is off. On laptops, charging devices from the USB port when it is not being powered from AC will drain the laptop battery faster. Desktop machines need to remain plugged into AC power for Sleep-and-charge to work.

As of June 2007, all new mobile phones applying for a license in China are required to use the USB port as a power port. This was the first standard to use the convention of shorting D+ and D − .

In September 2007, the Open Mobile Terminal Platform group (a forum of mobile network operators and manufacturers such as Nokia, Samsung, Motorola, Sony Ericsson and LG) announced that its members had agreed on micro-USB as the future common connector for mobile devices.

On 17 February 2009, the GSM Association (GSMA) announced that they had agreed on a standard charger for mobile phones. The standard connector to be adopted by 17 manufacturers including Nokia, Motorola and Samsung is to be the micro-USB connector (several media reports erroneously reported this as the mini-USB). The new chargers will be much more efficient than existing chargers. Having a standard charger for all phones means that manufacturers will no longer have to supply a charger with every new phone. The basis of the GSMAs Universal Charger Solution (UCS) is the technical recommendation from OMTP and the USB-IF battery charging standard.

On 22 April 2009, this was further endorsed by the CTIA – The Wireless Association.

In June 2009, many of the world's largest mobile phone manufacturers signed a Memorandum of Understanding (MoU), agreeing to make most data-enabled mobile phones marketed in the European Union compatible with a common External Power Supply (EPS) based on the GSMA/OMTP Universal Charging Solution.

On 22 October 2009, the International Telecommunication Union (ITU) announced that it had embraced the Universal Charger Solution as its "energy-efficient one-charger-fits-all new mobile phone solution," and added: "Based on the Micro-USB interface, UCS chargers will also include a 4-star or higher efficiency rating – up to three times more energy-efficient than an unrated charger."

Some USB devices require more power than is permitted by the specifications for a single port. This is common for external hard and optical disc drives, and generally for devices with motors or lamps. Such devices can use an external power supply, which is allowed by the standard, or use a dual-input USB cable, one input of which is used for power and data transfer, the other solely for power, which makes the device a non-standard USB device. Some USB ports and external hubs can, in practice, supply more power to USB devices than required by the specification but a standard-compliant device may not depend on this.

In addition to limiting the total average power used by the device, the USB specification limits the inrush current (i.e., that used to charge decoupling and filter capacitors) when the device is first connected. Otherwise, connecting a device could cause problems with the host's internal power. USB devices are also required to automatically enter ultra low-power suspend mode when the USB host is suspended. Nevertheless, many USB host interfaces do not cut off the power supply to USB devices when they are suspended.

Some non-standard USB devices use the 5 V power supply without participating in a proper USB network which negotiates power draws with the host interface. These are usually referred to as USB decorations. The typical example is a USB-powered keyboard light; fans, mug coolers and heaters, battery chargers, miniature vacuum cleaners, and even miniature lava lamps are available. In most cases, these items contain no digital circuitry, and thus are not Standard compliant USB devices at all. This can theoretically cause problems with some computers, such as drawing too much current and damaging circuitry; prior to the Battery Charging Specification, the USB specification required that devices connect in a low-power mode (100 mA maximum) and communicate their current requirements to the host, which would then permit the device to switch into high-power mode.

Some devices, when plugged into charging ports, draw even more power (10 watts or 2.1 Amps) than the Battery Charging Specification allows. The iPad and MiFi 2200 are two such devices. Barnes & Noble NOOK devices also require a special charger that runs at 1.9 Amps.

Powered USB uses standard USB signaling with the addition of extra power lines. It uses four additional pins to supply up to 6 A at either 5 V, 12 V, or 24 V (depending on keying) to peripheral devices. The wires and contacts on the USB portion have been upgraded to support higher current on the 5 V line, as well. This is commonly used in retail systems and provides enough power to operate stationary barcode scanners, printers, PIN pads, signature capture devices, and so on. This modification of the USB interface is proprietary and was developed by IBM, NCR, and FCI/Berg. It is essentially two connectors stacked such that the bottom connector accepts a standard USB plug and the top connector takes a power connector.

5.4.9 IEEE 1394

Apple was among the earliest one to develop this high-speed peripheral interface. Firewire is another name (trademark) of Apple. Before USB 3.0 was introduced, Firewire was faster than USB. It can be seen in many earlier digital video products. Now USB 3.0 supersedes Firewire in speed.

Same as USB, 1394 can be hot attached or removed without requiring shutting down the computer.

The power distribution for a 1394 port is much higher than the USB. The voltage can range between 8–33 V, and the current can be up to 1.5 Amps.

5.5 Network Interface Card

A NIC is a computer hardware component that connects a computer to a computer network.

A NIC implements the electronic circuitry required to communicate using a specific physical layer and data link layer standard such as Ethernet or Wi-Fi. This provides a base for a full network protocol stack, allowing communication among small groups of computers on the same LAN and large-scale network communications through routable protocols, such as IP.

Although other network technologies exist, Ethernet has achieved near-ubiquity since the mid-1990s.

Every Ethernet network controller has a unique 48-bit serial number called a MAC address, which is stored in read-only memory carried on the card for add-on cards. Every computer on an Ethernet network must have at least one controller. Each controller must have a unique MAC address. Normally it is safe to assume that no two network controllers will share the same address, because controller vendors purchase blocks of addresses from the Institute of Electrical and Electronics Engineers (IEEE) and assign a unique address to each controller at the time of manufacture.

Ethernet network controllers typically support 10 Mbit/s Ethernet, 100 Mbit/s Ethernet, and 1,000 Mbit/s Ethernet varieties. Such controllers are designated 10/100/1000 and this means they can support a notional maximum transfer rate of 10, 100 or 1000 Megabits per second (Hwang, 1993).

5.5.1 Basic NIC Architecture

The NIC allows computers to communicate over a computer network. It is both an OSI layer 1 (physical layer) and layer 2 (data link layer) device, as it provides physical access to a networking medium and provides a low-level addressing system through the use of MAC addresses. It allows users to connect to each other either by using cables or wirelessly. Figure 5.10 shows the block diagram of NIC architecture.

Most NICs have a DMA interface unit, a medium access control (MAC) unit, memory, and control logic. A block diagram of NIC architecture is shown in Figure 5.8. In the case of a programmable NIC, the control logic is one or more programmable processors that run compiled-code firmware. The DMA unit is

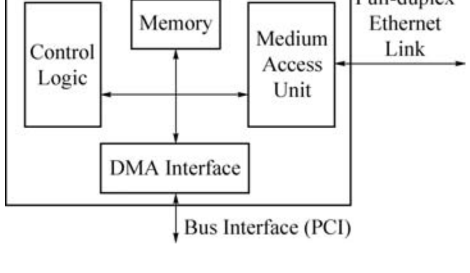


Figure 5.10 NIC architecture block diagram

directed by the onboard control logic to read and write data between the local NIC memory and the host's memory. The medium access unit interacts with the control logic to receive frames into local buffer storage and to send frames from local buffer storage out onto the network. The memory is used for temporary storage of frames, buffer descriptors, and other control data.

5.5.2 Data Transmission

NICs facilitate the transmission and receipt of data (frames) between the network and host operating system. Sending and receiving frames happens in a series of steps completed by the host and NIC.

Typically the host OS maintains a series of buffers, which are used for frame headers and contents. The OS also maintains a queue or ring of buffer descriptors. Each buffer descriptor indicates where in host memory the buffer resides and how big the buffer is.

5.5.2.1 Send

The steps of sending a single Ethernet frame are listed below. A diagram of sending process is shown in Figure 5.11.

1. The host operating system is informed that a frame is in host memory and is ready to be sent. The OS builds a buffer descriptor regarding this frame in host memory.
2. The OS notifies the NIC that a new buffer descriptor is in host memory and is ready to be fetched and processed. This is typically referred to as a "mailbox" event.
3. The NIC initiates a direct memory access (DMA) read of the pending buffer descriptor and processes it.

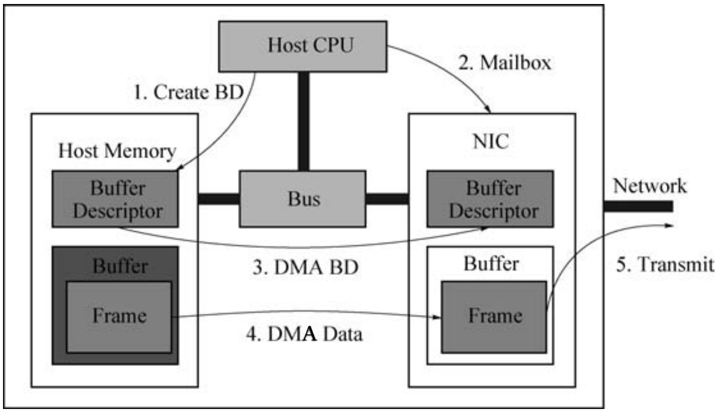


Figure 5.11 NIC send

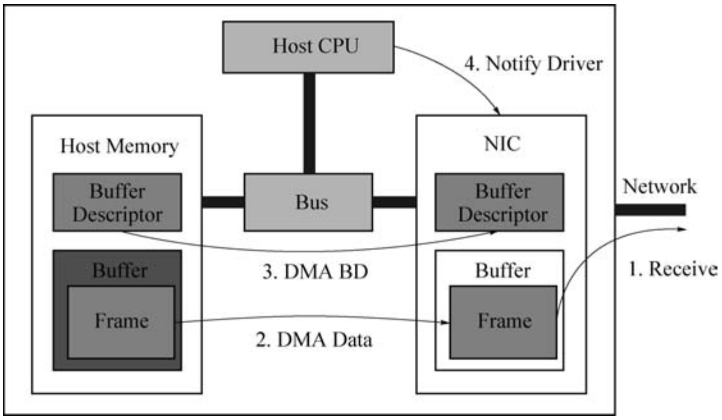


Figure 5.12 NIC receive

4. Having determined the host address of the pending frame, the NIC initiates a DMA read of the frame contents.
5. When all segments of the frame (which may use several buffer descriptors and buffers) have arrived, the NIC transmits the frame out onto the Ethernet.
6. Depending on how the OS has configured the NIC, the NIC may interrupt the host to indicate that the frame has completed.

5.5.2.2 Receive

The steps of receiving a single Ethernet frame are listed below. A diagram of receiving process is shown in Figure 5.12.

1. The NIC receives a frame from the network into its local receive buffer.
2. Presuming there is enough host memory available for this received frame, the NIC initiates a DMA write of the frame contents into host memory. The NIC determines what the starting host address is by examining the next free buffer descriptor (which it has previously fetched).
3. The NIC modifies the previously fetched buffer descriptor regarding the space that the new frame now occupies; the NIC fills in the frame length and possibly checksum information. After modifying this buffer descriptor, the NIC initiates a DMA write of it to the host.
4. Depending on how the OS has configured the NIC, the NIC may interrupt the host to indicate that a frame has arrived.

5.6 Keyboard, Video and Mouse (KVM) Interfaces

A computer usually has a keyboard, a mouse and one or more video displays. A keyboard and a mouse can be either connected directly to the computer, integrated into laptop computers or connected to a computer wirelessly. Depending on the graphic card, a computer can attach one display, two displays or even more. Usually laptops can connect an external display and the resolution on the external display is higher. Due to the high transmission rate requirement, monitors cannot be wirelessly connected to a computer system.

5.6.1 Keyboards

A keyboard contains a keyboard controller that scans the keyboard and reports key depressions and releases. A scan code is assigned to each key with a unique number. It has nothing to do with the ASCII code of the character.

5.6.2 Video Graphic Card

A video card is an expansion card that allows the computer to send graphical information to a video display device such as a monitor or projector.

Many modern computers do not have video expansion cards but instead have GPUs integrated directly onto the motherboard (Shiva, 2000). This allows for a less expensive computer but also for a less powerful graphics system. This option is wise for the average business and home user not interested in advanced graphics capabilities or the latest games.

The Digital Visual Interface (DVI) is a video interface standard covering the transmission of video between a computer and a display device. Most contemporary desktop PCs and LCD monitors feature a DVI interface, and many other devices (such as projectors and televisions) support DVI indirectly through HDMI, another video interface standard. Most laptops still have legacy VGA and, in some models, HDMI ports. DVI is designed to carry uncompressed digital video data to a display.

5.6.3 Mouses

In computing, a mouse is a pointing device that functions by detecting two-dimensional motion relative to its supporting surface. Some mouses provide three-dimensional information. It is usually seen in computer gaming and other areas that require three-dimensional coordinates. Physically, a mouse consists of an object held under one of the user's hands, with one or more buttons. It sometimes features other elements, such as "wheels," which allow the user to perform various system-dependent operations, or extra buttons or features that can add more control or dimensional input. The mouse's motion typically translates into the motion of a cursor on a display, which allows for fine control of a graphical user interface.

5.7 Input/Output Security

I/O security ranges from physical security, operation security and application security and so on. Security threats can affect all input and output devices. An unlocked keyboard is an example of physical security. An un-encrypted hard drive may expose sensitive data to unauthorized persons.

5.7.1 Disable Certain Key Combinations

Keyboards should always be locked if not in use. For a server, you may want to disable the ctrl-alt-del combination. An attacker may shut down a server by pressing the combination keys if it is not properly protected.

5.7.2 Anti-Glare Displays

If you are a frequent traveler, you may want to add an anti-glare screen filter so that nobody other than yourself can read the content on a computer display.

5.7.3 Adding Password to Printer

Printers can be protected by adding passwords so that only authorized users can print. Once the printing job is done, the documents should be collected immediately.

5.7.4 Bootable USB Ports

USB ports provide convenience to computer users. However, many USB flash drives contain a small operating system which is bootable. A bootable flash drive can start a virtual machine on top of the computer system. From the virtual machine, the attacker can easily read data stored on the host machine.

To prevent such attacks, all USB ports should be set not to bootable. Unused USB ports should be disabled.

5.7.5 Encrypting Hard Drives

Hard drives are also vulnerable to attackers. Sensitive data should be stored separately and encrypted. This will prevent any person to view the data by taking off the hard drive from the current computer and attach to another one.

For security related to network, there will be further discussions in later chapters.

5.8 Summary

In this chapter, we studied different kinds input and output devices and network interfaces. I/O devices contain several internal registers. The registers are used to receive command and data from the process. With DMA technology, data transmission between I/O devices and memory can be done directly without requiring CPU resources. To solve the low speed problem which in some cases may slow down the CPU operations, we use Interrupts so that an I/O device will not occupy CPU until the I/O job is done or its buffer (I/O register) is full.

USB is a common I/O interface and is becoming a standard interface on many I/O devices. The USB 3.0 is capable of transmitting and receiving at a speed of 400 MB/s.

The network interface card is a special type I/O device that manages the network data communication. Every NIC comes with a MAC address, a unique address to identify the device on the Internet. Some people have reservations to classify the network interface as an I/O device as it is so important and so different than regular I/O devices such as keyboards, mouses, displays and printers.

KVM are essential I/O devices for personal computers. On handheld devices such as smart phones and personal data assistants, gesture based input has become common. For example, sweeping your finger on an iPhone or iPad can turn the page.

Security problems for classical I/Os are not hard to manage. However network related security is a very broad topic that has drawn much attention and it is still constantly evolving. Chapter 9 will discuss this topic in details.

Exercises

5.1 What is DMA? Why DMA can reduce CPU work load?

5.2 When an interrupt happen, CPU will respond to the interrupt and execute the interrupt handling program. How does CPU know to resume current task?

5.3 A PC (=FE17) points to the top of the stack. The bottom of the stack is at address FFFF. After execute instructions: PUSH, PUSH and POP, what is the value in the PC?

5.4 A simple character printer could use programmed I/O reasonably well, since the printer speed is slow compared to the CPU. Yet most modern printers use DMA, Why?

5.5 What is a USB root hub? How many USB devices can be connected to a root hub?

5.6 A mouse with illuminate light was attached to a light weight laptop computer. However the mouse does not function. When plugging the same mouse to a desktop computer, it works perfectly. What could be the reason that caused the problem?

5.7 A bootable USB flash drive can setup a virtual machine on top of the host computer. On the virtual machine, a user may be able to access the files and data on the host computer. This is a convenient feather but sometimes it may cause threats to the security. Describe how to effectively address this problem?

5.8 A KVM device can allow two or more computers to share one display, one keyboard and one mouse. In earlier ones, people had to switch manually. The newer models allow you to switch with key combinations or mouse clicks. Draw a conceptual diagram of such a design (hint: PIO).

5.9 Data on a hard drive are protected by a password that is managed by the operating system. Describe the flaws of this type of "protection" technique.

5.10 At least one MAC address exists on a NIC card. What is the MAC address used for?

5.11 Describe the sending and receiving processes of a NIC card.

References

Dumas, J. D. (2006) *Computer Architecture: Fundamentals and Principles of Computer Design*, Taylor & Francis.

Foster, C. C. and Iberall, T. (1985) *Computer Architecture*, 3rd edn, Wan Nostrand Reinhold Company, New York.

Hwang, K. (1993) *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, Inc.

Randell, B. (ed.) (1982) *The Designs of Digital Computers*, Springer-Verlag.

Shiva, S. G. (2000) *Computer Design and Architecture*, 3rd edn, Marcel Dekker, Inc., New York.

6

Central Processing Unit

Central Processing Unit (CPU) is the brain of a computer system. A computer can be classified as a super computer, a mainframe computer, a minicomputer, a microcomputer, or a handheld computer. Each of the above mentioned computers may contain one or more processors. Usually enterprise computers (servers) contain more processors than personal computers. A CPU has a certain number of address, data, and control bus signals. The number of data bits that can be processed at one time is called a word. Unlike byte which is a fixed number equaled to 8 bits. The word size differs from computers to computers. For example, the word for a 32-bit computer is $32/8 = 4$ bytes, while the word for a 64-bit computer is $64/8 = 8$ bytes.

6.1 The Instruction Set

The instruction set provides an interface that allows users to make full use of the processor. The selection of the instructions is determined by the application for which the processor is intended. For example, a computational intensive computer may choose processors that are built with larger word size, more math functions and lots of memory management features. On the other hand, a processor designed for embedded systems such as TV remote controls and air conditioners may only contain less or simple instructions, small word size and easy addressing features.

6.1.1 Instruction Classifications

Instructions usually contain two components: operation code (**opcode**) and **operands**. An instruction can have one opcode and zero or more operands. Some common instruction formats are shown in Figure 6.1.

The operands are located either in registers (fast speed) or in a memory location. Table 6.1 lists some common classes of instructions.

The logic instructions and arithmetic instructions are a subset of the memory and register reference instructions.

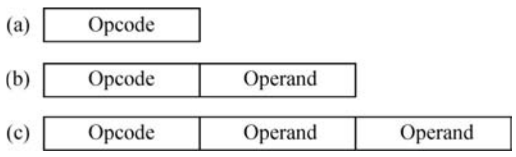


Figure 6.1 Opcode and operands

6.1.2 Logic Instructions

All processors have logical operation instructions. The common logical operations are AND, OR, NOT and Exclusive OR (XOR). AND needs both inputs equal to 1 in order to output 1. OR needs one or more input equal to 1 in order to output 1. In other words, OR output 0 iff (if and only if) both inputs are 0. NOT operation inverts the input. XOR is just like adding both inputs and discarding the carry (if there is one). So if two inputs are different then output 1 otherwise 0. Figure 6.2 shows the examples of the logical operation – XOR.

6.1.3 Arithmetic Instructions

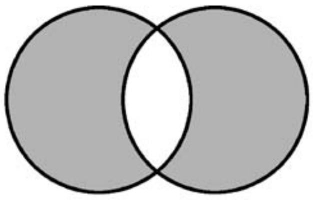
Arithmetic instructions include ADD, SUB, MUL, DIV, and so on.

The implementation of the arithmetic instructions varies from processor to processor. It mostly depends on the length of the instruction. Based on the number of operands, the instructions can be organized as follows:

- Three addresses (operands)
- Two addresses
- One address
- Zero address.

Table 6.1 Common classes of instructions

Memory to memory instructions	*Both operands are in memory*
Memory to register instructions	*One of the operand is in memory, another one is in register*
Register reference instructions	*The operation is performed on the contents of one or more registers*
Memory reference instructions	*The operation is performed on the contents of memory location.*
Control instructions	*Branching, halt, pause, and so on.*
Input/output instructions	*Input or output*
Macroinstructions	*A set of instructions*



$$A \oplus B = (A \wedge \neg B) \vee (B \wedge \neg A) = A\overline{B} + B\overline{A}$$

Figure 6.2 Logical operation – XOR

For a three-address machine, there are the two operands and the results are stored separately. Two-address machine will reuse one address as one of the operand and later holds the result. One address machine usually uses the accumulator (ACC) with one address (commonly a register) to perform the execution.

The word “address” here may refer to a register, a memory unit or both. Since registers are much faster than memory, instructions that use registers as operands are faster than those that use memory. Common processors use registers to perform arithmetic operations. Some imbedded chips use memory to do so.

Here are some examples of instructions with different operands

1. Three-address machine.

   $$ADD \quad A, B, C \qquad C \leftarrow A + B$$

   Each of the instructions needs three registers or memory access during each execution. In practice, the majority of operations are based on two operands with the result occupying the position of one of the operand. Therefore a two-address instruction may be good for such processors.
2. Two-address machine.

   $$ADD \quad A, B \qquad A \leftarrow A + B$$

   Two operands instruction are also common in processors. In addition to fetch-execution cycle, it adds a write back cycle to store the result to the location of one of the operands.
3. One-address machine.

   $$ADD \quad B \qquad ACC \leftarrow ACC + B$$

   A one-address processor is similar to a two-address processor except it uses the accumulator (ACC), a register in CPU, by default.

The addressing modes for operands can be direct, indirect or register. Direct addressing means the operand is a number. The indirect addressing means the operand is an address that points to a memory location. The register addressing means the operand is stored in a register.

- Direct addressing: $ADD \quad A, 3 \qquad A \leftarrow A + 3$
- Indirect addressing: $ADD \quad A, Pi \qquad A \leftarrow A + Pi$

In indirect addressing, the operand Pi (π) is a label or pointer pointing to the memory location of the actual number.

6.1.4 Intel 64/32 Instructions[*]

The data transfer instructions move data between memory and the general-purpose and segment registers. They also perform specific operations such as conditional moves, stack access, and data conversion.

6.1.4.1 Move Instructions

The MOV (move) instructions transfer data between memory and registers or between registers.

The MOV instruction performs basic load data and store data operations between memory and the processor's registers and data movement operations between registers.

The MOV instruction cannot move data from one memory location to another or from one segment register to another segment register. Memory-to-memory moves are performed with the MOVS (string move) instruction.

6.1.4.2 Stack Manipulation Instructions

The PUSH, POP, PUSHA (push all registers), and POPA (pop all registers) instructions move data to and from the stack. The PUSH instruction decrements the stack pointer (contained in the ESP register), then copies the source operand to the top of the stack. It operates on memory operands, immediate operands, and register operands (including segment registers). The PUSH instruction is commonly used to place parameters on the stack before calling a procedure. It can also be used to reserve space on the stack for temporary variables.

The PUSHA instruction saves the contents of the eight general-purpose registers on the stack. This instruction simplifies procedure calls by reducing the number of instructions required to save the contents of the general-purpose registers.

The registers are pushed on the stack in the following order: EAX, ECX, EDX, EBX, the initial value of ESP before EAX was pushed, EBP, ESI, and EDI.

The POP instruction copies the word or doubleword at the current top of the stack (indicated by the ESP register) to the location specified with the destination operand. It then increments the ESP register to point to the new top of the stack.

The destination operand may specify a general-purpose register, a segment register, or a memory location.

The POPA instruction reverses the effect of the PUSHA instruction. It pops the top eight words or doublewords from the top of the stack into the general-purpose registers, except for the ESP register. If the operand-size attribute is 32, the doublewords on the stack are transferred to the registers in the following order: EDI, ESI, EBP, ignore doubleword, EBX, EDX, ECX, and EAX. The ESP register is restored by the action of popping the stack. If the operand-size attribute is 16, the words on the stack are transferred to the registers in the following order: DI, SI, BP, ignore word, BX, DX, CX, and AX.

6.1.4.3 Shift Instructions

The SAL (shift arithmetic left), SHL (shift logical left), SAR (shift arithmetic right), SHR (shift logical right) instructions perform an arithmetic or logical shift of the bits in a byte, word, or doubleword.

The SAL and SHL instructions perform the same operation. They shift the source operand left by from 1 to 31 bit positions. Empty bit positions are cleared. The CF flag is loaded with the last bit shifted out of the operand.

The SHR instruction shifts the source operand right by from 1 to 31 bit positions. As with the SHL/SAL instruction, the empty bit positions are cleared and the CF flag is loaded with the last bit shifted out of the operand.

The SAR instruction shifts the source operand right by from 1 to 31 bit. This instruction differs from the SHR instruction in that it preserves the sign of the source operand by clearing empty bit positions if the operand is positive or setting the empty bits if the operand is negative. Again, the CF flag is loaded with the last bit shifted out of the operand.

6.1.4.4 Rotate Instructions

The ROL (rotate left), ROR (rotate right), RCL (rotate through carry left) and RCR (rotate through carry right) instructions rotate the bits in the destination operand out of one end and back through the other end. Unlike a shift, no bits are lost during a rotation. The rotate count can range from 0 to 31.

The ROL instruction rotates the bits in the operand to the left (toward more significant bit locations). The ROR instruction rotates the operand right (toward less significant bit locations).

The RCL instruction rotates the bits in the operand to the left, through the CF flag. This instruction treats the CF flag as a one-bit extension on the upper end of the operand. Each bit that exits from the most significant bit location of the operand moves into the CF flag. At the same time, the bit in the CF flag enters the least significant bit location of the operand.

The RCR instruction rotates the bits in the operand to the right through the CF flag. For all the rotate instructions, the CF flag always contains the value of the last bit rotated out of the operand, even if the instruction does not use the CF flag as an extension of the operand. The value of this flag can then be tested by a conditional jump instruction (JC or JNC).

6.1.4.5 Control Transfer Instructions

The processor provides both conditional and unconditional control transfer instructions to direct the flow of program execution. Conditional transfers are taken only for specified states of the status flags in the EFLAGS register. Unconditional control transfers are always executed.

For the purpose of this discussion, these instructions are further divided subordinate subgroups that process:

- Unconditional transfers
- Conditional transfers
- Software interrupts.

The JMP, CALL, RET, INT, and IRET instructions transfer program control to another location (destination address) in the instruction stream. The destination can be within the same code segment (near transfer) or in a different code segment (far transfer).

Jump instruction – The JMP (jump) instruction unconditionally transfers program control to a destination instruction. The transfer is one-way; that is, a return address is not saved. A destination operand specifies the address (the instruction pointer) of the destination instruction. The address can be a relative address or an absolute address.

A relative address is a displacement (offset) with respect to the address in the EIP register. The destination address (a near pointer) is formed by adding the displacement to the address in the EIP register. The displacement is specified with a signed integer, allowing jumps either forward or backward in the instruction stream.

An absolute address is an offset from address 0 of a segment. It can be specified in either of the following ways:

- An address in a general-purpose register – This address is treated as a near pointer, which is copied into the EIP register. Program execution then continues at the new address within the current code segment.
- An address specified using the standard addressing modes of the processor – Here, the address can be a near pointer or a far pointer. If the address is for a near pointer, the address is translated into an offset and copied into the EIP register. If the address is for a far pointe r, the address is translated into a segment selector (which is copied into the CS register) and an offset (which is copied into the EIP register).

In protected mode, the JMP instruction also allows jumps to a call gate, a task gate, and a task-state segment.

6.1.4.6 Call and Return Instructions

The CALL (call procedure) and RET (return from procedure) instructions allow a jump from one procedure (or subroutine) to another and a subsequent jump back (return) to the calling procedure.

The CALL instruction transfers program control from the current (or calling procedure) to another procedure (the called procedure). To allow a subsequent return to the calling procedure, the CALL instruction saves the current contents of the EIP register on the stack before jumping to the called procedure. The EIP register (prior to transferring program control) contains the address of the instruction following the CALL instruction. When this address is pushed on the stack, it is referred to as the return instruction pointer or return address.

The address of the called procedure (the address of the first instruction in the procedure being jumped to) is specified in a CALL instruction the same way as it is in a JMP instruction. The address can be specified as a relative address or an absolute address. If an absolute address is specified, it can be either a near or a far pointer.

The RET instruction transfers program control from the procedure currently being executed (the called procedure) back to the procedure that called it (the calling procedure). Transfer of control is accomplished by copying the return instruction pointer from the stack into the EIP register. Program execution then continues with the instruction pointed to by the EIP register.

The RET instruction has an optional operand, the value of which is added to the contents of the ESP register as part of the return operation. This operand allows the stack pointer to be incremented to remove parameters from the stack that were pushed on the stack by the calling procedure.

6.1.4.7 Loop Instructions

The LOOP, LOOPE (loop while equal), LOOPZ (loop while zero), LOOPNE (loop while not equal), and LOOPNZ (loop while not zero) instructions are conditional jump instructions that use the value of the ECX register as a count for the number of times to execute a loop. All the loop instructions decrement the count in the ECX register each time they are executed and terminate a loop when zero is reached. The LOOPE, LOOPZ, LOOPNE, and LOOPNZ instructions also accept the ZF flag as a condition for terminating the loop before the count reaches zero.

The LOOP instruction decrements the contents of the ECX register (or the CX register, if the address-size attribute is 16), then tests the register for the loop-termination condition. If the count in the ECX register is non-zero, program control is transferred to the instruction address specified by the destination operand. The destination operand is a relative address (that is, an offset relative to the contents of the

EIP register), and it generally points to the first instruction in the block of code that is to be executed in the loop. When the count in the ECX register reaches zero, program control is transferred to the instruction immediately following the LOOP instruction, which terminates the loop. If the count in the ECX register is zero when the LOOP instruction is first executed, the register is pre-decremented to FFFFFFFFH, causing the loop to be executed 2^{32} times.

The LOOPE and LOOPZ instructions perform the same operation (they are mnemonics for the same instruction). These instructions operate the same as the LOOP instruction, except that they also test the ZF flag.

If the count in the ECX register is not zero and the ZF flag is set, program control is transferred to the destination operand. When the count reaches zero or the ZF flag is clear, the loop is terminated by transferring program control to the instruction immediately following the LOOPE/LOOPZ instruction.

The LOOPNE and LOOPNZ instructions (mnemonics for the same instruction) operate the same as the LOOPE/LOOPPZ instructions, except that they terminate the loop if the ZF flag is set.

6.1.4.8 Random Number Generator Instruction

The RDRAND instruction returns a random number. All Intel processors that support the RDRAND instruction indicate the availability of the RDRAND instruction via reporting CPUID.01H:ECX.RDRAND[bit 30] = 1.

RDRAND returns random numbers that are supplied by a cryptographically secure, deterministic random bit generator DRBG. The DRBG is designed to meet the NIST SP 800-90 standard. The DRBG is re-seeded frequently from an on-chip non-deterministic entropy source to guarantee data returned by RDRAND is statistically uniform, non-periodic, and non-deterministic.

In order for the hardware design to meet its security goals, the random number generator continuously tests itself and the random data it is generating. Runtime failures in the random number generator circuitry or statistically anomalous data occurring by chance will be detected by the self test hardware and flag the resulting data as being bad. In such extremely rare cases, the RDRAND instruction will return no data instead of bad data.

Under heavy load, with multiple cores executing RDRAND in parallel, it is possible, though unlikely, for the demand of random numbers by software processes/ threads to exceed the rate at which the random number generator hardware can supply them. This will lead to the RDRAND instruction returning no data transitorily. The RDRAND instruction indicates the occurrence of this rare situation by clearing the CF flag.

The RDRAND instruction returns with the carry flag set (CF = 1) to indicate valid data is returned. It is recommended that software using the RDRAND instruction to get random numbers retry for a limited number of iterations while RDRAND returns CF = 0 and complete when valid data is returned, indicated with CF = 1. This will deal with transitory underflows. A retry limit should be employed to prevent a hard failure in the RNG (expected to be extremely rare) leading to a busy loop in software.

The intrinsic primitive for RDRAND is defined to address software's need for the common cases (CF = 1) and the rare situations (CF = 0). The intrinsic primitive returns a value that reflects the value of the carry flag returned by the underlying RDRAND instruction. The example below illustrates the recommended usage of an RDRAND instrinsic in a utility function, a loop to fetch a 64 bit random value with a retry count limit of 10. A C implementation might be written as follows:

```
#define SUCCESS 1
#define RETRY_LIMIT_EXCEEDED 0
#define RETRY_LIMIT 10
int get_random_64 ( unsigned __int 64 * arand)
{
    int i;
    for ( i = 0; i < RETRY_LIMIT; i ++) {
    if(_rdrand64_step(arand)) return SUCCESS;
    }
    return RETRY_LIMIT_EXCEEDED;
}
```

6.1.4.9 Program Environment Instructions

The programming environment for the general-purpose instructions consists of the set of registers and address space. The environment includes the following items:

- *General-purpose registers* – Eight 32-bit general-purpose are used in non-64-bit modes to address operands in memory. These registers are referenced by the names EAX, EBX, ECX, EDX, EBP, ESI EDI, and ESP.
- *Segment registers* – The six 16-bit segment registers contain segment pointers for use in accessing memory. These registers are referenced by the names CS, DS, SS, ES, FS, and GS.
- *EFLAGS register* – This 32-bit is used to provide status and control for basic arithmetic, compare, and system operations.
- *EIP register* – This 32-bit register contains the current instruction pointer.

General-purpose instructions operate on the following data types. The width of valid data types is dependent on processor mode:

- Bytes, words, doublewords
- Signed and unsigned byte, word, doubleword integers
- Near and far pointers
- Bit fields
- BCD integers.

6.2 Registers

Registers are designed for specific functions and are wired specifically for different purposes. A processor contains a number of registers to hold instruction and data. Some special registers are used to store the status of the current operation. Registers are the fastest memory in a computer system. The reason to use registers is to make the computation and handling fast. By reducing the times to access internal and external memories, the RISC machines usually contain a large number of registers to improve the speed.

Processors usually assign special roles to certain registers, including these registers:

- An Accumulator (ACC), which collects the result of computations (von Neumann, 2006).
- Address Registers, which keep track of where a given instruction or piece of data is stored in memory. Each storage location in memory is identified by an address.
- Data Registers, which temporarily hold data taken from or about to be sent to memory.
- Status Registers, which are used for store current CPU status.
- Program Counter (PC), which is used to point to the current instruction being executed.

Newly produced processors usually have more advanced control unit and dedicated control bits for security.

Intel Core i7-900 is based on 45 ns process technology. It contains an Execute Disable Bit that allows memory to be marked as executable or non-executable, when combined with a supporting operating system.

If code attempts to run in non-executable memory the processor raises an error to the operating system. This feature can prevent some classes of viruses or worms that exploit buffer over run vulnerabilities and can thus help improve the overall security of the system (Daswani et al., 2007).

Table 6.2 shows a few status registers on Intel Core i7. Those are only a small portion of registers used for enhancing system security.

RISC would always use registers for all operands for certain operations, for example, ADD. This makes programs run much faster. On the other hand, programmers have to load the value from memory to registers thus making the program lengthy.

6.2.1 General-Purpose Registers

The IA-32 architecture provides 16 basic program execution registers for use in general system and application programing. These registers can be grouped into four categories:

Table 6.2 Some status registers

Term	Description
RO	Read Only. If a register bit is read only, the hardware sets its state. The bit may be read by software. Writes to this bit have no effect
WO	Write Only. The register bit is not implemented as a bit. The write causes some hardware event to take place
RW	Read/Write. A register bit with this attribute can be read and written by software
RWL	Read/Write/Lock. A register bit with this attribute can be read or written by software. Hardware or a configuration bit can lock the bit and prevent it from being updated
RRW	Read/Restricted Write. This bit can be read and written by software. However, only supported values will be written. Writes of non supported values will have no effect
L	Lock. A register bit with this attribute becomes Read Only after a lock bit is set

- General-purpose registers. These eight registers are available for storing operands and pointers.
- Segment registers. These registers hold up to six segment selectors.
- EFLAGS (program status and control) register. The EFLAGS register report on the status of the program being executed and allows limited (application program level) control of the processor.
- EIP (instruction pointer) register. The EIP register contains a 32-bit pointer to the next instruction to be executed.

The 32-bit general-purpose registers EAX, EBX, ECX, EDX, ESI, EDI, EBP, and ESP are provided for holding the following items:

- Operands for logical and arithmetic operations
- Operands for address calculations
- Memory pointers.

Although all of these registers are available for general storage of operands, results, and pointers, caution should be used when referencing the ESP register. The ESP register holds the stack pointer and as a general rule should not be used for another purpose.

Many instructions assign specific registers to hold operands. For example, string instructions use the contents of the ECX, ESI, and EDI registers as operands. When using a segmented memory model, some instructions assume that pointers in certain registers are relative to specific segments. For instance, some instructions assume that a pointer in the EBX register points to a memory location in the DS segment.

The following is a summary of special uses of general-purpose registers:

- EAX – Accumulator for operands and results data
- EBX – Pointer to data in the DS segment
- ECX – Counter for string and loop operations
- EDX – I/O pointer
- ESI – Pointer to data in the segment pointed to by the DS register; source pointer for string operations
- EDI – Pointer to data (or destination) in the segment pointed to by the ES register; destination pointer for string operations
- ESP – Stack pointer (in the SS segment)
- EBP – Pointer to data on the stack (in the SS segment).

The lower 16 bits of the general-purpose registers map directly to the register set found in the 8086 and Intel 286 processors and can be referenced with the names AX, BX, CX, DX, BP, SI, DI, and SP. Each of the lower two bytes of the EAX, EBX, ECX, and EDX registers can be referenced by the names AH, BH, CH, and DH (high bytes) and AL, BL, CL, and DL (low bytes).

6.2.2 Segment Registers

The segment registers (CS, DS, SS, ES, FS, and GS) hold 16-bit segment selectors. A segment selector is a special pointer that identifies a segment in memory. To access a particular segment in memory, the segment selector for that segment must be present in the appropriate segment register.

When writing application code, programmers generally create segment selectors with assembler directives and symbols. The assembler and other tools then create the actual segment selector values associated with these directives and symbols. If writing system code, programmers may need to create segment selectors directly.

How segment registers are used depends on the type of memory management model that the operating system or executive is using. When using the flat (unsegmented) memory model, segment registers are loaded with segment selectors that point to overlapping segments, each of which begins at address 0 of the linear address space. These overlapping segments then comprise the linear address space for the program. Typically, two overlapping segments are defined: one for code and another for data and stacks. The CS segment register points to the code segment and all the other segment registers point to the data and stack segment.

When using the segmented memory model, each segment register is ordinarily loaded with a different segment selector so that each segment register points to a different segment within the linear address space. At any time, a program can thus access up to six segments in the linear address space. To access a segment not pointed to by one of the segment registers, a program must first load the segment selector for the segment to be accessed into a segment register.

Each of the segment registers is associated with one of three types of storage: code, data, or stack. For example, the CS register contains the segment selector for the code segment, where the instructions being executed are stored. The processor fetches instructions from the **code segment**, using a logical address that consists of the segment selector in the CS register and the contents of the EIP register. The EIP register contains the offset within the code segment of the next instruction to be executed. The CS register cannot be loaded explicitly by an application program. Instead, it is loaded implicitly by instructions or internal processor operations that change program control (such as, procedure calls, interrupt handling, or task switching).

The DS, ES, FS, and GS registers point to four **data segments**. The availability of four data segments permits efficient and secure access to different types of data structures. For example, four separate data segments might be created: one for the data structures of the current module, another for the data exported from a higher level module, a third for a dynamically created data structure, and a fourth for data shared with another program. To access additional data segments, the application program must load segment selectors for these segments into the DS, ES, FS, and GS registers, as needed.

The SS register contains the segment selector for the **stack segment**, where the procedure stack is stored for the program, task, or handler currently being executed. All stack operations use the SS register to find the stack segment. Unlike the CS register, the SS register can be loaded explicitly, which permits application programs to set up multiple stacks and switch among them.

6.2.3 EFLAGS Register

The 32-bit EFLAGS register contains a group of status flags, a control flag, and a group of system flags. Following initialization of the processor (either by asserting the RESET pin or the INIT pin), the state of the EFLAGS register is 00000002H. Bits 1, 3, 5, 15, and 22 through 31 of this register are reserved. Software should not use or depend on the states of any of these bits.

Some of the flags in the EFLAGS register can be modified directly, using special purpose instructions (described in the following sections). There are no instructions that allow the whole register to be examined or modified directly.

The following instructions can be used to move groups of flags to and from the procedure stack or the EAX register: LAHF, SAHF, PUSHF, PUSHFD, POPF, and POPFD. After the contents of the EFLAGS register have been transferred to the procedure stack or EAX register, the flags can be examined and modified using the processor's bit manipulation instructions (BT, BTS, BTR, and BTC).

When suspending a task (using the processor's multitasking facilities), the processor automatically saves the state of the EFLAGS register in the task state segment (TSS) for the task being suspended. When binding itself to a new task, the processor loads the EFLAGS register with data from the new task's TSS.

When a call is made to an interrupt or exception handler procedure, the processor automatically saves the state of the EFLAGS registers on the procedure stack. When an interrupt or exception is handled with a task switch, the state of the EFLAGS register is saved in the TSS for the task being suspended.

The status flags (bits 0, 2, 4, 6, 7, and 11) of the EFLAGS register indicate the results of arithmetic instructions, such as the ADD, SUB, MUL, and DIV instructions. The status flag functions are:

CF (bit 0) **Carry** flag – Set if an arithmetic operation generates a carry or a borrow out of the most-significant bit of the result; cleared otherwise. This flag indicates an overflow condition for unsigned-integer arithmetic. It is also used in multiple-precision arithmetic.

PF (bit 2) **Parity** flag – Set if the least-significant byte of the result contains an even number of 1 bits; cleared otherwise.

AF (bit 4) **Adjust** flag – Set if an arithmetic operation generates a carry or a borrow out of bit 3 of the result; cleared otherwise. This flag is used in binary-coded decimal (BCD) arithmetic.

ZF (bit 6) **Zero** flag – Set if the result is zero; cleared otherwise.

SF (bit 7) **Sign** flag – Set equal to the most-significant bit of the result, which is the sign bit of a signed integer. (0 indicates a positive value and 1 indicates a negative value.)

OF (bit 11) **Overflow** flag – Set if the integer result is too large a positive number or too small a negative number (excluding the sign-bit) to fit in the destination operand; cleared otherwise. This flag indicates an overflow condition for signed-integer (two's complement) arithmetic.

Of these status flags, only the CF flag can be modified directly, using the STC, CLC, and CMC instructions. Also the bit instructions (BT, BTS, BTR, and BTC) copy a specified bit into the CF flag.

The status flags allow a single arithmetic operation to produce results for three different data types: unsigned integers, signed integers, and BCD integers. If the result of an arithmetic operation is treated as an unsigned integer, the CF flag indicates an out-of-range condition (carry or a borrow); if treated as a signed integer (two's complement number), the OF flag indicates a carry or borrow; and if treated as a BCD digit, the AF flag indicates a carry or borrow. The SF flag indicates the sign of a signed integer. The ZF flag indicates either a signed- or an unsigned-integer zero.

When performing multiple-precision arithmetic on integers, the CF flag is used in conjunction with the add with carry (ADC) and subtract with borrow (SBB) instructions to propagate a carry or borrow from one computation to the next.

The condition instructions Jcc (jump on condition code cc), SETcc (byte set on condition code cc), LOOPcc, and CMOVcc (conditional move) use one or more of the status flags as condition codes and test them for branch, set-byte, or end-loop conditions.

6.3 The Program Counter and Flow Control

A program counter (PC) or instruction pointer is a special register to hold the current program execution location. The PC is incremented to point to the next instruction by adding **one word** after each instruction fetch (Hennessy and Patterson, 1998).

$$PC \leftarrow PC + 1$$

Here the number 1 means the next instruction. For a 32-bit processor, the next instruction is placed at $\mathrm{PC} + 4$. So PC always points to the next instruction after the current instruction is fetched, interpreted, and executed.

A program may not always run linearly. It may stop at some point to handle works (interrupts) or jump to another instruction, here we call branching. There are two types of branches, conditional and unconditional.

The goto statement is a common way in implementing branches. However there is a debate whether to use or not to use. Many computer languages support goto statement but some others may not, like Java. The structured program theorem proved that the goto statement is not necessary to write programs (Knuth, 1998). The goto statement is often combining with If statement in the form of:

```
IF condition THEN goto label;
```

Dijkstra, a renowned computer scientist, published a paper "A Case against the GO TO Statement." In the paper, he depreciates the goto statement and its effective replacement by structure language statement such as while loop statement (Dijkstra, 1968).

Kruth, the author of the famous book series *The Art of Computer Programming* wrote another paper "Structured Programming with go to Statements" suggests that it is optimal to use goto statement.

*6.3.1 Intel Instruction Pointer**

The instruction pointer (EIP) register on Intel processors contains the offset in the current code segment for the next instruction to be executed. It is advanced from one instruction boundary to the next in straight-line code or it is moved ahead or backwards by a number of instructions when executing JMP, Jcc, CALL, RET, and IRET instructions.

The EIP register cannot be accessed directly by software; it is controlled implicitly by control-transfer instructions (such as JMP, Jcc, CALL, and RET), interrupts, and exceptions. The only way to read the EIP register is to execute a CALL instruction and then read the value of the return instruction pointer from the procedure stack. The EIP register can be loaded indirectly by modifying the value of a return instruction pointer on the procedure stack and executing a return instruction (RET or IRET).

All IA-32 processors prefetch instructions. Because of instruction prefetching, an instruction address read from the bus during an instruction load does not match the value in the EIP register. Even though different processor generations use different prefetching mechanisms, the function of the EIP register to direct program flow remains fully compatible with all software written to run on IA-32 processors.

In 64-bit mode, the RIP register becomes the instruction pointer. This register holds the 64-bit offset of the next instruction to be executed. 64-bit mode also supports a technique called RIP-relative addressing. Using this technique, the effective address is determined by adding a displacement to the RIP of the next instruction.

*6.3.2 Interrupt and Exception**

Interrupts and exceptions are events that indicate that a condition exists somewhere in the system, the processor, or within the currently executing program or task that requires the attention of a processor. They typically result in a forced transfer of execution from the currently running program or task to a special software routine or task called an interrupt handler or an exception handler. The action taken by a processor in response to an interrupt or exception is referred to as servicing or handling the interrupt or exception.

Interrupts occur at random times during the execution of a program, in response to signals from hardware. System hardware uses interrupts to handle events external to the processor, such as requests to service peripheral devices. Software can also generate interrupts by executing the INT n instruction.

Exceptions occur when the processor detects an error condition while executing an instruction, such as division by zero. The processor detects a variety of error conditions including protection violations, page faults, and internal machine faults. The machine-check architecture of the Pentium 4, Intel Xeon, P6 family, and Pentium processors also permits a machine-check exception to be generated when internal hardware errors and bus errors are detected.

When an interrupt is received or an exception is detected, the currently running procedure or task is suspended while the processor executes an interrupt or exception handler. When execution of the handler is complete, the processor resumes execution of the interrupted procedure or task. The resumption of the interrupted procedure or task happens without loss of program continuity, unless recovery from an exception was not possible or an interrupt caused the currently running program to be terminated.

6.3.2.1 Source of Interrupts

The sources of interrupts received by the processor are two types: external (hardware generated) interrupts and software-generated interrupts.

External interrupts are received through pins on the processor or through the local APIC. The primary interrupt pins on Pentium 4, Intel Xeon, P6 family, and Pentium processors are the LINT[1:0] pins, which are connected to the local.

When the local APIC is enabled, the LINT[1:0] pins can be programmed through the APIC's local vector table (LVT) to be associated with any of the processor's exception or interrupt vectors.

When the local APIC is global/hardware disabled, these pins are configured as INTR and NMI pins, respectively. Asserting the INTR pin signals the processor that an external interrupt has occurred. The processor reads from the system bus the interrupt vector number provided by an external interrupt controller, such as an 8259A.

The processor's local APIC is normally connected to a system-based I/O APIC. Here, external interrupts received at the I/O APIC's pins can be directed to the local APIC through the system bus (Pentium 4, Intel Core Duo, Intel Core 2, Intel$^{®}$ Atom^{TM}, and Intel Xeon processors) or the APIC serial bus (P6 family and Pentium processors).

The I/O APIC determines the vector number of the interrupt and sends this number to the local APIC. When a system contains multiple processors, processors can also send interrupts to one another by means of the system bus (Pentium 4, Intel Core Duo, Intel Core 2, Intel Atom, and Intel Xeon processors) or the APIC serial bus (P6 family and Pentium processors).

The LINT[1:0] pins are not available on the Intel486 processor and earlier Pentium processors that do not contain an on-chip local APIC. These processors have dedicated NMI and INTR pins. With these processors, external interrupts are typically generated by a system-based interrupt controller (8259A), with the interrupts being signaled through the INTR pin.

Note that several other pins on the processor can cause a processor interrupt to occur. However, these interrupts are not handled by the interrupt and exception mechanism described in this chapter. These pins include the RESET#, FLUSH#, STPCLK#, SMI#, R/S#, and INIT# pins. Whether they are included on a particular processor is implementation dependent. Pin functions are described in the data books for the individual processors.

The INT n instruction permits interrupts to be generated from within software by supplying an interrupt vector number as an operand. For example, the INT 35 instruction forces an implicit call to the interrupt handler for interrupt 35.

Any of the interrupt vectors from 0 to 255 can be used as a parameter in this instruction. If the processor's predefined NMI vector is used, however, the response of the processor will not be the same as it would be from an NMI interrupt generated in the normal manner. If vector number 2 (the NMI vector) is used in this instruction, the NMI interrupt handler is called, but the processor's NMI-handling hardware is not activated.

Interrupts generated in software with the INT n instruction cannot be masked by the IF flag in the EFLAGS register.

6.3.2.2 Source of Exceptions

The processor receives exceptions from three sources:

- Processor-detected program-error exceptions
- Software-generated exceptions
- Machine-check exceptions.

The processor generates one or more exceptions when it detects program errors during the execution in an application program or the operating system or executive. Intel 64 and IA-32 architectures define a vector number for each processor-detectable exception. Exceptions are classified as faults, traps, and aborts.

The INTO, INT 3, and BOUND instructions permit exceptions to be generated in software. These instructions allow checks for exception conditions to be performed at points in the instruction stream. For example, INT 3 causes a breakpoint exception to be generated.

The INT n instruction can be used to emulate exceptions in software; but there is a limitation. If INT n provides a vector for one of the architecturally-defined exceptions, the processor generates an interrupt to the correct vector (to access the exception handler) but does not push an error code on the stack. This is true even if the associated hardware-generated exception normally produces an error code. The exception handler will still attempt to pop an error code from the stack while handling the exception. Because no error code was pushed, the handler will pop off and discard the EIP instead (in place of the missing error code). This sends the return to the wrong location.

The P6 family and Pentium processors provide both internal and external machine-check mechanisms for checking the operation of the internal chip hardware and bus transactions. These mechanisms are implementation dependent. When a machine-check error is detected, the processor signals a machine-check exception (vector 18) and returns an error code.

6.4 RISC Processors

A reduced instruction set computer (RISC) is a type of microprocessor that recognizes a relatively limited number of instructions. Until the mid-1980s, the tendency among computer manufacturers was to build increasingly complex CPUs that had ever-larger sets of instructions. At that time, however, a number of computer manufacturers decided to reverse this trend by building CPUs capable of executing only a very limited set of instructions. One advantage of reduced instruction set computers is that they can execute their instructions very fast because the instructions are so simple. Another, perhaps more important advantage, is that RISC chips require fewer transistors, which makes them cheaper to design and produce. Since the emergence of RISC computers, conventional computers have been referred to as CISCs (complex instruction set computers).

6.4.1 History

The first RISC projects came from IBM, Stanford, and UC-Berkeley in the late 1970s and early 1980s. The IBM 801, Stanford MIPS, and Berkeley RISC 1 and 2 were all designed with a similar philosophy which has become known as RISC. Certain design features have been characteristic of most RISC processors:

- *one cycle execution time:* RISC processors have a CPI (clock per instruction) of one cycle. This is due to the optimization of each instruction on the CPU and a technique called pipelining.
- *pipelining:* a technique that allows for simultaneous execution of parts, or stages, of instructions to more efficiently process instructions.
- *large number of registers:* the RISC design philosophy generally incorporates a larger number of registers to prevent in large amounts of interactions with memory.

6.4.2 Architecture and Programming

The simplest way to examine the advantages and disadvantages of RISC architecture is by contrasting it with its predecessor: Complex Instruction Set Computers (CISC) architecture.

RISC processors only use simple instructions that can be executed within one clock cycle. Thus, the "MULT" (multiply) command has to be divided into three separate commands: "LOAD," which moves data from the memory bank to a register, "PROD," which finds the product of two operands located within the registers, and "STORE," which moves data from a register to the memory banks. Figure 6.3 shows a diagram of the multiply execution. In order to perform the exact series of steps described in the CISC approach, a programmer would need to code four lines of assembly:

```
LOAD A, 2:3 ;load data 1 from memory 2:3 to register A
LOAD B, 5:2 ;load data 2 from memory 5:2 to register B
PROD A, B ;multiply
STORE 2:3, A ;store result to memory 2:3
```

The primary goal of CISC architecture is to complete a task in as few lines of assembly as possible. A CISC processor would come prepared with a MULT

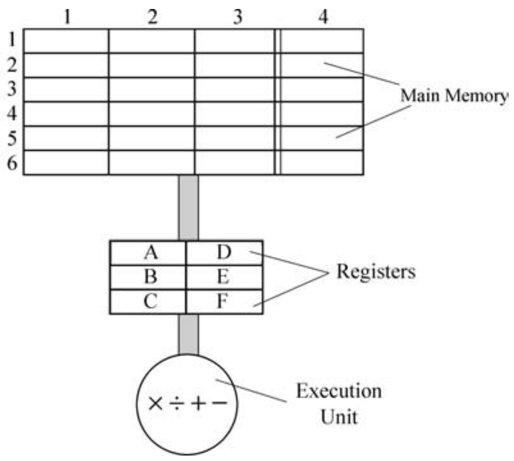


Figure 6.3 MULT instruction executions in RISC architecture

instruction. When executed, this instruction loads the two values into separate registers, multiplies the operands in the execution unit, and then stores the product in the appropriate register. Thus, the entire task of multiplying two numbers can be completed with one instruction:

```
MULT 2:3, 5:2
```

6.4.3 Performance

The following equation is commonly used for expressing a computer's performance ability:

$$\frac{time}{program} = \frac{time}{cycle} \times \frac{cycle}{instruction} \times \frac{instruction}{program}$$

The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction. RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program.

6.4.4 Advantages and Disadvantages

There is still considerable controversy among experts about the ultimate value of RISC architectures. Its proponents argue that RISC machines are both cheaper and faster, and are therefore the machines of the future. Skeptics note that by making the hardware simpler, RISC architectures put a greater burden on the software. They argue that this is not worth the trouble because conventional microprocessors are becoming increasingly fast and cheap anyway.

To some extent, the argument is becoming moot because CISC and RISC implementations are becoming more and more alike. Many of today's RISC chips support as many instructions as yesterday's CISC chips. And today's CISC chips use many techniques formerly associated with RISC chips.

6.4.5 Applications

RISC has not got the momentum on desktop computers and servers. However its benefits in embedded and mobile devices have become the main reason to be used widely in these areas including iPhone, BlackBerry, Android, and some gaming devices.

Well-known RISC families include DEC Alpha, AMD 29k, MIPS, PA-RISC, ARM, PowerPC, and SPARC and so on.

The hardware translation from x86 instructions into internal RISC-like micro-operations, which cost relatively little in microprocessors for desktops and servers, become significant in area and energy for mobile and embedded devices. Hence, ARM processors dominate cell phones and tablets today just as x86 processors dominate PCs. Atmel AVR used in a variety of products ranging from Xbox handheld controllers to BMW cars.

6.5 Pipelining

Traditional pipelining is a standard feature in RISC processors that is much like an assembly line. Because the processor works on different steps of the instruction at the same time, more instructions can be executed in a shorter period of time.

6.5.1 Different Types of Pipelines

Computer-related pipelines include:

- Instruction pipelines. Instruction pipelining allows overlapping execution of multiple instructions with the same circuitry (Almasi and Gottlieb, 1989). The circuitry is usually divided up into stages, including instruction decoding, arithmetic, and register fetching stages, wherein each stage processes one instruction at a time.
- Graphics pipelines. Graphics pipelining is found in most graphics cards, which consist of multiple arithmetic units, or complete CPUs, that implement the various stages of common rendering operations (perspective projection, window clipping, color and light calculation, rendering, etc.).
- Software pipelines. In software pipelining, commands can be written so that the output of one operation is automatically used as the input to the next, following operation. The Unix system call pipe is a classic example of this concept; although other operating systems do support pipes as well (Hockney et al., 1988).

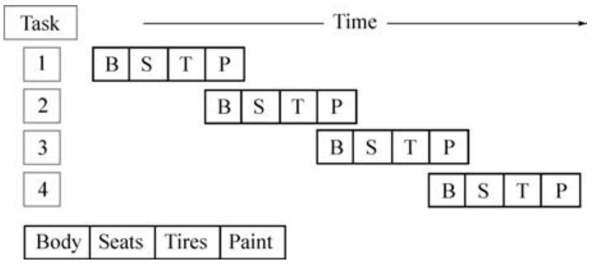


Figure 6.4 Linear car manufacture process

6.5.2 *Pipeline Performance Analysis*

A useful method of demonstrating pipelining is the car manufacture analogy. Let's say that there are four steps to make a car, body, seats, tires and painting. Suppose loading the body to the assembly line needs 20 minutes, installing seats needs 20 minutes, installing tires take 20 minutes, and then painting takes 20 minutes. When the first car is manufactured, the process starts again from loading the body to finally painting.

Since each car needs 80 minutes to finish, this assembly line can manufacture $60\,\text{min} \times 8$ hours$/80 = 6$ cars per day, that's equivalent to 180 cars a month. Figure 6.4 shows the linear car manufacture process.

However, a smarter approach to the problem would be to load the second car body after the first car moved to installing seats. When the second car body is loaded, then start loading the third car body. The process is shown in Figure 6.5.

Except at the very beginning and the end of the pipeline, the number of cars manufactured increased approximately four times compared with the linear assembly line.

Generally a car manufacture process can be divided into hundreds of steps. Let's take 80 steps as an example and assume each step takes the same time. Then on average the time needed to manufacture a car is one minute! Without using pipelining, no one can imagine a car can be manufactured in one minute!

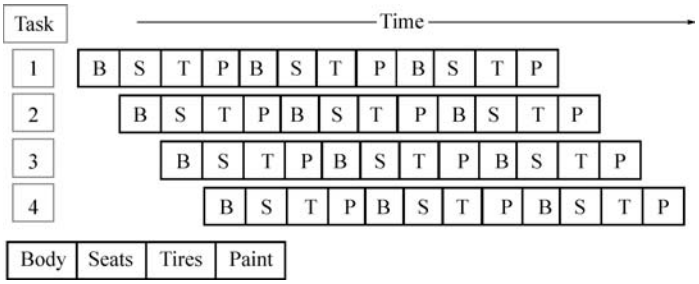


Figure 6.5 A car manufacture pipeline

In practice, each step is hardly to be divided exactly even. So the overall performance of a pipeline may not reach the ideal result as predicted. On the other hand, if at a certain step the pipeline is stalled, then resuming the pipeline needs extra time and therefore decreases the pipeline performance (Blaise, 2011).

6.5.3 Data Hazard

In a pipeline, a data dependency (Culler et al., 1999) occurs when an instruction depends on the results of previous instructions. A particular instruction may need data in a register which has not yet been stored since the preceding instruction has not yet reached the step in the pipeline.

For example:

$$\begin{gathered} \boldsymbol{R3} \leftarrow R1 + R2 \\ R5 \leftarrow \boldsymbol{R3} + R4 \end{gathered}$$

In this example, the second instructs is to add R3 and R4 and store the result in R5. When the second instruction is in the second stage, the processor will be attempting to read R3 and R4 from the registers. Remember that the first instruction is just one step ahead of the second, so the contents of R1 and R2 are being added, but the result has not yet been written into register R3. The second instruction therefore cannot read from the register R3 because it hasn't been written yet and must wait until the data is ready. Consequently, the pipeline is stalled and a number of empty instructions (known as bubbles) go into the pipeline. In this case, it is called read after write (RAW). Data dependency affects long pipelines more than shorter ones since it takes a longer period of time for an instruction to reach the final register-writing stage of a long pipeline.

6.6 CPU Security

Some tamper-resistant CPUs are packaged in a way that physical attempts to read or modify information inside the CPU cannot succeed easily. The CPU's tamper-resistant package protects its internal components, such as registers and cache, from hardware attacks. The CPU's internal cache is big enough (e.g., tens of megabytes) to contain a kernel and the working set of data for most programs, but not big enough to contain whole programs. Programs that require more memory use untrusted external memory. The CPU traps to a kernel when evicting or loading a cache line, so the kernel trap handler can protect data stored in external memory (Chen and Morris, 2003).

Such CPU has a corresponding public-private key pair. The private key is hidden in the CPU and never revealed to anyone else (including software that runs on the CPU).

The CPU reveals the public key in a CPU certificate signed by the manufacturer. A user trusts a CPU if the CPU certificate is signed by a trusted manufacturer. A CPU uses the private key to sign and decrypt data, and users use the public key to verify signatures and encrypt data for the CPU.

At the software level, the system uses a kernel to create and schedule processes onto the CPU and to handle traps and interrupts. The kernel runs in privileged mode; it can read or write all physical memory locations. Using page tables, the kernel partitions user-level processes into separate address spaces. The CPU applies conventional memory protection to prevent a program from issuing instructions that access or affect data in another address space. The page table of each address space is protected so only the kernel can change it. To prevent tampering of the kernel, the kernel text and some of its data reside in the secure CPU's cache and cannot be evicted. The secure CPU traps to the kernel when evicting data from its cache to external memory, or loading data from external memory into its cache. The kernel's trap handler decides, on a per-program basis, if and how the data should be protected. A program may ask the kernel to only authenticate or to both authenticate and copy-protect its instructions and data whenever they leave the secure CPU's internal cache.

The overhead of taking a trap on every cache event depends on a program's memory access pattern and miss rate, the cost of cryptographic operations, and the level of security a program demands. It is to believe that an increase in the size of a CPU's cache and using hardware assisted cryptographic operations decrease the overhead. Figure 6.6 shows what a system would look like. The kernel and some user-level servers that implement OS abstractions form a complete operating system. Users can also run virtualized operating systems such as VMWare in user space.

The system can also report the hardware and software configuration of a computer, so a user can decide if the hardware and software can be trusted to protect the user's program.

The CPU identifies each kernel by the content hash of the kernel's text and initialized data segment. If the kernel is modified before the system boots, its content hash would change. Because a kernel's text and data reside inside the tamper-resistant

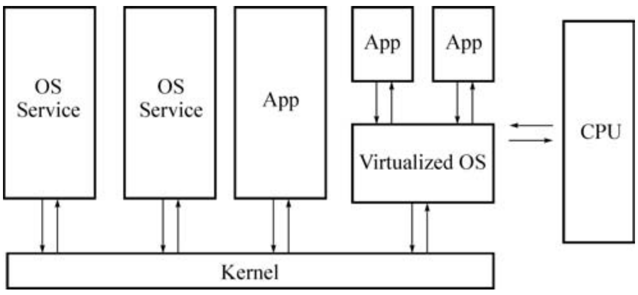


Figure 6.6 A secure processor diagram

CPU, they cannot be changed (e.g., by a DMA device) after the system boots. Kernel's content hash is also referred to as the kernel signature.

A processor-secured computer boots in several stages. On a hardware reset, the CPU computes the content hash of the BIOS and jumps to the BIOS code. Next, the BIOS computes the content hash of the boot loader, stored in the first sector of the computer's master hard drive, and jumps to the boot loader code. Finally, the boot loader code computes the kernel signature, and jumps into the corresponding kernel. Each stage uses a privileged CPU instruction to compute the content hash. The instruction stores the content hash in a register inside the CPU. The registers are protected so malicious programs cannot modify their content.

6.7 Virtual CPU

Virtual machine can be simply considered as "computers on a computer." It is a tightly isolated software container that can run its own operating system and applications as if it were a physical computer. A virtual machine behaves exactly like a physical computer and contains its own virtual CPU, RAM, hard disk and network interface card (NIC). Figure 6.7 shows a diagram of VM architecture.

A virtual CPU is a software based CPU that uses the resources of the host CPU and acts like a regular CPU (Englander, 2010). An operating system can't tell the difference between a virtual CPU/machine and a physical CPU/machine, nor can applications or other computers on a network. Nevertheless, a virtual machine is composed entirely of software and contains no hardware components whatsoever. As a result, virtual machines offer a number of distinct advantages over physical hardware.

For system virtual machines, multiple OS environments can co-exist on the same computer, in strong isolation from each other. The virtual machine can provide an instruction set architecture (ISA) that is somewhat different from that of the real machine in application provisioning, maintenance, availability, and disaster recovery.

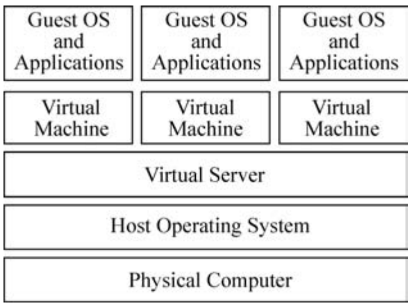


Figure 6.7 VM architecture diagram

Here are the main characteristics of VMs.

- Run multiple operating systems on a single computer including Windows, Linux and more.
- Reduce capital costs by increasing energy efficiency and requiring less hardware while increasing your server to admin ratio.
- Ensure enterprise applications perform with the highest availability and performance.
- Build up business continuity through improved disaster recovery solutions and deliver high availability throughout the datacenter.
- Improve enterprise desktop management and control with faster deployment of desktops and fewer support calls due to application conflicts.

Since virtual machines share the common resources with the host machine, it requires more memory and may have performance issues.

6.8 Summary

A central processing unit (CPU) has address, data and control signals. The width of the address and data buses is usually 64 for a 64-bit CPU. So a 32-bit CPU can have maximum memory of $2^{32} = 2^2 \times 2^{30} = 4\,\text{GB}$.

A CPU can have many different instructions. Some instructions (such as logic instructions) may take less time to execute than the others (such as arithmetic instructions). An instruction has two components, opcode and operand. The addressing mode for operands can be direct or indirect.

Registers are the fastest memory in computer systems. Some important registers include the accumulator (ACC), address registers, data registers, status registers and the program counter (PC). PC always points to the next instruction after the current instruction is fetched.

Reduced instruction set computers (RISC) have less instructions and each instruction takes one cycle. The goal is to make instructions so simple that they could easily be pipelined, in order to achieve a single clock throughput at high frequencies. This is favorable to using pipelines to speed up the execution time of a program. RISC processors are widely used in mobile electronics devices while traditional CISC processors are still leading the personal computers and servers.

Despite many successes, RISC has made few inroads into the desktop PC and commodity server markets, where Intel's x86 platform remains the dominant processor architecture.

There are researches in securing CPUs to prevent kernel being tampered with. One way to do this is to use encryption by adding a public-private key pair.

Virtual CPUs are software based CPUs that use the resources of the host CPU and act like regular CPUs. The virtual machine can provide an instruction set architecture (ISA) that functions like a regular machine. Since virtual machines share the common resources with the host machine. It requires more memory and may have performance issues.

Exercises

6.1 How much memory can a 32-bit CPU have?

6.2 List the algebra expression and truth table for logical operation XOR.

6.3 Subtraction usually involves two operands. How is subtraction done on a one-address machine?

6.4 A 64-bit CPU has PC pointing to memory location 8000:1000. After fetching a one word instruction, what is the value in the PC?

6.5 What is the performance of a CPU corresponding to the execution time?

6.6 Pipeline speed up can be expressed as

$$Speedup = \frac{Pipeline\ depth}{1 + Pipeline\ stall\ CPI} \times \frac{Cycle\ Time_{unpipelined}}{Cycle\ Time_{pipelined}}$$

where *CPI* stands for cycle per instruction.

Two RISC machines (CPI = 1). Machine B has 1.05 times fast clock rate than machine A and loads are 40% of instruction executed. Which machine is faster?

6.7 What is a virtual machine, what is a host machine?

6.8 Two computers run two tasks. The rate is in the following table. Computer the average and relative throughputs and specify which is faster.

System	Rate (Task 1)	Rate (Task 2)
A	10	20
B	20	10

References

Almasi, G. S. and Gottlieb, A. (1989) *Highly Parallel Computing*, Benjamin/Cummings, Redwood City, Calif.

Anderson, P. (2011) About SETI@home. In *SETI@home*. Retrieved November 14, 2011, from http://setiathome.berkeley.edu/sah_about.php.

Answers.com. (n.d.). Retrieved November 1, 2011, from http://www.answers.com/topic/serial-computer-1.

Blaise, B. (August 17, 2011) Introduction to Parallel Computing. In *Lawrence Livermore National Laboratory*. Retrieved November 12, 2011, from https://computing.llnl.gov/tutorials/parallel_comp/#Flynn.

Chen, B. and Morris, R., Certifying program execution with secure processors. Proc. of the 9th Conference on Hot Topics in Operating Systems, Vol. 9, pp. 23–29.

Culler, D. E., Singh, J.P., and Gupta, A. (1999) *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann Publishers, San Francisco.

Daswani, N., Kern, C., and Kesavan, A. (2007) *Foundations of Security: What Every Programmer Needs to Know*, Apress.

Dijkstra, E. W. (1968) A case against the GO TO statement. *ACM Communication*, **11**, 147–148.

Englander, I. (2010) *The Architecture of Computer Hardware, Systems Software, and Networking: An Information Technology Approach*, John Wiley & Sons, Hoboken, N.J.

Göhringer, D., Perschke, T., Hübner, M., and Becker, J. (2009) A taxonomy of reconfigurable single-/multiprocessor systems-on-chip. *International Journal of Reconfigurable Computing*, **2009**, 1–12.

GPUs: An Emerging Platform for General-Purpose Computation, Defense Technical Information Center, Ft. Belvoir.

Hennessy, J. L. and Patterson, D.A. (1998) *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann Publishers, San Francisco, Calif.

Hockney, R. W., Jesshope, C.R., and Hockney, R.W. (1988) *Parallel Computers 2: Architecture, Programming, and Algorithms*, A. Hilger, Bristol, England.

Johnson, E. (1988) Completing an MIMD multiprocessor taxonomy. *ACM Sigarch Computer Architecture News*, **16**(3), 44–47.

Knuth, D. E. (1998) *The Art of Computer Programming*, 3rd edn, Addison-Wesley.

Kruth, D. E. (1974) *Structured Programming with GO TO Statements*, Stanford University.

Mozdzynski, G. (2011) *Concepts of parallel computing* [PDF document]. Retrieved from Lecture Notes Online Web site: http://www.ecmwf.int/services/computing/training/material/hpcf/Para_concepts.pdf.

Schenck, A. (2006) (Photographer) Beowolf Cluster [Photograph], Retrieved November 16, 2011, from: http://en.wikipedia.org/wiki/File:Beowulf.jpg.

Shigeto, Y. and Sakai, M. (2011) Parallel computing of discrete element method on multi-core processors. *Particuology*, **9**(4), 398–405.

von Neumann (2006) Architecture [JGP Image]. Retrieved November 16, 2011, from: https://computing.llnl.gov/tutorials/parallel_comp/#Flynn.

7

Advanced Computer Architecture

Advanced computer architecture includes discussion of instruction set design (RISC and CISC), virtual memory system design, memory hierarchies, cache memories, pipelining, vector processing, I/O subsystems, coprocessor, and multiprocessor architectures. Some topics (such as vector processing, I/O subsystems etc.) are beyond the scope of this book. Interested readers can find books on advanced architecture for further study.

7.1 Multiprocessors

The central processor unit (CPU) was initially designed to include everything in a single chip. Over the years, more and more new processors have been developed with better speed and more functions. With the exponential growth for the need to further improve the performance of a computer system, the computer industry changes its approach from building faster chips to using multiple chips.

First, people started to explore the use of multiple microprocessors on the same motherboard. To do this, the hardware needs to support the manipulation of data so that the data can be sent to different central processing units. Later new architecture with multiple motherboards is used together with shared memory, storage, and interconnection networks.

7.1.1 Multiprocessing

Multiprocessing is a computer system that uses more than one CPU to handle data. Data needs to be broken into segments and then each processor is given a task to be performed. The division of work between the processors can be done in different ways. The idea of parallel central processing units was introduced in the 1970s. On the motherboard and the shared memory, each uses a CPU. Early on in the development it was classified as a math-coprocessor (as shown in Figure 7.1). In this arrangement one of the central processing units has to be parent and the other is the

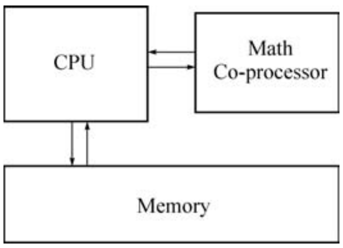


Figure 7.1 A computer has two processors, one CPU and one math co-processor

child. The parent would break up the tasks and then assign a task to the child processor and the results from the child would be passed back to the parent.

There were many problems to the first approach of multiprocessing units. The first approach had an overhead because the parent had to handle the flow of data and distribute it to the child. Both processors could not independently access the storage and make processing decisions. Though this was improvement to single central processing unit processor it had still another overhead issue of the distance between the processors and the time to move between them. To solve the second problem they moved the chip from its own socket on the motherboard and developed a stacked processor, with one chip on top of the other.

A multi-core processor is a single chip, or component, that contains two or more independent central processing units which can execute instructions independent of one another. The manufactures typically created integrated circuits that contained the central processing units, so that they were physically close together. They also gave the central processing units independent L1 and L2 caches.

7.1.2 Cache

A **cache** is a temporary memory device used by the central processing unit to store data that it could be working on. This type of memory is faster than main memory RAM because of how it is addressed and by its proximity to the central processing unit. The cache lessens the latency found when accessing the RAM. Most manufactures incorporated at least four independent caches to assist the processor so that it could function without sending requests down the bus to the main memory. One of these caches was the Data Cache which assists with the fetching of data stored in. L1 and L2 caches were pretty standard in the industry but some manufactures like IBM created an L3 cache. Another cache was the Instruction Cache whose purpose was to decrease the time it takes to fetch the instruction set. Figure 7.2 shows such a type of architecture.

Cache read has two possibilities, one is HIT and another one is MISS. If CPU can ready data directly from cache, then it is a HIT. If the data is not in the cache, then a MISS happens. Data will need to be moved to cache from memory. Figure 7.3 shows

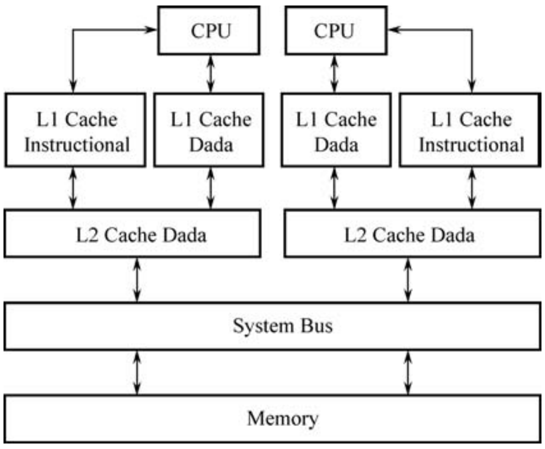


Figure 7.2 Architecture of a dual-core CPU with two level caches

a diagram of the interactions between CPU, cache, and memory. Cache is different than regular memory where individual data is accessed. Cache always moves in blocks. So whenever a MISS happens, data will be moved by blocks.

Once the development of multi-core processing chips was adapted into the industry, it took the concept and started the move away from physically separated dual processors. The term *Core* was introduced to refer to the number of central processing units in a chip. Intel and AMD started with the dual-core and moved to the quad-core which contains four core processors. The Hexa-core contains six cores and the Octal-core contains eight cores. The number of cores can scale up to 100. The limit is based mainly on how the data is transferred from the cores to the bus. Some manufactures have used switches in the cores to send data out of the central processing unit and into another core. Other manufacturers have used the mesh technique which links all of the cores data locations.

7.1.3 Hyper-Threading

Now that there is hardware that uses multiple central processing units, the industry had to adapt the operating systems to meet this new technology. The industry was led

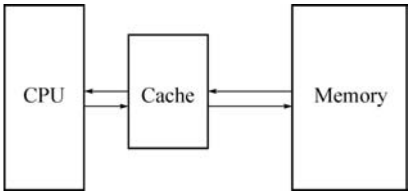


Figure 7.3 Cache memory

by Intel engineers, who created *Hyper-threading Technology*. This is the technology that is used to improve the flow of data to each processor. In order for each of the central processing units to be addressed by the Operation System efficiently, Intel created HTT. Hyper-threading Technology assists the parallelization of computations to the central processing unit.

7.1.4 Symmetric Multiprocessing

The computer industry created Symmetric Multiprocessing (SMP) which coordinates the use of two or more central processing units so that they can access a single shared memory (as shown in Figure 7.4). The authors of this book created dual-port memory (DPM) that is essentially the same as SMP. DPM allows two CPUs to connect the main memory simultaneously without worrying about conflicts. Without proper "treatment," two CPUs cannot be connected to one memory unit as conflicts will occur if one CPU write 0 and another CPU write 1 to the memory at the same time.

Processor coupling is to stack the processors together such that they can be defined as tightly-coupled Symmetric Multiprocessing (SMP). The term loosely-coupled multiprocessor systems are based on multiple standalone single or dual processors on the same motherboard, that are physically separated. If power consumption is a design consideration then the tightly-coupled central processing unit is a more efficient approach.

A shared bus is where the CPUs share a path to the RAM. The CPUs can be interconnected to the bus in a number of ways including mesh, multiplexed memory and crossbar switches.

7.1.5 Multiprocessing Operating Systems

One of the areas that slowed the development of a multiple processing chip was the operating system. Microsoft is one of the largest suppliers of operating systems who was late to come to the development of multiple processing. In the late 1980s and 1990s Microsoft was more focused on the user interface to achieve a large market

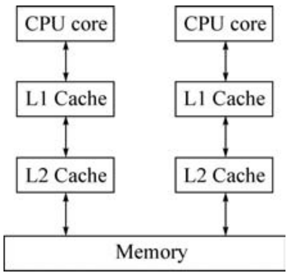


Figure 7.4 SMP diagram

share and was late in developing the multiple processor operating system. Early on, DOS and Windows did not have the ability to handle multiple central processing units. The operating system that did have the ability to handling multiple central processing units was UNIX. The UNIX operating system, with its roots in the mainframe, had the ability to be modified and was developed to handle multiple central processing units.

Sun Microsystems Corporation was the manufacturer that became the leader in this developmental race. Sun's approach to the computer market was to embrace the multiple processors and as such developed a computer that used this technology to stay on the cutting edge of computers. The Sun operating system is called Solaris and is a UNIX based language. In early 1987 Sun introduced the SPARC (Scalable Processor Architecture) which is a RISC instruction set. The system was based on their design and was initially a 32 bit structure but in 1995 it went up to 64 bits.

7.1.6 The Future of Multiprocessing

Multipurpose processing development fell into the classic supply/demand model. From the inception of the processor, the demand for more processing power was always pushing the market and driving innovation. This was greatly enhanced by the Microsoft approach to the computer market and was defined by its effort to make the computer a tool that could be used by people with little computer experience.

The future will be based on enabling the central processing unit reach the decision point. The industry has already done this by shifting the processing to a task. For example, the video of a computer was controlled by the CPU but today the raw video data for video display is sent to the video card. This video card will have a CPU and it will handle tasks that the main central processing units would have done in the past. A lot of programs require so much processing that multiple processors that were devoted to videos are now being sold on video cards themselves.

Other areas include the SSD, a device that will replace the legacy of the mechanical hard drive (HDD). It has an onboard processor to take the data from the main central processing units and store it on the SDD. The processor on the SDD handles the maintenance of storage and performs processes that run the SDD at its peak performance.

This concept has also been applied to networks. In the past, data went to the network and the hub would just broadcast the traffic. Now we use a router which contains multiple processing units to handle the rerouting of the data. With the new security model more and more central processing units are being deployed at the point of data exchange. With the implantation of DMZ and all new security devices, the industry is moving toward multiple central processing units at decision points. The evolution of the Smartphone is another example of moving multi-process closer to the decision point. Many Smartphones have multi-processors. So we will continue to see the industry implement more multi-processors at different levels of the network and user environment.

7.2 Parallel Processing

Parallel processing is the simultaneous usage of multiple machine resources. It often involves breaking down a very large problem, involving many calculations, into smaller ones, distributing them across multiple machines, and solving them concurrently. An example of a problem involving many calculations would be finding the 1 trillionth digit of pi. A single computer calculating this problem would take a massive amount of time. However, if the calculations where spread among 1,000 or a 100,000 machines, the time would be considerably less. Parallel computing involves architecture that is capable of allowing multiple machines to work together. Flynn's Taxonomy is representative of this architecture at many different levels: bit-level, instruction level, data level, and task level. Several different hardware configurations have been developed to maximize the potential of using several machine's resources; more aptly referred to as distributed computing. In other words, distributed systems are groups of networked computers, which have the same goal for their work.

7.2.1 History of Parallel Processing

Traditionally, computers were designed for serial computing; a single-processor computer that executes one instruction after the other. This is in stark contrast to parallel computing. Serial computing must complete one execution prior to beginning another and is generally very slow. Conversely, parallel computing is capable of executing many instructions simultaneously. This may seem very ordinary especially with today's technology, specifically processors with multiple cores. However, even with today's multi-core processors, an individual machine's ability to process many calculations is still limited to the amount of memory available to that operating system and motherboard. Below is a concept model created by Mozdzynski:

> An IFS T2047L149 forecast model takes about 5,000 seconds wall time for a 10 day forecast using 128 nodes of an IBM Power6 cluster. How long would this model take using a fast PC with sufficient memory (e.g., dual-core Dell desktop)? Answer: About 1 year. This PC would also need 2,000 GB of memory.

The previous illustration shows that parallel processing saves time. In saving time, you also save money associated with executing a task. Additional economic advantages come from the cost and value associated with current technology. PC chipmaker Intel's current flagship CPU is the Intel Xeon X7560 Nehalem-EX processor rated at a speed of 2.26 GHz combined with eight individual processing cores. The

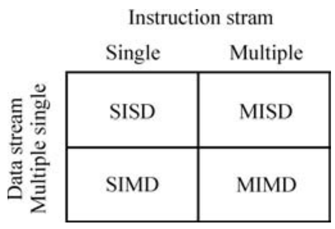


Figure 7.5 Flynn’s taxonomy

cost for one of these is approximately $4,000. However, Intel also has Core-i5 brand of processors, not nearly as robust, but has four individual processing cores, rated at a speed of 2.8 GHz for approximately $180. Roughly 22 Core-i5 CPUs may be purchased for the same price as one of their flagship CPUs giving you 11 times as many computational cores.

John von Neumann architecture is still the foundation of modern computers. It is important to note that parallel computers architecture remain unchanged, however an increase in the number of computers create the parallelism.

7.2.2 Flynn’s Taxonomy

Flynn’s taxonomy of computer systems has proved to be a useful tool for differentiating among uniprocessors (SISD), array and systolic machines (SIMD), and general multiprocessors (MIMD). Flynn classified programs and computers by “whether they were operating using a single set or multiple sets of instructions, whether or not those instructions were using a single or multiple sets of data.” SISD is the same as a system using sequential processing of a program’s instructions. SIMD is repetitive processing of a program over a large amount of data. MISD is not a popular architecture. In practice, few actual examples of this class of parallelism have ever existed such that each processing unit operates on the data independently via separate instruction steams and a single data stream is fed into multiple processing units. However, MIMD is the most popular, where many different calculations are carried out by many machines and used to calculate a very large amount of data. Figure 7.5 show the Flynn’s taxonomy.

7.2.3 Bit-Level Parallelism

Bit-level parallelism is a form of parallel computing based on increasing processor word size. It is done by performing the arithmetic on two numbers, bit by bit. The addition, for example, of two 32-bit numbers was performed in 32 machine cycles, by very simple circuitry capable only of adding one bit to another. Quite possibly the earliest use of parallelism in computers was to perform the operations on all bits simultaneously in parallel.

7.2.4 Instruction-Level Parallelism

Instruction-level parallelism performs many concurrent operations in a single computer program. For example:

1. $\boldsymbol{R3} \leftarrow R1 + R2$
2. $\boldsymbol{R6} \leftarrow R4 + R5$
3. $R7 \leftarrow \boldsymbol{R3} + \boldsymbol{R6}$.

The third operation is dependent on the prior two operations being complete. However, operations one and two are independent operations, so they can be calculated at the same time. The most efficient compiler and processor design would identify and take advantage of as much of this parallelism as possible. Ordinary programs are typically written under a sequential execution model where instructions execute one after the other and in the order specified by the programmer. Instruction level parallelism allows the compiler and the processor to overlap the execution of multiple instructions or even to change the order in which instructions are executed. Pipelining is an excellent example of this form of parallel processing. The idea of pipelining is that as each instruction completes a step, the following instruction moves into the stage just vacated. Thus, when the first instruction is completed, the next one is already one stage short of completion.

Data hazard occurs when processor fetches data c and f. Since c and f have not been written back, the fetch may access to the wrong data (previously stored in c and f, not the current ones). This is called “read after write” (RAW).

7.2.5 Data-Level Parallelism

Data-level parallelism is synonymous with Flynn’s SIMD architecture – single instruction (all processing units perform identical instruction) and multiple data (each processing unit can operate on varying data elements). Data-level parallelism is accomplished when each processor performs the same task on different data inputs. In some circumstances, one thread may control operations on several different pieces of data (as shown in Figure 7.6). In other situations, threads may control the operation, however they may execute the same code.

Data-level parallelism emphasizes the distributed (parallelized) nature of the data, as opposed to the processing (task parallelism). Most modern computers, particularly those with graphics processor units (GPUs), employ SIMD instructions and execution units.

7.2.6 Task-Level Parallelism

Task-level parallelism in a multiprocessor system is synonymous with Flynn’s MIMD architecture, which occurs when each CPU executes a different process on

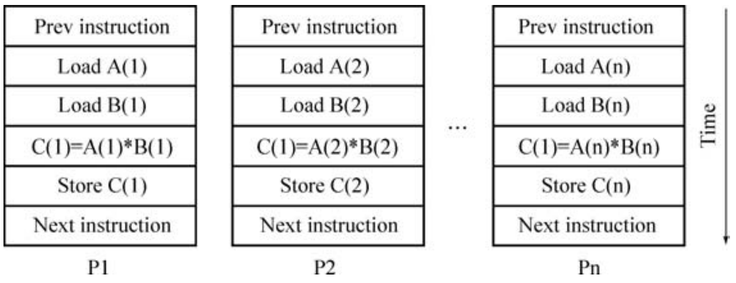


Figure 7.6 Data-level parallelism

the same or different data. The varying threads can execute the same or completely different code. In any case, however, varying threads must communicate with one another as they work. Communication takes place usually to pass data from one thread to the next as part of a workflow. Figure 7.7 shows a diagram of task-level parallelism.

Task parallelism emphasizes the distributed (parallelized) nature of the processing (i.e., threads), as opposed to the data (data parallelism). Most real programs fall somewhere on a continuum between task parallelism and data parallelism. Most supercomputers fall into this category.

7.2.7 Memory in Parallel Processing

In regard to parallelism, memory can either be shared or distributed. Shared memory typically is comprised of one of the following two architectures; Uniform Memory Access (or UMA) or Non-uniform Memory Access (or NUMA). Regardless of the specific architecture, shared memory is generally accessible to all the processors in the system and multiple processors may operate independently and continue to share the same memory.

Figure 7.8 illustrates UMA architecture. Multiple CPUs are capable of accessing one memory resource. This is also referred to as Symmetric Multiprocessor (or SMP). In contrast to this architecture is NUMA. NUMA is often made by physically

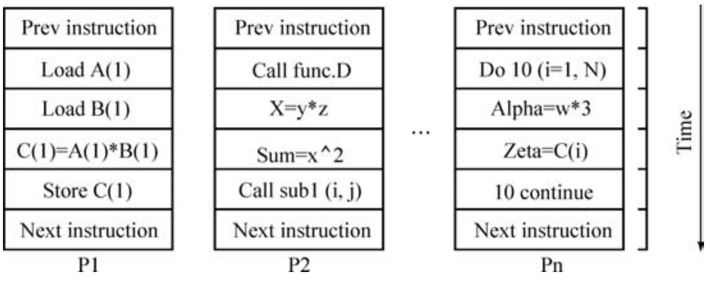


Figure 7.7 Task-level parallelism

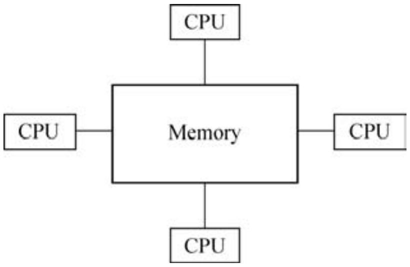


Figure 7.8 Shared memory system

linking two or more SMPs. One SMP is capable of directly accessing the memory of another SMP. An example of NUMA is pictured on the right.

Like shared memory systems, distributed memory systems (as shown in Figure 7.9) vary widely but share a common characteristic. Distributed memory systems require a communication network to connect inter-processor memory. In this type of architecture, processors have access to their own memory and do not share memory with another SMP. If a processor needs information from another data store, it has to communicate how and when it is to be accessed. This sharing generally comes about through a simple Ethernet connection. Hybrids of this technology have also been developed and are referred to as Distributed-Shared Memory architecture. However, this type of architecture is generally used in supercomputers.

7.2.8 Specialized Parallel Computers

Within parallel computing, there are devices that provide a niche market because of its capabilities. As previously discussed, a system may have multiple cores, but may continue to be limited because of the amount of memory available to those cores. General purpose computing on graphics processing units (GPGPU) is an inexpensive means of adding additional processing ability to a computer. A very beneficial and generous amount of memory is attached to video cards that are only accessible to its GPU. GPGPUs address the multiple cores and memory limitations as an add-on card to a single computer system. As long as a motherboard has available PCI-Express x16 slot available, it will be capable of hosting a very powerful GPGPU.

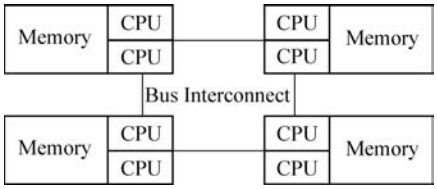


Figure 7.9 Distributed system

The front runners of the GPGPU market belong to Nvidia and AMD. Their graphics cards are particularly useful because it is designed to compute massive amounts of graphics and operate linear algebra matrices. The table below provides a very brief glimpse at the applications of the areas where GPUs are being used in computing:

- Physical based simulation and physics engines (usually based on Newtonian physics models).
- Weather forecasting.
- Global illumination – ray tracing, photon mapping, radiosity among others, sub-surface scattering.
- Fast Fourier transform.
- Medical Imaging.

7.2.9 The Future of Parallel Processing

The exponential growth of the personal computers will continue as competition for business in the industry persists. Many computer enthusiasts vie for the titles of world's fastest or best for fame or simple recognition. The next hurdle for computer enthusiasts would be to break the exaflops (10^{18}) barrier. That is a supercomputer capable of achieving a million trillion calculations per second. In order to achieve this massive parallel processing would need to be incorporated. Multiple computer nodes, with multiple processors, with multiple GPUs, attached to some form of dynamic RAM would accomplish this task.

With speed increases and increases in the number of cores present in a system, it may be possible for dedicated cores to ceaselessly navigate networks for intrusions and eliminate threats before they have a chance to propagate on a network. von Neumann's architecture has been around for so long, it's only a matter of time before a competing architecture becomes main stream and capable of disrupting hackers and threats. With this in mind, many new types of parallelism will be in our future with regard to this new up-and-coming architecture.

7.3 Ubiquitous Computing

Ubiquitous computing (**ubicomp**) is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. In the course of ordinary activities, someone "using" ubiquitous computing engages many computational devices and systems simultaneously, and may not necessarily even be aware that they are doing so. This model is usually considered an advancement from the desktop paradigm. More formally ubiquitous computing is defined as machines that fit the human environment instead of forcing humans to enter theirs. Figure 7.10 illustrates a diagram of ubicomp.

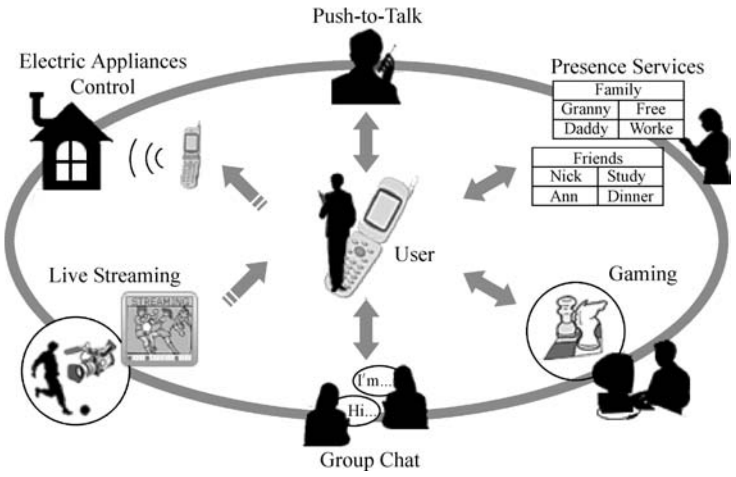


Figure 7.10 Ubiquitous computing

This paradigm is also described as pervasive computing, ambient intelligence, where each term emphasizes slightly different aspects. When primarily concerning the objects involved, it is also physical computing, the Internet of Things, haptic computing, and things that think. Rather than propose a single definition for ubiquitous computing and for these related terms, a taxonomy of properties for ubiquitous computing has been proposed, from which different kinds or flavors of ubiquitous systems and applications can be described.

7.3.1 Ubiquitous Computing Development

Intel's curiosity about how people use technology in cars is hardly surprising. Car makers are keen to install extra computing power in their vehicles in order to impress customers with a taste for technology, and Intel hopes that this will translate into a big new market for its chips. Ford, for instance, has already developed a service called SYNC, based on a Microsoft operating system. SYNC allows drivers to make calls, play music and do other things using voice commands. The car company has also created AppLink, a feature that lets people link their smartphones to a vehicle's voice-control system and operate their apps with it. For now the system works with only a handful of apps, such as Pandora, an Internet-radio service, but Ford is hoping to expand that number rapidly.

Japan's Toyota has also been working on an in-car system, called Entune, to which drivers will be able to connect their smartphones via Bluetooth wireless links and other means. And it plans to make driving even more personal by helping people's cars "talk" to them. The firm has announced plans for a Twitter-like private social network, called Toyota Friend, which will be integrated into some electric and hybrid vehicles in Japan. Based on software from Salesforce.com and Microsoft, this will enable a car to send a tweet-like message to its owner telling him that, say, its battery is running low or a maintenance check is due. People foresee many more product social networks that will create more intimate relationships between people and the devices they own.

It is not just vehicles that are becoming more connected. So are homes, public places like sports stadiums and even aircraft, where passengers are now sometimes offered in-flight Wi-Fi services for an extra charge. Cisco reckons that there could be almost 15 billion devices linked to the Internet in circulation by 2015, up from 7.5 billion in 2010. These will include everything from televisions and gaming consoles to coffee machines and cookers.

At their core, all models of ubiquitous computing share a vision of small, inexpensive, robust networked processing devices, distributed at all scales throughout everyday life and generally turned to distinctly common-place ends. For example, a domestic ubiquitous computing environment might interconnect lighting and environmental controls with personal biometric monitors woven into clothing so that illumination and heating conditions in a room might be modulated, continuously and imperceptibly. Another common scenario posits refrigerators "aware" of their suitably tagged contents, able to both plan a variety of menus from the food actually on hand, and warn users of stale or spoiled food.

7.3.2 Basic forms of Ubiquitous Computing

Ubiquitous computing presents challenges across computer science: in systems design and engineering, in systems modeling, and in user interface design. Contemporary human-computer interaction models, whether command-line, menu-driven, or GUI-based, are inappropriate and inadequate to the ubiquitous case. This suggests that the "natural" interaction paradigm appropriate to a fully robust ubiquitous computing has yet to emerge – although there is also recognition in the field that in many ways we are already living in an ubicomp world. Contemporary devices that lend some support to this latter idea include mobile phones, digital audio players, radio-frequency identification tags, GPS, and interactive whiteboards.

Mark Weiser proposed three basic forms for ubiquitous system devices, see also Smart device: tabs, pads and boards.

- *Tabs:* wearable centimeter sized devices
- *Pads:* hand-held decimeter-sized devices
- *Boards:* meter sized interactive display devices.

These three forms proposed by Weiser are characterized by being macro-sized, having a planar form and on incorporating visual output displays. If we relax each of these three characteristics we can expand this range into a much more diverse and potentially more useful range of Ubiquitous Computing devices.

7.3.3 Augmented Reality

Ubiquitous computing is roughly the opposite of virtual reality. Where virtual reality puts people inside a computer-generated world, ubiquitous computing forces the computer to live out here in the world with people. Virtual reality is primarily a horse power problem; ubiquitous computing is a very difficult integration of human factors, computer science, engineering, and social sciences.

Augmented reality (AR) is a live, direct or indirect, view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented) by a computer. As a result, the technology functions by enhancing one's current perception of reality. By contrast, virtual reality replaces the real world with a simulated one. An example of augmented reality is shown in Figure 7.11.

Augmentation is conventionally in real-time and in semantic context with environmental elements, such as sports scores on TV during a match. With the help of advanced AR technology (e.g., adding computer vision and object recognition) the information about the surrounding real world of the user becomes interactive and digitally manipulable. Artificial information about the environment and its objects can be overlaid on the real world. The term augmented reality is believed to have been coined in 1990 by Thomas Caudell, working at Boeing.

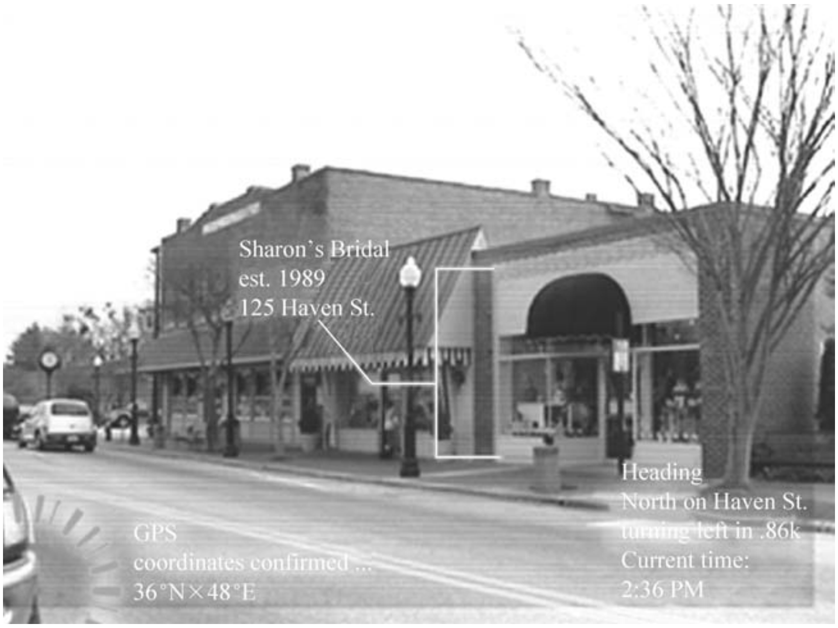


Figure 7.11 Augmented reality

Research explores the application of computer-generated imagery in live-video streams as a way to enhance the perception of the real world. AR technology includes head-mounted displays and virtual retinal displays for visualization purposes, and construction of controlled environments containing sensors and actuators.

7.3.4 Mobile Computing

The appearance of full-function laptop computers and wireless LANs in the early 1990s led researchers to confront the problems that arise in building a distributed system with mobile clients. The field of mobile computing was thus born. Although many basic principles of distributed system design continued to apply, four key constraints of mobility forced the development of specialized techniques. These constraints are: unpredictable variation in network quality, lowered trust and robustness of mobile elements, limitations on local resources imposed by weight and size constraints, and concern for battery power consumption.

Figure 7.12 shows a data collection and analysis system that can collect frequency, duration, accuracy and fluency data on the fly without requiring network connections. The system is able to run on different types of handheld devices including iPhone, iPad, Android and HP iPAQ and so on. It was from a Department of Education grant awarded to one of the book authors.

Mobile computing is still a very active and evolving field of research, whose body of knowledge awaits codification in textbooks. The results achieved so far can be grouped into the following broad areas:

- Mobile networking, including Mobile IP, *ad hoc* protocols, and techniques for improving TCP performance in wireless networks.

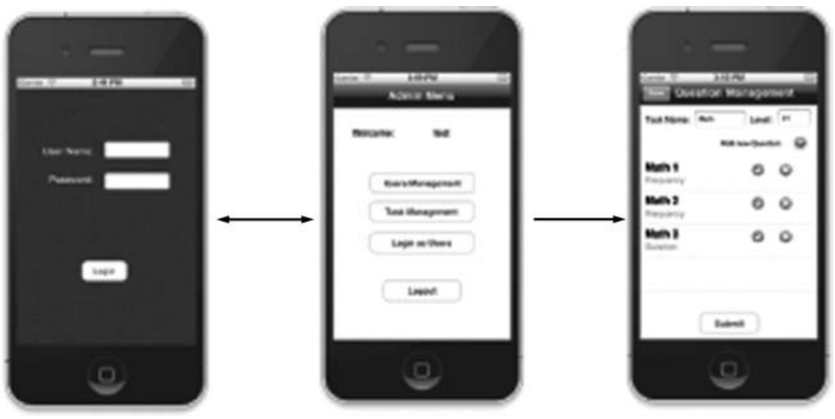


Figure 7.12 A standalone app to collect behavioral data

- Mobile information access, including disconnected operation, bandwidth-adaptive file access, and selective control of data consistency.
- Support for adaptive applications, including transcoding by proxies and adaptive resource management.
- System-level energy saving techniques, such as energy-aware adaptation, variable-speed processor scheduling, and energy-sensitive memory management.
- Location sensitivity, including location sensing and location-aware system behavior.

A future in which computers become pervasive, unobtrusive and almost invisible is being brought a step closer. Recent development shows that demonstrator and prototype sensor networks mark an important step forward in the cutting-edge field.

7.4 Grid, Distributed and Cloud Computing

Grid computing is a term referring to the federation of computer resources from multiple administrative domains to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. What distinguishes grid computing from conventional high performance computing systems such as cluster computing is that grids tend to be more loosely coupled, heterogeneous, and geographically dispersed. Although a grid can be dedicated to a specialized application, it is more common that a single grid will be used for a variety of different purposes. Grids are often constructed with the aid of general-purpose grid software libraries known as middleware.

Grid size can vary by a considerable amount. Grids are a form of distributed computing whereby a "super virtual computer" is composed of many networked loosely coupled computers acting together to perform very large tasks. For certain applications, "distributed" or "grid" computing, can be seen as a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface, such as Ethernet. This is in contrast to the traditional notion of a supercomputer, which has many processors connected by a local high-speed computer bus.

Grid computing combines computers from multiple administrative domains to reach a common goal, to solve a single task, and may then disappear just as quickly.

7.4.1 Characteristics of Grid Computing

One of the main strategies of grid computing is to use middleware to divide and apportion pieces of a program among several computers, sometimes up to many thousands. Grid computing involves computation in a distributed fashion, which may also involve the aggregation of large-scale cluster computing-based systems. Figure 7.13 shows a diagram of grid computing.

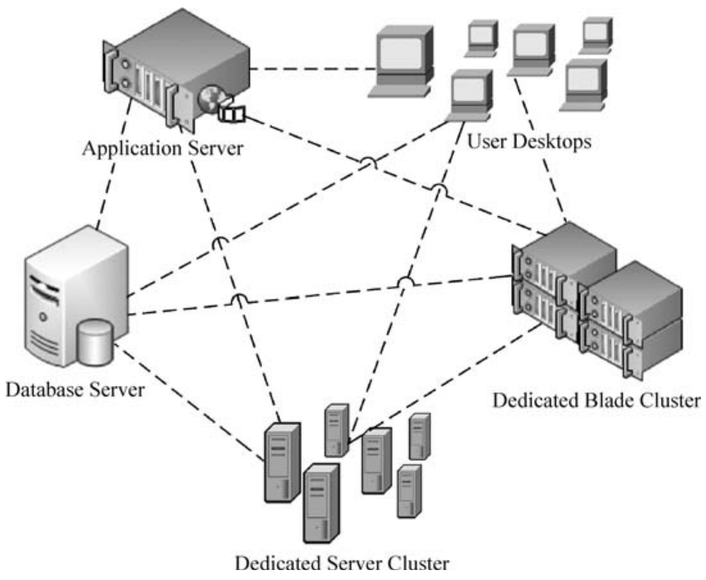


Figure 7.13 Grid computing

The size of a grid may vary from small – confined to a network of computer workstations within a corporation, for example – to large, public collaborations across many companies and networks. "The notion of a confined grid may also be known as an intra-nodes cooperation while the notion of a larger, wider grid may thus refer to an inter-nodes cooperation."

Grids are a form of distributed computing whereby a "super virtual computer" is composed of many networked loosely coupled computers acting together to perform very large tasks. This technology has been applied to computationally intensive scientific, mathematical, and academic problems through volunteer computing, and it is used in commercial enterprises for such diverse applications as drug discovery, economic forecasting, seismic analysis, and back office data processing in support for e-Commerce and Web services.

"Distributed" or "grid" computing in general is a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface, such as Ethernet. This is in contrast to the traditional notion of a supercomputer, which has many processors connected by a local high-speed computer bus.

7.4.2 The Advantages and Disadvantages of Grid Computing

The primary advantage of distributed computing is that each node can be purchased as commodity hardware, which, when combined, can produce a similar computing resource as multiprocessor supercomputer, but at a lower cost. This is due to the economies of scale of producing commodity hardware, compared to the lower efficiency of designing and constructing a small number of custom supercomputers. The primary performance disadvantage is that the various processors and local storage areas do not have high-speed connections. This arrangement is thus well-suited to applications in which multiple parallel computations can take place independently, without the need to communicate intermediate results between processors. The high-end scalability of geographically dispersed grids is generally favorable, due to the low need for connectivity between nodes relative to the capacity of the public Internet.

There are also some differences in programming and deployment. It can be costly and difficult to write programs that can run in the environment of a supercomputer, which may have a custom operating system, or require the program to address concurrency issues. If a problem can be adequately parallelized, a "thin" layer of "grid" infrastructure can allow conventional, standalone programs, given a different part of the same problem, to run on multiple machines. This makes it possible to write and debug on a single conventional machine, and eliminates complications due to multiple instances of the same program running in the same shared memory and storage space at the same time.

7.4.3 Distributed Computing

Distributed computing is a field of computer science that studies distributed systems. A distributed system consists of multiple autonomous computers that communicate through a computer network. The computers interact with each other in order to achieve a common goal. A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs.

Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers.

The word distributed in terms such as "distributed system," "distributed programming," and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing.

7.4.4 Distributed Systems

While there is no single definition of a distributed system, the following defining properties are commonly used:

- There are several autonomous computational entities, each of which has its own local memory.

- The entities communicate with each other by message passing.
- Have a common goal, such as solving a large computational problem.
- Each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users.
- The system has to tolerate failures in individual computers.
- The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program.

7.4.5 Parallel and Distributed Computing

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing," "parallel computing," and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed;" the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly-coupled form of distributed computing, and distributed computing may be seen as a loosely-coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

In parallel computing, all processors have access to a shared memory. Shared memory can be used to exchange information between processors.

In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

The situation is further complicated by the traditional uses of the terms parallel and distributed algorithm that do not quite match the above definitions of parallel and distributed systems; see the section on theoretical foundations below for more detailed discussion. Nevertheless, as a rule of thumb, high-performance parallel computation in a shared-memory multiprocessor uses parallel algorithms while the coordination of a large-scale distributed system uses distributed algorithms.

7.4.6 Distributed Computing Architectures

Various hardware and software architectures are used for distributed computing. At a lower level, it is necessary to interconnect multiple CPUs with some sort of network, regardless of whether that network is printed onto a circuit board or made up of loosely-coupled devices and cables. At a higher level, it is necessary to interconnect

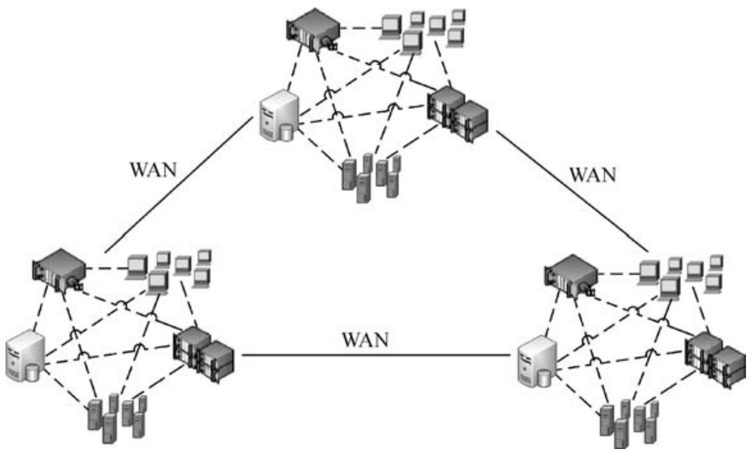


Figure 7.14 Distributed computing

processes running on those CPUs with some sort of communication system. Figure 7.14 shows a diagram of distributed computing.

Distributed programming typically falls into one of several basic architectures or categories: client–server, 3-tier architecture, n-tier architecture, distributed objects, loose coupling, or tight coupling.

Client–server: Smart client code contacts the server for data then formats and displays it to the user. Input at the client is committed back to the server when it represents a permanent change.

3-tier architecture: Three tier systems move the client intelligence to a middle tier so that stateless clients can be used. This simplifies application deployment. Most web applications are 3-tier.

N-tier architecture: n-tier refers typically to Web applications which further forward their requests to other enterprise services. This type of application is the one most responsible for the success of application servers.

Tightly coupled (clustered): refers typically to a cluster of machines that closely work together, running a shared process in parallel. The task is subdivided in parts that are made individually by each one and then put back together to make the final result.

Peer-to-peer: an architecture where there is no special machine or machines that provide a service or manage the network resources. Instead all responsibilities are uniformly divided among all machines, known as peers. Peers can serve both as clients and servers.

Space based: refers to an infrastructure that creates the illusion (virtualization) of one single address-space. Data are transparently replicated according to application needs. Decoupling in time, space and reference is achieved.

Another basic aspect of distributed computing architecture is the method of communicating and coordinating work among concurrent processes. Through various message passing protocols, processes may communicate directly with one another, typically in a master/slave relationship. Alternatively, a “database-centric” architecture can enable distributed computing to be done without any form of direct interprocess communication, by utilizing a shared database.

7.4.7 Cloud Computing

Cloud computing is an evolution from the notion of grid computing, which had come into vogue in the early 2000s, and involved the sharing of remote computing assets in a high performance computing environment. However, while an evolutionary technology, cloud computing is a disruptive technology, meaning it has the ability to radically transform an accepted and time-tested business practice. The disruptive aspect of cloud computing will occur most dramatically at the enterprise level, where cloud computing will alter the premise that large enterprises purchase their own computer hardware, develop proprietary software, and build large private networks to facilitate access to its data. In order to succeed, cloud computing must address multiple concerns before widespread acceptance becomes a market reality, namely issues surrounding access, security, data migration, resiliency, and regulatory. The cloud computing providers must also develop and refine their individual business models and promote transparency to address the many lingering concerns. Figure 7.15 illustrates a conceptual diagram of a cloud computing model.

The National Institute of Standards and Technology (NIST), in an effort to classify the cloud computing industry, has defined the three service models within the cloud computing industry as: i) Infrastructure as a Service (**IaaS**), which involves provisioning computer resources for customers on demand; ii) Software as a Service (**SaaS**), involving access to applications running on a cloud platform; and iii) Platform as a Service (**PaaS**), involving the ability to provision cloud infrastructure for a user’s company/personal requirements. The key to cloud computing is utilizing virtualization to provide de facto increases in computing power.

- IaaS (Infrastructure as a Service)
  This is the most basic of the cloud service models. Resources such as processing, storage, networks, and other fundamental computing can be provisioned by the consumer. The consumer would not have control of the cloud infrastructure but the use of the operating system or software applications.
- SaaS (Software as a Service)
  The highest layer in the cloud service model is SaaS (Software as a Service). With the use of this model the customer is purchasing the use of the provider’s application which is already running on the cloud infrastructure. Some examples of this are NetSuite and SalesForce.com.

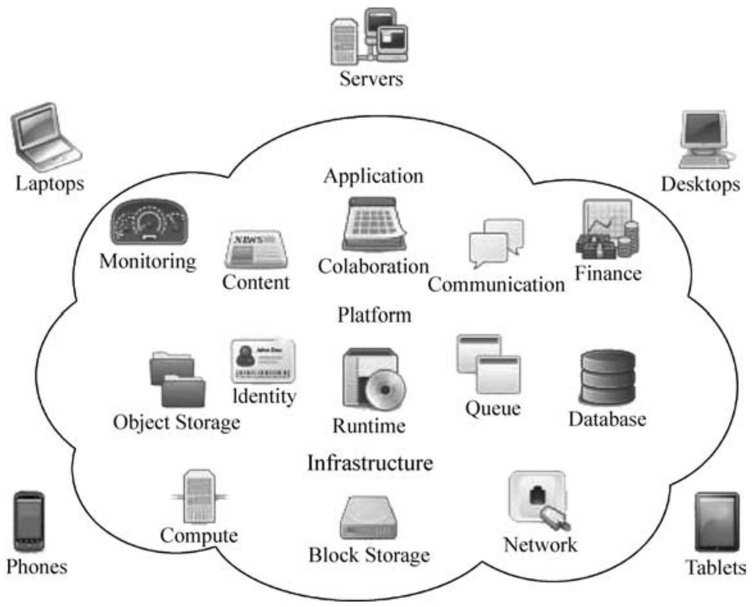


Figure 7.15 Cloud computing diagram

- PaaS (Platform as a Service)
  The next layer PaaS is where the customer purchases or creates an application that is fully functional to be placed on top of their infrastructure. Examples of this would be application stacks: Ruby on Rails, Java, or LAMP. Unlike IaaS, PaaS customers do not control or manage the cloud infrastructure, including network, servers, operating systems, or storage.

Utilizing the below cloud depiction, a "public cloud" is an example of SaaS, Amazon's Simple Storage solution is an example of PaaS, and the Amazon Elastic Compute Cloud would represent an example of IaaS. Of course, ceding one's control of a software service can be a dubious proposition for a firm, and factors such as technical feasibility, security, cloud resources, and transparency of the cloud service provider's (CSP) network policies and practices is paramount.

7.4.8 Technical Aspects of Cloud Computing

Cloud computing is a type of "parallel and distributed system," consisting of a collection of interconnected and virtualized computers, that are dynamically provisioned and presented as one or more unified computing resource. The computing relationship is based on service-level agreements established through negotiation between the CSP and the user, be it an enterprise, a government, an individual, or some other type of user. The cloud infrastructure typically consists of commodity hardware, such as blade servers, and specialized programs for virtualization and resource management. Cloud computing utilizes a pay as you go (or pay what you use) business model, and involves the dynamic provisioning of resources based on the requirements of the user, and the capability of the system.

Cloud computing incorporates virtual machine (VM) to manage its resources. While a mainframe computer may share the same operating system (O/S), its processing is treated as if there were multiple central processing units (CPU) inside the same computer. In effect, that is what happens, but in fact, the computer's central resources, such as memory, CPU, and arithmetic/logic unit (ALU), are being shared.

IaaS is a growth field within the research and computing-intensive scientific community. The use of virtualization with scientific research illustrates one of the useful applications of IaaS. Scientific research, which may utilize high throughput computing (HTC) workloads, often involves the use of a great deal of computer processing power, and such infrastructure is very expensive. Cloud computing, via its virtualization and pay as you go business model, presents a compelling case for such research. Virtualization permits encapsulating complex research applications found in scientific research, and often, depending on CPU usage, will have little performance degradation. CSPs utilize a scheduler to manage work flows and resource requests from their client base. This scheduler, incorporating a series of algorithms to include service level agreements, job prioritization, CPU utilization and so on, will queue individual jobs.

7.4.9 Security Aspects of Cloud Computing

Cloud computing, by its very nature, introduces a host of security challenges, both for the cloud service provider, and the users. At its core, the fundamental argument one might make is that an enterprise's data no longer resides on its servers within its domain, but rather on an amorphous platform, and is difficult, if not impossible, to know where the data actually resides. This implies an element of trust to the CSP that may or may not be warranted.

Ironically, cloud computing is often far more secure than traditional computing, because large cloud service providers, such as Google and Amazon, can attract and retain cyber security personnel of a higher quality than many companies or governmental agencies. Government employees commonly access cloud services such as Dropbox and Gmail in their personal lives. The use of such services by government employees on government computers may create a de facto security vulnerability, thus a case could be made that formally adopting such a service would be inherently more secure.

There are significant compliance aspects to the adoption of cloud computing, particularly for financial firms such as banks and insurance firms, and health care providers such as hospitals. There are certain baseline principles applicable to enterprises that collect, use, store or share customer information. It is the onus of the user entity to ensure its business practices and control of data is compliant with the relevant regulatory bodies/acts, and thus incumbent upon them to ensure their CSP likewise provides infrastructure, software, and security to ensure such compliance. However, CSPs may be reluctant to open up their infrastructure to those requiring access to conduct such information audits. An audit could be disruptive to a business model, and could violate agreements with customers to carefully restrict access to the data hierarchy, and lead to a denigration of service to other customers. Overall, this could impose a “brake” on the overall growth of the industry. At a minimum, customers will need to obtain industry standard certifications as to the adequacy of the internal controls of the provider and must insist on the ability of the regulators to conduct audits of the provider as may be required by such regulators.

In the European Union, the Payment Card Industry Data Security Standard (PCI DSS) is a major consideration for those CSPs working within the financial sector. This obligation is usually contained within the merchant agreements executed by the company and the various payment card brands. If payment card information is to be stored in the cloud, to remain PCI DSS compliant, a company will need to ensure that its service providers are PCI DSS compliant. For instance, the Gramm-Leach-Bliley Act (GLB Act) regulates the privacy and information security practices of financial institutions, including the obligation to disclose information on the means used to protect customer data. Similarly, the Health Insurance Portability and Accountability Act (HIPAA) addresses compliance aspects of personal health information.

The physical location of sensitive personal data is of paramount importance. Different countries have different regulatory and privacy issues for the use and storage of personal information. For example, privacy issues are much more sensitive in European Union countries than in the United States, thus storage of information on European citizens on servers located in the United States would be potentially problematic. Cloud services may result in customer data being replicated and/or separated into small portions and stored on multiple servers. If data is sent across international boundaries, privacy and data security concerns related to such cross-border data can be triggered. It is important for a company to know where these servers are located and whether the cloud computing service provider offers the capability to limit the transfer of data across specific or all international boundaries.

7.4.10 Ongoing and Future Elements in Cloud Computing

Moore’s law states that the number of transistors on a chip, or transistor density, doubles every 24 months. However, as transistor size decreases and transistor density and computing power increases, the heat generated by the chip becomes a major problem and multi-core processors become important. Consequently, the major chip manufacturers are now looking at doubling CPU performance by increasing the number of CPU cores (dual-core and quad-core processors), as against doubling the clock-speed of a single CPU. For applications to make effective use of these multiple cores, parallel programming code has to be written. Data centers housing cloud infrastructures are frequently made-up of thousands of such multi-core CPUs. Thus, cloud access to parallel applications deployed in such multi-core machines will generally enhance users' experience and has potential to positively influence Cloud adoption.

Cisco believes three areas are of paramount importance to ensure cloud computing's acceptance: security, service level agreements, and interoperability. Isolating network traffic via partitioning would increase security, latency and quality of service (QoS) must be addressed in service level agreements, and different types of data and data access must be included in discussions about interoperability.

7.4.11 Adoption of Cloud Computing Industry Drivers

Cloud computing is in its infancy, and new applications for the use of cloud computing are discovered regularly. One consideration is to consolidate on new, offshore data centers, which would be located where energy costs are low. Some US banks are looking seriously at rotating their heavy portfolio and risk calculations around the world, using grid computing to exploit cheaper and renewable sources of electricity, while other banks are considering locating server farms in places such as Iceland with renewable geothermal power. Such green initiatives are well suited to cloud computing, which potentially lends itself to reduced energy use and concomitant reduction in green house emissions.

Green computing is environment-friendly computing and one of its objectives is to make efficient use of electricity. Cloud computing infrastructures are generally deployed in very large data centers. For example, Google-designed data centers use about half the energy of a typical data center. People have increasingly become aware of the effects of computers on the environment, and this could be one of the key factors for broad acceptance of cloud computing. Other examples cite a European bank reduced its server count from 3,100 to 1,150 via virtualization, with a resultant 92% reduction in power usage. The Bank of Canada similarly virtualized 1,500 small to mid-range servers, with a tremendous savings due to the consolidation, greater resilience, and performance. The bank still has far to go with over 8,000 servers in total, running Solaris, Linux, and SQL databases.

Currently, major chip manufacturers are now focused on dual-core and quad-core processors. Concurrent development of software programming code utilizing parallel processing, coupled with virtualization, would yield significant performance improvements. Indeed, manufacturers such as Intel look to double CPU performance by increasing the number of CPU cores. Data centers housing cloud infrastructures are frequently made-up of thousands of such multi-core CPUs and have the potential to greatly enhance performance.

Initiatives to utilize low cost computers for less developed countries depend on cloud computing for storage and application resources. One such program is being developed in India. In order to keep costs down the LCACDs generally do not have in-built hard drives; data are stored on memory cards (much like the SD card used in smart phones) and/or inbuilt flash drives. Thus, the very configuration (this includes networking support, Internet browsing, media playing capabilities) of such devices make it ideal for cloud usage. Applications such as Google docs can also be key to the effective utilization of such devices. Another example is the Massachusetts Institute of Technology Media La and its development of low cost computing devices for developing countries. This work is based on the same principle, that is, moving data onto the Web and off the laptop/desktop.

Cloud computing has the potential to be a significant economic driver of business enterprise growth in the coming years. While there are significant obstacles to widespread adoption, the significant technology companies that have adopted cloud computing as a business model, provide the perfect opportunity for this initiative to succeed. While there are many obstacles yet to overcome, namely issues related to security, transparency, and interoperability, the industry is already well on its way to addressing those obstacles, and producing tangible mechanisms to address those shortcomings. The momentum the industry has at present is likely to accelerate over the next several years, until such time that it becomes standard business practice throughout the large, computationally intensive industry sectors, and adoption proceeds down the scale to individual users and small entities.

7.5 Internet Computing

Internet computing can gain significant flexibility and agility with migrating sensitive data into remote, worldwide data centers. Integration of documents on Web-based office suites can simplify processes and create a more efficient way of doing business. It will allow for cost savings, easy access and streamline most daily activities. Internet computing powering can also support different aspects of businesses, which also utilize Internet computing with forms accessible by Web-applications that electronically support online collaboration in real-time with other users. The information and communication systems used, whether networked or not, serves as a media to implement a business process. The objective of this paper is to discuss the benefits of an Internet computing environment that improves agility, integrating modern day technology (e.g., iPhone, iPad, Web interface phone etc.) and an example of Internet and computing online documents. It begins with an introduction of Internet computing by definition and its elements, referring to business benefits of utilizing of Internet computing, examples of Internet computing such as Web based online office suites and the integration of Internet computing and modern day technology.

7.5.1 Internet Computing Concept and Model

Internet computing is a model similar to the cloud computing model that enables convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, application and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Figure 7.16 shows a diagram of Internet computing.

The Internet computing model promotes availability and is composed of five essential characteristics, three service models and four deployment models.

- The Hardware Layer (Client) consists of computer hardware that relies on Internet computing for application delivery and that is in essence useless without it. Examples include some computers, phones and other devices, operating systems, and browsers.
- The Virtualization Layer can be viewed as part of an overall trend in enterprise IT that includes autonomic computing, a scenario in which the IT environment will be able to manage itself based on perceived activity, and utility computing, in which computer processing power is seen as a utility that clients can pay for only as needed.
- The IaaS Layer extends the virtualization layer by providing the mechanisms to provision and control the virtual machines in a utility computing manner.
- The PaaS Layer (Platform) extends and abstracts the IaaS layer by removing the hassle of managing individual virtual machine instances.

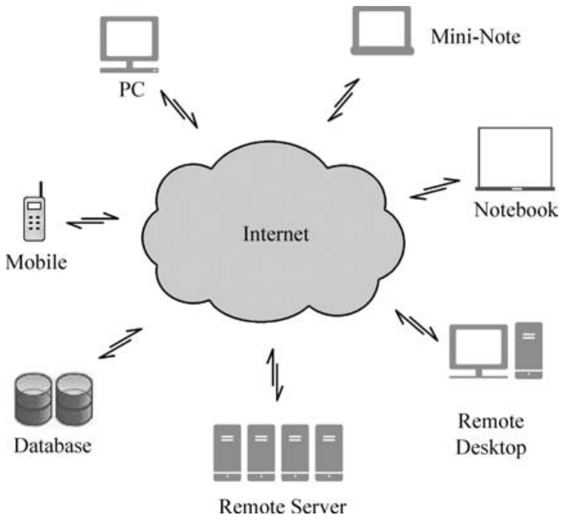


Figure 7.16 Internet computing

- The SaaS Layer (Application) sometimes referred to as "on-demand software," is a software delivery model in which software and its associated data are hosted centrally (typically in the cloud) and are typically accessed by users using a thin client, normally using a Web browser over the Internet.

A critical issue in implementing cloud computing is taking virtual machines, which contain critical applications and sensitive data, to public and shared environments. Below are the four deployment models.

- *Public* – describes computing in the traditional mainstream sense, whereby resources are dynamically provisioned to the general public on a fine-grained, self-service basis over the internet, via web applications/web services, from an off-site third-party provider who bills on a fine-grained utility computing basis.
- *Community* – shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally. the costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the benefits of cloud computing are realized.
- *Hybrid* – is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. it can also be defined as multiple cloud systems that are connected in a way that allows programs and data to be moved easily from one deployment system to another.
- *Private* – is infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally. they have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from lower up-front capital costs and less hands-on management, essentially lacking the economic model that makes cloud computing such an intriguing concept.

Each model comes with different options and a different price. A critical issue in implementing Internet computing is taking virtual machines, which contain critical applications and sensitive data, to public and shared environments. Internet computing offers obvious advantages, such as co-locating data with computations and an economy of scale in hosting the services.

The image below is an illustration of Internet computing components and shows a graphic depiction of how each element integrates with each other.

7.5.2 Benefit of Internet Computing for Businesses

Despite the fact that Internet computing is a relatively young concept with many questions still open, there is overwhelming consensus regarding the potential of this paradigm in advancing technology. New business solution such as Akamai Edge Computing Powered by WebSphere, will enable companies to supplement their existing IT infrastructure with virtualized Internet computing capacity on demand on a "pay-as-you-go" basis. A benefit of this business solution will allow data and programs to be swept up from desktop PCs and corporate server rooms and installed in "the compute Internet cloud." Implementing Internet computing will allow the elimination of the responsibility of updating software with each new release and configuration of desktops on installs. Since applications run in the cloud and not on individual computers or desktops, you don't need hard drive space or processing power demanded by traditional desktop software. Corporations do not have to purchase high powered personal computers (PC) and can purchase smaller hard disks, less memory and more efficient processors. Files will essentially be stored in the cloud, as well as software programs and this could assist in cutting overhead budgets substantially. The "Internet computing" trend of replacing software traditionally installed on business computers with applications delivered via the Internet is driven by aims of reducing IT complexity and cost.

The ease of collaborating and updating documents is also a benefit of implementing. Regardless of the location of the person, you will be able to access and update files upon demand. This will give access regardless of location, from any device, the power to collaborate, compare and clean Microsoft Office documents including Word, PowerPoint (including embedded objects) and Excel, as well as PDF files. The capabilities collaborating and sharing documents allow the same features as working on individual workstation. Some features are the following:

- Compare all Microsoft Office and native PDF documents, including complex Microsoft Word, Excel and PowerPoint documents with embedded objects and images.
- Compare Excel spreadsheets including both values and formulas.
- Remove metadata from all Microsoft Office and PDF documents to protect sensitive information and privacy.
- Convert Microsoft Office documents into a PDF file.
- Extract native PDF documents to Microsoft Word for editing or comparison and cleaning operations.
- Download resulting files online or have them sent to an e-mail account.

Businesses can also benefit from Internet computing using:

- Infrastructure – allows user to get the IT resources needed, only when needed. Pay only for what you use or additional storage.
- Flexibility – Adjust Internet computing-based resources up and down to meet real-time needs, or offload onsite data to the cloud as needed to improve operational efficiencies. And since this online-demand access is Internet-based, you can access

these resources from anywhere, supporting remote work and continuity of operations.
- Collaboration gives the ability to effectively communicate and collaborate to improve business process. In many cases, the spark to use Internet computing will come from rank-and-file employees themselves seeking to have access at work to the cloud services that they use in their personal lives. With both the application and the data stored in the cloud, it becomes easy for multiple users – located anywhere in the world – to work together on the same project.
- Disaster recovery/continuity of operations –With centralized data storage, management, and backups, data recovery in response to local business disruptions can be faster and easier.

Once an Internet protocol connection is established among several computers, it is possible to share services within any one of the aforementioned layers. Internet computing can completely transform business models and significantly improve a product, service or industry and in the process democratize them.

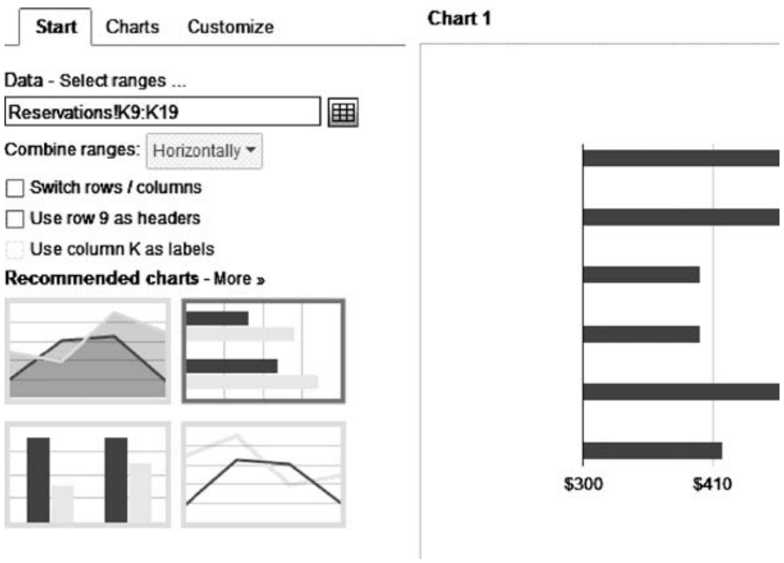
7.5.3 Examples of Internet Computing

Google App Engine enables you to build and host Web apps on the same systems that power Google applications. App Engine offers fast development and deployment; simple administration, with no need to worry about hardware, patches or backups; and effortless scalability. Google App Engine lets you run Web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve your users.

App Engine includes the following feature:

- dynamic Web serving, with full support for common Web technologies;
- persistent storage with queries, sorting and transactions;
- automatic scaling and load balancing;
- APIs for authenticating users and sending e-mail using Google Accounts;
- a fully featured local development environment that simulates Google App Engine on your computer;
- task queues for performing work outside of the scope of a Web request;
- scheduled tasks for triggering events at specified times and regular intervals.

Google Docs is a free, Web-based office suite, and data storage service offered by Google. It allows users to create and edit documents online while collaborating in real-time with other users. Google Apps collection such as Gmail, Google Calendar, Docs, Spreadsheets, Presentations and Drawings – with the latter four



Reservations!K9:K19

- [ ] Switch rows / columns
- [ ] Use row 9 as headers
- Use column K as labels

Figure 7.17 An example of Internet computing

collectively referred to as Google Documents. An example of Google spreadsheet is shown on Figure 7.17. Data storage of files up to 1 GB total in size was introduced on January 13, 2011, documents created inside Google Docs do not count towards this quota. Google Documents can store any type of file, up to 10 GB in size.

Google Docs is used simply as the G-Drive. Folders across multiple PCs and mobile devices just stay in sync automatically. Anything saved to these folders is automatically mirrored online and across all linked PCs.

7.5.4 Migrating Internet Computing

Internet computing also offers end users advantages in terms of mobility and collaboration. For the storage platform, all you need is a Web browser and an Internet connection to view your data. Migrating Internet (cloud) computing and mobile technology has allowed the ability to access data, emails or documents with just a push of a button. Google App Engine, Google's cloud computing platform, and the iPhone, Apple's mobile platform, has developed an application that synchronizes its data from "The Cloud." No longer do you just browse the Web, now you can fully interact with Internet solutions and you can create a desktop client solution that consumes data hosted in an Internet cloud service that can then be accessed from your phone. This marriage allows individuals to have access to whatever they want anytime from anywhere.

Internet computing relies on sharing computing resources rather than having local servers or personal devices to handle applications. It consists of four deployment models: Public, Community, Hybrid, and Private Cloud; and five layers: Hardware, Virtualization, IaaS, PasS, and SaaS. Not only does it support an efficient and productive environment, it also supports a convenient environment. Individuals are able to access their documents as needed anywhere on campus or on the current network. Businesses are benefiting from replacing traditional software with applications available via the Internet and utilizing Internet Computing. They are able to cut overhead costs by purchasing less software and upgrades. Cloud computing also supports integration with Web interface phones and other technology. The combination of Internet connected devices and Internet delivered services is the direction the industry is headed. It provides a platform that allows individuals to communicate from anywhere around the world. It's beneficial for working individuals who are on the go or need to collaborate with teammates anywhere in the world. Overall, Internet computing gives 24 hour access to needed documents and consistent involvement in a mobile office. This gives an opportunity for businesses to cut costs because they don't have to purchase as many desktops, hardware, and licenses for software. This gives the capability of investing funds in needed areas, for example, cutting edge technology. Gone are the days that the corporation communities have to set foot in the office and convene in a stuffy building. You essentially can be anywhere in the world and participate in a virtual conference room which give you the entire world as your business platform. Despite all the latest innovations in the area of cloud computing the reality remains that its current definition is somewhat limiting.

7.6 Virtualization

Virtualization is the creation of a virtual instance of an operating system, a server, a storage device or network resources. If you have ever divided your hard drive into different partitions, then you have used the virtualization techniques. This is an early form of virtualization. A partition is the logical division of a hard disk drive to create, in effect, two separate hard drives.

Operating system virtualization is the use of software to allow computer hardware to run multiple operating system images at the same time. The technology got its start on mainframes decades ago, allowing administrators to avoid wasting expensive processing power. Figure 7.18 shows a diagram of virtualization.

7.6.1 Types of Virtualization

Hardware virtualization or platform virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed

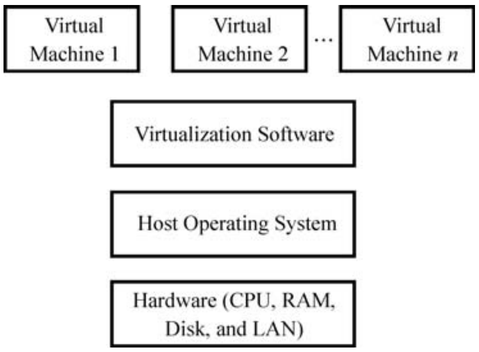


Figure 7.18 A diagram of virtualization

on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine.

In hardware virtualization, the host machine is the actual machine on which the virtualization takes place, and the guest machine is the virtual machine. The words host and guest are used to distinguish the software that runs on the actual machine from the software that runs on the virtual machine. The software or firmware that creates a virtual machine on the host hardware is called a hypervisor or Virtual Machine Monitor (VMM).

Virtualization software have been adopted faster than one ever imagined. There are three areas of IT where virtualization is making inroads: network virtualization, storage virtualization, and server virtualization.

- Network virtualization is a method of combining the available resources in a network by splitting up the available bandwidth into channels, each of which is independent from the others, and each of which can be assigned (or reassigned) to a particular server or device in real time. The idea is that virtualization disguises the true complexity of the network by separating it into manageable parts, much like your partitioned hard drive makes it easier to manage your files.
- Storage virtualization is the pooling of physical storage from multiple network storage devices into what appears to be a single storage device that is managed from a central console. Storage virtualization is commonly used in storage area networks (SANs).
- Server virtualization is the masking of server resources (including the number and identity of individual physical servers, processors, and operating systems) from server users. The intention is to spare the user from having to understand and manage complicated details of server resources while increasing resource sharing and utilization and maintaining the capacity to expand later.

7.6.2 History of Virtualization

In the mid 1960s, the IBM Watson Research Center was home to the M44/44X Project, the goal being to evaluate the then emerging time sharing system concepts. The architecture was based on virtual machines: the main machine was an IBM 7044 (M44) and each virtual machine was an experimental image of the main machine (44X). The address space of a 44X was resident in the M44's memory hierarchy, implemented via virtual memory and multi-programming.

IBM had provided an IBM 704 computer, a series of upgrades (such as the 709, 7090, and 7094), and access to some of its system engineers to MIT in the 1950s. It was on IBM machines that the Compatible Time Sharing System (CTSS) was developed at MIT. The supervisor program of CTSS handled console I/O, scheduling of foreground and background (offline-initiated) jobs, temporary storage and recovery of programs during scheduled swapping, monitor of disk I/O, and so on. The supervisor had direct control of all trap interrupts.

Around the same time, IBM was building the 360 family of computers. MITs Project MAC, founded in the fall of 1963, was a large and well-funded organization that later morphed into the MIT Laboratory for Computer Science. Project MACs goals included the design and implementation of a better time sharing system based on ideas from CTSS. This research would lead to Multics, although IBM would lose the bid and General Electric's GE 645 would be used instead.

Regardless of this "loss," IBM has been perhaps the most important force in this area. A number of IBM-based virtual machine systems were developed: the CP-40 (developed for a modified version of IBM 360/40), the CP-67 (developed for the IBM 360/67), the famous VM/370, and many more. Typically, IBMs virtual machines were identical "copies" of the underlying hardware. A component called the virtual machine monitor (VMM) ran directly on "real" hardware. Multiple virtual machines could then be created via the VMM, and each instance could run its own operating system. IBMs VM offerings of today are very respected and robust computing platforms.

7.6.3 Virtualization Architecture

Generically speaking, in order to virtualize, you would use a layer of software that provides the illusion of a "real" machine to multiple instances of "virtual machines." This layer is traditionally called the VMM. Figure 7.19 illustrates such an architecture.

There are many (often intertwined) high-level ways to think about a virtualization system's architecture. Consider some scenarios:

- A VMM could itself run directly on the real hardware – without requiring a "host" operating system. In this case, the VMM is the (minimal) OS.

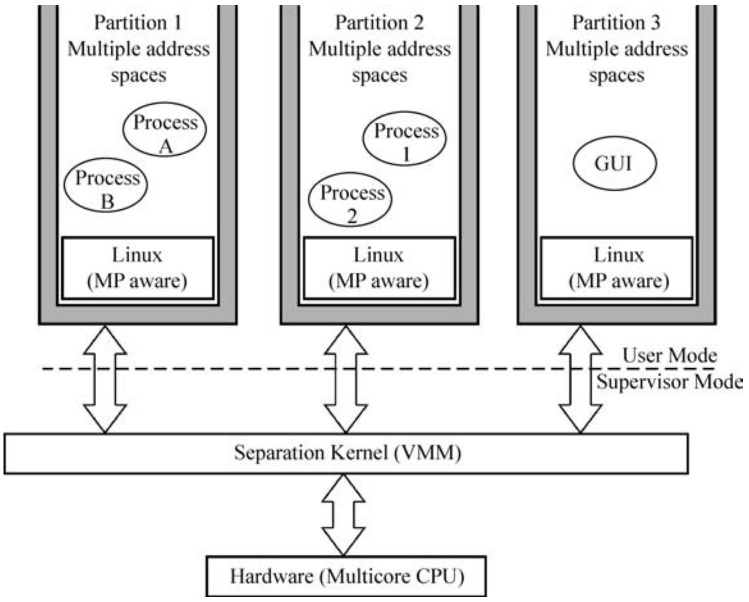


Figure 7.19 Virtualization architecture

- A VMM could be hosted, and would run entirely as an application on top of a host operating system. It would use the host OS API to do everything. Furthermore, depending on whether the host and the virtual machine's architectures are identical or not, instruction set emulation may be involved.

From the point of view of how (and where) instructions get executed: you can handle all instructions that execute on a virtual machine in software; you can execute most of the instructions (maybe even some privileged instructions) directly on the real processor, with certain instructions handled in software; you can handle all privileged instructions in software.

A different approach, with rather different goals, is that of complete machine simulation. SimOS and Simics, as discussed later, are examples of this approach.

Although architectures have been designed explicitly with virtualization in mind, a typical hardware platform, and a typical operating system, both are not very conducive to virtualization.

As mentioned above, many architectures have privileged and non-privileged instructions. Assuming the programs you want to run on the various virtual machines on a system are all native to the architecture (in other words, it would not necessitate emulation of the instruction set). Thus, the virtual machine can be run in non-privileged mode. One would imagine that non-privileged instructions can be directly executed (without involving the VMM), and since the privileged instructions would cause a trap (since they are being executed in non-privileged mode), they can be "caught" by the VMM, and appropriate action can be taken (they can be simulated by the VMM in software, say). Problems arise from the fact that there may be instructions that are non-privileged, but their behavior depends on the processor mode – these instructions are sensitive, but they do not cause traps.

7.6.4 Virtual Machine Monitor

When a process on a guest system running on a (hosted) virtual machine invokes a system call, it should not be handled by the host. The host should notify the guest operating system. One solution is for the virtual machine monitor to use *ptrace()* to trace process execution and identify system call entry. The VMM can then nullify the system call (say, by "converting" it to *getpid()*, or to an invalid system call), which is executed by the host. Upon system call exit, the VMM notifies the guest system kernel (through a signal, say), which can take appropriate action.

When a typical operating system kernel running on real hardware has nothing to do, it runs its idle thread, or loop. When the same kernel runs on a virtual machine, this behavior is undesirable, because the virtual machine is wasting its processor time. The virtual machine could have a mechanism to suspend itself, instead of running the idle loop. For example, the Denali Isolation Kernel uses a purely virtual instruction (idle-with-timeout) for this purpose.

7.6.5 Examples of Virtual Machines

VMware was founded in 1998. Its first product was VMware Workstation (1999). The GSX Server and ESX Server products were introduced in 2001.

VMware Workstation (as well as the GSX Server) has a hosted architecture: it needs a host operating system (such as Windows or Linux). In order to optimize the complex mix of performance, portability, ease of implementation, and so on, the product acts as both a virtual machine monitor (talking directly to the hardware), and as an application that runs on top of the host operating system. The latter frees the VMM from having to deal with the large number of devices available on the PCs (otherwise the VMM would have to include device drivers for supported devices).

VMW are Workstation's hosted architecture and includes the following components: a user-level application (VMApp), a device driver (VMDriver) for the host system, and a virtual machine monitor (VMM) that is created by VMDriver as it loads. Thereafter, an execution context can be either native (that is, the host's), or virtual (that is, belonging to a virtual machine). The VMDriver is responsible for switching this context. I/O initiated by a guest system is trapped by the VMM and forwarded to the VMApp, which executes in the host's context and performs the I/O using "regular" system calls. VMware uses numerous optimizations that reduce various virtualization overheads. GSX Server is also hosted, but is targeted for server deployments and server applications.

VMware ESX Server enables a physical computer to be available as a pool of secure virtual servers, on which operating systems can be run. This is an example of dynamic, logical partitioning. Moreover, ESX Server does not need a host operating system (like VMware workstation) – it runs directly on hardware (in that sense, it is the host operating system). ESX server was inspired by work on Disco and Cellular Disco, which virtualized shared memory multiprocessor servers to run multiple instances of IRIX. As mentioned earlier, the IA-32 architecture is not naturally virtualizable. Certain "sensitive" instructions must be handled by the VMM, and cannot be simply executed in non-privileged mode because they don't cause a General Protection exception. ESX Server solves this problem by dynamically rewriting portions of an operating system kernel's code to insert traps at appropriate places – in order to catch such sensitive instructions. ESX Server can run multiple virtual CPUs per physical CPU. Multiple physical network interface cards can be logically grouped into a single, high-capacity, virtual network device.

Mac-on-Linux, or simply MOL, is a virtual machine implementation that runs under Linux on most PowerPC hardware, and allows you to run Mac OS (7.5.2 to 9.2.2), Mac OS X, and Linux. Most of MOLs virtualization functionality is implemented as a kernel module. A user process takes care of I/O, and so on. There's even an (very limited) Open Firmware implementation within MOL.

Virtualization and the server consolidation that it delivers will be the top priority for chief information officers in 2012, according to a survey by the research firm IDC. Savings from server consolidation will be invested in new IT initiatives such as cloud computing, mobility, data analytics, and use of social media for business purposes.

When CIOs were asked by IDC to name their top three IT priorities for this year, nearly 40% of them picked virtualization and server consolidation, more than any other area of IT. After virtualization, investment in cloud services came in second, followed by collaboration tools, business analytics, and the consolidation of application portfolios.

Enterprises will also make more of an investment in "big data" analytics in 2012, IDC predicts, to analyze the petabytes of data their businesses generate in order to make business decisions.

"The final bricks in the foundation for the intelligent enterprise will be laid in 2012 in a number of key industries," IDC states, mentioning health care, utilities, and retail as industries making the most of analytics technology. Analytics jumped to number four on the list of CIO IT priorities for 2012, from ninth place in the previous survey.

IDC notes that the amount of data generated by enterprises is expected to grow by 48% this year and that 90% of it will be unstructured data. In the new, all-digital issue of InformationWeek Government: "As federal agencies close data centers, they must drive up utilization of their remaining systems. That requires a well-conceived virtualization strategy."

7.7 Biocomputers

Biocomputers use systems of biologically derived molecules, such as DNA and proteins, to perform computational calculations involving storing, retrieving, and processing data.

The development of biocomputers has been made possible by the expanding new science of nanobiotechnology. The term nanobiotechnology can be defined in multiple ways; in a more general sense, nanobiotechnology can be defined as any type of technology that uses both nano-scale materials, that is, materials having characteristic dimensions of 1–100 nanometers, as well as biologically based materials. A more restrictive definition views nanobiotechnology more specifically as the design and engineering of proteins that can then be assembled into larger, functional structures. The implementation of nanobiotechnology, as defined in this narrower sense, provides scientists with the ability to engineer biomolecular systems specifically so that they interact in a fashion that can ultimately result in the computational functionality of a computer.

7.7.1 Biochemical Computers

Biochemical computers use the immense variety of feedback loops that are characteristic of biological chemical reactions in order to achieve computational functionality. Feedback loops in biological systems take many forms, and many different factors can provide both positive and negative feedback to a particular biochemical process, causing either an increase in chemical output or a decrease in chemical output, respectively. Such factors may include the quantity of catalytic enzymes present, the amount of reactants present, the amount of products present, and the presence of molecules that bind to and thus alter the chemical reactivity of any of the aforementioned factors. Given the nature of these biochemical systems to be regulated through many different mechanisms, one can engineer a chemical pathway comprising a set of molecular components that react to produce one particular product under one set of specific chemical conditions and another particular product under another set of conditions. The presence of the particular product that results from the pathway can serve as a signal, which can be interpreted, along with other chemical signals, as a computational output based upon the starting chemical conditions of the system, that is, the input.

7.7.2 Biomechanical Computers

Biomechanical computers are similar to biochemical computers in that they both perform a specific output that can be interpreted as a functional computation based upon specific initial conditions which serve as input. They differ, however, in what exactly serves as the output signal. In biochemical computers, the presence or concentration of certain chemicals serves as the output signal. In biomechanical computers, however, the mechanical shape of a specific molecule or set of molecules under a set of initial conditions serves as the output. Biomechanical computers rely on the nature of specific molecules to adopt certain physical configurations under certain chemical conditions. The mechanical, three-dimensional structure of the product of the biomechanical computer is detected and interpreted appropriately as a calculated output.

7.7.3 Bioelectronic Computers

Biocomputers can also be constructed to perform electronic computing. Again, like both biomechanical and biochemical computers, computations are performed by interpreting a specific output that is based upon an initial set of conditions that serve as input. In bioelectronic computers, the measured output is the nature of the electrical conductivity that is observed in the bioelectronic computer, which comprises specifically designed biomolecules that conduct electricity in highly specific manners based upon the initial conditions that serve as the input of the bioelectronic system.

The new logic gates are formed from short strands of DNA and their complementary strands, which in conjunction with some simple molecular machinery mimic their electronic equivalent. Two strands act as the input: each represents a 1 when present or a 0 when absent. The response to their presence or absence represents the output, which can also be a 1 or 0.

Figure 7.20 is an "exclusive OR" or XOR logic gate. It produces an output when either of the two inputs is present but not when both are present or both are absent. To put the DNA version to the test, Willner and his team added molecules to both the complementary strands that caused them to fluoresce when each was present in isolation, representing a logical 1 as the output. But when both were present, the complementary strands combined and quenched the fluorescence, representing a 0 output. The inputs to an XOR logic gate are two complementary standards of DNA. If one or the other is present, the gate fluoresces, indicating an output of 1. If both are present, they bind together fluorescence, indicating an output of 0.

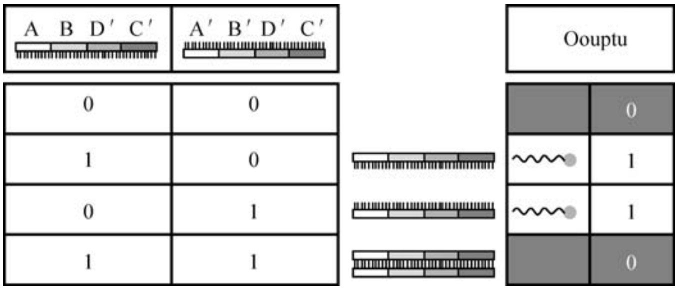


Figure 7.20 A biochemical logic gate

Currently, biocomputers exist with various functional capabilities that include operations of logic and mathematical calculations. Tom Knight of the MIT Artificial Intelligence Laboratory first suggested a biochemical computing scheme in which protein concentrations are used as binary signals that ultimately serve to perform logical operations. At or above a certain concentration of a particular biochemical product in a biocomputer chemical pathway indicates a signal that is either a 1 or a 0, and a concentration below this level indicates the other, remaining signal.

Many examples of simple biocomputers have been designed, but the capabilities of these biocomputers are still largely premature in comparison to commercially available non-bio computers. However, there is definitely great potential in the capabilities that biocomputers may one day acquire. Evidence of the true potential of the computing capabilities of biocomputers exists in the most powerful, complex computational machine known to currently exist: the biocomputer that is the human brain. Certainly, there is plenty of room to improve in the realm of biocomputer computational ability; one may reasonably expect the science of biocomputers to advance greatly in the years to come.

7.8 Summary

Advanced computer architecture including discussion of multiprocessors, multi-core processors, cache memory, parallel processing, multiprocessing, ubiquitous computing, mobile computing, grid computing, distributed computing, cloud computing, Internet computing, as well as other areas such as vector computing etc.

Readers may find out that many areas may not fall exactly in the architecture domain, such as parallel processing, cloud computing, and so on. However, those abovementioned computing models are all supported by various architectures. Nowadays, hardware and software are gradually merging. To draw a line between them is less important.

Cache is a type of fast memory, so using cache can improve the speed of a computer system. When a MISS happens, processors always move a block of data from memory to cache.

According to Flynn's Taxonomy, parallel processing computer systems can be classified into four categories: SISD (uniprocessor as common personal computers), SIMD (multiple processors perform the same operation on multiple data simultaneously often being referred to as data-level parallelism), MIMD (general multiprocessors and most common in parallel processing) and MISD (not popular).

Data dependency may occur for instruction-level parallelism when data being fetched currently have not been written back. This will affect the pipeline and reduce the overall performance. Read after write (RAW) is one type of data hazard that is attempting to read data before they are written back.

In parallel processing, memory can be all shared, all distributed or both. A system with both shared memory and distributed memory is common.

Ubiquitous computing is to enable users to access applications anywhere, anytime, and with any computing devices such as handheld, pads, tabs or boards. Augmented reality is the combination of virtual reality with real-time physical sensory. A solider is able to "see through" the buildings in front of him using the "super X-ray machine" is an example of augmented reality. The solider wears a head-mounted device seeing the virtual model inside the building with one eye and another eye sees the real world. The virtual scene inside the building is generated by a super computer gathering data from the sensor network inside the building. Those thousands of sensors can be placed in the building by a bomb to collect data and send them back to the super computer.

Grid computing, distributed computing, cloud computing, and Internet computing all have something in common. They are not new inventions. Each one is a combination of infrastructure, software and platform to provide various kinds of services to users.

In cloud computing, different organizations run applications (e.g., e-mail and calendar) on the same cloud. Security is the main concern even though there are different types of separations. Many originations still hesitate to move critical data (such as financial and marketing) into the public cloud.

A virtual machine is used to set up multiple computers on one physical computer. All virtual systems can share the resources with the host computer therefore providing the flexibility to test new applications and to run multiple platform applications on a single server.

Internet computing is a very promising area. With Google doc and other apps, users no longer need to purchase commercial word processing software. This may also lead to the potential wide use of thin clients. A thin client is a very simple computer with very limited computational power and memory. It is small, cheaper, fanless so there is no noise and it is "green." If more and more Internet-based applications come to earth, people can use thin clients to access all the applications easily without requiring new computers every two to three years. A thin client can be built very simply just to support a browser. In the near future, if you have a browser, you have a computer.

Exercises

7.1 What are the differences between multi-core processors and multi-processors?

7.2 Why can cache improve the performance of a computer system?

7.3 Hit rate is defined by the fraction of accesses found in cache. A processor has the miss rate of 1%. What does the number mean?

7.4 For cache read, if a MISS happens, the processor usually reads a block of data into the cache. What is the benefit of this?

7.5 According to Flynn's taxonomy, computer systems can be divided into four categories, SISD, MISD, SIMD, MIMD. For each category, provide an example of the architecture.

7.6 Look at the operations below:

1. $\boldsymbol{R3} \leftarrow R1 + R2$
2. $\boldsymbol{R6} \leftarrow R4 + R5$
3. $R7 \leftarrow \boldsymbol{R3} + \boldsymbol{R6}$.

When implementing instruction-level parallelism, data hazard occurs. How does this impact the performance of the pipeline and how should it be addressed?

7.7 Computer A is n times faster than B means:

[ ]Performance (A)/Performance (B) = n

[ ] Execution_time (B)/Execution_time (A) = n

[ ] Performance (A)/Performance (B) = 1/n

[ ] Execution_time (B)/Execution_time (A) = 1/n

7.8 Give an example of where augmented reality can help us in our daily life. (hint: car technologies)

7.9 Some people use smartphones to replace computers. Discuss the advantages and disadvantages of such a move.

7.10 Some countries charge different rates depending on the geographical locations of the websites you access. However when you access Gmail, you are always considered as "local." Explain why this happens?

7.11 Map each term on the left to the corresponding facts on the right.

IaaS	A cloud provides Windows and Linux
SaaS	Clients can choose which software to use
PaaS	Storage and other resources on a cloud

7.12 A computer has 4 GB memory serving as a host. Each virtual machine requires 1 GB memory. How many virtual machines can reside on the host computer?

7.13 Virtualization usually involves a physical computer and a host operating system. The host computer supports the virtual machines through a layer called the Virtual Machine Monitor (VMM). Can VMM replace the host operating system?

References

Answers.com. (n.d.). Retrieved November 1, 2011, from http://www.answers.com/topic/serial-computer-1.

Boggan, S. K. and Pressel, D. M. & Army Research Lab Aberdeen Proving Ground MD Computational and Information Science DIR.

Carter, N. P. (n.d.). *Schaum's Outline of Computer Architecture*, 1st edn, (Original work published 2001). Retrieved from http://books.google.com/books?id=24V00tD7HeAC.

Cohen, W. E. and Inaba, D. S. (2007) Downers, in *Upper, Downers, All Arounders*, CNS Publishing, Inc., Medford, p. 549.

Dandamudi, S. (n.d.) *Guide to RISC Processors: for Programmers and Engineers*, Springer (Original work published 2005).

De Leon, E. (2009) The Five Layers within Internet Computing. Retrieved November 11, 2011. http://cloudcomputing.sys-con.com/node/1200642.

Englander, I. (2009) *Computer Hardware, Systems Software & Networking*, John Wiley and Sons.

Furht, B. and Escalante, A. (2010) *Handbook of Internet Computing*, Springer Science Business+Media, LLC, New York.

Gift, N. (2009) Connecting Apple's iPhone to Google's cloud computing offerings. Retrieved November 20, 2011. http://www.ibm.com/developerworks/web/library/wa-aj-iphone/.

Golden, B. (2010) *What Cloud Computing Can Do*, CXO Media Inc.

Greengard, S. (2010) Cloud computing and developing nations. *Communications of the ACM*, **53**(5), 18–20. doi: 10.1145/1735223.1735232.

Hayes, B. (2008) Cloud Computing. N News Technology. Retrieved November 14, 2011. http://bit-player.org/bph-publications/CACM-2008-07-Hayes-cloud.pdf.

Markwire (2010) Document Lifecycle Management From the Cloud. Retrieved November 20, 2011. http://cloudcomputing.sys-con.com/node/1371929.

McAlpine, K. (2010) DNA logic gates herald injectable computers. Retrieved from http://www.news-cientist.com/article/dn18989-dna-logic-gates-herald-injectable-computers.html.

Mell, P. and Grance, T. (2010) The NIST definition of cloud computing. *Communications of the ACM*, **53**(6), 50.

Mircea, M. and Andrees, A. (2011) Using Cloud Computing: A Strategy to Improve Agility in the Current Financial Crisis. Retrieved November 6, 2011. http://www.ibimapublishing.com/journals/CIBIMA/2011/875547/875547.pdf.

O'Brien, J. A. and Marakas, G. M. (n.d.) *Management Information Systems*, 9th edn, McGraw-Hill Irwin (Original work published 2009).

Parsons, L. J. and Oja, D. (2009) *Computer Concepts*, 11th edn, Course Technology, Cengage Learning, Boston, MA.

Pocatilu, P., Alecu, F., and Vetrici, M. (2010) Measuring the Efficiency of Cloud Computing for E-Learning Systems. Retrieved November 12, 2011. http://www.wseas.us/e-ibrary/transaction/computers/2010/89-159.pdf.

Qin, Y. and Huang, S. (n.d.) Information fusion-oriented design and application of multiprocessor computer software and hardware. *Modern Applied Science*. doi: 10.5539/mas.v5n3p185.

Ryan, M. (2011) Cloud computing privacy concerns on our doorstep. *Communications of the ACM*, **54** (1),36–38. doi: 10.1145/1866739.1866751.

Salchow, K. Jr. (n.d.) Clustered *Multi-Processing: Changing the Rules of the Performance Game* [Article]. Retrieved from http://www.f5.com/pdf/white-papers/viprion-clustered-multiprocessing-wp.pdf.

Sasikala, S. and Prema, S. (2010) Massive centralized cloud computing (MCCC) exploration in higher education. *Advances in Computational Sciences and Technology*, **3**(2), 111–118.

Sultan, N. (2010) Cloud computing for education: A new dawn? *International Journal of Information Management*, 3.

Tout, Samir (2009) Cloud Computing and its Security in Higher Education. Retrieved November 9, 2011. http://proc.isecon.org/2009/2314/ISECON.2009.Tout.pdf.

WebSphere News Desk (2004) New IBM WebSphere Solution with Akamai Enables Customers to Tap into Virtualized Internet Computing Capacity. Retrieved November 20, 2011. http://cloudcomputing.sys-con.com/node/43350.

Wikipedia (2010) Cloud Computing. Retrieved November 19, 2011. http://en.wikipedia.org/wiki/Google_Documents.

8

Assembly Language and Operating Systems

Assembly language is to study not merely the programming skills, but most importantly how to take the full advantage of the processor architecture. So assembly language is mostly related to the specific hardware. For example an Intel processor uses different assembly language than that of an AMD processor. The goal of studying assembly language is to understand the basic instruction set of a processor, to know the data allocation statements, to master data transfer instruction and to become familiar with other instructions.

Usually programs written in assembly language are compiled into machine language that the processors can understand. Depending on the compiler, some assembly languages do have certain compatibilities. This makes it easy for programs written for one processor be easily converted to that of other processors.

Programs written in machine languages have the highest efficiency. However it is difficult to program therefore offers very low productivity. Assembly languages are very close to the machine languages. They are easy to remember and logically clear to programmers. Even though programs written in assembly languages run fast, programmers need to know the hardware well to write programs. As a result many high level computer languages such as Fortran, C and Java come into play.

An OS is a set of programs that manage computer hardware resources and provide common services for application software. The operating system is the most important type of system software in a computer system. A user cannot run an application program on the computer without an operating system, unless the application program is self booting. OS security is one of the most important topics in information security nowadays.

8.1 Assembly Language Basics

Programs written in assembly languages consist of different statements. Each statement will be translated into an instruction that can be executed in processors (Assembly language, 2012). For study purpose, we will use Intel processors as examples throughout the chapter unless otherwise specified. For Intel processors, General-purpose (GP) instructions are a subset of the IA-32 instructions that represent the fundamental instruction set for the Intel IA-32 processors. These instructions were introduced into the IA-32 architecture with the first IA-32 processors (the Intel 8086 and 8088). Additional instructions were added to the general-purpose instruction set in subsequent families of IA-32 processors (the Intel 286, Intel386, Intel486, Pentium, Pentium Pro, and Pentium II processors).

Intel 64 architecture further extends the capability of most general-purpose instructions so that they are able to handle 64-bit data in 64-bit mode. A small number of general-purpose instructions (still supported in non-64-bit modes) are not supported in 64-bit mode.

General-purpose instructions perform basic data movement, memory addressing, arithmetic and logical, program flow control, input/output, and string operations on a set of integer, pointer, and BCD data types. This chapter provides an overview of the general-purpose instructions.

General-purpose instructions are divided into the following subgroups:

- Data transfer
- Binary arithmetic
- Decimal arithmetic
- Logical
- Shift and rotate
- Bit and byte
- Control transfer
- String
- I/O
- Enter and Leave
- Flag control
- Segment register
- Miscellaneous.

Figure 8.1 shows the Intel IA-32 basic execution environment.

8.1.1 Numbering Systems

Most modern computer systems do not represent numeric values using the decimal system. Instead, they typically use a binary or two's complementary numbering

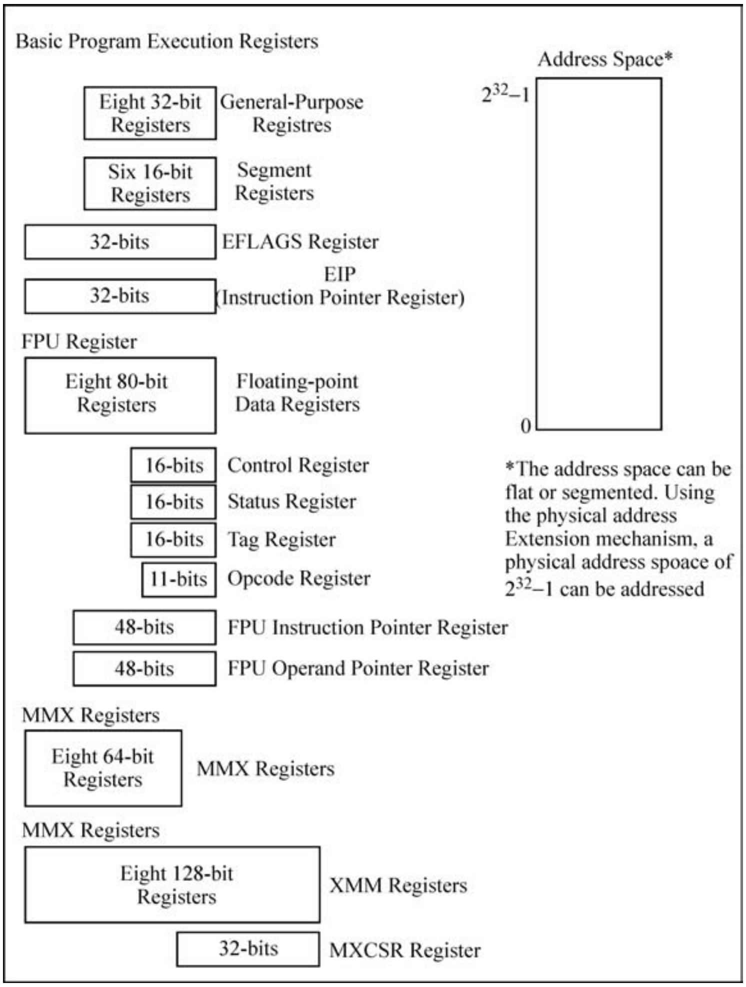


Figure 8.1 Intel IA-32 Basic Execution Environment

system. To understand the limitations of computer arithmetic, you must understand how computers represent numbers.

People have been using the decimal (base 10) numbering system for so long that we probably take it for granted. When you see a number like “753,” you don’t think about the value 753; rather, you generate a mental image of how many items this value represents. In reality, however, the number 753 represents:

$$\mathbf{7} * 10^2 + \mathbf{5} * 10^1 + \mathbf{3} * 10^0 \text{ or}$$
$$700 + 50 + 3$$

Each digit appearing to the left of the decimal point represents a value between zero and nine times an increasing power of ten. Digits appearing to the right of the decimal point represent a value between zero and nine times an increasing negative power of ten. For example, the value 246.359 means:

$$\mathbf{2} * 10^2 + \mathbf{4} * 10^1 + \mathbf{6} * 10^0 + \mathbf{3} * 10^{-1} + \mathbf{5} * 10^{-2} + \mathbf{9} * 10^{-3} \quad \text{or}$$
$$200 + 40 + 6 + 0.3 + 0.05 + 0.09$$

For any decimal number $\boldsymbol{m_{n-1}\ m_{n-2}\ m_{n-3} \ldots \quad \ldots m_1\ m_0}$, the value can be represented as

$$\sum_{i=0}^{n-1} m_i * 10^i$$

here the number 10 is called base. In base 10 (decimal) numbering system, there are ten unique numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

8.1.2 The Binary Numbering System and Base Conversions

Most modern computer systems operate using binary logic. The computer represents values using two voltage levels (usually 0 v and +5 v). With two such levels we can represent exactly two different values. These could be any two different values, but by convention we use the values zero and one. These two values, coincidentally, correspond to the two digits used by the binary numbering system.

The binary numbering system works just like the decimal numbering system, with two exceptions: binary is base 2 numbering system that only allows the digits 0 and 1 (rather than 0–9), and binary uses powers of two rather than powers of ten. Therefore, it is very easy to convert a binary number to decimal. For example, the binary value 11001010 represents:

$$\mathbf{1} * 2^7 + \mathbf{1} * 2^6 + \mathbf{0} * 2^5 + \mathbf{0} * 2^4 + \mathbf{1} * 2^3 + \mathbf{0} * 2^2 + \mathbf{1} * 2^1 + \mathbf{0} * 2^0$$

that is $128 + 64 + 8 + 2 = 202$ (base 10).

To convert a binary (base 2) number $\boldsymbol{m_{n-1}\ m_{n-2}\ m_{n-3} \ldots \quad \ldots m_1\ m_0}$ to a decimal number, here $m_i = \{0,1\}$, the value can be represented as:

$$\sum_{i=0}^{n-1} m_i * 2^i$$

The most commonly seen base number in a computer is 2, 8 and 16. To convert any base **b** number $\boldsymbol{m_{n-1}\ m_{n-2}\ m_{n-3} \ldots \quad \ldots m_1\ m_0}$ to a decimal number, the value can be represented as:

$$\sum_{i=0}^{n-1} m_i * b^i$$

8.1.3 The Hexadecimal Numbering System

A big problem with the binary system is verbosity. To represent the value 202 (decimal) requires eight binary digits. The decimal version requires only three decimal digits and, thus, represents numbers much more compactly than does the binary numbering system. This fact was not lost on the engineers who designed binary computer systems. When dealing with large values, binary numbers quickly become too unwieldy. Unfortunately, the computer thinks in binary, so most of the time it is convenient to use the binary numbering system. Although we can convert between decimal and binary, the conversion is not a trivial task. The hexadecimal (base 16) numbering system solves these problems. Hexadecimal numbers offer the two features we're looking for: they're very compact, and it's simple to convert them to binary and vice versa. Because of this, most binary computer systems use the hexadecimal numbering system. Since the radix (base) of a hexadecimal number is 16, each hexadecimal digit to the left of the hexadecimal point represents some value times a successive power of 16. For example, the number **1234** (hexadecimal) is equal to:

$$\mathbf{1} * 16^3 + \mathbf{2} * 16^2 + \mathbf{3} * 16^1 + \mathbf{4} * 16^0 \quad \text{or}$$
$$4096 + 512 + 48 + 4 = 4660 \text{ (decimal)}$$

Each hexadecimal digit can represent one of sixteen values between 0 and 15. Since there are only ten decimal digits, we need to invent six additional digits to represent the values in the range 10 through 15. Rather than create new symbols for these digits, we'll use the letters A through F. The following are all examples of valid hexadecimal numbers:

1234 DAD01 FEC0A BEEF 8A1DEAF

Table 8.1 provides the information to convert any hexadecimal number into a binary number or vice versa.

To convert a hexadecimal number into a binary number, simply substitute the corresponding four bits for each hexadecimal digit in the number. For example, to convert FEC79A (hex) into a binary value, simply convert each hexadecimal digit according to the table above:

F	E	C	7	9	A
1111	1110	1100	0111	1001	1010

To convert a binary number into hexadecimal format is almost as easy. The first step is to pad the binary number with zeros to make sure that there is a multiple of four bits in the number. For example, given the binary number **1011001010**, the first step would be to add two bits to the left of the number so that it contains 12 bits. The converted binary value is **001011001010**. The next step is to separate the binary value into groups of four bits, for example, **0010 1100 1010**. Finally, look up these binary values in the table above and substitute the appropriate hexadecimal digits, for example, **2CA**.

Table 8.1 Binary/Hex conversion

Binary	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Since converting between hexadecimal and binary is an operation you will need to perform over and over again, you should take a few minutes and memorize the table above. Even if you have a calculator that will do the conversion for you, you'll find manual conversion to be a lot faster and more convenient when converting between binary and hex.

8.1.4 Signed and Unsigned Numbers

So far, we've treated binary numbers as unsigned values. The binary number 00000 represents zero, 00001 represents one, 00010 represents two, and so on toward infinity. What about negative numbers? Now we discuss how to represent negative numbers using the binary numbering system.

To represent signed numbers using the binary numbering system we have to place a restriction on our numbers: they must have a finite and fixed number of bits. For our purposes, we're going to severely limit the number of bits to eight, 16, 32, or some other small number of bits.

With a fixed number of bits we can only represent a certain number of objects. For example, with eight bits we can only represent 256 different objects. Negative values are objects in their own right, just like positive numbers. Therefore, we'll have to use some of the 256 different values to represent negative numbers. In other words, we've got to use up some of the positive numbers to represent negative numbers. To make things fair, we'll assign half of the possible combinations to the negative values and half to the positive values. So we can represent the negative values −128 to −1 and the positive values 0 to 127 with a single 8-bit byte. With a 16-bit word we can represent values in the range −32,768 to +32,767. With a 32-bit double word we can represent values in the range −2,147,483,648 to +2,147,483,647. In general, with n bits we can represent the signed values in the range -2^{n-1} to $+2^{n-1} - 1$.

In the two's complement system, the most significant bit (MSB) of a number is a sign bit. If the MSB is zero, the number is positive; if the MSB bit is one, the number is negative. Examples:

For 16-bit numbers:

- 8000 hex is negative because the H.O. bit is one.
- 100 hex is positive because the H.O. bit is zero.
- 7FFF hex is positive.
- 0FFFF hex is negative.
- 0FFF hex is positive.

If the MSB bit is zero, then the number is positive and is stored as a standard binary value. If the H.O. bit is one, then the number is negative and is stored in the two's complement form. To convert a positive number to its negative, two's complement form, you use the following algorithm:

1. Invert all the bits in the number, that is, apply the logical NOT function.
2. Add one to the inverted result.

For example, to compute the 8-bit equivalent of −5:

1.	0000 0101	Five (in binary).
2.	1111 1010	Invert all the bits.
3.	1111 1011	Add one to obtain result.

If we take −5 and perform the two's complement operation on it, we get our original value, 00000101, back again, just as we expect:

1.	1111 1011	Two's complement for −5.
2.	0000 0100	Invert all the bits.
3.	0000 0101	Add one to obtain result (+5).

The following examples provide some positive and negative 16-bit signed values:

- 7FFF: +32,767, the largest 16-bit positive number.
- 8000: −32,768, the smallest 16-bit negative number.
- 4000: +16,384.

To convert the numbers above to their negative counterpart (i.e., to negate them), do the following:

7FFF:	0111	1111	1111	1111	+32,767
	1000	0000	0000	0000	Invert all the bits (8000 hex)
	1000	0000	0000	0001	Add one (8001 hex or −32,767)
8000:	1000	0000	0000	0000	−32,768
	0111	1111	1111	1111	Invert all the bits (7FFF)
	1000	0000	0000	0000	Add one (8000 hex or −32,768)
4000:	0100	0000	0000	0000	16,384
	1011	1111	1111	1111	Invert all the bits (BFFFh)
	1100	0000	0000	0000	Add one (0C000 or −16,384)

8000h inverted becomes 7FFF. After adding one we obtain 8000! Wait, what's going on here? −(−32,768) is −32,768? Of course not. But the value +32,768 cannot be represented with a 16-bit signed number, so we cannot negate the smallest negative value. If you attempt this operation, the same processors may complain about signed arithmetic overflow.

Why bother with such a miserable numbering system? Why not use the H.O. bit as a sign flag, storing the positive equivalent of the number in the remaining bits? The answer lies in the hardware. As it turns out, negating values is the only tedious job. With the two's complement system, most other operations are as easy as the binary system. For example, suppose you were to perform the addition $5 + (-5)$. The result is zero. Consider what happens when we add these two values in the two's complement system:

```
   00000101
   11111011
 ------------
 1 00000000
```

The result ends up with a carry into the ninth bit and all other bits are zero. As it turns out, if we ignore the carry out of the MSB, adding two signed values always produces the correct result when using the two's complement numbering system. This means we can use the same hardware for signed and unsigned addition and subtraction. This wouldn't be the case with some other numbering systems.

8.2 Operation Code and Operands

An opcode (operation code) is the portion of a machine language instruction that specifies the operation to be performed (Dumas, 2006). Their specification and format are laid out in the instruction set architecture of the processor in question (which may be a general CPU or a more specialized processing unit). Apart from the opcode itself, an instruction normally also has one or more specifiers for operands (i.e., data) on which the operation should act, although some operations may have implicit

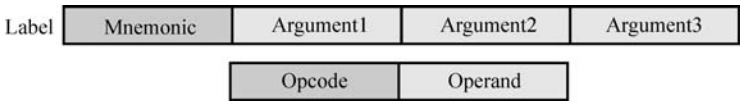


Figure 8.2 Instruction format

operands, or none at all. There are instruction sets with nearly uniform fields for opcode and operand specifiers, as well as others (the x86 architecture for instance) with a more complicated, varied length structure.

Assembly language uses mnemonics (opcode), instructions and operands to represent machine code (see Figure 8.2). This enhances the readability while still giving precise control over the machine instructions.
where:

- A label is an identifier which is followed by a colon.
- A mnemonic is a reserved name for a class of instruction opcodes which have the same function.
- The operands argument1, argument2, and argument3 are optional. There may be from zero to three operands, depending on the opcode. When present, they take the form of either literals or identifiers for data items. Operand identifiers are either reserved names of registers or are assumed to be assigned to data items declared in another part of the program (which may not be shown in the example).

When two operands are present in an arithmetic or logical instruction, the right operand is the source and the left operand is the destination.

Depending on architecture, the operands may be register values, values in the stack, other memory values, I/O ports, and so on, specified and accessed using more or less complex addressing modes. The types of operations include arithmetics, data copying, logical operations, and program control, as well as special instructions (such as CPUID and others).

Operands specify the data to be manipulated by the statement. The number of operands required depends on the specific statement. For example, executable statements may have zero, one, two, or three operands (see Figure 8.3).

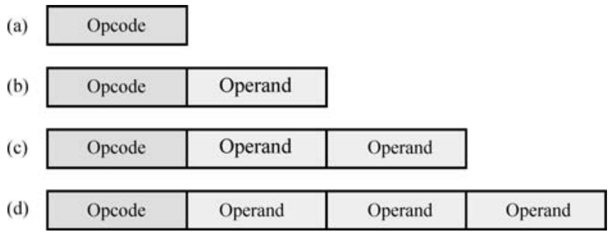


Figure 8.3 Opcode and operands

8.3 Direct Addressing

Some instructions use data encoded in the instruction itself as a source operand. These operands are called immediate operands (or simply immediates). For example, the following ADD instruction adds an immediate value of 14 to the contents of the EAX register:

ADD EAX, 14

All arithmetic instructions (except the DIV and IDIV instructions) allow the source operand to be an immediate value. The maximum value allowed for an immediate operand varies among instructions, but can never be greater than the maximum value of an unsigned doubleword integer (2^{32}).

Register Operands is source and destination operands can be any of the registers depending on the instruction being executed.

Some instructions (such as the DIV and MUL instructions) use quadword operands contained in a pair of 32-bit registers. Register pairs are represented with a colon separating them. For example, in the register pair EDX:EAX, EDX contains the high order bits and EAX contains the low order bits of a quadword operand.

Several instructions (such as the PUSHFD and POPFD instructions) are provided to load and store the contents of the EFLAGS register or to set or clear individual flags in this register. Other instructions use the state of the status flags in the EFLAGS register as condition codes for branching or other decision making operations.

The processor contains a selection of system registers that are used to control memory management, interrupt and exception handling, task management, processor management, and debugging activities. Some of these system registers are accessible by an application program, the operating system, or the executive through a set of system instructions. When accessing a system register with a system instruction, the register is generally an implied operand of the instruction.

8.4 Indirect Addressing

Memory Operands means source and destination operands in memory are referenced by means of a segment selector and an offset (see Figure 8.4). Segment selectors

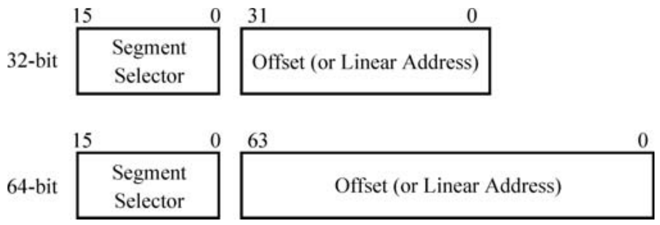


Figure 8.4 Memory Operand address (32-bit/64-bit)

Table 8.2 Default segment selection rules

Reference Type	Register Used	Segment Used	Default Selection Rule
Instructions	CS	Code Segment	All instruction fetches.
Stack	SS	Stack Segment	All stack pushes and pops
Local Data	DS	Data Segment	All data references
Destination String	ES	Data Segment pointed to with the ES register	Destination of string instructions.

specify the segment containing the operand. Offsets specify the linear or effective address of the operand.

Offsets can be 32 bits (represented by the notation m16:32) or 64 bits (represented by the notation m16:64).

The segment selector can be specified either implicitly or explicitly. The most common method of specifying a segment selector is to load it in a segment register rand then allow the processor to select the register implicitly, depending on the type of operation being performed. The processor automatically chooses a segment according to the rules given in Table 8.2. When storing data in memory or loading data from memory, the DS segment default can be overridden to allow other segments to be accessed. Within an assembler, the segment override is generally handled with a colon ":" operator. For example, the following MOV instruction moves a value from register EAX into the segment pointed to by the ES register. The offset into the segment is contained in the EBX register:

$$MOV\,ES : [EBX], EAX;$$

8.5 Stack and Buffer Overflow

A stack is a memory segment where data can only be accessed from one side. The BOTTOM of a stack is fixed or cannot be changed. The TOP of a stack will be adjusted based on the operations PUSH or POP. An important property for stack data is first-in-last-out (FILO) or last-in-first-out (LIFO).

The stack (see Figure 8.5) is a contiguous array of memory locations (Shiva, 2000). It is contained in a segment and identified by the segment selector in the SS register. When using the flat memory model, the stack can be located anywhere in the linear address space for the program. A stack can be up to 4 GBytes long, the maximum size of a segment.

Items are placed on the stack using the PUSH instruction and removed from the stack using the POP instruction. When an item is pushed onto the stack, the

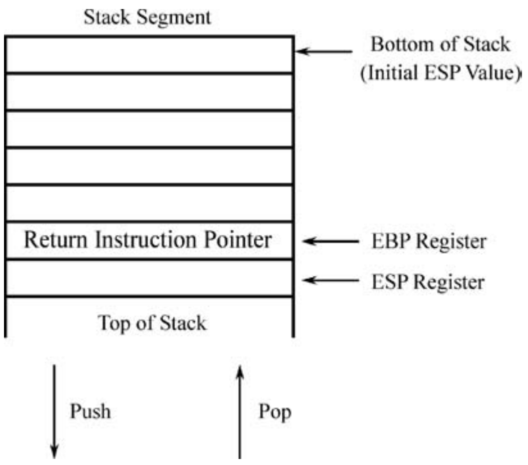

Stack Segment
Bottom of Stack (Initial ESP Value)
Return Instruction Pointer
EBP Register
ESP Register
Top of Stack
Push
Pop

Figure 8.5 Stack structure

processor decrements the ESP register, then writes the item at the new top of the stack. When an item is popped off the stack, the processor reads the item from the top of the stack and then increments the ESP register. In this manner, the stack grows down in memory (towards lesser addresses) when items are pushed on the stack and shrinks up (towards greater addresses) when the items are popped from the stack.

A program or operating system executive can set up many stacks. For example, in multitasking systems, each task can be given its own stack. The number of stacks in a system is limited by the maximum number of segments and the available physical memory.

Stack Manipulation Instructions – The PUSH, POP, PUSHA (push all registers), and POPA (pop all registers) instructions move data to and from the stack. The PUSH instruction decrements the stack pointer (contained in the ESP register), then copies the source operand to the top of the stack (see Figure 8.6). It operates on memory operands, immediate operands, and register operands. The PUSH instruction is commonly used to place parameters on the stack before calling a procedure. It can also be used to reserve space on the stack for temporary variables.

If a stack has reached the maximum reserved memory and there is another PUSH operation, the TOP will overwrite the data in the other memory segment. This is called stack overflow. Or if the stack has reached the BOTTOM and there is another POP operation, the data retrieved from the stack would be wrong. This is called stack underflow. If the POP operation is involved with an interrupt, then CPU may run into the unexpected program or data segment. If this was preset by a hacker, then it may launch a malicious program. This is called stack buffer overflow attack.

In software, stack buffer overflow occurs when a program writes to a memory address on the program's call stack outside of the intended data structure; usually a

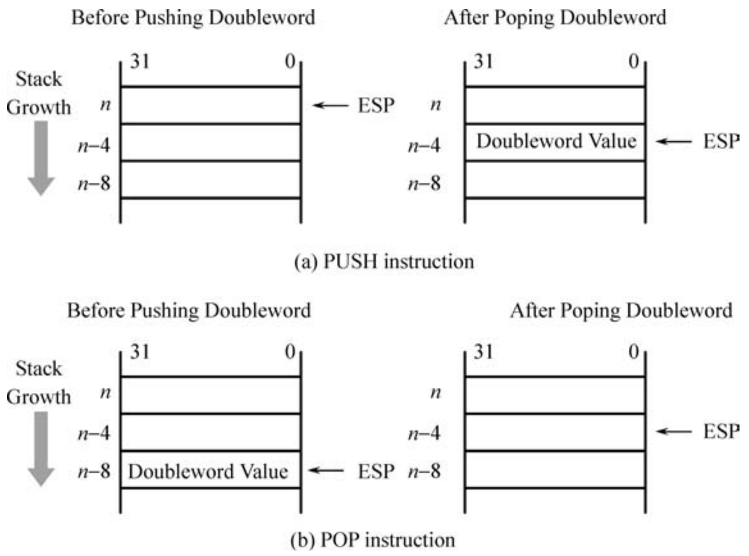


Figure 8.6 Operation of PUSH/POP instructions

fixed length buffer. Stack buffer overflow bugs or attacks are caused when a program writes more data to a buffer located on the stack than was actually allocated for that buffer. This almost always results in corruption of adjacent data on the stack, and in cases where the overflow was triggered by mistake, will often cause the program to crash or operate incorrectly. This type of overflow is part of the more general class of programming bugs known as buffer overflows.

If the affected program is running with special privileges, or accepts data from untrusted network hosts (e.g., a Web server) then the bug is a potential security risk. If the stack buffer is filled with data supplied from an untrusted user then that user can corrupt the stack in such a way as to inject executable code into the running program and take control of the process. This is one of the oldest and more reliable methods for hackers to gain unauthorized access to a computer.

8.5.1 Calling Procedures Using CALL and RET (Return)

The CALL instruction allows control transfers to procedures within the current code segment (near call) and in a different code segment (far call). Near calls usually provide access to local procedures within the currently running program or task. Farcalls are usually used to access operating system procedures or procedures in a different task.

The RET instruction also allows near and far returns to match the near and far versions of the CALL instruction. In addition, the RET instruction allows a program to increment the stack pointer on a return to release parameters from the stack.

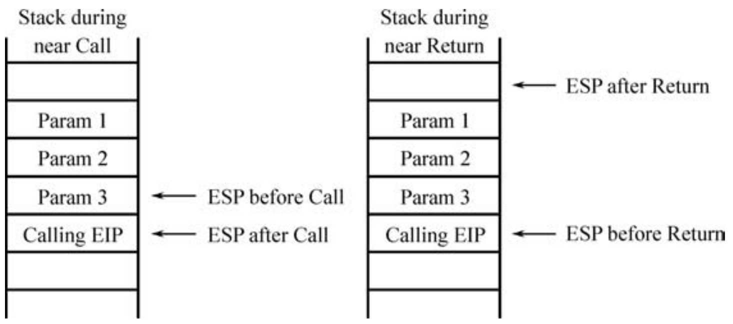


Figure 8.7 Stack change using CALL and RET

When executing a near call, the processor does the following (see Figure 8.7):

1. Pushes the current value of the EIP register on the stack.
2. Loads the offset of the called procedure in the EIP register.
3. Begins execution of the called procedure.

When executing a near return, the processor performs these actions:

1. Pops the top-of-stack value (the return instruction pointer) into the EIP register.
2. If the RET instruction has an optional n argument, increments the stack pointer by the number of bytes specified with the n operand to release parameters from the stack.
3. Resumes execution of the calling procedure.

8.5.2 *Exploiting Stack Buffer Overflows*

The canonical method for exploiting a stack based buffer overflow is to overwrite the function return address with a pointer to attacker-controlled data (usually on the stack itself). This is illustrated in the example below:

An example with strcpy

```
#include <string.h>
void foo (char *bar)
{
char c[12];
strcpy(c, bar); // no bounds checking...
} int main (intargc, char **argv)
{
foo(argv[1]);
}
```

This code takes an argument from the command line and copies it to a local stack variable c. This works fine for command line arguments smaller than 12 characters (as you can see in Figure 8.8b below). Any arguments larger than 11 characters long will result in corruption of the stack. (The maximum number of characters that is safe is one less than the size of the buffer here because in the C programming language strings are terminated by a zero byte character. A 12-character input thus requires thirteen bytes to store, the input followed by the sentinel zero byte. The zero byte then ends up overwriting a memory location that's one byte beyond the end of the buffer.)

Notice in Figure 8.8, when an argument larger than 11 bytes is supplied on the command line *foo()* overwrites local stack data, the saved frame pointer, and most importantly, the return address. When *foo()* returns it pops the return address off the stack and jumps to that address (i.e., starts executing instructions

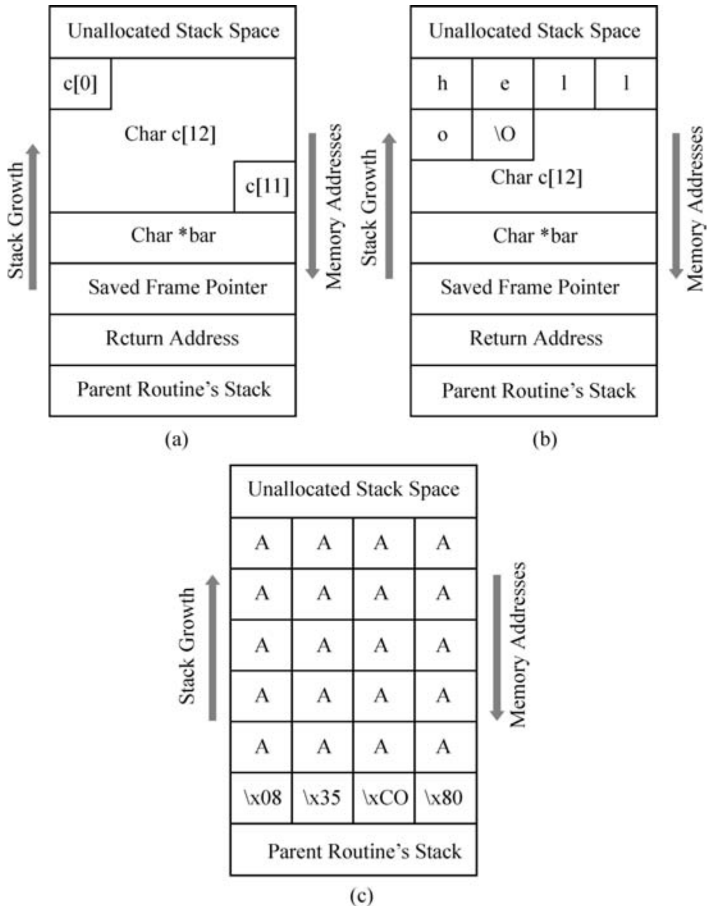


Figure 8.8 Program stack in foo() with various inputs (a) before data is copied; (b) "hello" is the first command line argument; (c) "AAAAAAAAAAAAAAAAAAAA\x08\x35\xC0 \x80 is the first command line argument

from that address). As you can see in Figure 8.8, the attacker has overwritten the return address with a pointer to the stack buffer *char c[12]*, which now contains attacker supplied data. In an actual stack buffer overflow exploit the string of "A"'s would be replaced with shellcode suitable to the platform and desired function. If this program had special privileges (e.g., the SUID bit set to run as the superuser), then the attacker could use this vulnerability to gain superuser privileges on the affected machine.

The attacker also can modify internal variables values to exploit some bugs. With the following example:

```
#include <string.h>
#include <stdio.h>
void foo (char *bar)
{
Float My_Float = 10.5;          // Addr = 0x0023FF4C
char c[12];                   // Addr = 0x0023FF30
  // Will print 10.500000
printf("My Float value = %f\n", My_Float);
   /* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Memory map:
@ : c allocated memory
# :My_Float allocated memory
    - : other memory
     *c                        *My_Float
0x0023FF30                    0x0023FF4C
   |                            |
   @@@@@@@@@@@@------------#####
foo("my string is too long !!!!! XXXXX");
memcpy will put 0x1010C042 in My_Float value.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
memcpy(c, bar, strlen(bar)); // no bounds checking...
// Will print 96.031372
printf("My Float value = %f\n", My_Float);
}

int main (intargc, char **argv){
foo("my string is too long !!!!! \x10\x10\xC0\x42");
return 0;
}
```

8.5.3 Stack Protection

One approach to preventing stack buffer overflow exploitation is to enforce memory policy on the stack memory region to disallow execution from the stack. This means that in order to execute shellcode from the stack an attacker must either find a way to disable the execution protection from memory, or find a way to put her/his shellcode payload in a non-protected region of memory.

This method is becoming more popular now that hardware support for the no-execute flag is available in most desktop processors.

Intel uses a stack segment register (SS) to manage PUSH and POP operations. Any memory reference which uses ESP or EBP uses general purpose register as a base register.

8.6 FIFO and M/M/1 Problem

A queue is another important data structure in mathematics and computer science (Hwang, 1993). A useful queuing model represents a real-life system with sufficient accuracy and is analytically tractable. A queuing model based on the Poisson process and its companion exponential probability distribution often meets these two requirements. A Poisson process models random events (such as a customer arrival, a request for action from a Web server, or the completion of the actions requested of a Web server) as emanating from a memoryless process. That is, the length of the time interval from the current time to the occurrence of the next event does not depend upon the time of occurrence of the last event. In the Poisson probability distribution, the observer records the number of events that occur in a time interval of fixed length. In the (negative) exponential probability distribution, the observer records the length of the time interval between consecutive events. In both, the underlying physical process is memoryless.

8.6.1 FIFO Data Structure

A queue is different than a stack as it has a FRONT and BACK. Data always move into queue from BACK and leave queue from FRONT. So it is a first-in-first-out (FIFO) data structure.

Queues are very common in our daily life. When people go through the checkpoint at the airport, they all follow the first-in-first-out (FIFO) model (if only one checkpoint is open at that time). If too many passengers arrived, then the line would be very long or the waiting time would be very long.

Queues provide services in computer science, transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer.

Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists.

In Figure 8.9, the customer or packet at the front of the line was the first one to enter, while the one at the end of the line is the last to have entered. Every time a customer finishes paying for their items that object leaves the queue from the front. This represents the queue “dequeue” function. Every time another object or customer enters the line to wait, they join the end of the line and represent the

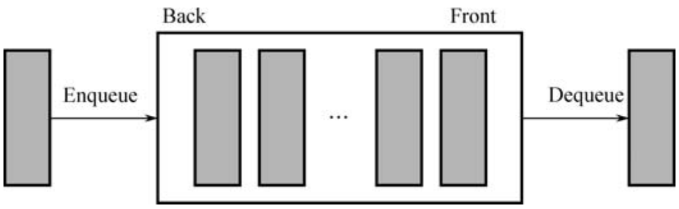


Figure 8.9 A queue model is using the FIFO data structure

"enqueue" function. The queue "size" function would return the length of the line, and the "empty" function would return true only if there was nothing in the line.

8.6.2 M/M/1 Model

In queuing theory, a discipline within the mathematical theory of probability, an M/M/1 queue represents the queue length in a system having a single server, where arrivals are determined by a Poisson process and job service times have an exponential distribution. The model name is written in Kendall's notation. Figure 8.10 shows a diagram of an M/M/1 model. The model is the most elementary of queuing models and an attractive object of study as closed-form expressions can be obtained for many metrics of interest in this model.

An M/M/1 queue is a stochastic process whose state space is the set {0, 1, 2, 3, . . . } where the value corresponds to the number of customers in the system, including any currently in service.

- Arrivals occur at rate λ according to a Poisson process and move the process from state i to i + 1.
- Service times have an exponential distribution with parameter μ in the M/M/1 queue.
- A single server serves customers one at a time from the front of the queue. When the service is complete the customer leaves the queue and the number of customers in the system reduces by one.
- The buffer is of infinite size, so there is no limit on the number of customers it can contain.

This is the same continuous time Markov chain as in a birth–death process. The state space diagram for this chain is shown as Figure 8.11.

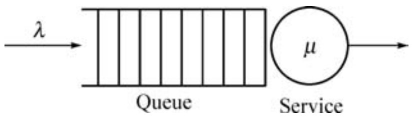


Figure 8.10 M/M/1 model

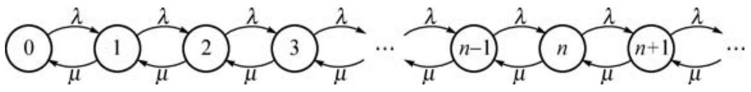


Figure 8.11 Markov chain

The model is considered stable only if $\lambda < \mu$. If, on average, arrivals happen faster than service completions the queue will grow indefinitely long and the system will not have an equilibrium. Various performance measures can be computed explicitly for this model. We write $\rho = \lambda/\mu$ for the utilization of the buffer and require $\rho < 1$ for the queue to be stable. ρ represents the average proportion of time which the server is occupied.

It can be mathematically proved that the number of customers in the system is

$$\frac{\rho}{1-\rho}$$

and the variance of number of customers in the system is $\rho/(1 - \rho)2$.

For customers who arrive and find the queue as a stationary process, the average time (the sum of both waiting time and service time) spent waiting is

$$\frac{1}{\mu-\lambda} - \frac{1}{\mu} = \frac{\rho}{\mu-\lambda}$$

8.7 Kernel, Drivers and OS Security

8.7.1 Kernel

A kernel is the main component of most computer operating systems; it is a bridge between applications and the actual data processing done at the hardware level. Figure 8.12 shows a diagram of a kernel in a computer system. The kernel's responsibilities include managing the system's resources (hardware, software and the communication between them). Usually as a basic component of an operating system, a

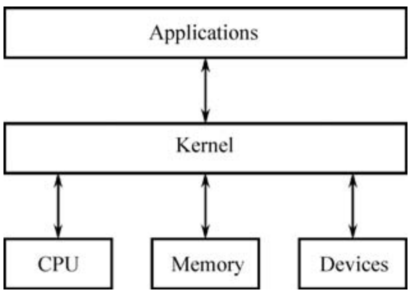


Figure 8.12 Kernel is a bridge between hardware and the applications

kernel can provide the lowest-level abstraction layer for the resources (especially processors and I/O devices) that application software must control to perform its function. It typically makes these facilities available to application processes through inter-process communication mechanisms and system calls.

Operating system tasks are done differently by different kernels, depending on their design and implementation. While monolithic kernels execute all the operating system code in the same address space to increase the performance of the system, micro-kernels run most of the operating system services in user space as servers, aiming to improve maintainability and modularity of the operating system. A range of possibilities exists between these two extremes.

The kernel's primary function is to manage the computer's resources and allow other programs to run and use these resources. Typically, the resources consist of:

- The Central Processing Unit
- The computer's memory
- Any I/O devices present in the computer, such as keyboard, mouse, disk drives, printers, displays, and so on.

8.7.2 BIOS

The System BIOS or ROM BIOS is a de facto standard defining a firmware interface. BIOS software is stored on a non-volatile ROM chip on the motherboard. It is specifically designed to work with each particular model of computer, interfacing with various devices that make up the complementary chipset of the system. In modern computer systems the BIOS chip's contents can be rewritten without removing it from the motherboard, allowing BIOS software to be upgraded in place. The author of this book invented this non-volatile ROM as early as 1985.

The BIOS software is built into the PC, and is the first code run by a PC when powered on ("boot firmware"). When the PC starts up, the first job for the BIOS is the power-on self-test, which initializes and identifies system devices such as the video display card, keyboard and mouse, hard disk drive, optical disc drive and other hardware. The BIOS then locates boot loader software held on a peripheral device (designated as a "boot device"), such as a hard disk or a CD/DVD, and loads and executes that software, giving it control of the PC. This process is known as booting, or booting up, which is short for bootstrapping.

A BIOS has a user interface (UI), typically a menu system accessed by pressing a certain key on the keyboard when the PC starts. In the BIOS UI, a user can:

- configure hardware
- set the system clock
- enable or disable system components
- select which devices are eligible to be a potential boot device

- set various password prompts, such as a password for securing access to the BIOS UI functions itself and preventing malicious users from booting the system from unauthorized peripheral devices.

The BIOS provides a small library of basic input/output functions used to operate and control the peripherals such as the keyboard, text display functions and so forth, and these software library functions are callable by external software. From around 2010 the BIOS firmware of PCs started to be replaced by a Unified Extensible Firmware Interface (UEFI).

8.7.3 Boot Loader

BIOS is a boot loader (bootstrap loader). This small program's only job is to load other data and programs which are then executed from RAM. Often, multiple-stage boot loaders are used, during which several programs of increasing complexity load one after the other in a process of chain loading.

The boot process can be considered complete when the computer is ready to interact with the user, or the operating system is capable of running system programs or application programs. Typical modern personal computers boot in about one minute, of which about 15 seconds are taken by a power-on self-test (POST) and a preliminary boot loader, and the rest by loading the operating system and other software. Time spent after the operating system loading can be considerably shortened to as little as 3 seconds by bringing the system up with all cores at once, as with coreboot. Large servers may take several minutes to boot and start all their services.

Many embedded systems must boot immediately. For example, waiting a minute for a digital television or GPS satellite to start is generally unacceptable. Therefore such devices have software systems in ROM or flash memory so the device can begin functioning immediately. For these types of embedded systems little or no loading is necessary, since the loading can be precomputed and stored on the ROM when the device is made.

Most computers are also capable of booting over a computer network. In this scenario, the operating system is stored on the disk of a server, and certain parts of it are transferred to the client using a simple protocol such as the Trivial File Transfer Protocol. After these parts have been transferred, the operating system then takes over control of the booting process.

Once the BIOS has found a bootable device it loads the boot sector to linear address 0x7C00 and transfers execution to the boot code. In the case of a hard disk, this is referred to as the master boot record (MBR) and is often not operating system specific. The conventional MBR code checks the MBRs partition table for a partition set as bootable. If an active partition is found, the MBR code loads the boot sector code from that partition and executes it. The main function is to load and execute the operating system kernel, which continues startup.

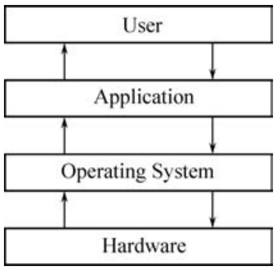


Figure 8.13 Device driver diagram

8.7.4 *Device Drivers*

A device driver or software driver is a computer program allowing higher-level computer programs to interact with a hardware device. Figure 8.13 shows a device driver diagram. A driver typically communicates with the device through the computer bus or communications subsystem to which the hardware connects. When a calling program invokes a routine in the driver, the driver issues commands to the device. Once the device sends data back to the driver, the driver may invoke routines in the original calling program. Drivers are hardware-dependent and operating-system-specific. They usually provide the interrupt handling required for any necessary asynchronous time-dependent hardware interface.

Writing a device driver requires an in-depth understanding of how the hardware and the software of a given platform function. Drivers operate in a highly privileged environment and can cause disaster if they get things wrong. In contrast, most user-level software on modern operating systems can be stopped without greatly affecting the rest of the system. Even drivers executing in user mode can crash a system if the device is erroneously programmed. These factors make it more difficult and dangerous to diagnose problems.

Device drivers, can run in kernel-mode or in user-mode. The primary benefit of running a driver in user mode is improved stability, since a poorly written user mode device driver cannot crash the system by overwriting kernel memory. On the other hand, user/kernel-mode transitions usually impose a considerable performance overhead, thereby prohibiting user mode-drivers for low latency and high throughput requirements.

Because of the diversity of modern hardware and operating systems, drivers operate in many different environments. Drivers may interface with:

- printers
- video adapters
- network cards
- sound cards
- local buses of various sorts – in particular, for bus mastering on modern systems

- low-bandwidth I/O buses of various sorts (for pointing devices such as mice, keyboards, USB, etc.)
- computer storage devices such as hard disk, CD-ROM and floppy disk buses (ATA, SATA, SCSI)
- implementing support for different file systems
- image scanners
- digital cameras.

An application programming interface (API) is a source code-based specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables.

An API specification can take many forms, including an International Standard or vendor documentation such as the Microsoft Windows API, or the libraries of a programming language (Wang et al., 2008), for example, Standard Template Library in C++ or Java API.

8.8 Summary

Unlike other high level computer languages, assembly languages are closely related with the architecture of the processors (or a series of processors). Assembly languages can be easily translated into machine languages that consist of instructions processors can execute directly.

Intel-32 and Intel-64 architecture has many general purpose instructions such as arithmetic, logical and I/O, and so on. The execution environment contains general-purpose registers, segment registers, instruction pointer register (EIP, or program counter), Single-instruction multiple data instruction set (MMX) registers and so on.

An opcode specifies the type of command for an instruction. The operands are arguments or numbers to be used during the operations. Some operands are immediate or direct. Some operands are stored in memory. The operands are pointers (or offset) pointing to the data location.

Stack is a very important data structure. The BOTTOM is fixed or in other words cannot be accessed; data can only be accessed from the TOP by following the LIFO rule. Stack can be used in I/O interfaces to connect the low speed I/O devices to the high speed processors. Procedures can use stack to save the current execution location and return from the subroutine when finished.

Due to the limited capacity, a stack may overflow if it pushes data to the stack once the stack is full, or a stack may underflow if pop data once the stack is empty. If the stack overflow (or underflow) is caused by an attack, then the computer could be compromised and the data stored on the computer may be in danger.

Using non-executable memory for stack is one method to reduce the stack buffer overflow exploitation.

Queue is another important data structure. It has a FRONT and a BACK. Data always moves into the queue from the back and leaves queue from the front (FIFO). The single-server queuing model (M/M/1) is widely used in many areas. The arrivals rate λ should be less than the service rate μ with traffic intensity $\rho = \lambda/\mu < 1$. The number of customers in the system is $\rho/(1-\rho)$ and the average time spent waiting is $\rho/(\mu-\lambda)$.

Computer architecture (no matter whether it is a quad-core or a pipeline) cannot function well without the support of the operating system. A kernel is the center of operating systems working directly with the hardware and the resources. The BIOS is a boot firmware (bootstrap loader) in personal computers that is stored on a non-violate memory. BIOS is first executed followed by the kernel after a cool start.

Device drivers are programs that on one side communicate directly with the specific device and on the other side provide interface to high-level computer programs. Application program interface (API) is a software interface to let other programs call the functions inside the software. A complied software application without API can hardly be modified or expanded.

Exercises

8.1 Today most programs are written either in C and Java. Is there any benefit to writing an assembly language program?

8.2 Addition is to add two numbers together. Explain how to do addition using instructions with one operand.

8.3 A PC (=FE17) points to the top of the stack. The bottom of the stack is at address FFFF. After execute instructions: PUSH, PUSH and POP, what is the value in the PC?

8.4 Read Section 8.5.2 and answer the questions: How to modify the program and to prevent the buffer overflow problem.

8.5 A queue has two pointers front and back. Write two functions:
(1) Enqueue () – move data into queue
(2) Dequeue () – move data out of queue.

8.6 A cell phone service provider has the capacity of connecting 1,000 people per second. Due to an emergency, the incoming requests reach to 999 people per second. Calculate the number of people waiting in line to be served. Explain why the result shows almost nobody is served?

8.7 There are researches to develop driver-free devices. One way is to store the driver into the device. When a device is connected, the driver is loaded automatically. Find an example of this device and explain how it works.

8.8 A programmer wants to add features to a software application developed by another company. Unfortunately there is no source code that can be revised. What he should do to finish his job?

References

Assembly language (2012) *Wikipedia*. Retrieved February 2, 2012 at http://en.wikipedia.org/wiki/Assembly_language.

Dumas, J. D. (2006) *Computer Architecture: Fundamentals and Principles of Computer Design*, Taylor & Francis.

Foster, C. C. and Iberall, T. (1985) *Computer Architecture*, 3rd edn, Wan Nostrand Reinhold Company, New York.

Hwang, K. (1993) *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, Inc.

Randell, B. (ed.) (1982) *The Designs of Digital Computers*, Springer-Verlag.

Shiva, S. G. (2000) *Computer Design and Architecture*, 3rd edn, Marcel Dekker, Inc., New York.

Wang, S., Dong, Z., Chen, J., and Ledley, R. (2008) PPL – A whole image processing language. *Computer Language, System and Structures*, **34** (1), 18–24.

9

TCP/IP and Internet

Data communication is considered one of the fastest growing technologies in the world. Our society will become even more dependent on the sharing and manipulation of information. In addition, people will demand the most expedient transmission as the digitized flow of information that increasingly impacts the critical aspects of society's information infrastructure.

TCP/IP is one of the most important prototypes in digital communications and is the foundation of the Internet. Routers, switches; and gateways are key equipment for a network. Wireless communication networks provide the flexibility of easy Internet connection anywhere, anyplace and at any time. Since security has been a concern for almost all computer systems, protecting data stored on a computer has become more and more important.

9.1 Data Communications

Data communication is to study the transmission of digital messages to devices besides the message source. The devices or channels are generally thought of as being independently powered circuitry that exists beyond the chassis of a computer or other digital message source. In general, the maximum permissible transmission rate of a message is directly proportional to signal power, and inversely proportional to channel noise. It is the aim of any communications system to provide the highest possible transmission rate at the lowest possible power and with the least possible noise.

For instance, there are many government and state agencies that rely on the Internet to function. The United States Department of Homeland Security is an example of an agency that relies on the Internet to communicate and share critical mission related information with sub-units in the field, and with other outside agencies important to the completion of its mission. Thus, part of its mission is dependent on the efficiency, power and effectiveness of the Internet, which likewise is dependent on the efficient flow of data communications. So, the importance of development of a computer network protocol to maintain this efficiency cannot be ignored.

The goal of data communication is to provide a means of reliable and efficient data communication between two end nodes or hosts. Communications between these entities are in the form of messages composed and received by senders and receivers. This concept may appear quite simple on the surface; however, it actually involves unique protocol models that are keys to ensuring the success of the transmission of the message. There are various models that can be used to facilitate this process. One model in particular is the Transmission Control Protocol/Internet Protocol (TCP/IP), which is a universal suite that offers reliable software that facilitates communications between diverse vendor equipment, and which is now the basis for the operation of the Internet.

9.1.1 Signal, Data, and Channels

Electromagnetic **signals**, which are capable of propagation on a variety of transmission media, can be used to convey **data**. We use the term data to represent something meaningful. We use the term signal to stand for physical transmission of data along various mediums. In the real world, we have two types of signals, analog signal and digital signal. The word channel refers either to a physical transmission medium such as a wire, or to a logical connection such as a radio channel. A **channel** has a certain capacity for transmitting information, often measured by its bandwidth in Hz or its data rate in bits per second (bps).

An analog signal is a format of representing data with continuously varying electromagnetic wave. An old telephone system transmits analog signals from the source to the destination. A digital signal is a format of representing data with a sequence of voltage pulses. Nowadays most network signals are all digital signals. Most IP phones also transmit digital signals. Figure 9.1 shows an analog signal and a digital signal.

Digital communication has the advantages of cost-effective, data integrity, large capacity, security and privacy, and easy integration. Digital data, voice and video can be easily integrated and transmitted through a single communication channel.

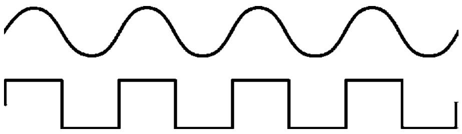


Figure 9.1 Analog signal and digital signal

9.1.2 Signal Encoding and Modulation

Analog data are often encoded and modulated during transmission. The amplitude modulation (AM) is used by many radio stations to broadcast news information. The frequency modulation (FM) is often used by radio stations to broadcast music, since FM can have better voice quality compared with AM. There is another modulation method that modulates phases of the signal (PM). Figure 9.2 shows the AM, FM and PM signals.

In telecommunications, a **carrier** is a waveform (usually sinusoidal) that is modulated with an input signal for the purpose of conveying information. This carrier wave is usually of a much higher frequency than the input signal. The purpose of the carrier is usually either to transmit the information through space as an electromagnetic wave (as in radio communication), or to allow several carriers at different frequencies to share a common physical transmission medium by frequency division multiplexing.

An example of this is the telephone system where many people can talk through one cable. Since human voices fall between 20 and 20 KHz, there is no way to transmit multiple voice signals over one cable without modulation. The carrier helps to modulate the voice signals to different consecutive frequencies so there is no overlap between any voice signals. At the destination, the modulated signal is de-modulated to restore the signal to the original (audible) signal.

Digital signals can also be encoded to reduce errors during transmission. An analog/digital (A/D) converter is used to encode the analog signal to digital signal.

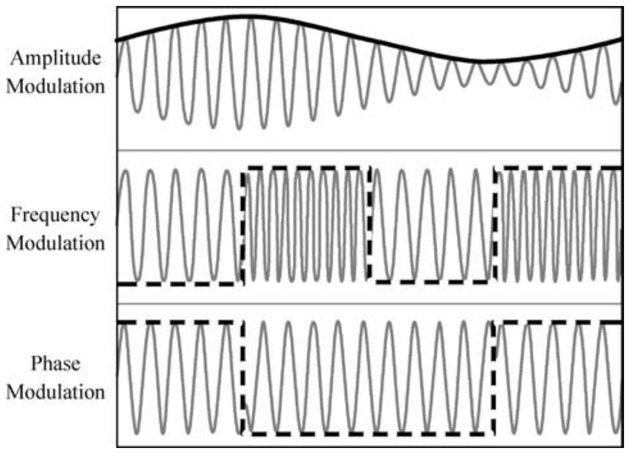


Figure 9.2 AM, FM and PM signals

9.1.3 Shannon Theorem

The Nyquist–**Shannon** sampling theorem is a fundamental result in the field of information theory, in particular telecommunications and signal processing. Sampling is the process of converting a signal (for example, a function of continuous time or space) into a numeric sequence (a function of discrete time or space). Shannon's version of the theorem states:

> If a function x(t) contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart. If the highest frequency B in the original signal is known, the theorem gives the lower bound on the sampling frequency for which perfect reconstruction can be assured. This lower bound to the sampling frequency, 2B, is called the Nyquist rate.

In summary, the Shannon theorem states: If the original signal has the highest frequency B, the sample rate should be 2B in order to reconstruct the original signal losslessly.

For CDs, the sample rate is 44,100 Hz. The reason to choose this sample rate is because the human hearing range is roughly 20 Hz to 20,000 Hz. According to Shannon' sample theorem, the sample rate should be no less than twice the highest frequency, or 40,000 Hz. In addition to this, signals must be low-pass filtered (LPF) and LPF is hardly perfect, so we keep a transition band 2,000 Hz. As a result it requires additional 4 k Hz sampling. The extra 100 Hz is added to comply with the video recording methods. For NTSC, 245 lines $*$ 60 frames $*$ 3 samples = 44,100 samples per second. For PAL, 294 lines $*$ 50 frames $*$ 3 samples = 44,100 samples per second.

9.2 TCP/IP Protocol

The Transmission Control Protocol/Internet Protocol architecture originated during the 1970s, as a result of the United States Defense Department's need for a wide-area communication system, covering the United States and allowing the interconnection of heterogeneous hardware and software systems. It is a result of protocol research and development conducted on the experimental packet-switched network, ARPANET, funded by the Defense Advanced Research Projects Agency (DARPA), and is generally referred to as the TCP/IP protocol suite. Although it was at first used to facilitate consistent communications with governments, military and educational sites together, commercial companies were eventually allowed to access this realm. Eventually, this protocol would become the foundation of the Internet.

9.2.1 Network Topology

There is a concept of using layers to describe Internet protocols. There are no physical layers in the protocol, the layer is a logical concept or in other words it is virtual. The common five layers model contains application, transport, network, data link, and physical. The ISO model contains seven layers, application, presentation, session, transport, network, data link, and physical. An easy way to remember OSI model is "All People Seem To Need Data Processing." Figure 9.3 shows the TCP/IP network topology.

TCP/IP revolutionized and greatly improved the transmission of digitized information throughout the Internet. It is used as a tool to transmit data in the form of a message or file through the Internet, and consists of several levels of functionality. TCP/IP has become the most common suite for telecommunications as the Internet was originally created around it, and the access to the Internet would not be possible without it. The communication layers associated with TCP/IP consist of the application layer, host-to-host or transport layer, Internet layer, network access layer and Physical layer. The most commonly used protocol at the transport layer is the transmission control protocol (TCP). TCP provides a reliable stream delivery and virtual

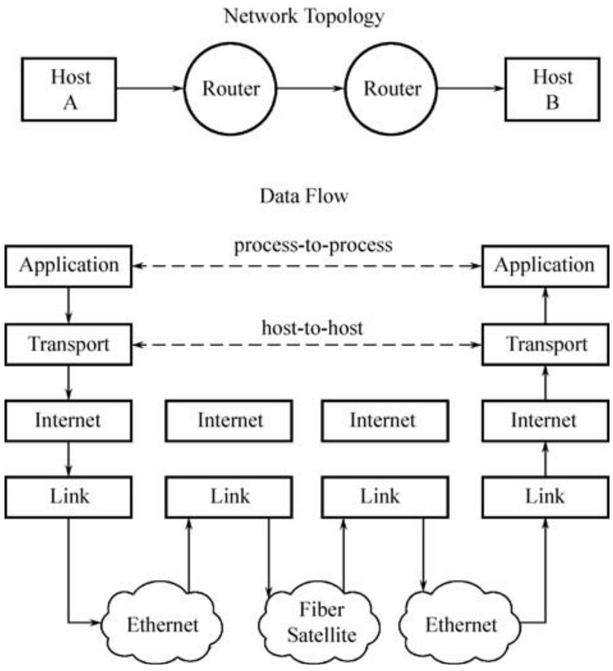


Figure 9.3 TCP/IP network topology

connection service to applications through the use of sequenced acknowledgment with retransmission of packets when necessary. Since many network applications may be running on the same machine with one single IP address, computers need something to make sure that the correct software application on the destination computer gets the data packets from the source machine and the replies get routed back to the correct application on the source computer, and this is accomplished through the use of the TCP "port numbers" and the combination of IP address of a network station and its port number is known as a "socket" or an "endpoint."

9.2.2 Transmission Control Protocol (TCP)

TCP is one of the core protocols of the Internet Protocol Suite. TCP is one of the two original components of the suite, complementing the Internet Protocol (IP), and therefore the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. TCP is the protocol that major Internet applications such as the World Wide Web, e-mail, remote administration and file transfer rely on. Other applications, which do not require reliable data stream service, may use the User Datagram Protocol (UDP), which provides a datagram service that emphasizes reduced latency over reliability.

There are several types of applications designed to complement TCP. These applications are the Simple Mail Transfer Protocol (SMTP), the File Transfer Protocol (FTP), and the Secure Shell (SSH). The SMTP operates in a similar way as the gathering, organizing, packaging and delivering of mail in the traditional sense. Traditionally speaking, when someone places their physical letter (message) in a mail box, or at the post office, a mail room employee takes possession of that mail, completes the necessary action consistent with mailroom protocol procedures, places it in the queue for dispatching it to the intended destination. The Simple Mail Transfer Protocol operates in that same vein in an electronic sense by accepting a message and using TCP to send it to an SMTP module on another host, and then the SMTP makes use of a local electronic mail package to store the incoming message in a user's mailbox.

The File Transfer Protocol relies on the use of TCP when uploading and transferring text and binary files that are to and from multiple host systems on the Internet, and the Secure Shell is actually what its name implies, a secure method of logging on to a system, that is, a terminal or other computerized device, with the ability to encrypt data ensuring that no one is able to contaminate, taint or corrupt that data.

There are other protocol suites that serve the same purposes as TCP/IP, such as the Open Systems Interconnection (OSI) reference model. However, TCP has led the way in effortless communications and the exchange of data across the Internet spectrum. OSI was developed by the International Organization for Standardization (ISO) as a model for a computer protocol architecture and as a framework for developing protocol standards. OSI was also based on the use of layering with each layer performing a related subset of the functions required to communicate with another system. It was anticipated that the OSI model would bypass other known communication suites, however, this system came up short and the TCP/IP is still the leading architecture in this area of technology.

9.2.3 The User Datagram Protocol (UDP)

UDP is one of the core members of the Internet Protocol Suite, the set of network protocols used for the Internet. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without requiring prior communications to set up special transmission channels or data paths.

UDP uses a simple transmission model without implicit handshaking dialogues for providing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagrams may arrive out of order, appear duplicated, or go missing without notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system. If error correction facilities are needed at the network interface level, an application may use the Transmission Control Protocol (TCP) which is designed for this purpose.

UDPs stateless nature is also useful for servers answering small queries from huge numbers of clients. Unlike TCP, UDP supports packet broadcast (sending to all on local network) and multicasting (send to all subscribers).

Common network applications that use UDP include: the Domain Name System (DNS), streaming media applications such as IPTV, Voice over IP (VoIP), Trivial File Transfer Protocol (TFTP), IP tunneling protocols and many online games.

9.2.4 Internet Protocol (IP)

The Internet protocol (IP) is another layer or critical piece in the successful transmission of data. It is designed to handle the address part of a packet so that it gets to the right destination. It is used at the Internet layer to provide routing function across multiple networks, implemented not only in the end systems but also in routers. Routers are essential to the entire process of effecting efficient communications in that they are used to connect multiple networks, and disseminate information from network to network on various routes on the Internet from the sender to the receiver. The network layer involves the routing of data to its specified location and the physical layer pertains to the actual hardware that is used to physically generate the data that is to be routed.

TCP/IP communication protocol is a prime example of the cutting edge technology that has evolved since the development of the Internet. It is considered one of the most powerful tools to have evolved during the era of computer technology, and its development has essentially revolutionized the speed, clarity, and authenticity of data communications like never before. Since its conception, the efficiency of computer networking has grown by leaps and bounds. As the world continues to seek methods of increasing efficiency of data communications, and as new applications are defined, there is no doubt that TCP/IP will continue being one of the leading protocol suites ensuring enhanced Internet connectivity and easy communications between people in the Inter-networking realm of cyber space.

9.3 Network Switches

A network **switch** is a computer networking device that connects network segments. There are basically two types of switches: unmanaged and managed.

An unmanaged switch works right out of the box. It's not designed to be configured, so you don't have to worry about installing or setting it up correctly. Unmanaged switches have less network capacity than managed switches. You'll usually find unmanaged switches in home networking equipment.

Managed switches create a network. Routers connect networks. A router links computers to the Internet, so users can share the connection. A router acts as a dispatcher, choosing the best path for information to travel so it's received quickly.

Switches and routers are the building blocks for all business communications, from data to voice and video to wireless access. Switches are able to connect different types of networks, including Ethernet, Fiber Channel, ATM, ITU-T G.hn and 802.11. Figure 9.4 shows a switched network topology.

9.3.1 Layer 1 Hubs

A network hub, or repeater, is a simple network device. Hubs do not manage any of the traffic that comes through them. Any packet entering a port is broadcast out or "repeated" on every other port, except for the port of entry. Since every packet is repeated on every other port, packet collisions affect the entire network, limiting its capacity.

There are specialized applications where a hub can be useful, such as copying traffic to multiple network sensors. High end switches have a feature which does the same thing called port mirroring. The throughput of a hub is very limited. For example, an 8-port hub connects to 100 M network using the uplink connector. Then each port can only have the maximum speed of 100M/8 = 12.5 M.

Hubs and switches appear very much alike. Both also serve the same basic function, to provide a centralized connection for local networks and other network equipment. In fact, hubs and switches operate quite differently. Replacing a hub with a switch is definitely worthwhile.

A network hub is essentially a "dumb" device that has no knowledge of what devices are plugged into it, where data is coming from or where it's going.

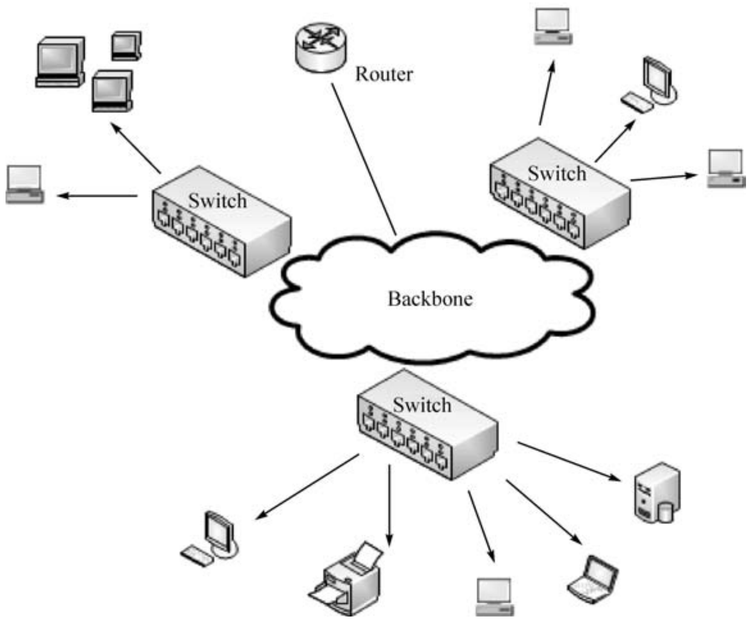


Figure 9.4 Switched network

Therefore, when Computer A sends data to Computer B through a hub, the hub simply retransmits, or “repeats,” the data to each and every port in a process called a “broadcast.” Each device connected to the hub then checks to see if the data, known as a frame, is addressed to it in order to determine whether to accept or ignore it.

The problem is that the hub’s “send everything to everyone” approach is that it generates lots of unnecessary network traffic, which in turn causes network congestion and seriously limits performance. Since all devices connected to a hub must take turns sending or receiving data, they spend lots of time waiting for network access or retransmitting data that was stepped on by another transmission. The lights on a hub indicate such “collisions.”

9.3.2 Ethernet Switch

A network switch is an intelligent device that can actively manage the data going through it. Unlike a hub, a switch knows which computers are connected to each of its ports as well as the source and destination address of each data frame in encounters. The switch identifies each computer by its unique MAC address.

When Computer A sends data to Computer B over a switch, the switch sends only the data to the port Computer B is connected to. This limits unnecessary traffic, provides better performance, and leaves other devices connected to the switch free to simultaneously communicate via their respective ports. So even while Computer A and B are exchanging data, Computer C and Computer D (and maybe others) can still transmit and receive data.

In addition, a hub is a shared-bandwidth device. It provides 10 or 100 Mbps of total data throughput that is shared among all of its ports. Each port on a switch gets a portion of the 10 or 100 Mbps of bandwidth.

Switches were considerably more expensive than hubs, so the hub was commonly used in situations where cost was as important a consideration as performance. Nowadays, switches have become quite inexpensive and common. A basic 16 port 10/100 Mbps switch can be had for as little as $60. While faster Gigabit (1,000 Mbps) switches and those that provide more advanced features such as data prioritization or network management do cost more, they're still relatively inexpensive.

9.4 Routers

Routers are one of the most critical components involved in the success of the Internet. They allow for packets to flow from source to destination with little delay. The process used is packet switching, which builds a virtual circuit between two connection points. While other protocols, standards, and technologies are a major part of the Internet that we enjoy today; the router has to be credited with the Internet's existence. The device allows network engineers to break large networks into smaller logical sub-networks. Then these routing devices are able to communicate with each other to ensure that packets can traverse the Internet and reach their ultimate destination. The Internet has truly made the world a smaller place, and has allowed efficient communications between any two locations in the world. On the other hand, routers are vulnerable to all kinds of intruders. The solution to security on the Internet is often attributed to firewalls and anti-virus programs. Perhaps the routers themselves can also play a critical role in network security.

A router is a device that forwards data packets between computer networks, creating an overlay Internet work. A router is connected to two or more data links from different networks. When a data packet comes in on one of the links, the router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the Internet work until it gets to its destination node.

The most familiar type of routers are home and small office routers that simply pass data, such as Web pages and e-mail, between the home computers and the owner's cable or DSL modem, which connects to the Internet (ISP). However, more sophisticated routers range from enterprise routers which connect large business or ISP networks up to the powerful core routers that forward data at high speed along the optical fiber lines of the Internet backbone.

9.4.1 History of Routers

Much of the credit for the invention of the router is unfairly given solely to Cisco systems, but it is important to also credit William Yeager. He was a researcher at Stanford when in 1980 his boss assigned him the task of connecting the computer science department, medical center, and the department of electrical engineering. His first router operated on a 3 megabit Ethernet. He later updated his routing system to function with the Internet Protocol (IP).

Shortly after Yeager created his routing configuration, Len Bosak and Sandy Lerner started a company called Cisco. They also worked with Yeager, and were provided access to his code and improved upon it. They released the first commercially available router product and Hewlett-Packard was their first customer. Cisco went on to become the largest distributer of routing products.

9.4.2 Architecture

Routers range in size from small home devices to large rack mounted units. The larger units are extremely costly, but can handle a high rate of traffic. Cisco CRS-1 is capable of routing 40 Gigabits per slot, and has a capability of housing 16 slots. The CRS series of router can even be run in a multi-rack configuration with a total routing capability of up to 92 terabits per second. Obviously this type of router is intended for a location that is processing tremendous traffic.

Figure 9.5 shows the system architecture of Cisco CRS-1 router. There are mainly three components inside the router: router processors, multi-stage switch fabric, and 40 Gbps link cards.

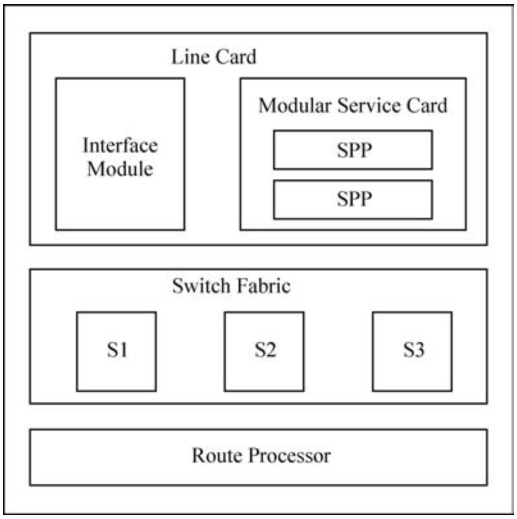


Figure 9.5 Cisco CRS-1 architecture

A line card consists of interface module and Modular Services Card (MSC). The interface module provides the physical connections to the network. The module services card is a high-performance layer 3 forwarding engine. Each MSC is equipped with two high-performance Cisco Silicon Packet Processors (SPP). The SPP is a sophisticated ASIC consisting of 188 32-bit RISC processors per chip with 40 Gbps processing power.

The switch fabric provides the communications path between line cards and is a three-stage, self-routed architecture with 1,296 × 1,296 buffering. The three stages of switching follow:

- S1 is connected to the ingress link card
- S2 supports multicast replication
- S3 connected to the egress line card.

The route processor is available to execute algorithms such as Border Gateway Protocol (BGP) and supports up to 4 GB DRAM, 40 GB hard drive (HDD) and 2 × 32 GB solid state drive (SSD). The system software is built on memory-protected software architecture.

Most office routers are nothing more than a specialized computer that is optimized for the function of routing traffic at a highly efficient rate. A router generally contains seven major internal components: CPU, RAM, NVRAM, Flash Memory, ROM, Console, and interfaces. The router itself can have one or more different types of interfaces such as fiber optic, coaxial, and Ethernet. The Ethernet connection is one of the most common types of interfaces.

The smaller home routers are better suited for handling a few computers, but still offer many advanced features. A Linksys E-series router has integrated switch ports. In some models these ports are as fast as 1,000 megabits per second. This unit is also capable of connecting to both wired and wireless devices. This is because a wireless access point (WAP) is built inside.

In order to allow for inter-networking, routers have to accomplish a few fundamental tasks. Stallings lists the following essential functions:

- Connect or link multiple networks
- Route and deliver data between the end systems attached to the networks
- Not require modification of architecture of the attached networks in order to accomplish these tasks.

Routers are fundamentally different from other networking devices such as hubs, switches and bridges. This is because routers provide the inter-networking capabilities, while these other devices are used to segment a single network. Figure 9.6 shows a diagram of a simple Internet work that includes a switch. The router is the device that is connecting the two networks and allowing an Internet to work between them.

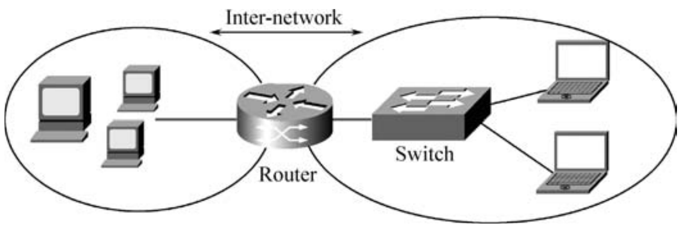


Figure 9.6 Inter-networking diagram

9.4.3 *Internet Protocol Version 4 (IPv4)*

At the core of a router is a simple process. When a packet arrives at an input port, the processor makes a decision on where the packet is to be directed and sets the switch to direct the packet to the correct output port. In order to be able to make these routing decisions, a router must function on protocols. If it were not for standardized protocols, the Internet would not be where it is today. These protocols allow devices from different vendors to transmit data with confidence to another device knowing that the other device will be able to understand the message and reply. There are many different protocols, but the following are some of the most important to understand.

The Internet Protocol Version 4 (**IPv4**) is an Open Systems Interconnection (OSI) layer 3 addressing protocol that uses 32-bits to identify a device. All packets sent on the network contain a header for layer 3 addressing. The address is similar to an address for a house, and the packet is like a letter that is being mailed by the postal system. Routers use this addressing to determine which network the packet belongs to. This is because the packet is split into two different portions: the network bits and the host bits. To identify which portion is network and which is host, a number called a subnet mask is configured on the computer or router. There is a one-to-one correlation between the bits in the IPv4 address and the subnet mask. All address bits that correspond with a binary 1 in the subnet-mask are part of the network. All bits that line up with a 0 are the host bits.

Routers use IP addresses to send packets to other routers. Routing protocols help the router make the decision on the best path to take. IPv4 itself does have built-in security. Therefore, the network administrator can configure the router to use access control lists (ACL) to block or allow specific addresses. This can allow an administrator to segment their network into different layers. One suggested method is to divide a network into a public, application, and data layer. Then servers can be placed in one of the three layers based on which systems will be able to reach the server. In some cases this access control measure is only slightly protective, because an attacker can spoof an IP address and pretend to be an authorized machine.

A better method of providing security with IPv4 is through the use of IP security (IPSec), which allows for both authentication and confidentiality. When IPSec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. IPSec allows, with the use of authentication headers (AH), for two hosts to prove their authenticity. This provides assurance that the device on the other end is the intended recipient. IPSec also allows, with the use of encapsulating security payload (ESP), for the IP packet's data to be encrypted. This ensures that only the intended recipient can view the traffic. IPSec is an excellent choice to use when traffic over an unsecure network of routers is necessary. IPSec is a technology that came about to solve the need for security and is not being currently used by most home users and many enterprise users.

9.4.4 Internet Protocol Version 6 (IPv6)

Due to IPv4 only using 32-bits for addressing, there are not enough unique addresses to assign to every public device on the Internet. $2^{32} = 2^2 \times (2^{10})^3$ is roughly four billion (4×10^9). Many have been able to stretch the life of IPv4 by using technologies such as network address translation (NAT) along with private IPv4 addresses, but this is not a permanent solution to the problem. The real solution is to use a larger address space. IPv6 solves this problem by using 128-bits for an address ($2^{128} \approx 3.4 \times 10^{38}$) theoretically. In addition to solving the issue of the number of addresses, **IPv6** is also including IPSec as part of the protocol's standard. This will ensure that IPv6 traffic through a router is secure from sniffing attacks.

More and more modern routers are being made with IPv6 capabilities built in, but many older devices are not ready for the IPv6 transition. It is important to upgrade older routers to support IPv6 in order to take advantage of these new capabilities. It may be advantageous to upgrade older routers because newer routers include advanced security features.

9.4.5 Open Shortest Path First

Another dynamic routing protocol that is used by routers is open shortest path first (**OSPF**), which uses the Dijkstra algorithm. This protocol will calculate the shortest path first and then create the routing table, which will result in the best path for network traffic.

The Dijkstra algorithm involves the process of checking the difference between every link in search of the quickest route. This process is repeated on each link or series of links until a table is built that identifies the cost associated with each path. The router can use this table to find the best route for each packet that it processes. OSPF takes advantage of the Dijkstra algorithm in order to ensure that traffic can travel on the quickest router to its final destination. It is important for a router to have a clear and correct routing table so that it can maintain a fast processing rate.

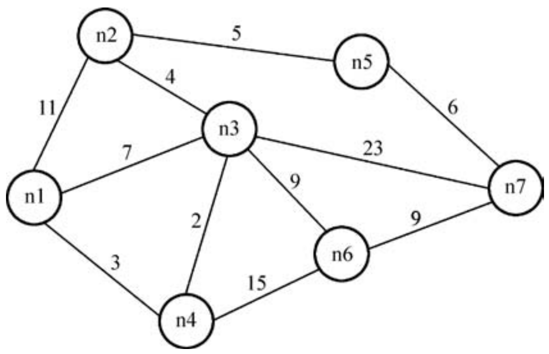


Figure 9.7 Shortest path algorithm

In Figure 9.7 for example, to find the shortest path from vertex n1 to vertex n7, the Dijkstra algorithm provides a way by just looking one step ahead to find the solution. This is particularly useful for writing a program running on computers. The conceptual algorithm is as follows:

```
function Dijkstra(Graph, source):
    for each vertex v in Graph:            // Initializations
        dist[v] := infinity ;              // Unknown distance function from source to v
        previous[v] := undefined ;         // Previous node in optimal path from source
    end for ;

    dist[source] := 0 ;                    // Distance from source to source
    Q := the set of all nodes in Graph ;   // All nodes in the graph are unoptimized

    while Q is not empty:                  // The main loop
        u := vertex in Q with smallest distance in dist[] ;
        if dist[u] = infinity:
            break ;                        // inaccessible from source
        end if ;
        remove u from Q ;

        for each neighbor v of u:          // where v has not yet been removed from Q.
            alt := dist[u] + dist_between(u, v) ;
            if alt < dist[v]:              // Relax (u,v,a)
                dist[v] := alt ;
                previous[v] := u ;
                decrease-key v in Q;       // Reorder v in the Queue
            end if ;
        end for ;
    end while ;
    return dist[] ;
end Dijkstra.|
```

When the program runs, it checks the neighbors (v) of the current vertex (u) to see whether distance from the current vertex (u) plus the distance between u and the neighbor (v) is the smallest. If neighbor (v) has another path which is greater than the sum of the distance u plus the distance between u and v, then the distance for the neighbor (v) is updated with the small value.

```
for each neighbor v of u:
    alt := dist[u] + dist_between(u, v) ;
    if alt < dist[v]:
        dist[v] := alt ;|
    end if ;
end for ;
```

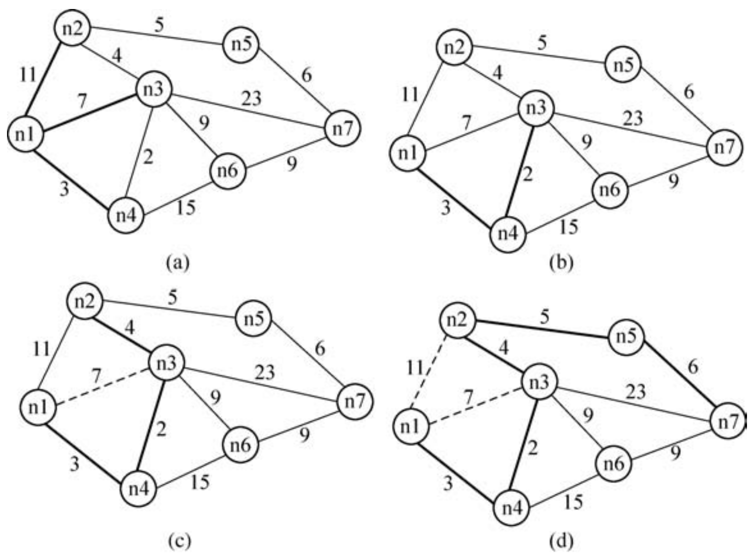


Figure 9.8 Dijkstra shortest path algorithm. (a) initial value, (b) update n1 → n3, (c) update n1 → n2, (d) final

Using Dijkstra's shortest path algorithm, the shortest path from n1 to n7 can be obtained as follows.

Starting from n1, initially the shortest path from n1 to n2, n3, and n4 is 11, 7 and 3 respectively. This is shown in Figure 9.8(a). However, there is another path n1 → n4 → n3 its cost is less than n1 → n3 directly. So we take the better one (3 + 2) shown in (b). Next we found out that n3 → n2 is better than n1 → n2. So we take the new value of 9, shown in (c). Continue the loop statement, we got the shortest path from n1 to n7 is 20 (n1 → n4 → n3 → n2 → n5 → n7) shown in (d).

9.4.6 Throughput and Delay

The two most important factors for a router are to find the best route and to have highest throughput. The Dijkstra algorithm answers the first question. To increase the throughput, we need to look at how to reduce the delay while a packet is at a router. We will use the M/M/1 model that we discussed in Chapter 8 to look into the problem. In Figure 9.9 shows a single queue single server system.

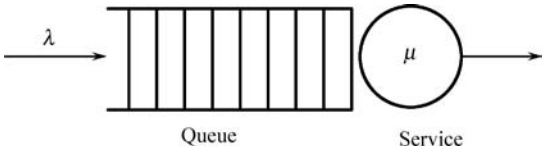


Figure 9.9 Single queue single server system

Here we define traffic intensity

$$\rho = \frac{\lambda}{\mu}(<1) \tag{9.1}$$

The expected number of users N in the queue is given by

$$\mathrm{N} = \frac{\rho}{1-\rho} \tag{9.2}$$

The total expected waiting time (include queue and servers) is

$$T = \frac{1}{\mu - \lambda} \tag{9.3}$$

To verify the equations are correct, first look at equation (9.2). When the arrival rate λ is approaching the processing rate μ, then the traffic intensity is approaching the limit $\rho \to 1$. When ρ is approaching to 1, the number of users in the queue N is approaching to infinity. This means that more and more packets will wait in the queue not being processed. For equation (9.3), when the arrival rate λ is approaching to the processing rate μ, then the waiting time T is approaching to infinity. This means packets will stay in the queue forever. Ideally the arrival rate λ should less than the processing rate μ. In practice ρ should keep no more than 80% to get the best performance. If traffic intensity reaches 90% or higher, then the waiting time will increase dramatically.

9.5 Gateways

In a communications network, **gateway** is a network node that is used for interfacing with another network that uses different protocols. A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It also requires the establishment of mutually acceptable administrative procedures between both networks. A protocol translation/mapping gateway interconnects networks with different network protocol technologies by performing the required protocol conversions. Gateways, also called protocol converters, can operate at any network layer. The activities of a gateway are more complex than that of the router or switch as it communicates using more than one protocol.

For the TCP/IP network, a gateway is a node that serves as an access point to another network. A **default gateway** on a network is a node that routes traffic from a workstation to another network segment. The default gateway commonly connects the internal networks and the outside network. Sometimes the gateway node could also act as a proxy server and a firewall. The gateway is also associated with both a router, which uses headers and forwarding table to determine where packets are sent, and a switch, which provides the actual path to the packet in and out of the gateway. In other words, a default gateway provides an entry point and an exit point in a network.

9.6 Wireless Networks and Network Address Translation (NAT)

Wireless network by its name is a computer network without using any cables. Data are transmitted and received through radio signals in the air. It is easy to setup and costs less money than Ethernet networks.

Wireless local area network is the technology that is rapidly improving and quickly being adopted by IT industries and professionals. The study of wireless LAN includes definitions, history, architecture, protocols, war driving, and security issues. The study on NAT includes configurations, administration, and security issues.

The modern IT industry needs professionals with understanding of the IT communications, structural design, computer hardware, software, networking, new concepts, and tools. Some of the notions were most significant and considerable to the growing IT industry. The use of wireless networks extended in a way that it has become part of our life. It is essential to the IT professionals to understand the basics of wireless networking.

9.6.1 Wireless Networks

A typical wireless local area network is a network which is connecting multiple devices or computers with no wires. It uses spread-spectrum technology which is radio waves to connect among devices in a limited region. It became popular because of the portability and simplicity of installation. Businesses like MacDonald, Starbucks, and shopping malls have begun to offer wireless access to their customers.

Wireless LAN (VLAN) connects multiple computers or devices over a short distance. It connects to the Internet through a router or access point. The spread-spectrum technologies gives portability to devices while remaining connected to the network. 802.11 comes with several flavors of 802.11a and transmits at 5 GHz and may increase up to 54 megabits of data/sec. WLAN uses **orthogonal frequency-division multiplexing or** OFDM technology. Table 9.1 compares different wireless networks.

Table 9.1 Different wireless networks

	PAN	LAN	MAN	WAN
Standards	Bluetooth	802.11 HiperLAN2	802.11 WiMAX (.16)	GSM, GRPS CDMA, 3G/4G
Speeds	<1 Mbps	11–54 Mbps	11–100+ Mbps	10 to 384 Kbps 1.8/3.6–7.2 M
Range	Short	Medium	Medium-Long	Long
Application	Peer-to-Peer Device-Device	Enterprise Networks	E1 replacement Last mile	Mobile Phones Cellular data

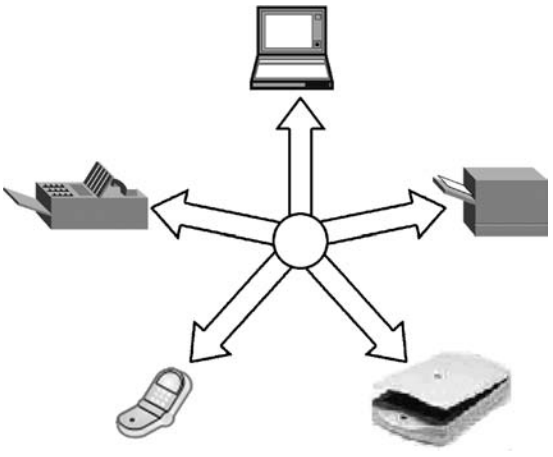


Figure 9.10 Wireless Personal Area Networks

Wireless Personal Area Networks (WPAN) connects multiple devices in a comparatively shorter distance (Figure 9.10). For instance, many of us use Bluetooth headsets which use invisible Infrared light to connect to your phone or computer. WPAN equipped devices are becoming more popular nowadays because of the size and price of the device.

Wireless Metropolitan Area Networks (WMAN) connects numerous wireless networks. (See Figure 9.11) For instance, WiMAX which is a WMAN established by IEEE 802.16 standards.

Wireless wide area networks (WWAN) cover huge areas, long distances like from cities, states, and countries. These types of networks uses point to point microwave

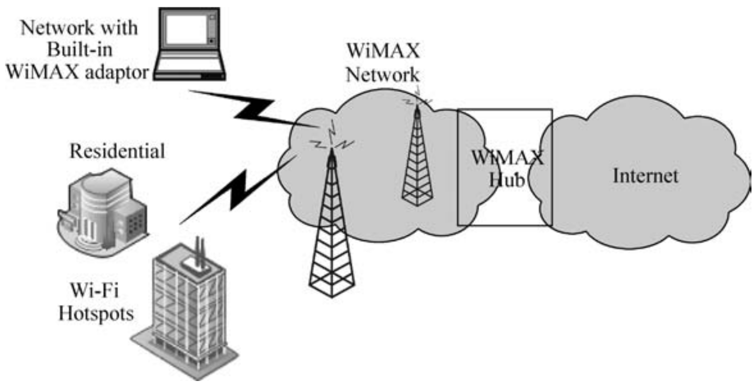


Figure 9.11 Wireless Metropolitan Area Network

links using parabolic dishes on the 2.4 GHz band where MAN and WAN uses antennas. A WWAN includes base station gateways, access points, and wireless bridging relays. For example–Cell phone networks, 3G, 4G, GSM, and GPRS.

Wireless mesh networks are created by radio nodes and structured by a mesh topology. Every node can send messages on behalf of other nodes. If the power is lost it has the capability to re-route to another node.

Cell phone networks:

- Personal Communications Service (PCS) uses a radio band that can be used by mobile phones in North America and South Asia. For example, Sprint first established a PCS network.
- Digital Advanced Mobile Phone Service (D-AMPS) is an upgraded version of AMPS, is being phased out due to advancement in technology. The newer GSM networks are replacing the older system.
- Global System Mobile (GSM) is divided into three parts such as switching, base station, and operation support systems. It is a common standard for most cell phone companies.

9.6.2 Wireless Protocols

IEEE 802.11 is a set of standards for implementing wireless local area network (WLAN) computer communication. IEEE 802.11 is a subset of IEEE 802 which is the standard for LAN and MAN. The 802.11 standard (Table 9.2) provides the foundation for wireless network products using the Wi-Fi band.

Table 9.2 Different 802.11 standards

Protocol	**Release Date**	**Op. Frequency**	**Date Rate (Typical)**	**Date Rate (Max)**	**Range (Indoor)**
Legacy	1997	2.4–2.5 GHz	1 Mbit/s	2 Mbit/s	?
802.11a	1999	5.15–5.35/5.47–5.725/5.725–5.875 GHz	25 Mbit/s	54 Mbit/s	∼30 meters (∼100 feet)
802.11b	1999	2.4–2.5 GHz	6.5 Mbit/s	11 Mbit/s	∼50 meters (∼150 feet)
802.11g	2003	2.4–2.5 GHz	11 Mbit/s	54 Mbit/s	∼30 meters (∼100 feet)
802.11n	2006 (draft)	2.4 GHz or 5 GHz	200 Mbit/s	540 Mbit/s	∼50 meters (∼150 feet)

- 802.11 legacy
  It is now obsolete. It is the original version of the standard IEEE 802.11 first invented in 1997. It can transfer data up to 2 megabits/sec. It has three physical layer technologies:
  – Diffuse infrared operating at 1 Mbps.
  – Frequency-hopping spread spectrum up to at 2 Mbps.
  – Direct-sequence spread spectrum up to at 2 Mbps.
- 802.11b
  The highest data transfer rate is 11 Mbps. It uses the media access method which is the same as the original standard and direct extension of the modulation technique. It first came onto the market in 2000. 11b technology is widely accepted in the market because of its lower price and has an increased throughput compared to the original standard. But it has interference issues with other products operating in the 2.4 GHz band such as microwave ovens, Bluetooth devices, baby monitors, and cordless telephones.
- 802.11g
  802.11g first came out it 2003 and has less interference issues with 2.4 GHz band products than the 802.11b has. It uses the same OFDM method as 802.11a. It has a maximum speed of 54 Mbps at physical layer. 11g has an average throughput of 22 Mbps and it is backwards compatible with 802.11b. 802.11g standard was widely accepted by customers because of the cost and faster data transfer rate. At the end of 2003, the majority of dual-band 802.11a/b products become dual-band/ tri-mode which means a NIC card or router will support all a, b and g.
- 802.11a
  It uses data link layer protocol. 802.11a protocol can transmit at 5 GHz to 54 megabits/sec of data. It applies orthogonal frequency-division multiplexing (OFDM), which splits that radio signal into numerous sub-signals before they reach a receiver. The range of 802.11a is less than 802.11b/g. 802.11a signals are absorbed more by buildings, and solid objects in their way than a smaller wavelength.
- 802.11n
  In 2009, 802.11n came out with better performance than the previous versions of a, b, and g. It has new multiple-input multiple-output antennas (MIMO) which previous versions did not have. 802.11n is able to work with 2.4 GHz band products with less interference.

9.6.3 WLAN Handshaking, War Driving, and WLAN Security

Below is the handshaking process when browsing the Web using WLAN:

1. Wireless NIC card sends out a radio signal to search for the local router.
2. When the router is found, then the router takes the instruction and sends it to a gateway server using WAP.

3. Then the gateway server regains the data using HTTP.
4. The gateway server encodes the HTTP data as WML.
5. The WML-encoded data is sent back to the client computer.
6. Then the page is ready to be viewed.

The WLANs became popular because of convenience, cost efficiency, portability, and simplicity of incorporation with other network devices and access points. Mostly new computers come with WLAN adapters. Some of the benefits include:

- *Portability* – Users can easily move around with the laptops or smartphones. Users can access the Internet even when they are outside of the home network. Many restaurants, hotels, shopping malls offer free Wi-Fi connections. A user can buy a 4G air card from cellular carriers and enjoy high-speed Internet almost anywhere.
- *Pricing* – The cost of wireless products is a little bit higher than wired products. The Wi-Fi devices price is continually decreasing.
- *Productivity* – It is to be seen that in Starbucks, McDonald, Wendys, Taco-bell, employees use wireless headsets to take customers' orders with faster than traditional speed.
- *Installation* – To setup a wireless network requires setting up wireless routers, switches and antennas. But wired networks are hard to install and on many occasions it is hard to take connections from one building to another.
- *Expandability* – The network admin can easily increase connections using the existing equipment. But in a wired network, new employees will require extra cabling.

Years ago, two people drove around the city with laptop computers. They went to different areas where wireless networks were detected and started capturing packets using their pre-configured laptop. They used the software name *NetStumbler 0.4.0* and they captured 802.11b/g packets using *Link Ferret 3.10* software. They found roughly 50% of the wireless traffic is not encrypted. Among the 50% of encrypted wireless networks, some are encrypted with WEP. The 104-bit WEP key can be easily cracked by using Aircrack 2.1 software or many others that can be downloaded free from the Internet.

Wireless gateway attacks and rogue access point installations are among some of the common attacks on wireless networks. A hacker may set a proxy server in between the wireless client and gateway. When a user tries to connect to the Internet using wireless NIC card, it will connect to the proxy instead. The proxy will establish an SSL (Secure Socket Layer) connection. Next, the proxy will create an SSL connection to the wireless gateway. The hacker will see the wireless gateway and authenticate to it without the owner's knowledge.

DoS Attacks and Session Hijacking are other types of attack. The hackers can insert traffic into the radio network without login to a wireless router. The 802.11

MAC is designed to allow multiple networks to share the same space and radio channels. DoS attacks can be done to WLANs to overwhelm networks.

MAC Spoofing attack is the hacker who is are able to change the MAC or Ethernet address of a wireless NIC card, by using software such as Mac Makeup.

In ARP (Address Resolution Protocol) Poisoning can be done using Cain and Abel software, an attacker can exploit the ARP Cache and intercept network traffic between two computers in the network.

9.6.4 *Security Measures to Reduce Wireless Attacks*

Here are some security measures recommended to reduce being a victim of wireless attacks:

- Use WPA with TKIP/AES and use the CCMP with AES in future.
- Turn off the SSID broadcast.
- Change default WEP settings.
- Change default SSID.
- Change default IP address.
- Change default login/password.
- Use MAC filtering in AP level or in RADIUS server.
- Position and shield the antenna to direct the radio waves to a limited space.
- Use limited DHCP clients to control the number of users who may connect to the WLAN.
- Use a firewall between AP and the wired LAN to secure the wired LAN from further intrusion.
- Enable the accounting and logging to locate and trace.
- Use good intrusion detection software to monitor the network activity.
- Use VPN, IPSec and secure tunneling.
- Use honey pots or fake APs in the regular network to confuse the hacker.
- Use biometric authentication such as fingerprinting.
- Update drivers and firmware of the routers.
- Use strong passwords.

9.6.5 *The Future of Wireless Network*

The future of WLAN will change because of the success of wireless air-cards or portable hotspots which use 3g, 4g, WiMAX or cellular networks. Telecoms are rapidly improving the speeds of their networks with fourth-generation technology. Some people claim that MiFi, a device that can support up to five users to access Internet through cellular network (Figure 9.12), could eventually replace wired broadband subscriptions in the same way that Americans are canceling home phone lines in favor of cell phones.

The future of wireless network is expected to be a meeting of different kinds of wireless technologies, such as cellular technologies, wireless local area networks,

Figure 9.12 AMiFi device

wireless metropolitan area networks, wireless sensor networks, and traditional wired networks.

Although users will be oblivious to the specific underlying network being used by their applications, the networks should be able to provide the resource (bandwidth) with guaranteed quality of service (QoS) (Didi et al., 2009). Users should be able to move seamlessly among different networking technologies, for example, among Ethernet, WLANs, WiMAX, and 2G/3G/4G, with stringent QoS requirements.

- *Mobility*: The existing Internet which is built for stationary end-hosts, does not handle mobility easily within the Internet architecture. The issue of mobility relates to handling changes in location and underlying network connectivity of mobile end-systems at each protocol layer.
- *Multi-homing*: In the past, most hosts/nodes or computers had only one networking interface. Hosts stayed within one network with one egress path. However, multi-homed hosts or devices having multiple networking interfaces are becoming more common.
- *Routing Scalability*: A common solution for IP network sites to allow their service providers to change is to use Provider Independent (PI) addresses. However, these addresses are not aggregatiable and lead to an exponential increase in size of the routing table.
- *Deploy Ability*: To deploy a new system from scratch can be hectic. On many occasions companies lack an appropriate and realistic deployment plan.

9.6.6 Network Address Translation

NAT (Network Address Translation) is used for a set of IP addresses for internal and external traffic. NAT acts as a mediator between the Internet and LAN. It uses only one IP address for an entire LAN (Figure 9.13). It is a solution to the shortage of IP addresses also. NAT has three functionalities:

- It hides the internal IP and acts as a firewall.
- It organizes the use of internal IP and removes IP conflict with other networks.
- It can join numerous ISDN connections to a single Internet connection.

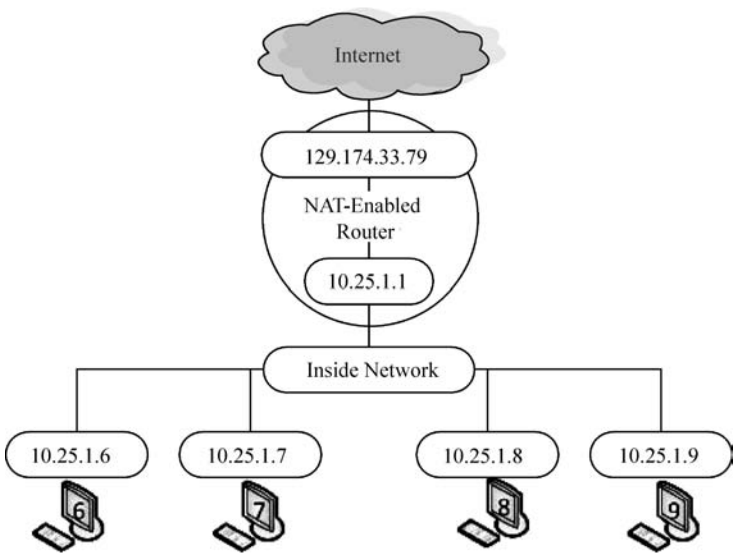


Figure 9.13 A wireless router with NAT

The router in Figure 9.14 uses the IP 10.25.1.1 for the inside and 129.174.33.79 for outside network. When users try to connect to Internet from any of these computers, the NAT will convert the 10.25.1.x to 129.174.33.79.

Static NAT changes an unregistered IP address to registered IP. It is useful when someone from outside is trying to access inside network. Dynamic NAT changes a registered IP address to an unregistered IP. Figure 9.14 shows the static (a) and dynamic (b) address translation.

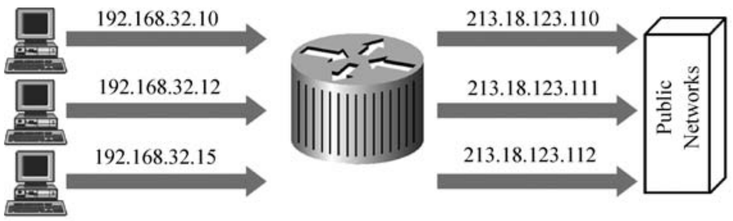
Dynamic NAT acts as a firewall between a client computer and the Internet. Outside users can only connect to the inside computer by using NAT. NAT prevents outside users accessing inside computers directly but lets inside computers connect to the Internet.

Static NAT allows external devices to initiate connections to computers on the stub domain. It allows a host in stub domain to keep a specific IP address when connecting to the Internet.

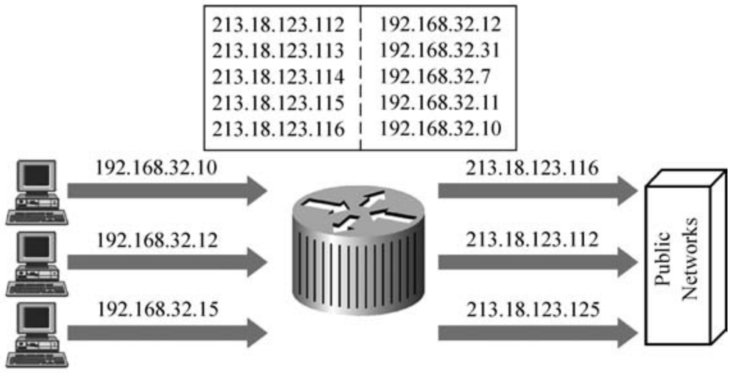
9.6.7 Environmental and Health Concerns Using Cellular and Wireless Devices

There have been increased concerns about the safety of wireless communication. National Cancer Institute (NCI) at the National Institute of Health (NIH) has the following statements regarding the cell phone with cancer risk:

- Cell phones emit radio frequency energy, a form of non-ionizing electromagnetic radiation, which can be absorbed by tissues closest to where the phone is held.



(a) Using static NAT, 192.168.32.12 will always translate to 213.18.123.111



213.18.123.112	192.168.32.12
213.18.123.113	192.168.32.31
213.18.123.114	192.168.32.7
213.18.123.115	192.168.32.11
213.18.123.116	192.168.32.10

(b) Using a dynamic NAT, 192.168.32.15 will translate to the first available address in the range from 213.18.123.100 to 213.18.123.150

Figure 9.14 NAT address translations

- The amount of radio frequency energy a cell phone user is exposed to depends on the technology of the phone, the distance between the phone's antenna and the user, the extent and type of use, and the user's distance from cell phone towers.
- Studies thus far have not shown a consistent link between cell phone use and cancers of the brain, nerves, or other tissues of the head or neck. More research is needed because cell phone technology and how people use cell phones have been changing rapidly.

A team of 31 scientists from 14 countries, including the United States, made the decision after reviewing peer-reviewed studies on cell phone safety. The team found enough evidence to categorize personal exposure as "possibly carcinogenic to humans." The type of radiation coming out of a cell phone is called non-ionizing. It is not like an X-ray, but more like a very low-powered microwave oven.

What microwave radiation does in most simplistic terms is similar to what happens to food in microwaves, essentially cooking the brain. So in addition to leading to a development of cancer and tumors, there could be a whole host of other effects like cognitive memory function, since the memory temporal lobes are where we hold our cell phones.

The European Environmental Agency has pushed for more studies, saying cell phones could be as big a public health risk as smoking, asbestos, and leaded gasoline. The head of a prominent cancer-research institute at the University of Pittsburgh sent a memo to all employees urging them to limit cell phone use because of a possible risk of cancer.

"When you look at cancer development – particularly brain cancer – it takes a long time to develop. I think it is a good idea to give the public some sort of warning that long-term exposure to radiation from your cell phone could possibly cause cancer," a statement from a professor who has studied radiation for more than 30 years.

Results from the largest international study on cell phones and cancer was released in 2010. It showed participants in the study who used a cell phone for 10 years or more had doubled the rate of brain glioma, a type of tumor. Children's skulls and scalps are thinner. So the radiation can penetrate deeper into the brain of children and young adults. Their cells are at a dividing faster rate, so the impact of radiation can be much larger.

The Apple iPhone 4 safety manual says users' radiation exposure should not exceed FCC guidelines: "When using iPhone near your body for voice calls or for wireless data transmission over a cellular network, keep iPhone at least 15 millimeters (5/8 inch) away from the body."

BlackBerry Bold advises users to "keep the BlackBerry device at least 0.98 inch (25 millimeters) from your body when the BlackBerry device is transmitting."

The logic behind such recommendations is that the further the phone is from the body, the less radiation is absorbed. Users can also use the speakerphone function or a wired earpiece to gain some distance. Users can text instead of talk if they want to keep the phone away from their faces.

9.7 Network Security

As Internet use has expanded over the decade with an overwhelming number of users. The crimes and ethical issues related to computing and its use has also become an issue. Organizations and Internet users implement different mechanisms to protect their information or data over the network from malicious users. Firewall technology is no doubt the widely used network equipment to detect and prevent attacks. Networks are inherently insecure. Therefore, strong security measures must be taken independently of the network to protect the components of the network as well as the data flowing through the network.

Since the frequency of attacks on networks has increased, denial of services (DoS) and IP spoofing are very common. DoS and distributed denial of services (DDoS) are major problems because they are very hard to detect.

The prompt growth of computer networks has transformed the prospect of network security. An easy availability causes computer networks to be vulnerable against various and possibly devastating threats from hackers. Researchers have developed Intrusion Detection Systems (IDS) capable of discovering attacks in numerous available environments. Intrusion Prevention Systems (IPS) evolved to resolve uncertainties in passive network monitoring by assigning detection systems on the line of attack. IDS and IPS are capable of providing prevention commands to firewalls and access control changes to routers. They can also make access control assessments based on application content, rather than IP address or ports as traditional firewalls do. The next advancement is the blend of IDS and IPS known as Intrusion Detection and Prevention Systems (IDPS) capable of identifying and averting attacks from happening.

9.7.1 Introduction

Internet has already become an essential part of our lives. It is where we work, shop, enjoy, or communicate with one another. It is where we access our banking records, credit card statements, tax returns, and other highly sensitive personal information. Regardless of the business, an increasing number of users on private networks are demanding access to Internet services. In addition, corporations want to offer websites for public access on the Internet. Moreover, its purpose and use is growing quickly and exponentially, and the need for security has also increased substantially over the years.

9.7.1.1 Network Security Basics

As the number of computers users all over the world has increased dramatically over the years, the numbers of malicious users and attackers have also increased. And that security has become one of the primary concerns when an individual or organization connects its computer or private network to the Internet. Network administrators, responsible for maintaining system efficiency and security, have an increasing concern about the security of their network when they expose private data and networking infrastructure to Internet malicious users. Thus, to provide the required level of protection, an organization needs a security policy to prevent unauthorized users from accessing resources on the private network and to protect against the unauthorized export of private information. Even if an organization is not connected to the Internet, it may still want to establish an internal security policy to manage user access to portions of the network and protect sensitive or secret information.

A **firewall** is anything, whether hardware or software (or a combination of hardware and software), that can filter the transmission of packets of digital information as they attempt to pass through a boundary of a network. Figure 9.15 shows a diagram of a firewall and network.

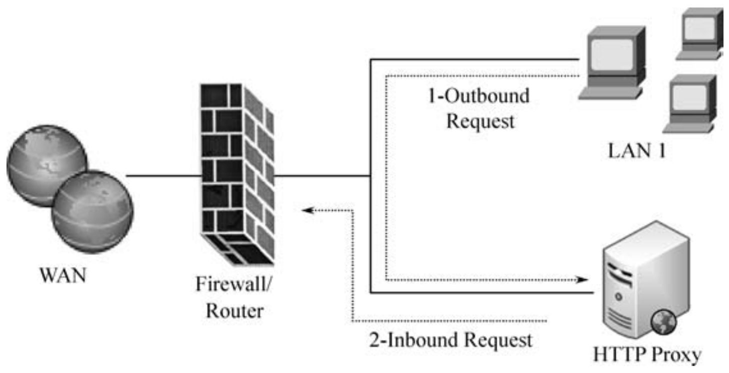


Figure 9.15 Firewall and network

9.7.1.2 Information Hacking

It is difficult to describe a typical hacker attack because intruders have different levels of technical expertise and many different motivations. Some hackers are intrigued by the thrill or challenge involved, others just want to make life more difficult for others, and still others are out to steal sensitive data for profit or self financial gains. Moreover there are many sites which can teach how to hack, trace IP addresses, and port scanning and that is why the concern of security has increased. There are different categories of hackers:

- *Neophyte hackers* – a set of people full of stuff and enthusiasm but low on experience, usually hack for thrill.
- *Script kiddy hackers* – is a group of people who use existing and frequently well-known and easy-to-find techniques and programs or scripts to search for and exploit weaknesses in other computers on the internet – often randomly and with little regard or perhaps even understanding of the potentially harmful consequences.
- *White hat hackers* – refers to an ethical hacker or a computer security expert, who specializes in penetration testing and in other testing methodologies to ensure the security of an organization's information systems.
- *Gray hat hackers* – sometimes arguably act illegally, though in good will, on how they disclose vulnerabilities. they usually do not hack for personal gain or have malicious intentions, but may be prepared to technically commit crimes during the course of their technological exploits in order to achieve better security.
- *Blue hat hackers* – refer to outside computer security consulting firms that are employed to bug test a system prior to its launch, looking for exploits so they can be closed.
- *Black hat hackers* – refer to hackers who break into a computer system or network with malicious intent or for self financial gain.

System administrators and developers adopt different network security standards and mechanisms to prevent their network from different hackers and malicious users. One of the mechanisms used to protect private network from outside Internet network is Internet firewall.

9.7.1.3 Benefits of Using Firewalls

Firewalls are helping protect intrusions and provide a secure and reliable environment from unauthorized users. In addition, here are some of the benefits of having a firewall from a security point of view.

- Monitor and record the access information to a network.
- Firewalls are used to secure and filter e-mail viruses and malware attacks.
- Firewalls are used to give role based access to different parts of a network and hence manage use of the resources on the network.
- Firewalls allow calculate usage of the Internet, mainly to check individual users and performance effects.
- Allows computing ethics issues to be managed and enforced for an organization.

9.7.1.4 Firewall and Firewall Technologies

An Internet firewall is a system or group of systems that enforces a security policy between an organization's network and the Internet. The firewall determines which inside services may be accessed from the outside, which outsiders are permitted access to the permitted inside services, and which outside services may be accessed by insiders (see Figure 9.16). It inspects all incoming traffic when it passes through it. The firewall must permit only authorized traffic to pass, and the firewall itself must be immune to penetration. Unfortunately, a firewall system cannot offer any protection once an attacker has gotten through or around the firewall. Firewalls offer a convenient point where Internet security can be monitored and alarms generated. It should be noted that for organizations that have connections to the Internet, the question is not whether but when attacks will occur.

Several types of firewall technologies are available. And their capabilities are also different based on the TCP/IP protocol layer that each is able to examine. Firewalls can inspect application traffic and use it as the basis for policy. Basic firewalls

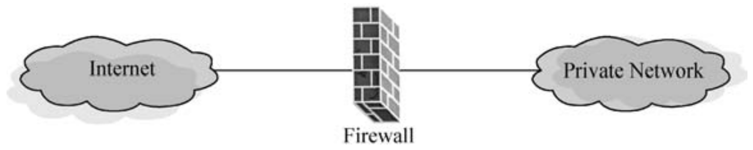


Figure 9.16 Firewall in a network

operate on one or a few layers – typically the lower layers (Network layer, and transport layer) – while more advanced firewalls examine all of the layers. Those firewalls that examine more layers can perform more granular and thorough examinations. Firewalls that understand the application layer can potentially accommodate advanced applications and protocols and provide services that are user-oriented. For example, a firewall that only handles lower layers cannot usually identify specific users, but a firewall with application layer capabilities can enforce user authentication and log events to specific users.

Firewalls are often placed at the perimeter of a network. Such a firewall can be said to have an external and internal interface, with the external interface being the one on the outside of the network. These two interfaces are sometimes referred to as unprotected and protected, respectively. Firewalls are often combined with other technologies – most notably routing. Many firewalls also include content filtering features to enforce organization policies not directly related to security. Some firewalls include intrusion prevention system (IPS) technologies, which can react to attacks that they detect to prevent damage to systems protected by the firewall.

Firewall devices at the edge of a network are also sometimes required to do more than block unwanted traffic. A common requirement for these firewalls is to encrypt and decrypt specific network traffic flows between the protected network and external networks. This nearly always involves virtual private networks (VPN), which use additional protocols to encrypt traffic and provide user authentication and integrity checking. VPNs are most often used to provide secure network communications across not trusted networks.

9.7.2 Firewall Architecture

A firewall quality is measured by the components integrated inside it. A typical firewall is composed of one or more of the following building blocks in addition to the software enabling it:

- Packet filtering routers
- Application level gateways or proxy server
- Circuit-level gateway.

9.7.2.1 Packet Filtering Routers

A packet filtering router in firewalls make a permit or deny decision for each packet that it receives. The router examines every packet's header information to determine whether it matches one of its packet-filtering rules. The header information consists of the IP source address, the IP destination address, the encapsulated protocol (TCP, UDP, ICMP, or IP Tunnel), the TCP/UDP port source port, the TCP/UDP destination port, the ICMP message type, and the incoming interface of the packet. If a match is found and the rule permits the packet, the packet is forwarded according to the information in the routing table. If a match is found and the rule denies the packet, the packet is discarded. If there is no matching rule, a user-configurable default parameter determines whether the packet is forwarded or discarded. There are two types of packet filtering:

1. *Service-dependent filtering* – Where packet-filtering rules allow a router to permit or deny traffic based on a specific service, since most service listeners reside on well-known tcp/udp port numbers. For example a telnet server listens for remote connections on port 23 and smtp listens to port 25. Some typical filtering rules include:
   - Permit incoming Telnet sessions only to a specific list of internal hosts.
   - Permit incoming FTP sessions only to specific internal hosts.
   - Deny all incoming traffic from specific external networks, and so on.
2. *Service-independent filtering* – These are used to filter attacks that are difficult to identify using basic packet header information because the attacks are service independent. Routers can be configured to protect these types of attacks and they are difficult to specify. Some of these types of attacks include:
   - Source IP Address Spoofing Attack
   - Source Routing Attacks
   - Tiny Fragment Attacks.

9.7.2.2 Application Level Gateways

Application-level gateways are often referred to as a "bastion host" and is a designated system that is specifically armored and protected against attacks. Application level gateways allow information to flow between systems but do not allow the direct exchange of packets, unlike packet filtering.

9.7.2.3 Circuit Level Gateways

A circuit-level gateway is a specialized function that can be performed by an application-level gateway. A circuit-level gateway simply relays TCP connections without performing any additional packet processing or filtering.

Practical examples of firewalls include:

- *Packet–filter router firewall.* The most common Internet firewall system consists of a packet-filtering router deployed between a private network and the Internet. (Figure 9.17)
- *Screened host firewall.* Firewalls which employ both a packet-filtering router and an application level gateway (bastion host). (Figure 9.18)
- *Demilitarized zone (DMZ) or screened-subnet firewall.* These kinds of firewalls employ two packet-filtering routers and an application level gateway (bastions hosts). Figure 9.19 shows a DMZ and the network.

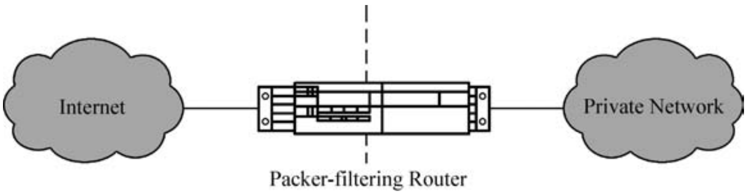


Figure 9.17 Packet Filtering Router Firewall

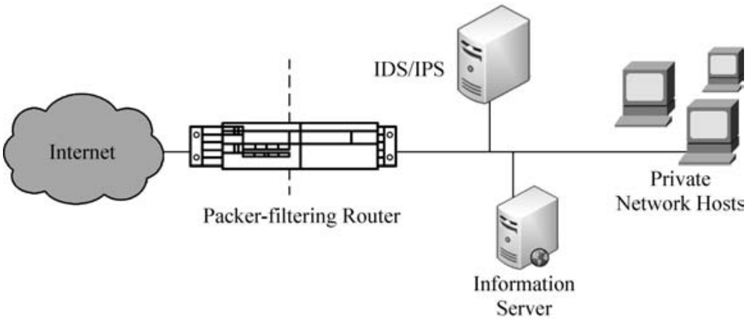


Figure 9.18 Screened Host Firewall

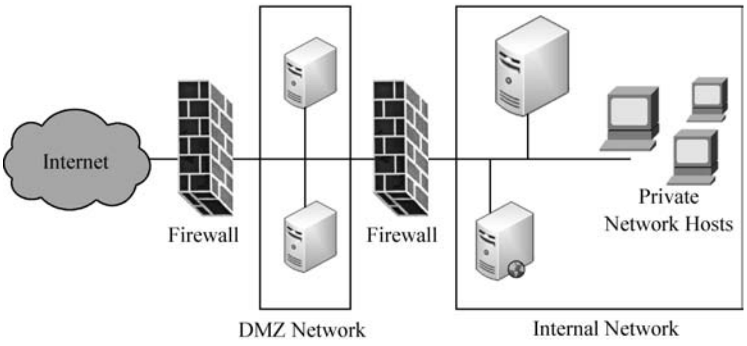


Figure 9.19 Screened Subnet Firewall

9.7.3 Constraint and Limitations of Firewall

Below are some constraints and limitations of firewalls.

- Firewalls cannot protect against attacks that do not go through the firewall.
- Internet firewalls cannot protect against the type of threats posed by traitors or unwitting users.

- Firewalls do not prohibit traitors or corporate spies from copying sensitive data and leaving the building.
- Internet firewalls cannot protect against the transfer of virus-infected software or files. Concerned organizations should deploy anti-virus software at each desktop to protect.
- Internet firewalls cannot protect against data-driven attacks – attacks that occur when seemingly harmless data is mailed or copied to an internal host and is executed to launch an attack.

A firewall management program can be designed in one of two basic ways:

1. *A default-deny policy.* The firewall administrator lists all the allowable network services, and everything else is denied.
2. *A default-allow policy.* The firewall administrator lists network services which are prohibited, and everything else is accepted.

A default-deny approach to firewall security is the most secure, but many networks use the default-allow approach instead.

9.7.4 Enterprise Firewalls

There are various types of enterprise firewalls of network security devices that help optimize IT operational and financial risk management to maximize IT performance for business. Here are a few examples of the most popular versions:

- *Cisco firewall* – Cisco is a manufacturer of a widespread range of computer security and networking products. They offer routers for joining public and private IP networks for mobile, data, voice, and video including switches, IP phones, access points, and servers. They also provide services for application networking and network security products including firewall, intrusion detection prevention, virtual private network (VPN) and e-mail security products.
- *Barracuda firewall* – Contain security products such as spam and virus firewall, storage and data protection. Barracuda Networks products are easy to use and deploy. There is no software installation or network modifications are required. It also features Web filtering and Web application firewall products, network access control and application delivery is managed through a simple to use interface which can help secure networks from e-mail, Web and Internet messaging threats.
- *Checkpoint firewall* – This option provides top-tier firewall software for enterprise security. It features a stateful packet inspection, VPNs and content filtering and also employs all the standard features expected from a commercial grade firewall. Checkpoint's enterprise-wide firewall management interface is a first-rate choice

for companies needing to manage a geographically scattered network from remote locations.

- *Juniper networks netscreen* – A maker of routers, switching equipment, firewall, networking security hardware and security systems, that integrates firewall, VPN, traffic management, denial of service (DoS) and distributed DoS security intended for large enterprise, carrier and data center networks.
- *Windows ISA server* – Microsoft Internet Security and Acceleration Server (ISA) is multi-layered firewall network security that employs both stateful packet inspection and application-layer filtering to deliver Internet security.
- *Sonicwall* – Provides intrusion prevention, malware protection and application control, and is a producer of comprehensive network security solutions providing appliances for e-mail security, remote access and data security to meet enterprise network management requirements for a total firewall solution which includes VPN, anti-virus, and spyware protection.

Firewall plays a key role in traditional security architecture, since it controls most of the incoming and outgoing traffic of an enterprise. Essentially the firewall is almost a must-have in each enterprise. To review the challenges for the traditional architecture, undoubtedly it is necessary to address the limitation of traditional firewalls.

9.8 Summary

Data communication is to study the transmission of data over various channels or networks. The goals are to be high throughput, fast speed, error-free, and reliable. In order to let many signals share a common communication channel, signal encoding and modulation are needed. Shannon theorem states that in order to restore the original signal at destination lossless, signal must be sampled at double the frequency of the highest frequency of the original signal.

The transmission control protocol and Internet protocol (TCP/IP) is the foundation of the modern computer network. Based on the functions, people divided network with layers. Among the layer models, four layer, five layer and seven layers models are common.

TCP provides reliable stream delivery and virtual connection service to applications. It is connection-based so there are some overhead or in order word delay. UDP is connection-less and the overhead is small so the speed is faster. Unlike TCP, UDP does not guarantee the packet delivery so there may be packet loss. For file transfer, we need to guarantee there is no error otherwise the file would not be open again. In this case TCP is the perfect choice. For Voice over IP (VoIP) such as IP phone, a millisecond error would not be noticed as long as the voice is synchronized with the speaker. So UDP is definitely the perfect choice.

Network switches are the basic devices used to set up networks and connect network segments. A gateway serves as a bridge to interface with different protocols. Besides, it also connects the internal networks and the outside networks.

Routers find the shortest paths from source to destinations and guarantee with least delay. It is a piece of key equipment to set up virtual circuits between sources and destinations. Usually people use graph theory to study network. Dijkstra's algorithm is a very useful program to find the shortest paths between sources and destinations. The queue theory is another important tool to analyze the throughput of a router and therefore to reduce the packets delay.

Wireless networks have many advantages over the Ethernet networks. Even though the speed generally cannot match the broad band Internet, it can satisfy the need for most occasions and it is getting better. WLAN provides flexibility to enable people and different kinds of devices to be connected anywhere at any time. Due to radio frequencies being spread in the air, enhancing the security is an important task for most users.

NAT let several users share one IP and separate an internal network from outside networks. It is a technology commonly seen in home and office wireless networks.

Network security involves securing information, data, and network equipment from all kinds of attacks and intruders. Firewall, IDS, and IPS are common technologies used to secure networks and devices. Distributed denial of services (DDoS) attacks are still common and hard to be detected and prevented.

To enhance the network security, organizations usually use white hat hackers (or in other words ethical hacker) to find the vulnerabilities of an IT system. On the other hand, there are many black hat hackers on the Internet who are constantly trying to break into a computer system or network to steal data.

A firewall provides a single point of defense between two networks – it protects one network from the other. Usually, a firewall protects the company's private network from the public or shared networks to which it is connected.

Exercises

9.1 What are the differences between signals, channels, data, and information?

9.2 Why do most music radio stations use FM instead of AM?

9.3 The voice of an opera singer contains the highest frequency component of 20 Hz. In order to re-play on a high fidelity (Hi-Fi) amplifier with the best quality, what is the sample rate to consider? If the voice is traveling through a telephone, what does the sound quality (band-width) need to be?

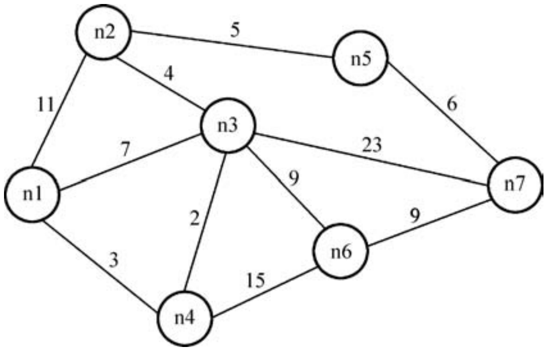
9.4 Describe the functions of each layer for the five-layer TCP/IP protocol.

9.5 Some people say TCP is better since it is connection-based which means it can correct errors. Some people say UDP is better because it is faster. What is your opinion on this?

9.6 With regard to the throughput, what is the difference between Ethernet switches and hubs?

9.7 What are the most important factors for a router?

9.8 Use the Dijkstra's algorithm and the graph below to find the shortest path from n6 to n1.



9.9 A single queue single server router has the capacity of processing 1,000 Mbps. When the packet arrival rate is 500 Mbps, find out the number of packets in the router waiting to be processed, and the average waiting time (ms). When the packet arrival rate reaches to 990 Mbps, what will happen?

9.10 For Internet access, most families have only one public IP. However each family may have many devices, computers, smartphones, pads, printers, and so on. How should multiple devices be handled over one public IP?

9.11 Amazon, Visa and many other websites have suffered DDoS attacks. Find the most recent incident of DoS or DDoS attack and describe how to stop them.

9.12 For CISSP certification, the study of information security is divided into ten domains. List the 10 domains in information security.

9.13 Most companies have security policies in place. False positive means there was an attack trigger and action was taken but actually it was not an attack. False negative means there was an attack but it was not detected. Provide examples of why sometimes false positive is good and sometimes it is not.

9.14 Describe how VPN works and why it is considered a secure connection.

9.9 Virtual Cyber-Security Laboratory

As we know, distant learning provides a flexible learning environment where people can learn from books and other reference material, from instructors if a distance education class is formed, and from each other by collaboration. On the other hand, distant learning lacks tools and environment for people to participate in lab activities and gain hands on experience.

To solve the problem, the book authors have started working on a series of projects. The goals are to let readers explore the vulnerabilities of information systems and to gain hands-on experiences so that information systems can be well managed and protected. The first one of the series is completed and is named Virtual Cyber-Security Lab (01). It contains experiments where interested readers can practice some basic hacking techniques based on injections.

The virtual lab is hosted on www.wiley.com/go/wang/comp_arch and can be accessed using the access key along with this book.

References

A brief history of Wi-Fi (2004) Economist, 371(8379), 26. http://www.economist.com/.

Austin, R., Holden, G., Mattord, H., and Whitman, M. (2009) *Guide to Firewalls and Network Security: With Intrusion Detection and VPNs*, 2nd edn, Course Technology, Cengage Learning.

Bangeman, E. (2008) Canadian university says no to WiFi over health concerns. Arstechnica.com. http://arstechnica.com/news.ars/post/20060222-6235.html.

Basu, A. and Riecke, J. (2001) Stability issues in OSPF routing. SIGCOMM '01 Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications. doi: 10.1145/383059.383077.

Bragg, R., Rhodes-Ousley, M., and Strassberg, K. (2004) *Network Security: The Complete Reference*, McGraw-Hill, Emeryville, CA.

Briere, D. D., Bruce, W.R., and Hurley, P.J. (2009) *Wireless Home Networking for Dummies*, John Wiley & Sons, Inc., US.

Cisco (2011) Cisco Carrier Routing System. http://www.cisco.com/en/US/prod/collateral/routers/ps5763/prod_brochure0900aecd800f8118.pdf.

Cisco Systems (2011) Linksys E-Series Routers: User Guide. http://homesupport.cisco.com/en-us/support/routers/E4200.

CNNTech (2011) MiFi the future of wireless internet. http://articles.cnn.com/2010-11-18/tech/mifi.wireless.hotspots_1_wi-fi-verizon-wireless-mifi?_s=PM:TECH.

Didi, F., Labiod, H., Pujolle, G., and Feham, M. (n.d). Mobility and QoS of 802.11 and 802.11e.

Dix, J. (2006, March 27) Router man. NetworkWorld. http://www.networkworld.com/supp/2006/anniversary/032706-routerman.html.

Dye, M.A., McDonald, R., and Rufi, A.W. (2008) *Network Fundamentals*, Cisco Press, Indianapolis, IN.

Egevang, K. and Francis, P. (1994) The IP Network Address Translator (NAT). Cray Communications. http://www.studentwebstuff.com/mis/showthread.php?t=8111.

None to claim their bones: IBM i5/OS Information Center. http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?topic=%2Frzatj%2Ficnat.htm.

IEEE 802.11. Wikipedia. (2011) http://en.wikipedia.org/wiki/IEEE_802.11.

Issac, B., Jacob, S.M., and Mohammed, L.A. (2005) The art of war driving and security threats—a malaysian case study. Proceedings of the IEEE International Conference on Networks (ICON 2005), 124–129. Kuala Lumpur, Malaysia, November 16–18, 2005.

Jain, R., Paul, S., and Pan, J. (2010) Future Wireless Networks: Key Issues and a Survey. Retrieved from http://www.cs.wustl.edu/~jain/papers/ftp/fwn_ijcn.pdf.

Lammle, T. (2007) *Cisco Certified Network Associate Study Guide*, 6th edn, Wiley Publishing, Indianapolis, IN.

Lar, S.-U., Liao, X., Ma, Q., and urRehman, A. (2011) Proactive security mechanism and design for firewall. *Journal of Information Security*, **2**(3), 122.

Liu, H.H. (2009) *Software Performance and Scalability: A Quantitative Approach*, John Wiley & Sons, Hoboken, NJ.

Lowe, D. (2004) *Networking All-in-one Desk Reference for Dummies*, John Wiley & Sons, Inc, US.

Mohammed, L.A. and Issac, B. (2005) DoS Attacks and Defense Mechanisms in Wireless Networks. Proceedings of the IEE Mobility Conference 2005 (Mobility 2005), Guangzhou, China, November 15–17, 2005.

Pcmag (2003) WLAN time line. http://www.pcmag.com/article2/0,2817,1275516,00.asp#fbid=9JdY3VPacno.

Pfleeger, C.P. and Pfleeger, S.L. (2007) *Security in Computing*, 4th edn, Pearson, Boston, MA.

Phifer, L. (2007). WPA PSK Crackers: Loose Lips Sink Ships. http://www.wi-fiplanet.com/tutorials/article.php/3667586.

Protocol Guide: TCP/IP Protocols: Transport Layer Protocols: TCP: Transmission Control Protocol. (2007) *Network Protocols Handbook*, pp. 48–49, http://www.developerdotstar.com/mag/articles/PDF/DevDotStar_VanDamme_OO_CASE.pdf.

Reynders, D. and Wright, E. (2003) *Practical TCP/IP and Ethernet Networking*, Elsevier.

Rodriguez, E. (2011) Cisco Router Hardware. http://www.skullbox.net/routers.php.

Runnels, T. (2005) History of Wireless Networks. Arp Sparnet. http://www.arp.sprnet.org/default/inserv/trends/history_wireless.htm.

Sharma, R., Singh, G., and Agnihotri, R. (2010) Comparison of performance analysis of 802.11a, 802. 11b and 802. 11g standard. *International Journal On Computer Science & Engineering*, **1**(6), 2042–2046, http://www.enggjournals.com/ Chart 1.

SSI service strategies (2011) Add an Additional Layer of Protection with NAT. http://www.ssimail.com/Netaddtran.htm.

Stallings, W. (2009) *Business Data Communications*, Pearson Education, Upper Saddle River, NJ.

Tulloch, M. and Tulloch, I. (2002) *Microsoft Encyclopedia of Networking*, 2nd edn, Microsoft Press, Redmond, WA.

Tyson, J. (2010) How Network Address Translation Works. http://computer.howstuffworks.com/nat.htm/printable.

WAP (1999) What Is It? The Wireless Applications Protocol. Director (00123242).53(4) p. 132, http://www.iod.com.

White, G. (2003) *Security + Certification*, McGraw-Hill, Emeryville, CA.

Wireless LAN Standards. *International Arab Journal of Information Technology*, **6**(4), 403–411, http://www.enggjournals.com/.

Wireless LANs (2011) Wireless LANs – A definition, and WLAN technology further explained.

Wireless network. Wikipedia (2011) http://en.wikipedia.org/wiki/Wireless_network.

WLAN time line. http://www.ieee802.org/11/Reports/802.11_Timelines.htm.

10

Design and Implementation: Modifying Neumann Architecture

In this chapter, we will discuss the design and implementation of an invention of a new type of computer architecture that is capable of preventing data stored on computers from being stolen.

Modified Neumann architecture has shown promise in solving computer security problems. The architecture separates the network communication component from the other parts of the computer system with a separate system bus. Data exchange between the two system buses can only be performed through the Bus Controller via a command issued by the computer operator. A secured micro-operating system is used to guarantee the integrity of communications.

The micro-OS that runs on a microprocessor resides in the "red" zone. It uses a firmware image that automatically loads to the microprocessor memory during booting process. A watch-dog monitors the system and resets it whenever a preset threshold is met. In addition, memory protection and virtualization techniques are used to enhance runtime security and prevent code injection.

The invention has been prototyped and tested and was awarded a national high-tech grant.

10.1 Data Security in Computer Systems

Computers nowadays are very easily broken into via a network especially through the Internet. Therefore, information stored on a computer such as SSN, credit cards, bank accounts, and personal privacy information and so on, is vulnerable to computer hackers.

Firewalls to some extent can prevent information stored on a computer from being stolen. However it can only be effective in a certain period of time. Some firewalls are mere software; some others even though they use "hardware" to setup a "wall" between the computer and the outside world, the core components are based on algorithms or, in other words, software. On the other hand, a firewall is not designed to use on personal computers or handheld devices. So it cannot guarantee that the information stored on a computer will never be stolen.

10.1.1 Computer Security

Computer security is to study and to enhance the confidentiality, integrity, and availability of computer systems (Bishop, 2002). Certified Information Systems Security Professionals (CISSP) classifies computer and information security into ten domains (Gregg, 2009):

- Access Control
- Application Development Security
- Business Continuity and Disaster Recovery Planning
- Cryptography
- Information Security Governance and Risk Management
- Legal, Regulation, Investigations, and Compliance
- Operations Security
- Physical Security
- Security Architecture and Design
- Telecommunications and Network Security.

Computer security also can be classified into physical security and technological security (ISC^2, 2010). Technological security can be divided into five aspects: application security, operating system security, network security, architecture security, and data security.

However, if the applications running on the Web server are not properly designed to prevent attacks, then hackers may be able to break into the database server from the Web server. One of such attacks is called SQL injection attack.

We are all tired of installing security updates on our computers. This is an example of operating system security. An operating system (no matter whether it is Windows, Mac iOS, or Linux) contains hundreds of millions of lines of source code. It is very likely that there are bugs and other vulnerabilities. If an attacker has very good knowledge about the operating system and has discovered its vulnerabilities, security bleach is inevitable.

Data security is to guarantee the safety of data stored on a computer system (Daswani et al., 2007). There are all kinds of data on computers, from Web browsing data (cookies, history, etc.) to sensitive data such as passwords, banking information or even SSN. Identity theft will happen if those data are not properly protected (Merkow and Brethaupt, 2005).

10.1.2 Data Security and Data Bleaches

Privacy is one of the biggest concerns nowadays. Some employers use centralized monitoring software to monitor employee's emails and other private information. Google operates under a streamlined privacy policy that enables the Internet's most powerful company to dig even deeper into the lives of its more than 1 billion users. Google will share their users' data across Gmail, Google Plus, YouTube and other products.

Identity theft is a more serious problem which has drawn attention recently from Congress. Nearly 10 million people were victimized by identity theft last year, according to *Time Magazine*, the loss reached $5 billion. In early March 2005, the nation's largest data miner ChoicePoint with 19 billion data files including drivers' license, SSN, credit histories, birth certificates, real estate deeds, and even thumbprints and DNA was broken into and some 145,000 people's data was extracted. In Senator Charles Schumer's words, "Our system of protecting people's identity is virtually nonexistent in this country." His staff was able to download personal information on the likes of Dick Cheney and Brad Pitt from a ChoicePoint rival, Westlaw.

In a recent letter US Senator Orrin Hatch wrote to one of the book authors in responding to the new invention in computer security, "Identity theft is a serious problem that has drawn much attention recently in Congress. As we know, the damage caused can go beyond money and privacy and become a real threat to our national security."

So far, many current researches or inventions may have some impact on reducing the risk of information theft in one way or another, however, those solutions have not solved the information security problems thoroughly due to the limitation of the computer architecture they used. There is a problem that exists in the John von Neumann computer architecture model – the foundation of computer architecture. If this problem is not solved, information stored on a computer will hardly be secure.

The main goal for the invention is to propose a new type of secure computer system with a microprocessor-based hardware-assistant and a micro-OS that can not only monitor the system security but also enable computers to prevent intruders from getting data stored in the computer system. In a recent pending patent, the book authors proposed a new computer architecture model – modified Neumann model. Based on this new model, the network communication component is separated from the other parts of a computer system with a separate system bus. All components in a computer system (except network) reside on another system bus. Data exchange between those two system buses can only be performed through the Bus Controller via a command issued by the computer operator. So, data stored in this computer (main storage) can only be accessed by the computer operator. In other words, user data is isolated from outside networks and therefore cannot be accessed even when the computer is compromised or taken over from outside networks.

In addition to preventing information theft, the system contains a security agent that can monitor and report any security related events. The recorded security events can be transmitted to or viewed by the central monitoring system in real-time.

A test bed has been developed and initial experiments show that the system is very promising. The major technology breakthrough is that it can prevent unauthorized access of any information in a protected computer system. Security is guaranteed as the system is implemented using the new patent pending secure computer architecture (hardware).

The theme of the research is stated as following:

- Study the widely used John Neumann computer architecture model.
- Modify the Neumann model and proposed a new secure architecture model.
- Complete the technical details and the implementation.

An add-on security board is constructed by using a co-processor, FPGA and other digital circuits together with kernel software. A multiport I/O and a dual-port memory interface circuits are designed in combing with the add-on circuit board. The dual-bus system can be switched over one another through another add-on circuit name Bus Controller. A micro-OS manages the add-on operations and monitors the system security. In the following sections, each of the new designs will be discussed in detail.

10.1.3 Researches in Architecture Security

There are many researches related to the secure computer architecture area. Largman et al. (2004) proposed "automatically create multiple sequentially or concurrently and intermittently isolated and/or restricted computing environments method to prevent viruses, malicious, or even computer or device corruption and failure." According to this method, untrusted content is only exposed in the user processor logic environment in a temporary storage. The question remaining for this method is how to determine which content is trusted and which is not. There might be a predetermination process.

Anderson put "removable trusted (hardware) gateway devices" between each of the inputs/outputs and the bus to secure the file transmission. As described, the approval of access data is depended on a so-called "LOCK." Once the lock is stolen, intercepted or hacked, sensitive data is then open to those hackers.

Hewlett-Packard (HP-Compaq, 2002) have been working on a new type of Secure Platform Architecture (SPA). It is a set of software interfaces built on top of HP's Itanium-based product line. SPA will enable operating systems and device drivers to run as unprivileged tasks and will allow services to be authenticated and identified. The problem exists in the SPA and is that, as the company described, it uses a set of software interfaces to authenticate and identify the tasks. Once the system is compromised, SPA will not be able to function well.

Sean Smith and Steve Weingart (Smith and Weingart, 1999) developed a prototype using a high-performance, programmable secure coprocessor. It is a type of software, hardware, and cryptographic architecture (Suh et al., 2005). This architecture addressed some issues especially on how to secure programs running on coprocessors and system recovery. In terms of secure information and data, there is a lot of work which needs to be done.

Recently, MIT researchers proposed secure processors that enable new applications by ensuring private and authentic program execution even in the face of physical attack.

10.2 Single-Bus View of Neumann Architecture

Neumann architecture is the foundation of modern computer systems. It is a **single bus**, stored program computer architecture that consists of a CPU, memory, I/O, and storage. The CPU is composed of a control unit (CU) and arithmetic logical unit (ALU) (von Neumann, 1945). Almost all modern computers are Neumann computers which is characterized as a single system bus (control, data, address) with all circuits attached to it.

10.2.1 John von Neumann Computer Architecture

John von Neumann wrote "First Draft of a Report on the EDVAC" in which he outlined the architecture of a stored-program computer. He proposed a concept that has characterized mainstream computer architecture since 1945. Figure 10.1 shows the Neumann model.

A "system bus" representation of the Neumann model is shown in Figure 10.2. This is just another view of the Neumann model, with the introduction of the concept of DMA.

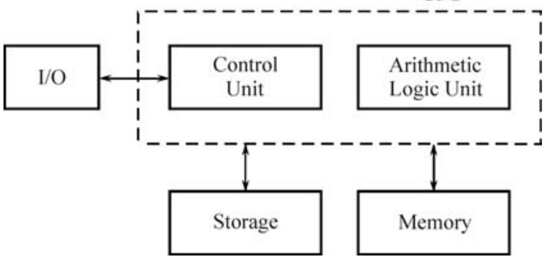


Figure 10.1 Block diagram of John von Neumann's computer architecture model

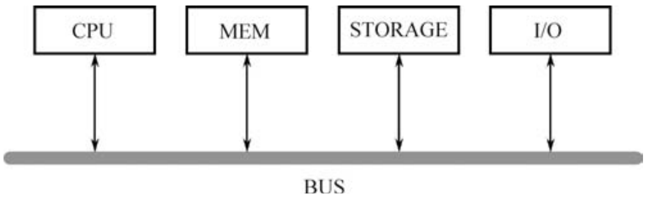


Figure 10.2 A “system bus” representation of the Neumann model. It is equivalent to Figure 10.1 with the introduction of DMA

10.2.2 Modified Neumann Computer Architecture

Since the 1990s, computer networks, especially the Internet, have been widespread around the world. Computers are no longer only being used to compute as a stand-alone machine. The feature of information exchange through a network is a vital component in today’s computers. Unfortunately John Neumann was not able to foresee this change. One can argue that we can consider a network is part of an input/output device which is already included in the Neumann model. However, the network interface is so important that it is not appropriate to classify it as in the general I/O device category. Furthermore, an I/O device in the Neumann model refers to those devices such as a keyboard, a display, and a printer and so on which are used for direct interaction with the computers. Now, the way people use a computer is quite different from that of sixty years ago. So a modification of Neumann’s computer architecture model is necessary to reflect this change. Figure 10.3 shows the modified Neumann model. In Figure 10.3, a network unit (interface) is added to the computer system bus so that the I/O unit only deals with input and output devices such as keyboard, mouse, display, and so on. Separating a network unit from the general I/O offers great advantages.

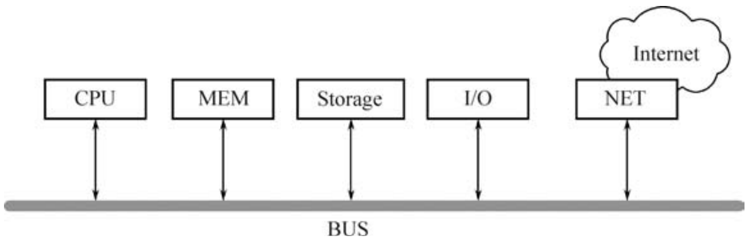


Figure 10.3 Modified Neumann computer architecture model. Here a network interface is added to the Neumann model and is separated from the general input and output devices

10.2.3 Problems Exist in John Neumann Model

As we all know, Newton's three laws and theories of gravitation make essentially identical predictions as long as the strength of the gravitational field is weak, which is our usual experience. They were so dominant that no one dared to doubt them until Einstein predicted that the direction of light propagation would be changed in a gravitational field. This discovery modified Newton's law and made the modern theory of gravity possible.

The Neumann model is so dominant that no one has dared to challenge it since its birth in 1945. However if we look into the Neumann model from a security perspective, we would find out that it does have some drawbacks.

In the Neumann model, CPUs, memory, I/O, external storage, and network interface are all connected to one single system bus which includes control bus, data bus, and address bus. Once intruders break into the system from any network location, they can totally take over the computer system and do whatever they want.

For the Neumann model, the concept of CPU is a centralized control and arithmetic unit. Even though nowadays a computer with multiprocessors is very common, those processors are merely coordinated together by software to perform one task or a series of tasks. In other words, they share the same system bus. Intruders can take over the whole system once they break into the system from any network port.

10.3 A Dual-Bus Solution

The main idea for this invention was to propose a new computer architecture that enables computers to prevent intruders from getting data stored in the computer system. Based on the modified Neumann model, the network communication component is separated from the other parts of a computer system with a separate system bus. All components in the computer system (except network) are run on another system bus. Data exchange between those two system buses can only be performed through the Bus Controller via a command issued by the computer operator. So, data stored on this computer (main storage) can only be accessed by the computer operator. In other words, user data is isolated from outside networks and therefore cannot be accessed even when the computer is compromised or taken over from outside the network.

A computer platform constructed in accordance with the principles of the present invention is an intrusion-free, information and data secure computer system. It comprises:

1. Two **zones** (red zone and green zone) with two separated system buses.
2. The network interface is only attached on one bus in red zone.
3. Each bus has its own CPU and private memory.
4. Main (protected) external storage is attached only on one bus in green zone.

5. One cache storage (temporary external storage or **dual-port** external storage) is connected to both internal system buses via a Bus Controller.
6. A **Bus Controller** connects two internal system buses between the red zone and green zone.
7. Input and output devices such as keyboard, mouse, and display and so on.

In Figure 10.3, a network interface is added to the Neumann model. Even though a network interface can be considered as an input/output device, adding this interface to the system bus and separating it from other parts (even the general I/O port) has many advantages. The modification made it possible for this invention to isolate a network from other parts within a computer system while data can still be transmitted through the network.

Figure 10.4 depicts a functional block diagram of such intrusion-free, information and data secure computer system architecture (Wang, 2005). Normally the computer is in the state of green zone where all computation works are performed. In the green zone, the network is disabled. When data transmission is needed, the Bus Controller switches to red zone where another CPU is taken over the job. In the red zone, there is no external storage, all data is stored on cache storage via the Bus Controller. The Bus Controller is managed by the computer operator or delegates (programs) assigned by the computer operator.

Looking from the network side (outside), this intrusion-free, information and data secure computer has one or more CPUs, internal memory, input/output devices such as a keyboard and a mouse, network ports (Ethernet or wireless) and cache storage. Because the red zone only deals with the network communication, suppose a hacker breaks into the system from the Internet, what the hacker will see is just the temporary data on the cache storage and maybe some of the system data. It is impossible for the intruder to see data on the main (protected) storage.

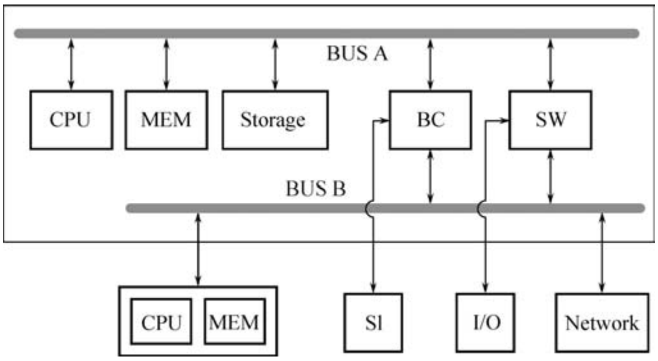


Figure 10.4 Block diagram of intrusion-free computer architecture. User data are stored on the main storage which will never expose to the network

10.4 Bus Controller

Figure 10.5 is the block diagram of the Bus Controller. Bus A in the green zone can access the cache storage only if the EN 1 signal is enabled. Similarly, the Bus B from the red zone can access the cache storage only if the EN 2 signal is enabled. Notice that EN 1 and EN 2 are controlled by the computer operator. Intruders cannot make any enable actions without directly operating the computer.

10.4.1 Working Mechanism of the Bus Controller

Computer operators can automatically enable the data access to the cache storage. To automatic enable the data access to the cache storage, an operator sets the default to Bus A (green zone) so that data can be accessed directly from/to the cache storage. When network communication is needed such as launching an Internet explorer, the EN 2 is automatically enabled so that Bus B is connected and Bus A is disconnected from the system so that main storage is isolated from the system.

A **multiport interface** is used to switch the keyboard/mouse and display devices between those two buses automatically. For automatic switching, the switching process is synchronized with the Bus Controller.

Combining the cache storage or temporary external storage with the Bus Controller forms the dual-port storage which can be accessed by two computer system buses. It is different from so-called dual-port external storage devices which for example have one USB port and one FireWare port. In that case you can only attach one port at a time. Attaching two ports simultaneously would damage the system.

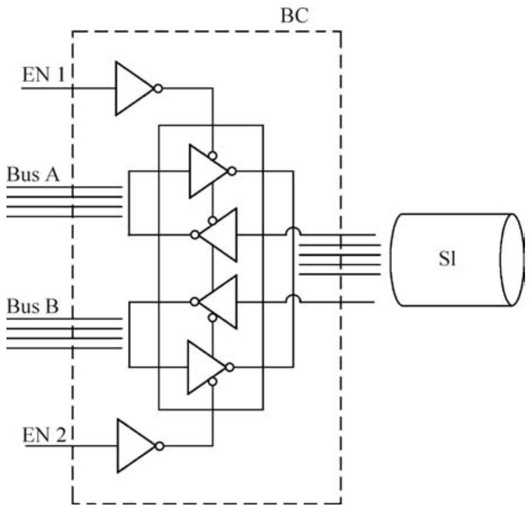


Figure 10.5 Block diagram of Bus Controller that connects two buses and a cache (dual-port) storage device for data exchange between the red zone and green zone

When the cache storage is attached onto Bus A in the green zone, the files are displayed and then the trusted files are ready to be copied to the main storage. After the operation, the cache storage is erased (Wang and Ledley, 2006). User data can then be copied to the cache storage if network transmission is further required. When the cache storage is switched to the Bus B in the red zone, the data is displayed and is ready to be transmitted. Data download from the network or Internet can then be stored on the cache storage. All data have to pass through the Bus Controller which is controlled by the computer operator.

10.4.2 Co-processor Board

The coprocessor board contains a coprocessor, an FPGA, flash memory, multiport memory interface, multiport I/O interface, a Bus Controller and kernel program that enable the add-on board. The kernel program also coordinates the communication between the add-on board and the current computer system.

FPGA solutions from Lattice deliver unique features, high performance, and excellent value for FPGA designs. LatticeXP FPGA devices utilize a combination of non-volatile FLASH cells and SRAM technology to deliver a single-chip solution supporting "instant-on" start-up and infinite re-configurability. A non-volatile FLASH cell array distributed within the LatticeXP FPGA device stores the device configuration. At power-up the configuration is transferred from FLASH memory to configuration SRAM in less than 1mS providing an instant-on FPGA. In addition, LatticeXP FPGA devices provide security by eliminating the need for an external configuration bit-stream and by providing non-volatile security features. Non-volatile, reprogrammable FPGAs are well suited for implementing system logic for this project.

The LatticeXP architecture contains an array of logic blocks surrounded by programmable I/O Cells (PIC). Interspersed between the rows of logic blocks are rows of sysMEM bedded Block RAM (EBR).

On the left and right sides of the PFU array, there are Non-volatile Memory Blocks. In configuration mode this non-volatile memory is programmed via the IEEE 1149.1 TAP port or the sysCONFIG^{TM} peripheral port. On power up, the configuration data is transferred from the Non-volatile Memory Blocks to the configuration SRAM. With this technology, expensive external configuration memories are not required and designs are secured from unauthorized read-back. This transfer of data from non-volatile memory to configuration SRAM via wide buses happens in microseconds, providing an "instant-on" capability that allows easy interfacing in many applications.

There are two kinds of logic blocks, the Programmable Functional Unit (PFU) and Programmable Functional unit without RAM/ROM (PFF). The PFU contains the building blocks for logic, arithmetic, RAM, ROM and register functions. The PFF block contains building blocks for logic, arithmetic and

ROM functions. Both PFU and PFF blocks are optimized for flexibility, allowing complex designs to be implemented quickly and efficiently. Logic blocks are arranged in a two-dimensional array. Only one type of block is used per row. The PFU blocks are used on the outside rows. The rest of the core consists of rows of PFF blocks interspersed with rows of PFU blocks. For every three rows of PFF blocks there is a row of PFU blocks.

Each PIC block encompasses two PIOs (PIO pairs) with their respective sysIO interfaces. PIO pairs on the left and right edges of the device can be configured as LVDS transmit/receive pairs. sysMEM EBRs are large dedicated fast memory blocks. They can be configured as RAM or ROM. The PFU, PFF, PIC, and EBR blocks are arranged in a two-dimensional grid with rows and columns. The blocks are connected with many vertical and horizontal routing channel resources. The place and route software tool automatically allocates these routing resources.

At the end of the rows containing the sysMEM Blocks are the sysCLOCK Phase Locked Loop (PLL) blocks. These PLLs have multiply, divide and phase shifting capability; they are used to manage the phase relationship of the clocks. The LatticeXP architecture provides up to four PLLs per device.

Every device in the family has a JTAG Port with internal Logic Analyzer (ispTRACY) capability. The sysCONFIG port allows for serial or parallel device configuration. The LatticeXP devices are available for operation from 3.3 V, 2.5 V, 1.8 V and 1.2 V power supplies, providing easy integration into the overall system.

- *PFU and PFF blocks.* The core of the LatticeXP devices consists of PFU and PFF blocks. The PFUs can be programmed to perform Logic, Arithmetic, Distributed RAM and Distributed ROM functions. PFF blocks can be programmed to perform Logic, Arithmetic and ROM functions. Except where necessary, the remainder of the data sheet will use the term PFU to refer to both PFU and PFF blocks.

  Each PFU block consists of four interconnected slices, numbered 0–3 as shown in Figure 10.6. All the interconnections to and from PFU blocks are from routing. There are 53 inputs and 25 outputs associated with each PFU block.
- Slice. Each slice contains two LUT4 lookup tables feeding two registers (programmed to be in FF or Latch mode), and some associated logic that allows the LUTs to be combined to perform functions such as LUT5, LUT6, LUT7 and LUT8. There is control logic to perform set/reset functions (programmable as synchronous/asynchronous), clock select, chip-select and wider RAM/ROM functions. Figure 10.7 shows an overview of the internal logic of the slice. The registers in the slice can be configured for positive/negative and edge/level clocks.

  There are 14 input signals: 13 signals from routing and one from the carry-chain (from adjacent slice or PFU). There are seven outputs: six to routing and one to carry-chain (to adjacent PFU).

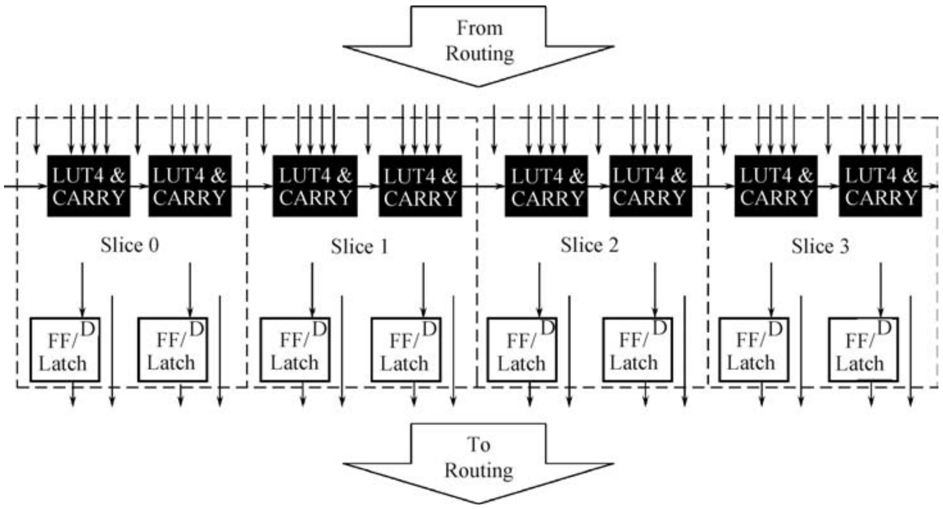


Figure 10.6 PFU Diagram

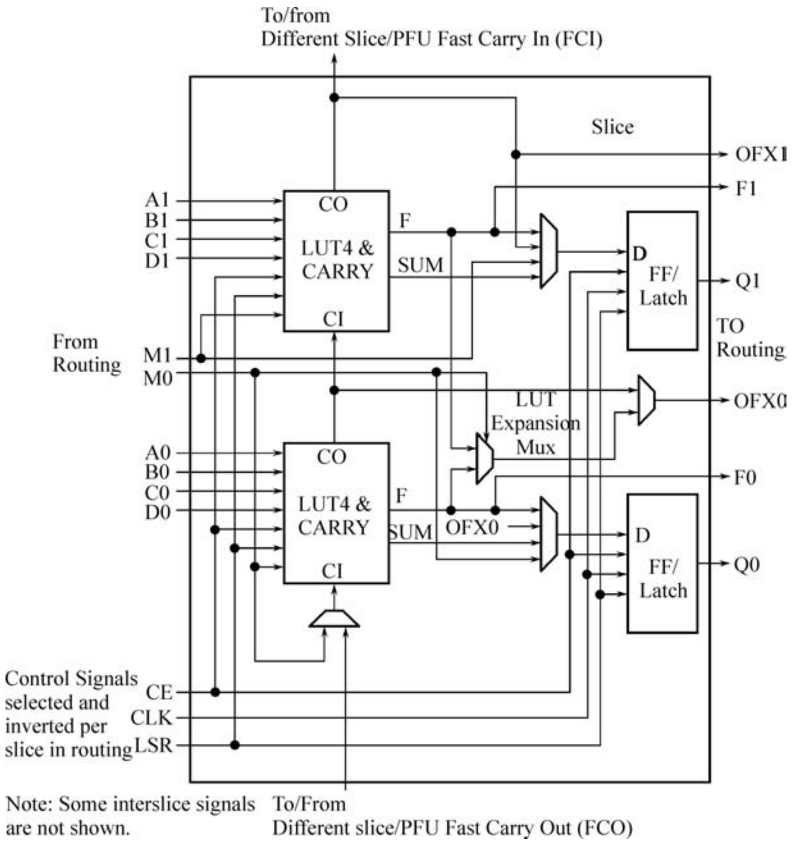

To/from Different Slice/PFU Fast Carry In (FCI)
Slice
OFX1
F1
A1
B1
C1
D1
CO
LUT4 & CARRY
F
SUM
CI
D
FF/ Latch
Q1
TO Routing
From Routing
M1
M0
OFX0
LUT Expansion Mux
A0
B0
C0
D0
F0
OFX0
Q0
Control Signals selected and inverted per slice in routing
CE
CLK
LSR
Note: Some interslice signals are not shown.
To/From Different slice/PFU Fast Carry Out (FCO)

Figure 10.7 Slice Diagram

10.5 Dual-Port Storage

Computer memory and storage are mostly single-port. This means that they can only be attached to one processor. Dual port storage is a type of external memory which can be accessed by two processors simultaneously without worrying about the read/ write conflicts.

Motorola's MPC8260 is a chip that contains a 64-bit PowerPC microprocessor and a versatile communications processor module (CPM). The MPC8260 is used in a wide array of applications, especially those in the communications and networking markets. Examples include remote access servers, regional office routers, cellular base stations, and SONET transmission controllers.

A Lattice's ispGDX2^{TM} Generic Digital Crosspoint Switch is used as a multiport interface. The ispGDX2 device can interface the MPC8260 with an external master and a number of slaves including SDRAM and FLASH. The control logic for the SDRAM and FLASH is built in a CPLD which is used to interface the MPC8260 to the ispGDX2 device and to control the read/write to the memory. This function can be implemented in Lattice CPLDs.

The PowerPC core of the 8260 (the PowerPC 603e) can be replaced by other processors or ASIC. The memory controller within the MPC8260 is utilized in this design. Figure 10.8 shows the diagram using MPC 8260 with the multiport interface.

Figure 10.9 shows in detail the function, internal logic, and cross-connections that the ispGDX2 performs in the design. This section includes the signal list and descriptions of all signals used in this design and also provides a functional description of the design.

10.6 Micro-Operating System

Software design involves controlling and monitoring the communication between two zones. A micro-operating system is designed. It runs on a microprocessor which

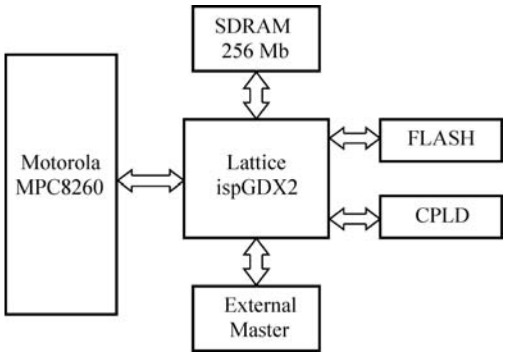


Figure 10.8 Diagram of ispGDX2 Multiport Interface

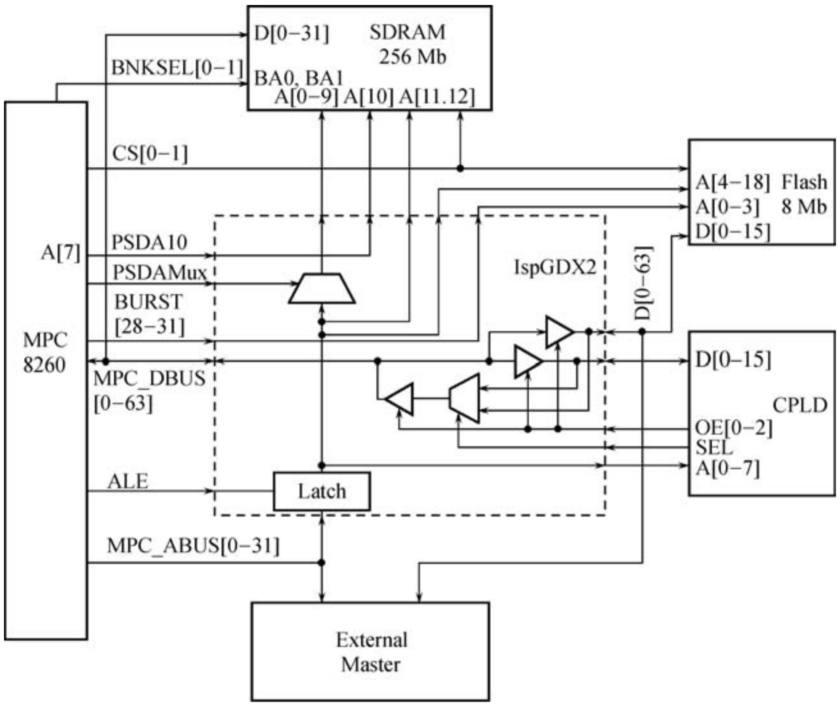


Figure 10.9 Detailed Functional Block Diagram

resides in the “red” zone. The **micro-OS** uses a firmware image that automatically loads to the microprocessor memory during the booting process. A **watch-dog** monitors the system and resets the system whenever a preset threshold is met. In addition, memory protection and virtualization techniques are used to enhance runtime security and prevent code injection.

10.7 Summary

A **prototype** computer based on the newly proposed secure computer architecture has been built (Figure 10.10). Preliminary tests show that it meets the design goals. The Windows firewall was intentionally removed and all security updates were declined in order to test the data security on this computer system. Several intrusion tools have been used in the tests including Key Loggers, Spyware, Spyware cookies, Trojans, Worms and Virus and so on. The system has also undergone a series of tests by senior security professionals and “white hat” attackers. So far, no data bleaches have been found. The initial experiments prove that the proposed secure computer architecture model can enhance computer security. And hopefully it can be adapted to modern personal computers.

Figure 10.10 Prototype of a computer system based on the modified Neumann architecture

Further research may result in the possibility of extending the scope of this architecture from personal computers to server systems. In order to make the system to be able to be widely used in the market, further software development and hardware improvements are needed to make the system not only safe from intruders, but also secure and can be monitored for any security related incidents.

Exercises

10.1 What are the similarities and differences between the study of computer security, information security, data security, and network security?

10.2 What are the ten domains of information system security defined by CISSP?

10.3 An attacker breaks into an information system by modifying data through the Web application. What type of attack is it?

10.4 Modern computers have many buses such as ISA, EISA, PCI, AGP, USB, SATA, SCSI, and so on. Why do we still consider these computers to be single-bus computers?

10.5 Many people consider network interface to be essentially an I/O device. Explain why separating the network interface from the general I/O devices is better when studying computer security?

10.6 The dual-bus computer discussed in this chapter use two (main) buses. What will happen if those two buses want to access the CPU or memory at the same time? How can this be prevented from happening?

10.7 A dual-port memory is defined such that two processors can be attached to the memory together. A control bit is used to enable one processor to access the memory while disabling the other. Draw a diagram of such an implementation.

10.8 Projects

Here are some projects on which practice your architecture and hardware design skills. Hints and help can be found on www.wiley.com/go/wang/comp_arch.

10-1 Computer memory is composed of billions of single memory unit. This single memory unit can only represent 0 or 1. Design and build a circuit of 1-bit memory.

10-2 Design and implement an Easy-Pass system with RFID.

10-3 Design and implement a digital medical identification card with a smart card.

References

Anderson and Stephen, M. (2000) Secure computer architecture. U.S. patent: US 6,115,819. USPTO.

Bishop, M. (2002) *Computer Security: Art and Science*, Addison-Wesley.

Dandamudi, S.P. (2003) *Fundamentals of Computer Organization and Design*, Springer, New York.

Daswani, N., Kern, C., and Kesavan, A. (2007) *Foundations of Security: What Every Programmer Needs to Know*, Apress.

Dumas, J. D. (2006) *Computer Architecture: Fundamentals and Principles of Computer Design*, Taylor & Francis.

Foster, C.C. and Iberall, T. (1985) *Computer Architecture*, 3rd edn, Wan Nostrand Reinhold Company, New York.

Gregg, M. (2009) *CISSP Exam Cram*, 2nd edn, QUE.

Hennessy, J.L. and Patterson, D.A. (2006) *Computer Architecture: A Quantitative Approach*, 4th edn, Morgan Kaufmann, Elsevier.

HP-Compaq Sets Platform Security (2002) eWeek, June 3, 2002.

Hwang, K. (1993) *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, Inc.

(ISC)^2 (2010) *Fundamentals of Information Systems Security*, Jones Y Bartlett Learning, LLC.

Largman, K., More, A.B., and Blair, J. (2004) Computer system architecture and method providing operating-system independent virus-, hacker-, and cyber-terror-immune processing environments. U. S. patent: US 2004-0236874. USPTO.

Merkow, M. and Breithaupt, J. (2005) *Information Security: Principles and Practices*, Pearson Education, Inc., New Jersey.

Randell, B. (ed.) (1982) *The Designs of Digital Computers*, Springer-Verlag.

Shiva, S. G. (2000) *Computer Design and Architecture*, 3rd edn, Marcel Dekker, Inc., New York.

Smith, S.W. and Weingart, S. (1999) Building a high-performance, programmable secure coprocessor. *Computer Networks*, **31**, 831–860.

Suh, G. E., O'Donnell, C.W., Sachdev, I., and Devadas, S. (2005) Design and implementation of the AEGIS single-chip secure processor using physical random functions. Proceedings of 32nd International Symposium on Computer Architecture, ISCA '05 pp. 25–36.

von Neumann, J. (1945) *First Draft of a Report on the EDVAC. Moore School of Electrical Engineering*, University of Pennsylvania.

Wang, S. (2005) Intrusion-free Secure Computer Architecture for Information and Data Security. U.S. patent pending.

Wang, S. and Ledley, R.S. (2006) Connputer – A Framework of Intrusion-Free Secure Computer Architecture. WORLDCOMP Intl. Conf. on Security and Management (SAM'06).

Appendix A

Digital Logic Simulators

Logic simulation is the use of a computer program to simulate the operation of a digital circuit. Logic simulation is the primary tool used for verifying the logical correctness of a hardware design. In many cases logic simulation is the first activity performed in the process of taking a hardware design from concept to realization. Modern hardware description languages are both simulatable and synthesizable.

A.1 CEDAR Logic Simulator

CEDAR LS is an interactive digital logic simulator to be used for the teaching of logic design or testing simple digital designs. It features both low-level logic gates as well as high-level components, including registers and a Z80 microprocessor emulator.

Features:

- Drag-and-drop interface
  - Gates are dragged from a categorized palette onto a tabbed circuit canvas
  - Wire connections are made by dragging from one connection point to another interactive wiring system.
- Continuing simulation shows color-coded wires for 5-state logic
  - Wires rubber-band when connected gates are moved
  - Wire segments may be dragged to fit circuit design.
- Circuit time-step control
  - Simulation may be continuous with a user-specified time per step
  - Play/Pause control.
- User-friendly circuit navigation
  - Keyboard scrolling
  - Miniature map display for large circuits
  - Zoom in/out control

– Group select
– Copy/Paste for wires, gates, and selected groups
– Circuits on different tabs may be connected with named bridge gates
– Print and Export to Bitmap functions.

Download: SourceForge (free)

A.2 Logisim

Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller subcircuits, and to draw bundles of wires with a single mouse drag, Logisim can be used (and is used) to design and simulate entire CPUs for educational purposes.

Logisim is used by students at colleges and universities around the world in many types of classes, ranging from a brief unit on logic in general-education computer science surveys, to computer organization courses, to full-semester courses on computer architecture.

Features:

- Open-source (GPL).
- It runs on any machine supporting Java 5 or later; special versions are released for MacOS X and Windows. The cross-platform nature is important for students who have a variety of home/dorm computer systems.
- The drawing interface is based on an intuitive toolbar. Color-coded wires aid in simulating and debugging a circuit.
- The wiring tool draws horizontal and vertical wires, automatically connecting to components and to other wires. It's very easy to draw circuits!
- Completed circuits can be saved into a file, exported to a GIF file, or printed on a printer.
- Circuit layouts can be used as "subcircuits" of other circuits, allowing for hierarchical circuit design.
- Included circuit components are inputs and outputs, gates, multiplexers, arithmetic circuits, flip-flops, and RAM memory.
- The included "combinational analysis" module allows for conversion between circuits, truth tables, and Boolean expressions.

Download: http://ozark.hendrix.edu/~burch/logisim/ (free)

A.3 Digital Logic Simulator v0.4

Digital Logic Simulator simulates electronic circuits and creates, saves, loads, and imports scenes. It can also simulate computer chips that allow the user to build simple or complex computer components such as RAM, Adders, ALU, or anything you can wrap your brain around. The program allows users to build a scene, save it, and import that entire scene as a single chip. It is great for any computer engineering students or enthusiasts.

Features:

- Simulate electronic circuits.
- Build complex components.
- Create, save and import scenes.

Download: http://bradwarestudios.com/downloads/fun/Digital_Logic_Simulator/ (free)

A.4 Logicly

Logicly is a tool to teach logic gates and circuits effectively. It is able to keep students engaged and put them hands-on.

Features:

- Drag, Drop, and Create
  Build and simulate basic logic circuits with just a few mouse clicks. Drag components into the editor. Then draw connections between them. It's easy to zoom, pan, and rotate too.
- Save and Open Circuits
  When you're finished designing your circuit, you can save it to your hard drive. Great for handing in homework assignments and providing a starting point for lab activities.
- Control the Simulation
  Watch the simulator run in real time, or pause it to advance step by step at your own pace. Control clock components and drive signal propagation with a click of your mouse.
- Cross-Platform
  Install Logicly on Windows, Mac OS X, or Linux. Perfect for student personal computers and environments with a variety of different hardware and operating systems.

Download: http://logic.ly/ (fee)

Other logic simulators include ModelSim, Silos, HALOTIS, STEED, SirSim, FORTE, Atanua, Hades, Simlo, LogicCircuit, Deeds, and so on.

Appendix B

Computer Security Tools

Computer security tools are categorized into many categories. The common tools include password crackers, sniffers, vulnerability scanners, Web scanners, wireless, exploration, packet crafters and so on. Below is a short list of software. More security tools can be found on www.wiley.com/go/wang/comp_arch. A security lab is also set up for interested readers who want to practice some hands-on skills.

B.1 Wireshark (Ethereal)

Wireshark (also known as Ethereal) is a fantastic open source multi-platform network protocol analyzer. It allows you to examine data from a live network or from a capture file on disk. You can interactively browse the capture data, delving down into just the level of packet detail you need. Wireshark has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session. It also supports hundreds of protocols and media types. A tcpdump-like console version named tshark is included. One word of caution is that Wireshark has suffered from dozens of remotely exploitable security holes, so stay up-to-date and be wary of running it on untrusted or hostile networks (such as security conferences).

B.2 Metasploit

Metasploit took the security world by storm when it was released in 2004. It is an advanced open-source platform for developing, testing, and using exploit code. The extensible model through which payloads, encoders, no-op generators, and exploits can be integrated has made it possible to use the Metasploit Framework as an outlet for cutting-edge exploitation research. It ships with hundreds of exploits, as you can see in their list of modules. This makes writing your own exploits easier, and it certainly beats scouring the darkest corners of the Internet for illicit shellcode of dubious quality.

Metasploit was completely free, but the project was acquired by Rapid7 in 2009 and it soon sprouted commercial variants. The Framework itself is still free and open source, but they now also offer a free-but-limited Community edition, a more advanced Express edition ($3,000 per year per user), and a full-featured Pro edition ($15,000 per user per year). Other paid exploitation tools to consider are Core Impact (more expensive) and Canvas (less).

The Metasploit Framework now includes an official Java-based GUI and also Raphael Mudge's excellent Armitage. The Community, Express, and Pro editions have Web-based GUIs.

B.3 Nessus

Nessus is one of the most popular and capable vulnerability scanners, particularly for UNIX systems. It was initially free and open source, but they closed the source code in 2005 and removed the free "Registered Feed" version in 2008. It now costs $1,200 per year, which still beats many of its competitors. A free "Home Feed" is also available, though it is limited and only licensed for home network use.

Nessus is constantly updated, with more than 46,000 plugins. Key features include remote and local (authenticated) security checks, a client/server architecture with a Web-based interface, and an embedded scripting language for writing your own plugins or understanding the existing ones. The open-source version of Nessus was forked by a group of users who still develop it under the OpenVAS name.

B.4 Aircrack

Aircrack is a suite of tools for 802.11a/b/g WEP and WPA cracking. It implements the best known cracking algorithms to recover wireless keys once enough encrypted packets have been gathered. The suite comprises over a dozen discrete tools, including airodump (an 802.11 packet capture program), aireplay (an 802.11 packet injection program), aircrack (static WEP and WPA-PSK cracking), and airdecap (decrypts WEP/WPA capture files).

B.5 Snort

This network intrusion detection and prevention system excels at traffic analysis and packet logging on IP networks. Through protocol analysis, content searching, and various pre-processors, Snort detects thousands of worms, vulnerability exploit attempts, port scans, and other suspicious behavior. Snort uses a flexible rule-based language to describe traffic that it should collect or pass, and a modular detection engine. Also check out the free Basic Analysis and Security Engine (BASE), a Web interface for analyzing Snort alerts.

While Snort itself is free and open source, parent company SourceFire offers their VRT-certified rules for $499 per sensor per year and a complementary product line of software and appliances with more enterprise-level features. Sourcefire also offers a free 30-day delayed feed.

B.6 Cain and Abel

UNIX users often smugly assert that the best free security tools support their platform first, and Windows ports are often an afterthought. They are usually right, but Cain & Abel is a glaring exception. This Windows-only password recovery tool handles an enormous variety of tasks. It can recover passwords by sniffing the network, cracking encrypted passwords using dictionary, brute-force and cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, revealing password boxes, uncovering cached passwords and analyzing routing protocols. It is also well documented.

B.7 BackTrack

This excellent bootable live CD Linux distribution comes from the merger of Whax and Auditor. It boasts a huge variety of Security and Forensics tools and provides a rich development environment. User modularity is emphasized so the distribution can be easily customized by the user to include personal scripts, additional tools, customized kernels, and so on.

B.8 Netcat

This simple utility reads and writes data across TCP or UDP network connections. It is designed to be a reliable back-end tool to use directly or easily drive by other programs and scripts. At the same time, it is a feature-rich network debugging and exploration tool, since it can create almost any kind of connection you would need, including port binding to accept incoming connections.

The original Netcat was released by Hobbit in 1995, but it hasn't been maintained despite its popularity. It can sometimes even be hard to find a copy of the v1.10 source code. The flexibility and usefulness of this tool prompted the Nmap Project to produce Ncat, a modern reimplementation which supports SSL, IPv6, SOCKS and http proxies, connection brokering, and more. Other takes on this classic tool include the amazingly versatile Socat, OpenBSD's nc, Cryptcat, Netcat6, pnetcat, SBD, and so-called GNU Netcat.

B.9 Tcpdump

Tcpdump is the network sniffer we all used before (Wireshark) came on the scene, and many of us continue to use it frequently. It may not have the bells and whistles (such as a pretty GUI and parsing logic for hundreds of application protocols) that

Wireshark has, but it does the job well and with less security risk. It also requires fewer system resources. While Tcpdump doesn't receive new features often, it is actively maintained to fix bugs and portability problems. It is great for tracking down network problems or monitoring activity. There is a separate Windows port named WinDump. tcpdump is the source of the Libpcap/WinPcap packet capture library, which is used by Nmap and many other tools.

B.10 John the Ripper

John the Ripper is a fast password cracker for UNIX/Linux and Mac OS X. Its primary purpose is to detect weak Unix passwords, though it supports hashes for many other platforms as well. There is an official free version, a community-enhanced version (with many contributed patches but not as much quality assurance), and an inexpensive pro version.

Appendix C

Patent Application: Intrusion-Free Computer Architecture for Information and Data Security

CROSS-REFERENCE TO RELATED APPLICATIONS: None
FEDERALLY SPONSORED RESEARCH: None
SEQUENCE LISTING: None

C.1 Background of the Invention

Computers nowadays are very easy to be intruded via a network especially through the Internet. Therefore, information stored on a computer such as ssn, credit cards, bank accounts, and personal privacy information and so on, is vulnerable to computer hackers.

Firewalls to some extent can prevent information stored on a computer from being stolen. However it can only be effective in a certain period of time. Because some firewalls are just software and some others even though they use "hardware" to set up a "wall" between the computer and the outside world, the core components are based on algorithms or in other words software. On the other hand, a firewall is not designed to be used on personal computers or handheld devices. So it cannot guarantee that the information stored on a computer will never be stolen.

Privacy is one of the biggest concerns nowadays. Some employers use centralized monitoring software to monitor employee's emails and other private information.

Identity theft is a more serious problem which drew the attention recently bofy the US Congress. Nearly 10 million people were victimized by identity theft last year, according to *Time* magazine, the loss reached $5 billion. In early March 2005, the nation's largest data miner ChoicePoint with 19 billion data files including driver's license, ssn, credit history, birth certificate, real estate deed, and even thumbprint and DNA was broken into and some 145,000 people's data was extracted. In Senator Charles Schumer's words, "Our system of protecting people's identity is virtually nonexistent in this country." His staff was able to download personal information on the likes of Dick Cheney and Brad Pitt from a ChoicePoint rival, Westlaw. The damage caused by intrusions goes far beyond money and privacy, it also becomes a real threat to national security.

There are many previous researches related to this area. Largman et al. [US 2004-0236874] proposed "automatically create multiple sequentially or concurrently and intermittently isolated and/or restricted computing environments method to prevent viruses, malicious, or even computer or device corruption and failure." According to this method, untrusted content is only exposed in the user processor logic environment in a temporary storage. The question remains how to determine which content is trusted and which is not. There might need to be a pre-determination process. Another problem is "concurrently computing environment" reduces the processing power dramatically. It is generally not suitable for PCs.

Anderson [6,115,819] put "removable trusted (hardware) gateway devices" between each of the inputs/outputs and the bus to secure the file transmission. As described, the approval of accessing the data is dependent on a so-called "LOCK." Once the lock is stolen, intercepted or hacked, sensitive data is then open to those hackers.

So far, many current inventions may have some impact on reducing the risk of information theft in one way or another. However, those solutions have not solved the information security problems thoroughly due to the limitation of the computer architecture they used. There is a problem which exists in John von Neumann's computer architecture model – the foundation of computer architecture. If this problem is not solved, information stored on a computer will not be secure.

C.1.1 John von Neumann Computer Architecture Model

John von Neumann wrote "First Draft of a Report on the EDVAC" in which he outlined the architecture of a stored-program computer. He proposed a concept that has characterized mainstream computer architecture since 1945. Figure C.1 shows the Neumann model.

A "system bus" representation of Neumann model is shown in Figure C.2. This is just another view of the Neumann model, with the introduction of the concept of Direct Memory Access (DMA).

C.1.2 Modified Neumann Computer Architecture

Since the 1990s, computer networks, especially the Internet, have spread around the world. Computers no longer work as stand alone machines. The feature of information exchange through a network is a vital component in today's

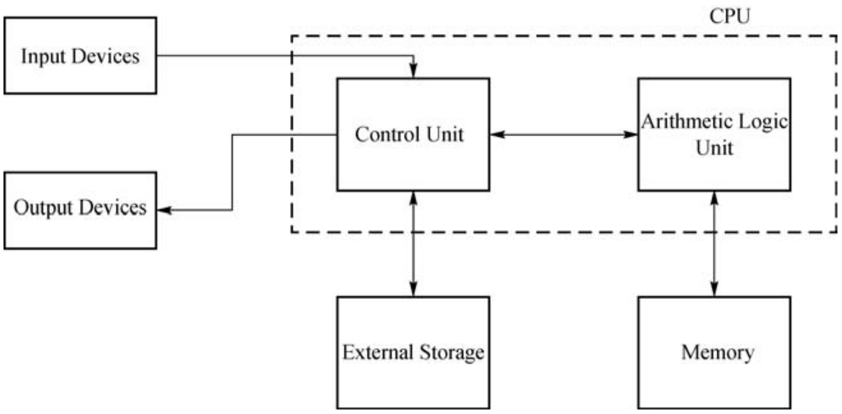


Figure C.1 The block diagram of John von Neumann's computer architecture model.

computers. Unfortunately John Neumann was not able to foreseen this change. One can argue that we can consider a network is part of an input/output device which is already included in the Neumann model. However, the network interface is so important that it is not appropriate to include it in the general I/O device category. Furthermore, an I/O device in the Neumann model refers to those devices such as a keyboard, a display and a printer and so on, which are used for direct interaction with the computers. Now, the way people use a computer is quite different to sixty years ago. So a modification of Neumann's computer architecture model is necessary to reflect this change. Figure C.3 shows the modified Neumann model. In Figure C.3, a network unit (interface) is added to the computer system bus so that the I/O unit only deals with input and output devices such as keyboard, mouse, display, and so on. Separating network unit from the general I/O offers great advantages.

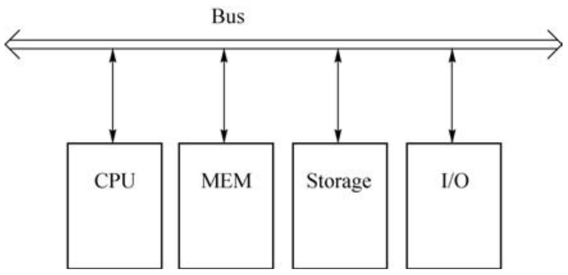


Figure C.2 The block diagram of system bus representation of Neumann's computer architecture model. Figure C.2 is identical to Figure C.1 with the invention and introduction of DMA.

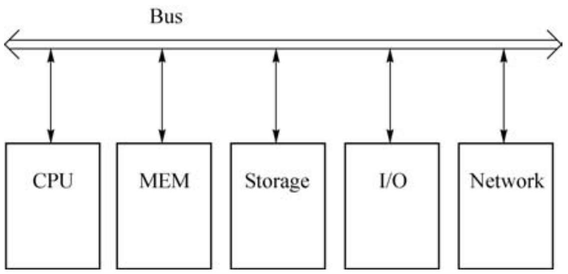


Figure C.3 The block diagram of the modified Neumann computer architecture model. Here network interface is added to the Neumann model and is separated from the general input and output devices.

Newton's three laws and theories of gravity make essentially identical predictions as long as the strength of the gravitational field is weak, which is our usual experience. No one would dare doubt the theories until Einstein predicted that the direction of light propagation should be changed in a gravitational field. This discovery modified Newton's law and made the modern theory of gravity possible.

C.1.3 Problems Existed in the John Neumann Model

In the Neumann model, CPUs, Memory, I/O, external storage and network interface are all connected to one single system bus which includes control bus, data bus, and address bus. Once intruders break into the system from the network locations, they can take over the computer system and do whatever they want.

For the Neumann model, the concept of CPU is a centralized control and arithmetic unit. Even though nowadays a computer with multiprocessors is very common, those processors are merely coordinated together by software to perform one task or a seres of tasks. In other words, they share the same system bus. Intruders can take over the whole system once they break into it from any network ports.

C.1.4 The Goal of the Invention

The main goal for this invention is to propose a new computer architecture that enables computers to prevent intruders from getting data stored in the computer system. Based on the modified Neumann model, the network communication component is separated from the other parts of a computer system with a separate system bus. All components in the computer system (except network) are run on another system bus. Data exchange between those two system buses can only be performed through the Bus Controller via a command issued by the computer operator. So, data stored on this computer (main storage) can only be accessed by the computer operator. In other words, user data is isolated from outside networks and therefore cannot be accessed even when the computer is hacked or taken over from outside the network.

C.2 Field of Invention

This invention related to intrusion-free computer architecture in which network connection is separated from the normal computation tasks. All computations are performed on the protected CPU and user data is stored on a protected external storage which is isolated from the network. Data exchange between protected storage and un-protected storage are managed by the Bus Controller which can only be controlled by the computer operator. The un-protected storage (cache storage) will be erased after data exchange is accomplished. The main protected storage will never be exposed to the intruders even when the system is broken into from the network.

C.3 Detailed Description of the Invention

A computer platform constructed in accordance with the principles of the present invention is an intrusion-free, information and data secure computer system. It comprises

1. Two zones (red zone and green zone) with two separated system buses.
2. The network interface is only attached on one bus in red zone.
3. Each bus has its own CPU and private memory.
4. Main (protected) external storage is attached only on one bus in green zone.
5. One cache storage (temporary external storage or dual-port external storage) is connected to both internal system buses via a Bus Controller.
6. A Bus Controller connects two internal system buses between the red zone and green zone.
7. Input and output devices such as keyboard, mouse and display and so on.

Figure C.3 shows a block diagram of the modified Neumann computer architecture model. A network interface is added to the Neumann model. Even though a network interface can be considered as an input/output device, adding this interface to the system bus and separate it from other parts (even the I/O port) has many advantages. The modification makes it possible for this invention to isolate the network from other parts within a computer system while data can still be transmitted through the network.

Figure C.4 depicts a functional block diagram of intrusion-free, information and data secure computer system architecture. Normally the computer is in the state of green zone where all computation works are performed. In the green zone, the network is disabled. When data transmission is needed, the Bus Controller switches to red zone where another CPU takes over the job. In the red zone, there is no external storage, all data is stored on cache storage via the Bus Controller. The Bus Controller is managed by the computer operator.

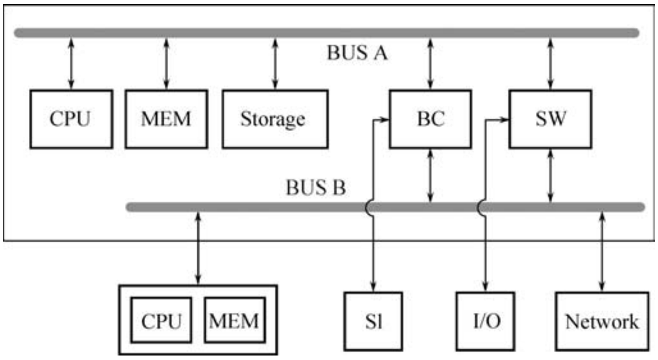


Figure C.4 The block diagram of an intrusion-free computer architecture in accordance with the invention. User data is stored on the main storage which will never be exposed to the network. Data exchange between Bus A and Bus B are controlled by Bus Controller via a command by computer operator only. Information on cache storage will be erased after data exchange.

Looking from the network side (outside), this intrusion-free, information and data secure computer has one or more CPUs, internal memory, input/output devices such as a keyboard and a mouse, network ports (Ethernet or wireless) and cache storage. Because the red zone only deals with the network communication, if a hacker breaks into the system from the Internet, what the hacker will see is just the temporary data on the cache storage and maybe the system data on a hard drive in the red zone. It is impossible for the intruder to see data on the main (protected) storage.

Figure C.5 is the block diagram of the Bus Controller. Bus A in the green zone can access the cache storage only if the EN 1 signal is enabled. Similarly, the Bus B from the red zone can access the cache storage only if the EN 2 signal is enabled. Notice that EN 1 and EN 2 are controlled by the computer operator. Intruders cannot make any enabled actions without direct operation with the computer.

Computer operators can either manually or automatically enable the data access to the cache storage. To automatically enable the data access to the cache storage, an operator sets the default to Bus A (green zone) so that data can be accessed directly from/to the cache storage. When network communication is needed such as launching Internet Explorer, the EN 2 is automatically enabled so that Bus B is connected and Bus A is disconnected from the system so that main storage is isolated from the system.

A switch is used to switch the keyboard/mouse and display devices between those two buses either automatically or manually. For automatic switching, the switching process is synchronized with the Bus Controller.

Combining the cache storage or temporary external storage with the Bus Controller forms the dual-port storage which can be accessed by two computer system

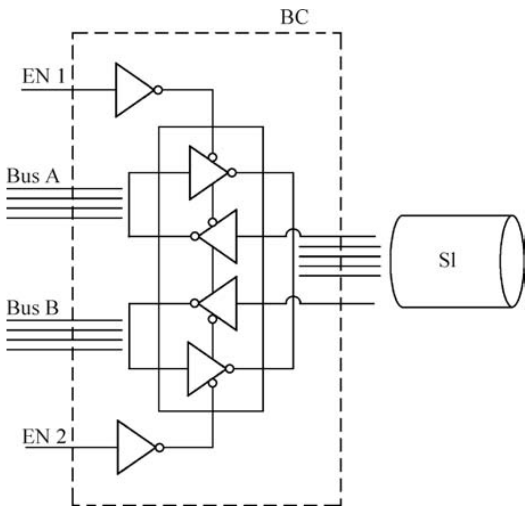


Figure C.5 The block diagram of Bus Controller that connects two buses and a cache (dual-port) storage device for data exchange between the red zone and green zone.

buses. It is different from so-called dual-port external storage devices, which, for example, have one USB port and one FireWare port. In this case you cannot just attach the device to two system buses without synchronizing them.

When the cache storage is attached onto Bus A in green zone, the files are displayed and then the trusted files are ready to be copied to the main storage. After the operation, the cache storage is formatted. User data can then be copied to the cache storage if network transmission is further required. When the cache storage is switched to the Bus B in red zone, the data is displayed and is ready to be transmitted. Data download from a network or the Internet can then be stored on the cache storage. All data have to pass through the Bus Controller which is controlled by the computer operator.

C.4 Claim

What is claimed is:

1. A computer platform constructed in accordance with the principles of the present invention is an intrusion-free computer system. It comprises
   (a) Two zones (red zone and green zone) with two separated system buses.
   (b) The network interface is attached only on one bus in the red zone.
   (c) Each system bus has its own CPU and its own memory.
   (d) Main (protected) external storage is attached only on one bus in the green zone.

   (e) One cache storage (temporary external storage or dual-port external storage) is connected to both internal buses via a Bus Controller.
   (f) A Bus Controller connects two internal system buses between the red zone and green zone.
   (g) Input and output devices.
2. The computer platform of claim1 wherein at least two system buses each have their own CPU(s) and memory.
   (a) All components in the green zone are connected together with Bus A to form a fully feathered computer system which has its own operating system and application software.
   (b) In the green zone, one or more main external storages is attached that only can be accessed by the computer operator.
   (c) In the red zone, system Bus B is used to connect another set of components to form the second computation environment.
   (d) All tasks performed in the red zone are only limited to network transmission or Internet access. There can be a stand alone browser, browsers or network protocols running on an operating system.
3. Data exchange between red zone and green zone is conducted by controlling the Bus Control unit.
   (a) Storage means a hard drive, a USB flash drive, or any media that can store data onto it.
   (b) Main storage or protected storage is a storage that is only attached onto Bus A, therefore is only available to the computer operator.
   (c) CPU here means one or more CPUs.
   (d) Cache storage can be a hard drive, a flash drive or any media that can store data.
   (e) A Bus Controller means an IC composed of several three-state gates, a trigger, a digital switch or simply an on/off switch.

Index