notes by abhishek kumar

BSCS - IV

**COMPUTER ARCHITECTURE AND ORGANIZATION**

**CHAPTER 02 COMPUTER EVOLUTION AND PERFORMANCE**

Table of Contents

[2.1 A BRIEF HISTORY OF COMPUTERS 2](#_Toc130988470)

[Von-Neumann Architecture 2](#_Toc130988471)

[IAS Computer 2](#_Toc130988472)

[2.1 Designing for Performance 3](#_Toc130988473)

[Speeding it up – Microprocessor Speed 3](#_Toc130988474)

[Pipelining, branch prediction, superscalar execution, data flow analysis, and speculative execution 4](#_Toc130988475)

[Performance Balance 5](#_Toc130988476)

[Problem with Connection between the Processor and Main Memory 5](#_Toc130988477)

[Solutions: 5](#_Toc130988478)

[Handling of I/O Devices 5](#_Toc130988479)

[Improvements in Chip Organization and Architecture 6](#_Toc130988480)

[Problems with Clock Speed and Logic Density 6](#_Toc130988481)

[Two main strategies have been used 6](#_Toc130988482)

[New Approach – Multiple Cores 8](#_Toc130988483)

[Many Integrated Core (MIC) vs Graphics Processing Unit (GPU) 8](#_Toc130988484)

[2.3 THE EVOLUTION OF THE INTEL x86 ARCHITECTURE 9](#_Toc130988485)

[Introduction 9](#_Toc130988486)

[Evolution of Intel Microprocessor 9](#_Toc130988487)

[2.4 EMBEDDED SYSTEMS AND THE ARM 11](#_Toc130988488)

[Embedded Systems 11](#_Toc130988489)

[Benefits and Drawbacks 11](#_Toc130988490)

[ARM 12](#_Toc130988491)

[ARM Evaluation 12](#_Toc130988492)

[ARM Design Categories 13](#_Toc130988493)

[CICS vs RISC 13](#_Toc130988494)

[2.5 PERFORMANCE ASSESSMENT 15](#_Toc130988495)

[System Clock 15](#_Toc130988496)

[Amdahl’s Law 15](#_Toc130988497)

# 2.1 A BRIEF HISTORY OF COMPUTERS

## Von-Neumann Architecture

Von Neumann architecture is a theoretical design for a stored-program digital computer that was proposed by mathematician and physicist John von Neumann in the late 1940s. This architecture is characterized by a single memory that stores both instructions and data, and a central processing unit (CPU) that fetches and executes instructions one at a time. The von Neumann architecture is still used in modern computers today.

## IAS Computer

The IAS computer was designed by John von Neumann and his colleagues at Princeton Institute for Advanced Studies in the mid-1940s. It was the first computer to use the stored-program concept, which allowed the computer to store both data and instructions in memory, making it easier to enter and alter programs. The IAS computer had a main memory, an arithmetic and logic unit (ALU), a control unit, and input/output (I/O) equipment. The memory consisted of 1000 storage locations, called words, of 40 binary digits each, where data and instructions were stored. Each instruction was a binary code consisting of an 8-bit operation code (opcode) and a 12-bit address. The IAS computer served as the prototype for subsequent general-purpose computers, and the von Neumann architecture is still widely used today.

# 2.1 Designing for Performance

The cost of computer systems has significantly decreased over the years while their performance and capacity have significantly increased. Laptops today are as powerful as IBM mainframes from 10 or 15 years ago. Processors are now so affordable that we even throw away microprocessors. Desktop applications that require a lot of processing power, such as image processing, 3D rendering, and speech recognition, can now be easily run on microprocessor-based systems. Businesses now use powerful servers to handle transaction and database processing and support client/server networks that have replaced the huge mainframe computer centers of the past. Cloud service providers have high-performance servers to support a wide range of clients with high-volume and high-transaction-rate applications.

## Speeding it up – Microprocessor Speed

Computer processors, like the ones made by Intel and IBM, are able to do so many calculations so quickly because the people who make them are always trying to make them faster. They do this by adding more tiny electronic parts to the processor chip, and by making the distances between the parts smaller. This has made processors four or five times faster every three years since 1978.

But a processor can only be as powerful as the work it's given to do. If it's not given enough work, it won't be as fast as it could be. So, the manufacturers of processor chips are increasing their speed, and the designers of processors are simultaneously developing ways to ensure that they constantly receive instructions to perform tasks. They've come up with many different techniques to do this, like predicting what work the processor will need to do next, and doing that work before the processor is even asked to do it.

Pipelining, branch prediction, superscalar execution, data flow analysis, and speculative execution are all techniques used by processor designers to maximize the power and efficiency of processors.

1. **Pipelining:** A processor's ability to move data or instructions through multiple stages of a pipeline for simultaneous processing.
2. **Branch prediction:** Branch prediction is a technique that allows the processor to predict which instructions are likely to be processed next. This helps to keep the pipeline full, so the processor can work at maximum capacity.
3. **Superscalar execution:** Superscalar execution allows processors to issue more than one instruction in every clock cycle by using multiple parallel pipelines. This makes the processor work faster and more efficiently.
4. **Data flow analysis:** Data flow analysis is a technique that allows the processor to analyze which instructions are dependent on each other's results or data.
5. **Speculative execution:** Speculative execution uses both branch prediction and data flow analysis to allow the processor to execute instructions ahead of their actual appearance in the program execution. This technique holds the results in temporary locations, keeping the execution engines as busy as possible.

In summary, branch prediction is used to predict which branch is likely to be processed next, while speculative execution goes a step further by executing instructions speculatively based on that prediction.

## Performance Balance

### Problem with Connection between the Processor and Main Memory

The power of computer processors has increased very quickly, but other parts of the computer haven't kept up. This means that there is a need to balance performance. One important issue is the connection between the processor and main memory. Although processors have become faster, the connection between memory and the processor hasn't kept up. This connection is very important because it carries instructions and data back and forth. If the connection is too slow, the processor has to wait and can't work as quickly.

### Solutions:

Computer designers have come up with different ways to solve the problem of slow data transfer between the processor and main memory.

* They can increase the amount of data retrieved by making DRAMs wider and using wide bus data paths.
* Another way is to change the DRAM interface by adding a cache or buffering scheme on the DRAM chip.
* They can also reduce the frequency of memory access by using complex and efficient cache structures between the processor and main memory.
* Another solution is to increase the bandwidth between processors and memory by using higher-speed buses and a hierarchy of buses to structure data flow.

### Handling of I/O Devices

As computers get faster and can do more things, there are now more advanced devices that require a lot of data to be transferred between them and the computer. While current processors can handle this data but there remains the problem moving data between processor & peripherals. To solve this problem, designers use strategies like caching and buffering, faster buses, and more complex structures of buses. They also sometimes use multiple-processor setups to help handle the high data demands of these devices.

## Improvements in Chip Organization and Architecture

There are three ways to make computer processors faster.

First, you can increase the hardware speed of processor by making the logic gates smaller and packing more of them together. This reduces the time it takes for signals to travel and increases the clock rate, which means operations can be done more quickly.

Second, you can make the caches bigger and faster by dedicating part of the processor chip to them, which makes it faster to access important data.

Third, you can change the organization and architecture of the processor to make it more efficient at executing instructions in parallel, which means it can do more work at the same time.

### Problems with Clock Speed and Logic Density

Increasing clock speed and logic density have been the dominant factors in performance gains, but they also lead to obstacles such as power density (heat), RC delay, and memory latency. Therefore, designers have turned to new approaches, such as placing multiple processors on the same chip with a large shared cache. This strategy increases performance without increasing clock rate and allows for the effective use of multiple processors to double performance. In the future, most new processor chips will likely have multiple processors.

### Two main strategies have been used

To improve computer performance beyond just increasing clock speed, two strategies have been used.

The first is to increase the size of the cache, which is a type of memory that stores frequently used data. By putting more cache on the computer chip, it can access important data faster.

The second strategy is to make the computer smarter by allowing it to do more things at the same time. This involves using pipelines, which work like assembly lines and allow different stages of execution of different instructions to occur at the same time. Additionally, superscalar architecture allows multiple pipelines within a single processor, so instructions that do not depend on one another can be executed in parallel.

## New Approach – Multiple Cores

In order to increase performance, a new approach called "multicore" has been developed. This involves using two or more simpler processors on the same chip instead of one more complex processor. This approach saves the need to increase the clock rate. With multiple processors, larger caches can be used. Additionally, the power consumption of memory logic is less than that of processing logic. An example of this approach is the IBM POWER4, which has two cores based on PowerPC.

### Many Integrated Core (MIC) vs Graphics Processing Unit (GPU)

The MIC (Many Integrated Core) approach is a way to make computer processors faster by putting multiple general-purpose processors on a single chip. This can make the computer run much faster, but it also presents challenges in developing software that can effectively utilize so many cores.

GPU (Graphics Processing Unit): A GPU is a specialized processor that is designed to handle graphics-related tasks, such as rendering 2D and 3D graphics and processing video. GPUs are typically found on plug-in graphics cards and are used in a variety of applications that require repetitive computations, such as gaming and scientific computing.

# 2.3 THE EVOLUTION OF THE INTEL x86 ARCHITECTURE

## Introduction

The text is discussing two types of computer architectures: the Intel x86 and the ARM.

The x86 is an example of a complex instruction set computer (CISC), which has been developed for many years and incorporates design principles from mainframes and supercomputers.

The ARM, on the other hand, is an example of a reduced instruction set computer (RISC) and is commonly used in embedded systems.

Intel is the leading maker of microprocessors for non-embedded systems, and the evolution of its flagship product, the x86, shows how computer technology has advanced over time. Intel used to release new processors every four years, but has recently increased the pace to stay ahead of competitors.

## Evolution of Intel Microprocessor

Intel has a long history of producing innovative microprocessors. Here are some of the highlights of the evolution of Intel's product line:

* 8080: The first general-purpose microprocessor, an 8-bit machine that was used in the first personal computer, the Altair.
* 8086: A much more powerful 16-bit machine that introduced the x86 architecture. It featured an instruction cache, or queue, that prefetches a few instructions before they are executed.
* 80286: An extension of the 8086 that enabled addressing a 16-MByte memory instead of just 1 MByte.
* 80386: Intel's first 32-bit machine that could run multiple programs at the same time, supporting multitasking.
* 80486: This processor introduced much more powerful cache technology and instruction pipelining. It also offered a built-in math coprocessor.
* Pentium: With the Pentium, Intel introduced superscalar techniques, allowing multiple instructions to execute in parallel.
* Pentium Pro: Continued the move into superscalar organization begun with the Pentium, with aggressive use of register renaming, branch prediction, data flow analysis, and speculative execution.
* Pentium II: Incorporated Intel MMX technology, which is designed to process video, audio, and graphics data efficiently.
* Pentium III: Incorporated additional floating-point instructions to support 3D graphics software.
* Pentium 4: Included additional floating-point and multimedia enhancements.
* Core: The first Intel x86 microprocessor with a dual-core, referring to the implementation of two processors on a single chip.
* Core 2: Extended the architecture to 64 bits and provided four processors on a single chip with the Core 2 Quad.

# 2.4 EMBEDDED SYSTEMS AND THE ARM

The ARM architecture refers to a processor architecture that has evolved from RISC design principles and is used in embedded systems

## Embedded Systems

An embedded system is a computer system that's built into a device to perform a specific task. It's different from a regular computer because it's designed for a single purpose, like controlling a car's antilock braking system. Embedded systems are often part of a bigger device and are typically made up of hardware, software and even other parts.

Embedded systems have different needs and limitations, depending on what they are used for. For example, some need to be very cheap, while others need to be very safe and reliable.

They may also need to work in real-time, which means they have to respond very quickly to the environment they are in. They are designed to work closely with their environment and interact with it in specific ways.

## Benefits and Drawbacks

Benefits:

* Embedded systems are designed for specific tasks, making them more efficient and effective.
* They can operate in harsh environments and extreme conditions.
* They consume less power and can be more affordable than general-purpose computers.
* They are often smaller and more compact than traditional computer systems.
* They can be used in a wide range of applications, from automobiles to medical equipment.

Drawbacks:

* Embedded systems can be more difficult to program and require specialized knowledge.
* They are not as flexible as general-purpose computers and cannot perform a wide range of tasks.
* Upgrades and maintenance can be more challenging due to their specialized design.
* They may be more vulnerable to security threats and attacks.
* They may not be as powerful as general-purpose computers, which can limit their performance in certain applications.

## ARM

In the 1980s, a company called Acorn Computers created the very first RISC processor that could be sold to the public. They called it the Acorn RISC Machine (ARM) that was later renamed to Advanced RISC Machine

ARM is a type of microprocessor and microcontroller designed by ARM Inc. in Cambridge, England. The company doesn't actually make processors, but instead designs them and licenses them to manufacturers. ARM chips are known for their high speed and low power requirements, and are used in a wide range of consumer products, including handheld devices like phones and PDAs. They are also used in Apple's iPod and iPhone. ARM is one of the most widely used embedded processor architectures in the world,

## ARM Evaluation

* ARM1: In 1985, the first version of ARM, ARM1, was created and used for internal research and development as well as being used as a coprocessor in the BBC machine.
* ARM2: Also in 1985, Acorn released the ARM2, which had greater functionality and speed within the same physical space.
* ARM3: In 1989, the ARM3 was released and further improvements were achieved.
* ARM6 and Beyond: A new company was organized with Acorn, VLSI, and Apple Computer as founding partners, known as ARM Ltd. The Acorn RISC Machine became the Advanced RISC Machine, and the new company's first offering, an improvement on the ARM3, was designated ARM6. Subsequently, the company has introduced a number of new families with increasing functionality and performance.

## ARM Design Categories

ARM processors are designed to meet the needs of three main system categories.

The first category is embedded real-time systems, which are used in storage, automotive, industrial, and networking applications.

The second category is application platforms, which are devices running open operating systems such as Linux, Palm OS, Symbian OS, and Windows CE. These platforms are used in wireless, consumer entertainment, and digital imaging applications.

The third category is secure applications, which are used in smart cards, SIM cards, and payment terminals.

## CICS vs RISC

CISC:

1. Large and complex instruction set
2. Instructions can take different numbers of clock cycles to execute
3. Manipulation directly in memory
4. Emphasis on hardware-based optimization
5. Higher power consumption and heat generation
6. Good for general-purpose computing
7. Used in desktops, laptops, and servers

RISC:

1. Simple and limited instruction set
2. Instructions take the same number of clock cycles to execute
3. Manipulation in registers
4. Emphasis on software-based optimization
5. Lower power consumption and heat generation
6. Good for specific-purpose computing
7. Used in embedded systems, mobile devices, and game consoles

Overall, CISC processors are better suited for general-purpose computing where a wider range of instructions and hardware optimization can lead to better performance. RISC processors, on the other hand, are more specialized and efficient in specific-purpose computing, making them ideal for embedded systems and mobile devices where power consumption and heat generation are important considerations.

# 2.5 PERFORMANCE ASSESSMENT

When choosing a processor for a new system, performance is an important consideration, along with cost, size, security, reliability, and power consumption. However, comparing the performance of different processors can be difficult, even among processors in the same family.

Performance not only depends on the clock speed but also depends on other factors such as the instruction set, implementation language, compiler efficiency, and programming skill. So, when evaluating processor performance, it's important to consider all of these factors.

## System Clock

Operations performed by a processor are controlled by a system clock.

The clock signal is produced by a quartz crystal, which produces a constant signal wave that are converted into a constant digital flow for the processor.

Each pulse is known as a clock cycle or tick, and the number of clock cycles per second (Hz) is referred to as the clock rate or clock speed or speed of processor. For example, a 1 GHz processor receives one billion pulses per second.

# Amdahl’s Law

Amdahl's law is about how much a program can be speed up when using multiple processors. It assumes that some parts of the program can't be parallelized.

The law says that adding more processors won't always speed up the program linearly. At some point, adding more processors won't make the program faster anymore.