NAME: VIVIAN KERUBO MOSOMI

REGISTRATION NO: SCT212-0062/2021

UNIT: COMPUTER ARCHITECTURE

FUNDAMENTALS OF COMPUTER DESIGN

1.1 INTRODUCTION

Factors that led to an increasing fraction of the computer business being based on microprocessors.

1.The ability of the microprocessor to ride the improvements in integrated circuit technology led to a higher rate of improvement and growth in performance
2.Cost advantages of a mass-produced microprocessor.
3.In addition, two significant changes in the computer marketplace made it easier than ever before to be commercially successful with a new architecture. These are:
i) The virtual elimination of assembly language programming reduced the need for object-code compatibility.
ii) Second, the creation of standardized, vendor-independent operating systems, such as UNIX and its clone, Linux, lowered the cost and risk of bringing out a new architecture.

These changes made it possible to develop successfully a new set of architectures with simpler instructions called RISC (Reduced Instruction Set Computer)

The RISC-based machines focused the attention of designers on two critical performance techniques:

i)The exploitation of instruction level parallelism (initially through pipelining and later through multiple instruction issue)
ii)The use of caches (initially in simple forms and later using more sophisticated organizations and optimizations).

The RISC-based computers raised the performance bar, forcing prior architectures to keep up or disappear. The combination of architectural and organizational enhancements led to 16 years of sustained growth in performance at an annual rate of over 50%—a rate that is unprecedented in the computer industry. The effect of this dramatic growth rate in the 20th century has been twofold:

i) It has significantly enhanced the capability available to computer users. For many applications, the highest-performance microprocessors of today outperform the supercomputer of less than 10 years ago.
ii) Led to the dominance of microprocessor-based computers across the entire range of the computer design

1.2 CLASSES OF COMPUTERS

Year	Type of Computer	Features	Uses
1960	Large MainFrame	Stored in computer rooms with multiple operators overseeing their support Costed millions of dollars	Business data processing and large-scale scientific computing
1970	Minicomputer	A smaller-sized computer	Applications in scientific laboratories, but rapidly branching out with the popularity of time sharing—multiple users sharing a computer interactively through independent terminals
1970	Supercomputer	High processing power, high storage capacity and can handle multiple users at the same time	High-performance computers for scientific computing
1980	Desktop Computer	Based on Microprocessors, in the form of both personal com puters and workstations	Replaced time-sharing and led to the rise of servers—computers that provided larger-scale services such as reliable, long-term file storage and access, larger memory, and more computing power

i) Personal Mobile Device

Personal mobile device (PMD) - A collection of wireless devices with multimedia user interfaces such as cell phones, tablet computers, and so on.

Key features of PMDs:

i) Responsiveness

ii) Predictability

iii) The need to minimize memory and the need to use energy efficiently. Energy efficiency is driven by both battery power and heat dissipation. The memory can be a substantial portion of the system cost, and it is important to optimize memory size in such cases

ii) Desktop Computing

The first, and probably still the largest market in dollar terms, is desktop comput ing. Desktop computing spans from low-end netbooks that sell for under $300 to high-end, heavily configured workstations that may sell for $2500.

Throughout this range in price and capability, the desktop market tends to be driven to **optimize price-performance.** This combination of performance (measured primarily in terms of compute performance and graphics performance) and price of a system is what matters most to customers in this market, and hence to computer designers.

As a result, the newest, highest-performance microprocessors and cost-reduced microprocessors often appear first in desktop systems

iii) Servers

As the shift to desktop computing occurred, the role of servers grew to provide larger-scale and more reliable file and computing services. The World Wide Web accelerated this trend because of the tremendous growth in the demand and sophistication of Web-based services. Such servers have become the backbone of large-scale enterprise computing, replacing the traditional mainframe.

Key features of Servers:

i) Dependability - Considering the servers running Google, taking orders for Cisco, or running auctions on eBay. Failure of such server systems is far more catastrophic than failure of a single desktop, since these servers must operate seven days a week, 24 hours a day.

ii) Scalability - Server systems often grow in response to an increasing demand for the services they support or an increase in functional requirements. Thus, the ability to scale up the computing capacity, the memory, the storage, and the I/O bandwidth of a server is crucial

iii) Efficient Throughput - That is, the overall performance of the server—in terms of transactions per minute or Web pages served per second.

iv) Clusters/Warehouse-Scale Computers

The growth of Software as a Service (SaaS) for applications like search, social networking, video sharing, multiplayer games, online shopping, and so on has led to the growth of a class of computers called **clusters.**
Clusters - Collections of desktop computers or servers connected by local area networks to act as a single larger computer. Each node runs its own operating system, and nodes communicate using a networking protocol. The largest of the clusters are called warehouse-scale computers (WSCs), in that they are designed so that tens of thousands of servers can act as one

Critical features of WSC

1.Price performance and power.
2. Availability. For example, Amazon.com had $13 billion in sales in the fourth quarter of 2010. As there are about 2200 hours in a quarter, the average revenue per hour was almost $6M. During a peak hour for Christmas shopping, the potential loss would be many times higher
3. Minimize power.Optimizing power is often critical in embedded applications

WSCs emphasize interactive applications, large-scale storage, dependability, and high Internet bandwidth.

v) Embedded Computers

Embedded computers are the fastest growing portion of the computer market. These devices range from everyday machines—most microwaves, most washing machines, most printers, most networking switches, and all cars contain simple embedded microprocessors—to handheld digital devices, such as cell phones and smart cards, to video games and digital set-top boxes

Features of Embedded Computers

1.Have the widest spread of processing power and cost.
They include 8-bit and 16-bit processors that may cost less than a dime, 32-bit microprocessors that execute 100 million instructions per second and cost under $5, and high-end processors for the newest video games or network switches that cost $100 and can execute a billion instructions per second.
2. Minimize memory
3. Minimize power.Optimizing power is often critical in embedded applications

Classes of parallelism and Parallel Architecture

There are basically two kinds of parallelism in applications:

1. Data-Level Parallelism (DLP) arises because there are many data items that can be operated on at the same time.
2. Task-Level Parallelism (TLP) arises because tasks of work are created that can operate independently and largely in parallel.

Computer hardware in turn can exploit these two kinds of application parallelism in four major ways:

1. Instruction-Level Parallelism exploits data-level parallelism at modest levels with compiler help using ideas like pipelining and at medium levels using ideas like speculative execution.
2. Vector Architectures and Graphic Processor Units (GPUs) exploit data-level parallelism by applying a single instruction to a collection of data in parallel.
3. Thread-Level Parallelism exploits either data-level parallelism or task-level parallelism in a tightly coupled hardware model that allows for interaction among parallel threads.
4. Request-Level Parallelism exploits parallelism among largely decoupled tasks specified by the programmer or the operating system

1.3 DEFINING COMPUTER ARCHITECTURE

The computer designer needs to determine what attributes are important for a new computer, then design a computer to maximize performance while staying within cost, power, and availability constraints. This task has many aspects, including instruction set design, functional organization, logic design, and implementation. The implementation may encompass integrated circuit design, packaging, power, and cooling. Optimizing the design requires familiarity with a very wide range of technologies, from compilers and operating systems to logic design and packaging

Instruction Set Architecture(ISA)

ISA - Refers to the actual programmer visible instruction set. The ISA serves as the boundary between the software and hardware.

The seven dimensions of an ISA:

i) Class of ISA - Nearly all ISAs today are classified as general-purpose register architectures, where the operands are either registers or memory locations.

ii) Memory Addressing - Virtually all desktop and server computers, including the 80x86 and MIPS, use byte addressing to access memory operands. Some architectures, like MIPS, require that objects must be aligned.

iii) Addressing modes - In addition to specifying registers and constant operands, addressing modes specify the address of a memory object.

iv) Types and size of operands - —Like most ISAs, MIPS and 80x86 support operand sizes of 8-bit (ASCII character), 16-bit (Unicode character or half word), 32-bit (integer or word), 64-bit (double word or long integer), and IEEE 754 floating point in 32-bit (single precision) and 64-bit (double pre cision)

Instruction Set Architecture(ISA)

v) Operations - The general categories of operations are data transfer, arith metic logical, control (discussed next), and floating point. MIPS is a simple and easy-to-pipeline instruction set architecture, and it is representative of the RISC architectures being used in 2006.

vi) Control Flow Instructions - Virtually all ISAs, including 80x86 and MIPS, support conditional branches, unconditional jumps, procedure calls, and returns. Both use PC-relative addressing, where the branch address is speci fied by an address field that is added to the PC.

vii) Encoding an ISA - There are two basic choices on encoding: fixed length and variable length. All MIPS instructions are 32 bits long, which simplifies instruction decoding.

The Rest of Computer Architecture: Designing the Organization and Hardware to Meet Goals and Functional Requirements

The implementation of a computer has two components:

i) Organization - Includes the high-level aspects of a computer's design, such as the memory system, the memory interconnect, and the design of the internal processor or CPU (central processing unit—where arithmetic, logic, branching, and data transfer are implemented)

ii) Hardware - Refers to the specifics of a computer, including the detailed logic design and the packaging technology of the computer. Often a line of computers contains computers with identical instruction set architectures and nearly identical organizations, but they differ in the detailed hardware implementation.

The word architecture covers all three aspects of computer design—**instruction set architecture, organization, and hardware**. Computer architects must design a computer to meet functional requirements as well as price, power, performance, and availability goals.

1.4 TRENDS IN TECHNOLOGY

If an instruction set architecture is to be successful, it must be designed to survive rapid changes in computer technology. An architect must plan for technology changes that can increase the lifetime of a successful computer. To plan for the evolution of a computer, the designer must be aware of rapid changes in implementation technology

i) Integrated circuit logic technology—Transistor density increases by about 35% per year, quadrupling in somewhat over four years. Increases in die size are less predictable and slower, ranging from 10% to 20% per year.
ii) Semiconductor DRAM (dynamic random-access memory) - Capacity increases by about 40% per year, doubling roughly every two years
iii) Magnetic disk technology - Prior to 1990, density increased by about 30% per year, doubling in three years. It rose to 60% per year thereafter, and increased to 100% per year in 1996. Since 2004, it has dropped back to 30% per year.
iv) Network Technology - Network performance depends both on the perfor mance of switches and on the performance of the transmission system.

These rapidly changing technologies shape the design of a computer that, with speed and technology enhancements, may have a lifetime of three to five years. Key technologies such as DRAM, Flash, and disk change sufficiently that the designer must plan for these changes. Indeed, designers often design for the next technology, knowing that when a product begins shipping in volume that the next technology may be the most cost-effective or may have performance advan tages. Traditionally, cost has decreased at about the rate at which density increases.

Performance Trends: Bandwidth vs Latency

Bandwidth or throughput **-** The total amount of work done in a given time, such as megabytes per second for a disk transfer.
Latency or response time - The time between the start and the completion of an event, such as milliseconds for a disk access

Scaling of Transistor Performance and Wires

1. Feature Size and Transistor Density
Feature size refers to the smallest dimension of a transistor or wire (x or y direction).
Smaller feature sizes mean higher transistor density (quadratic increase).
More transistors allow for advances in computing, like moving from 4-bit to 64-bit processors and enabling multi-core chips, SIMD units, and improved caching.
The feature size has shrunk from 10 microns in 1971 to 32 nanometers in 2011.

2. Transistor Performance and Scaling
Smaller transistors improve performance but introduce complex challenges:

- They shrink horizontally (x, y) and vertically (z).
- Vertical shrinking requires lower operating voltage for proper function and reliability.
- Overall, performance improves linearly with smaller feature sizes.

1.5 Trends in Power and Energy in Integrated Circuits

Today, power is the biggest challenge facing the computer designer for nearly every class of computer. First, power must be brought in and distributed around the chip, and modern microprocessors use hundreds of pins and multiple inter connect layers just for power and ground. Second, power is dissipated as heat and must be removed.

Power and Energy: A Systems Perspective

How should a system architect or a user think about performance, power, and energy? From the viewpoint of a system designer, there are three primary concerns:

i) What is the maximum power a processor ever requires?

Meeting this demand can be important to ensuring correct operation. For example, if a processor attempts to draw more power than a power supply system can provide (by drawing more current than the system can supply), the result is typically a voltage drop, which can cause the device to malfunction. Modern processors can vary widely in power consumption with high peak currents; hence, they provide voltage indexing methods that allow the processor to slow down and regulate voltage within a wider margin. Obviously, doing so decreases performance.

ii) What is the sustained power consumption?

This metric is called the thermal design power (TDP), since it determines the cooling requirement. TDP is neither peak power, which is often 1.5 times higher, nor is it the actual average power that will be consumed during a given computation, which is likely to be lower still. A typical power supply for a system is usually sized to exceed the TDP, and a cooling system is usually designed to match or exceed TDP. Failure to provide adequate cooling will allow the junction temperature in the processor to exceed its maximum value, resulting in device failure and possibly permanent damage

iii) Energy and energy efficiency.

Power is simply energy per unit time: 1 watt = 1 joule per second. Which metric is the right one for comparing processors: energy or power? In general, energy is always a better metric because it is tied to a specific task and the time required for that task. In particular, the energy to execute a workload is equal to the average power times the execution time for the workload.

Thus,to know which of two processors is more efficient for a given task, we should compare energy consumption (not power) for executing the task. For example, processor A may have a 20% higher average power consumption than processor B, but if A executes the task in only 70% of the time needed by B, its energy consumption will be $1.2 \times 0.7 = 0.84$, which is better.

Energy and Power within a Microprocessor

How Microprocessors Consume Power:

Most of the energy in CMOS chips is used to switch transistors.

The energy required per transistor depends on:

- Capacitive load (number of transistors connected to an output).
- Voltage (higher voltage = more power).
- Switching frequency which is how often transitions occur).
- Formula :
  - Energy per switch $\propto$ ½ × Capacitive Load × Voltage²
  - Power per transistor $\propto$ ½ × Capacitive Load × Voltage² × Frequency

Power-Saving Techniques in Modern Microprocessors

Modern microprocessors conserve energy by disabling inactive components when they are not in use. This technique is known as “Do Nothing Well" and it helps reduce unnecessary power consumption.

- Turning off unused parts: If a component is not actively needed, the processor disables its clock signal, preventing it from consuming power.
- Example – Floating-Point Unit (FPU): If a program is not performing floating-point calculations, the processor automatically shuts down the floating-point unit to save energy.
- Idle cores save power: In multi-core processors, if some cores are not required for processing, their clocks are stopped or slowed down, significantly reducing power consumption and heat generation.

1.6 Trends in Cost

Although costs tend to be less important in some computer designs—specifically supercomputers—cost-sensitive designs are of growing significance. Indeed, in the past 30 years, the use of technology improvements to lower cost, as well as increase performance, has been a major theme in the computer industry

The Impact of Time, Volume, and Commoditization

- The cost of a manufactured computer component decreases over time even with out major improvements in the basic implementation technology. The underlying principle that drives costs down is the learning curve—manufacturing costs decrease over time. The learning curve itself is best measured by change in yield—the percentage of manufactured devices that survives the testing procedure. Whether it is a chip, a board, or a system, designs that have twice the yield will have half the cost.

- Understanding how the learning curve improves yield is critical to projecting costs over a product's life. One example is that the price per megabyte of DRAM has dropped over the long term. Since DRAMs tend to be priced in close relation ship to cost—with the exception of periods when there is a shortage or an oversupply—price and cost of DRAM track closely. Microprocessor prices also drop over time, but, because they are less standardized than DRAMs, the relationship between price and cost is more complex.

Volume is a second key factor in determining cost. Increasing volumes affect cost in several ways including:

i) Decreasing the time needed to get down the learning curve, which is partly proportional to the number of systems (or chips) manufactured.

ii) Volume decreases cost, since it increases purchasing and manufacturing efficiency.

As a rule of thumb, some designers have estimated that cost decreases about 10% for each doubling of volume. Moreover, volume decreases the amount of development cost that must be amortized by each computer, thus allowing cost and selling price to be closer

- Commodities are products that are sold by multiple vendors in large volumes and are essentially identical. Virtually all the products sold on the shelves of grocery stores are commodities, as are standard DRAMs, Flash memory, disks, monitors, and keyboards. Because many vendors ship virtually identical products, the market is highly competitive. Thus, this competition decreases the gap between cost and selling price, but it also **decreases cost**.

- Reductions occur because a commodity market has both volume and a clear product definition, which allows multiple suppliers to compete in building components for the commodity product. As a result, the overall product cost is lower because of the competition among the suppliers of the components and the volume efficiencies the suppliers can achieve. This rivalry has led to the low end of the computer business being able to achieve better price-performance than other sectors and yielded greater growth at the low end, although with very limited profit

Cost of an Integrated Circuit

Although the costs of integrated circuits have dropped exponentially, the basic process of silicon manufacture is unchanged.Thus the cost of a packaged integrated circuit is:

Cost of integrated circuit = (Cost of die + Cost of testing die + Cost of packaging) / final test Final test yield

The cost of dies, summarizing the key issues in testing and packaging at the end:
Cost of die = (Cost of wafer) / (Dies per wafer) * Die yield

- Given the tremendous price pressures on commodity products such as DRAM and SRAM, designers have included redundancy as a way to raise yield. For a number of years, DRAMs have regularly included some redundant memory cells, so that a certain number of flaws can be accommodated. Designers have used similar techniques in both standard SRAMs and in large SRAM arrays used for caches within microprocessors. Obviously, the presence of redundant entries can be used to boost the yield significantly.

What should a computer designer remember about chip costs?

The manufacturing process dictates the wafer cost, wafer yield, and defects per unit area, so the sole control of the designer is die area. In practice, because the number of defects per unit area is small, the number of good dies per wafer, and hence the cost per die, grows roughly as the square of the die area. The computer designer affects die size, and hence cost, both by what functions are included on or excluded from the die and by the number of I/O pins.

There is, however, one very important part of the fixed costs that can significantly affect the cost of an integrated circuit for low volumes (less than 1 million parts), namely, **the cost of a mask set**. Each step in the integrated circuit process requires a separate mask. Thus, for modern high-density fabrication processes with four to six metal layers, mask costs exceed $1M. This large fixed cost affects the cost of prototyping and debugging runs and, for small-volume production, can be a significant part of the production cost. Since mask costs are likely to continue to increase, designers may incorporate reconfigurable logic to enhance the flexibility of a part or choose to use gate arrays (which have fewer custom mask levels) and thus reduce the cost implications of masks.

Cost Versus Price

With the commoditization of computers, the margin between the cost to manufacture a product and the price the product sells for has been shrinking. Those margins pay for a company's research and development (R&D), marketing, sales, manufacturing equipment maintenance, building rental, cost of financing, pretax profits, and taxes.

Cost of Manufacturing versus Cost of Operation

cost means the cost to build a computer and price means price to purchase a computer. With the advent of warehouse scale computers, which contain tens of thousands of servers, the cost to operate the computers is significant in addition to the cost of purchase.

the amortized purchase price of servers and networks is just over 60% of the monthly cost to operate a warehouse-scale computer, assuming a short lifetime of the IT equipment of 3 to 4 years. About 30% of the monthly operational costs are for power use and the amortized infrastructure to distribute power and to cool the IT equipment, despite this infrastructure being amortized over 10 years. Thus, to lower operational costs in a warehouse-scale computer, computer architects need to use energy efficiently.