Національний технічний університет України

“Київський політехнічний інститут”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

РЕФЕРАТ

з дисципліни “Англійська мова професійного спрямування”

на тему: “Сentral processing unit”

Виконав: Бута С.О.

Група ІО-12

Київ 2014

Content

[Introduction 3](#_Toc389638140)

[History 4](#_Toc389638141)

[Transistor and integrated circuit CPUs 5](#_Toc389638142)

[Microprocessors 6](#_Toc389638143)

[Operation 6](#_Toc389638144)

[Design and implementation 7](#_Toc389638145)

[Control unit 7](#_Toc389638146)

[Integer range 7](#_Toc389638147)

[Clock rate 8](#_Toc389638148)

[Parallelism 8](#_Toc389638149)

[Instruction level parallelism 9](#_Toc389638150)

[Thread-level parallelism 9](#_Toc389638151)

[Data parallelism 9](#_Toc389638152)

[Performance 9](#_Toc389638153)

[Summary 10](#_Toc389638154)

[Glossary 11](#_Toc389638155)

# Introduction

A central processing unit (CPU) (formerly also referred to as a central processor unit) is the hardware within a computer that carries out the instructions of a computer program by performing the basic arithmetical, logical, and input/output operations of the system. The term has been in use in the computer industry at least since the early 1960s. The form, design, and implementation of CPUs have changed over the course of their history, but their fundamental operation remains much the same.

A computer can have more than one CPU; this is called multiprocessing. All modern CPUs are microprocessors, meaning contained on a single chip. Some integrated circuits (ICs) can contain multiple CPUs on a single chip; those ICs are called multi-core processors. An IC containing a CPU can also contain peripheral devices, and other components of a computer system; this is called a system on a chip (SoC).

Two typical components of a CPU are the arithmetic logic unit (ALU), which performs arithmetic and logical operations, and the control unit (CU), which extracts instructions from memory and decodes and executes them, calling on the ALU when necessary.

Not all computational systems rely on a central processing unit. An array processor or vector processor has multiple parallel computing elements, with no one unit considered the "center". In the distributed computing model, problems are solved by a distributed interconnected set of processors.

# History

Computers such as the ENIAC had to be physically rewired to perform different tasks, which caused these machines to be called "fixed-program computers". Since the term "CPU" is generally defined as a device for software (computer program) execution, the earliest devices that could rightly be called CPUs came with the advent of the stored-program computer.

The idea of a stored-program computer was already present in the design of J. Presper Eckert and John William Mauchly's ENIAC, but was initially omitted so that it could be finished sooner. On June 30, 1945, before ENIAC was made, mathematician John von Neumann distributed the paper entitled First Draft of a Report on the EDVAC. It was the outline of a stored-program computer that would eventually be completed in August 1949. EDVAC was designed to perform a certain number of instructions (or operations) of various types. Significantly, the programs written for EDVAC were to be stored in high-speed computer memory rather than specified by the physical wiring of the computer. This overcame a severe limitation of ENIAC, which was the considerable time and effort required to reconfigure the computer to perform a new task. With von Neumann's design, the program, or software, that EDVAC ran could be changed simply by changing the contents of the memory.

Early CPUs were custom-designed as a part of a larger, sometimes one-of-a-kind, computer. However, this method of designing custom CPUs for a particular application has largely given way to the development of mass-produced processors that are made for many purposes. This standardization began in the era of discrete transistor mainframes and minicomputers and has rapidly accelerated with the popularization of the integrated circuit (IC). The IC has allowed increasingly complex CPUs to be designed and manufactured to tolerances on the order of nanometers. Both the miniaturization and standardization of CPUs have increased the presence of digital devices in modern life far beyond the limited application of dedicated computing machines. Modern microprocessors appear in everything from automobiles to cell phones and children's toys.

While von Neumann is most often credited with the design of the stored-program computer because of his design of EDVAC, others before him, such as Konrad Zuse, had suggested and implemented similar ideas. The so-called Harvard architecture of the Harvard Mark I, which was completed before EDVAC, also utilized a stored-program design using punched paper tape rather than electronic memory. The key difference between the von Neumann and Harvard architectures is that the latter separates the storage and treatment of CPU instructions and data, while the former uses the same memory space for both. Most modern CPUs are primarily von Neumann in design, but CPUs with the Harvard architecture are seen as well, especially in embedded applications; for instance, the Atmel AVR microcontrollers are Harvard architecture processors.

Relays and vacuum tubes (thermionic valves) were commonly used as switching elements; a useful computer requires thousands or tens of thousands of switching devices. The overall speed of a system is dependent on the speed of the switches. Tube computers like EDVAC tended to average eight hours between failures, whereas relay computers like the (slower, but earlier) Harvard Mark I failed very rarely.

## Transistor and integrated circuit CPUs

The design overall of CPUs increased as various technologies facilitated building smaller and more reliable electronic devices. The first such improvement came with the advent of the transistor. Transistorized CPUs during the 1950s and 1960s no longer had to be built out of bulky, unreliable, and fragile switching elements like vacuum tubes and electrical relays. With this improvement more complex and reliable CPUs were built onto one or several printed circuit boards containing discrete (individual) components.

During this period, a method of manufacturing many interconnected transistors in a compact space was developed. The integrated circuit (IC) allowed a large number of transistors to be manufactured on a single semiconductor-based die, or "chip". At first only very basic non-specialized digital circuits such as NOR gates were miniaturized into ICs. CPUs based upon these "building block" ICs are generally referred to as "small-scale integration" (SSI) devices.

## Microprocessors

In the 1970s the fundamental inventions by Federico Faggin (Silicon Gate MOS ICs with self-aligned gates along with his new random logic design methodology) changed the design and implementation of CPUs forever. Since the introduction of the first commercially available microprocessor (the Intel 4004) in 1970, and the first widely used microprocessor (the Intel 8080) in 1974, this class of CPUs has almost completely overtaken all other central processing unit implementation methods. Mainframe and minicomputer manufacturers of the time launched proprietary IC development programs to upgrade their older computer architectures, and eventually produced instruction set compatible microprocessors that were backward-compatible with their older hardware and software. Combined with the advent and eventual success of the ubiquitous personal computer, the term CPU is now applied almost exclusively[a] to microprocessors. Several CPUs (denoted 'cores') can be combined in a single processing chip.

# Operation

The fundamental operation of most CPUs, regardless of the physical form they take, is to execute a sequence of stored instructions called a program. The instructions are kept in some kind of computer memory. There are four steps that nearly all CPUs use in their operation: fetch, decode, execute, and writeback.

The first step, fetch, involves retrieving an instruction from program memory. Its location (address) in program memory is determined by a program counter (PC), which stores a number that identifies the address of the next instruction to be fetched.

The instruction that the CPU fetches from memory is used to determine what the CPU is to do. In the decode step, the instruction is broken up into parts that have significance to other portions of the CPU. The way in which the numerical instruction value is interpreted is defined by the CPU's instruction set architecture (ISA

After the fetch and decode steps, the execute step is performed. During this step, various portions of the CPU are connected so they can perform the desired operation. If, for instance, an addition operation was requested, the arithmetic logic unit (ALU) will be connected to a set of inputs and a set of outputs.

The final step, writeback, simply "writes back" the results of the execute step to some form of memory. Very often the results are written to some internal CPU register for quick access by subsequent instructions. In other cases results may be written to slower, but cheaper and larger, main memory.

After the execution of the instruction and writeback of the resulting data, the entire process repeats, with the next instruction cycle normally fetching the next-in-sequence instruction because of the incremented value in the program counter.

# Design and implementation

The basic concept of a CPU is as follows:

Hardwired into a CPU's design is a list of basic operations it can perform, called an instruction set. Each of these basic operations is represented by a particular sequence of bits; this sequence is called the opcode for that particular operation.

The actual mathematical operation for each instruction is performed by a subunit of the CPU known as the arithmetic logic unit or ALU.

In many CPU designs, an instruction set will clearly differentiate between operations that load data from memory, and those that perform math.

## Control unit

The control unit of the CPU contains circuitry that uses electrical signals to direct the entire computer system to carry out stored program instructions. The control unit does not execute program instructions; rather, it directs other parts of the system to do so. The control unit must communicate with both the arithmetic/logic unit and memory.

## Integer range

The way a CPU represents numbers is a design choice that affects the most basic ways in which the device functions. Some early digital computers used an electrical model of the common decimal (base ten) numeral system to represent numbers internally.

Related to number representation is the size and precision of numbers that a CPU can represent. In the case of a binary CPU, a bit refers to one significant place in the numbers a CPU deals with. The number of bits (or numeral places) a CPU uses to represent numbers is often called "word size", "bit width", "data path width", or "integer precision" when dealing with strictly integer numbers (as opposed to floating point). This number differs between architectures, and often within different parts of the very same CPU.

## Clock rate

The clock rate is the speed at which a microprocessor executes instructions. Every computer contains an internal clock that regulates the rate at which instructions are executed and synchronizes all the various computer components. The faster the clock, the more instructions the CPU can execute per second.

Most CPUs, and indeed most sequential logic devices, are synchronous in nature.That is, they are designed with, and operate under, assumptions about a synchronization signal. This signal, known as a clock signal, usually takes the form of a periodic square wave. By calculating the maximum time that electrical signals can move in various branches of a CPU's many circuits, the designers can select an appropriate period for the clock signal.

## Parallelism

The description of the basic operation of a CPU offered in the previous section describes the simplest form that a CPU can take. This type of CPU, usually referred to as subscalar, operates on and executes one instruction on one or two pieces of data at a time.

This process gives rise to an inherent inefficiency in subscalar CPUs. Since only one instruction is executed at a time, the entire CPU must wait for that instruction to complete before proceeding to the next instruction. As a result, the subscalar CPU gets "hung up" on instructions which take more than one clock cycle to complete execution. Even adding a second execution unit (see below) does not improve performance much; rather than one pathway being hung up, now two pathways are hung up and the number of unused transistors is increased. This design, wherein the CPU's execution resources can operate on only one instruction at a time, can only possibly reach scalar performance (one instruction per clock). However, the performance is nearly always subscalar (less than one instruction per cycle).

### Instruction level parallelism

One of the simplest methods used to accomplish increased parallelism is to begin the first steps of instruction fetching and decoding before the prior instruction finishes executing. This is the simplest form of a technique known as instruction pipelining, and is utilized in almost all modern general-purpose CPUs. Pipelining allows more than one instruction to be executed at any given time by breaking down the execution pathway into discrete stages. This separation can be compared to an assembly line, in which an instruction is made more complete at each stage until it exits the execution pipeline and is retired.

### Thread-level parallelism

Another strategy of achieving performance is to execute multiple programs or threads in parallel. This area of research is known as parallel computing. In Flynn's taxonomy, this strategy is known as Multiple Instructions-Multiple Data or MIMD.

One technology used for this purpose was multiprocessing (MP). The initial flavor of this technology is known as symmetric multiprocessing (SMP), where a small number of CPUs share a coherent view of their memory system. In this scheme, each CPU has additional hardware to maintain a constantly up-to-date view of memory. By avoiding stale views of memory, the CPUs can cooperate on the same program and programs can migrate from one CPU to another.

### Data parallelism

A less common but increasingly important paradigm of CPUs (and indeed, computing in general) deals with data parallelism. The processors discussed earlier are all referred to as some type of scalar device. As the name implies, vector processors deal with multiple pieces of data in the context of one instruction. This contrasts with scalar processors, which deal with one piece of data for every instruction.

# Performance

The performance or speed of a processor depends on the clock rate (generally given in multiples of hertz) and the instructions per clock (IPC), which together are the factors for the instructions per second (IPS) that the CPU can perform. Many reported IPS values have represented "peak" execution rates on artificial instruction sequences with few branches, whereas realistic workloads consist of a mix of instructions and applications, some of which take longer to execute than others. The performance of the memory hierarchy also greatly affects processor performance, an issue barely considered in MIPS calculations. Because of these problems, various standardized tests, often called "benchmarks" for this purpose—such as SPECint – have been developed to attempt to measure the real effective performance in commonly used applications.

Processing performance of computers is increased by using multi-core processors, which essentially is plugging two or more individual processors (called cores in this sense) into one integrated circuit. Ideally, a dual core processor would be nearly twice as powerful as a single core processor. In practice, however, the performance gain is far less, only about 50%, due to imperfect software algorithms and implementation. Increasing the number of cores in a processor (i.e. dual-core, quad-core, etc.) increases the workload that can be handled. This means that the processor can now handle numerous asynchronous events, interrupts, etc. which can take a toll on the CPU (Central Processing Unit) when overwhelmed. These cores can be thought of as different floors in a processing plant, with each floor handling a different task. Sometimes, these cores will handle the same tasks as cores adjacent to them if a single core is not enough to handle the information.

# Summary

A modern CPU is usually small and square with many short, rounded, metallic connectors on its underside. Some older CPUs have pins instead metallic connectors.

The CPU attaches directly to a CPU "socket" (or sometimes a "slot") on the motherboard. The CPU is inserted into the socket pin-side-down and a small lever helps to secure the processor.

After running even a short while, modern CPUs can get very hot. To help dissipate this heat, it is necessary to attach a heat sink and a fan directly on top of the CPU. Typically, these come bundled with a CPU purchase.

# Glossary

1. Processing - Обробка
2. Hardware - Апаратні засоби
3. carries out - здійснює
4. performing - виконання
5. remains - залишки
6. multiprocessing - Багатопроцесорна обробка
7. microprocessor - мікропроцесор
8. contained - міститься
9. integrated circuits - інтегральні схеми
10. peripheral - периферійний
11. ALU - АЛП
12. CU – модуль контролю
13. IC – ІС – інтегральна схема
14. CPU – ЦП – центральний процесор
15. extracts - вибирає
16. decodes - декодує
17. executes - виконує
18. necessary - необхідно
19. rely - покладатися
20. array - масив
21. vector - вектор
22. multiple - багаторазовий
23. distributed - розподілений
24. interconnected - взаємопов'язані
25. set - набір
26. rewired - перемонтований
27. fixed-program computers – комп’ютери з фіксованими програмами
28. stored-program computer – комп’ютери з можливістю збереження програм
29. initially - спочатку
30. omitted -опущений
31. entitled –без заголовку
32. certain - певний
33. specified - зазначений
34. overcame -переміг
35. effort -зусилля
36. task -задача
37. custom-designed -спеціально розроблений
38. particular - конкретний
39. mass-produced -масове виробництво
40. standardization -стандартизація
41. discrete -дискретний
42. transistor - транзистор
43. mainframe – велика ЕОМ
44. rapidly -швидко
45. accelerated -прискорений
46. popularization -популяризація
47. increasingly -все більше і більше
48. complex -комплекс
49. tolerances -допуски
50. nanometers -нанометрів
51. increased -збільшений
52. limited -обмежений
53. beyond –за
54. far -далеко
55. appear -з'являтися
56. architecture -архітектура
57. utilized -використовується
58. punched paper tape -перфострічки
59. storage -зберігання
60. treatment -лікування
61. AVR –тип архітектури для пристроїв з малим енергоспоживанням
62. relays - реле
63. vacuum -вакуум
64. tube -трубка
65. thermionic - термоелектронний
66. valve -клапан
67. switching - перемикання
68. overall -загальний
69. facilitated -полегшений
70. manufacturing -виробництво
71. semiconductor-based die -на базі напівпровідникової підложки
72. small-scale integration - низька ступінь інтеграції

# Literature

1. Central processing unit [Електронний ресурс] – Режим доступу http://en.wikipedia.org/wiki/Central\_processing\_unit – Заголовок з екрану.
2. CPU - Central Processing Unit [Електронний ресурс] – Режим доступу http://www.webopedia.com/TERM/C/CPU.html – Заголовок з екрану.
3. CPU [Електронний ресурс] – Режим доступу http://www.computerhope.com/jargon/c/cpu.htm – Заголовок з екрану.
4. Central Processing Unit (CPU): Parts, Definition & Function

[Електронний ресурс] – Режим доступу http://education-portal.com/academy/lesson/central-processing-unit-cpu-parts-definition-function.html#lesson – Заголовок з екрану.