KEY PRINCIPLOF COMPUTATION

CHAPTER 1: FUNDAMENTALS OF COMPUTATION

1.1. Introduction

Amongst the most significant advancement of the 20th century is the electronic computer. The computer, and the communication and information technologies constructed on it, have altered business, government, society, and science, and have impacted practically every area of our life, just as industrialization did in the 19th century. The essential principles and procedures utilized in the advancements of computer apps are detailed in this literature, which covers the discipline of computing (Abramowitz & Stegun, 1965).

Getting into a novel sector like computers is going to a workstation in a foreign country for the first time. Whereas all countries have certain fundamental characteristics, like the need for language and cultural and trade preferences, the vast variances in such characteristics from one country to the next may be confusing and even devastating for newbies. Furthermore, describing the characteristics of a country in any universal way is challenging since they differ widely and changes with time. Similarly, joining the world of computers may be unsettling, and defining its characteristics may be challenging.

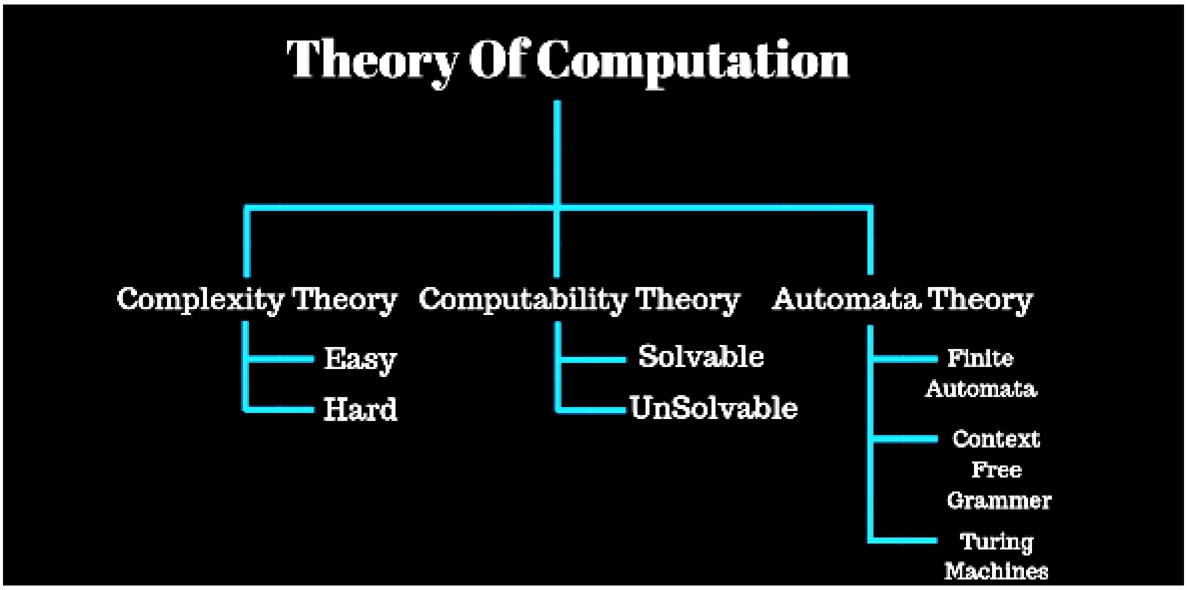


Figure 1.1. The computation theories

Source: https://www.proprofs.com/quiz-school/story.php?title=njuyodkw2z8j

Although there are basic notions that underpin the subject of computers which may be defined, learnt and applied practically (Acar, 2009). All calculation is dependent upon the associated utilization of computer types of equipment, referred to as hardware, and the programs of computers that make them referred to as software. All the software apps have been designed utilizing data and the specifications of the process, referred to as algorithms and data structures, and all hardware devices are manufactured by utilizing algorithms and data structures. Although software and hardware technology have advanced continuously during the history of computing, and novel paradigms for process and data descriptions have emerged regularly, such foundations have maintained relatively consistent throughout that period (Agrawal et al., 2008).

It begins with defining the idea of computing, and then moves on to examine the notions of software and hardware, before concluding with a foreword to the creation of software (also known as the programming of the computer). After that, most work is devoted to the production of computer software, with an in-depth explanation of software concepts and a glimpse of contemporary culture in the software development area. For the first half of the work, the author makes extensive usage of Processing, a Java-based programming environment; after that, the author makes extensive usage of the complete Java development environment.

1.2. Computing

As previously said, defining computing is difficult, however, according to the curriculum of Computing 2005: The Overview Report issued through a joint committee of AIS, ACM, and IEEE, computing is defined as follows: "We may define computing in a broad sense as any goal-oriented activity that needs, advantages from, or creates computers. It is a wide idea that contains computer hardware creation, the development of the application, and the development of software. The production of the software of a computer is the latest of such businesses and the subject of this text (Agrawal et al., 2006).

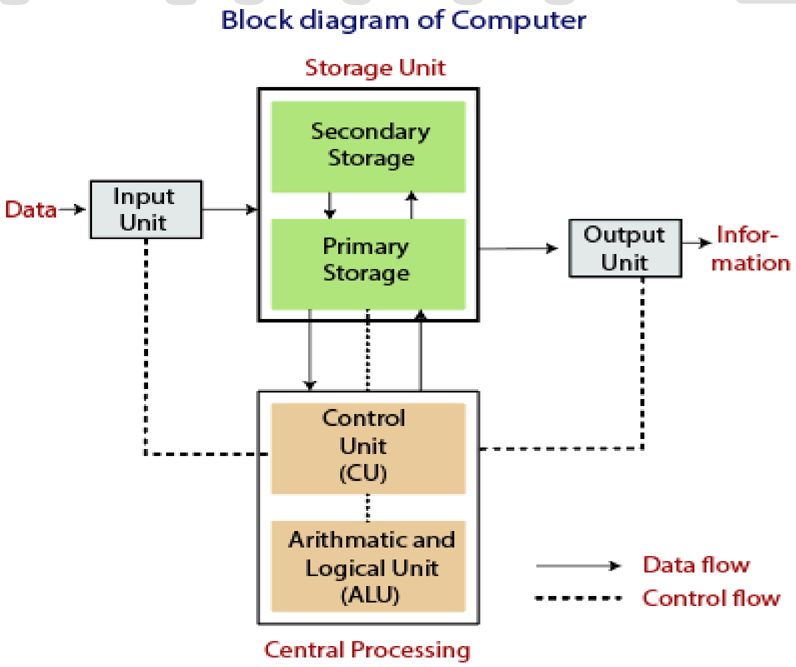


Figure 1.2. The computer's block diagram.

Source: https://www.tutorialandexample.com/block-diagram-of-a-computer/

Since the development of software is reliant on computer hardware, we will cover the basics of computer hardware and how it relates to software to better get ready students for the construction of software. The authors believe that it would connect a fresh era of software engineers, such as not just scientifically and mathematically inclined students who seem to be prevalent in the courses of programming, as well as a younger youth of students in the social sciences and civilizations that have been discovering that calculation is as pertinent to their disciplines as it is always in the fields of science (Agrawal et al., 2004).

1.3. Hardware

The word "computer" was first used in the 1600s. Till the 1950s, the word nearly exclusively applied to a human who did calculations. The task of conducting massive quantities of calculations is hard, time-consuming, and error-prone for humans. As a result, the wish to automate mathematics is a long-standing human ambition (Aho et al., 1983).

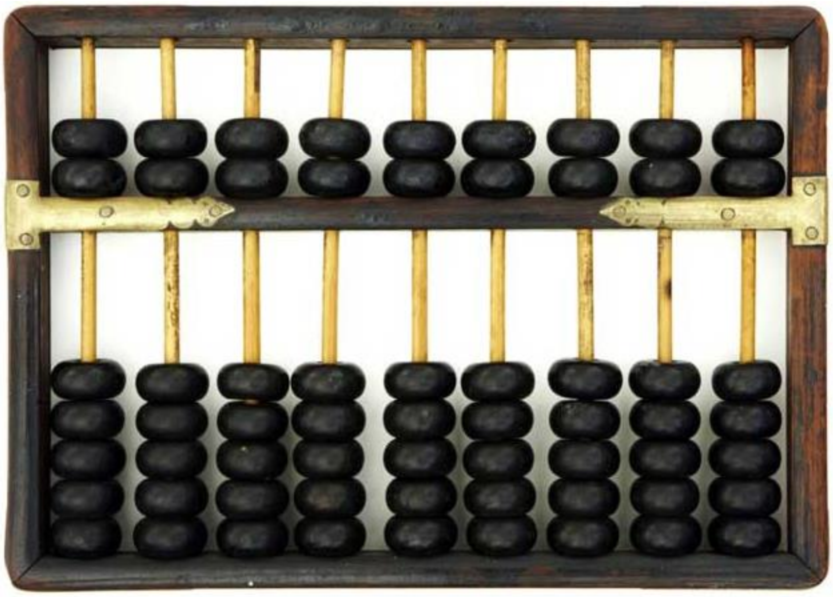


Figure 1.3. The demonstration of the ancient abacus

Source: https://www.opencolleges.edu.au/informed/learning-strategies/why-we-need-ancient-forms-of-learning-in-the-21st-century/

The abacus, which had been in usage in Indian, ancient Mesopotamia, Persian, Asian, Mesoamerican communities and Mesoamerican communities, and Greco-Roman is still in use worldwide nowadays, was among the first instruments devised for easing human arithmetic (Ahuja et al., 1989). An abacus is a digitized arithmetic device, similar to the contemporary computer since its actions imitate the variations in digits that happen when people perform fundamental arithmetic operations. It is made up of an orderly collection of stones or beads moving along rods or in grooves. A few of these cultures employed base-60, base-20, or base-16 numeral systems; therefore not all abacus systems utilized decimal–base-10–numerals.

Figure 1.4. Pascal's computer

Source: https://www.britannica.com/technology/Pascaline

Blaise Pascal (1623-1662), a young French mathematician, built one of the earliest adding machines which are gear-based to aid in the massive number of computations required in tax calculation. The decimal edition of the Pascaline operated similarly to a type of calculator popular among grocery store consumers in the United States and internationally throughout the late 1950s and early 1960s (Ahuja et al., 1995).

Charles Babbage (1792-1871), an English mathematician, introduced the 1st stage of his anticipated "Difference Engine" in 1822; it is similarly employed 10-position gears to express decimal digits. This had been able of further sophisticated computations as compared to addition machines such as Pascaline's basic arithmetic. Although, the engineering of the Difference Engine proved so difficult that Babbage discontinued the project for this and several other reasons (Ajtai et al., 2001).

There are 2 primary challenges here, each exhibiting two important computing ideas. Firstly, such gadgets were mechanical, requiring physically moving and connecting pieces. A device with moving components is usually always slower, more likely to fail, and more complex to build as compared to the equipment which is immoveable (Akra & Bazzi, 1998).

Electronic equipment, on the other hand, like **vacuum tubes** utilized in primitive radios, have no moving components by design.

As a result, the ENIAC, one of the first electronic digital computers, depicted every decimal digit with a column of 10 vacuum tubes that might electrically switch off and on to depict the 0 to 9 counting order of a decimal-digit without necessitating a physical movement.

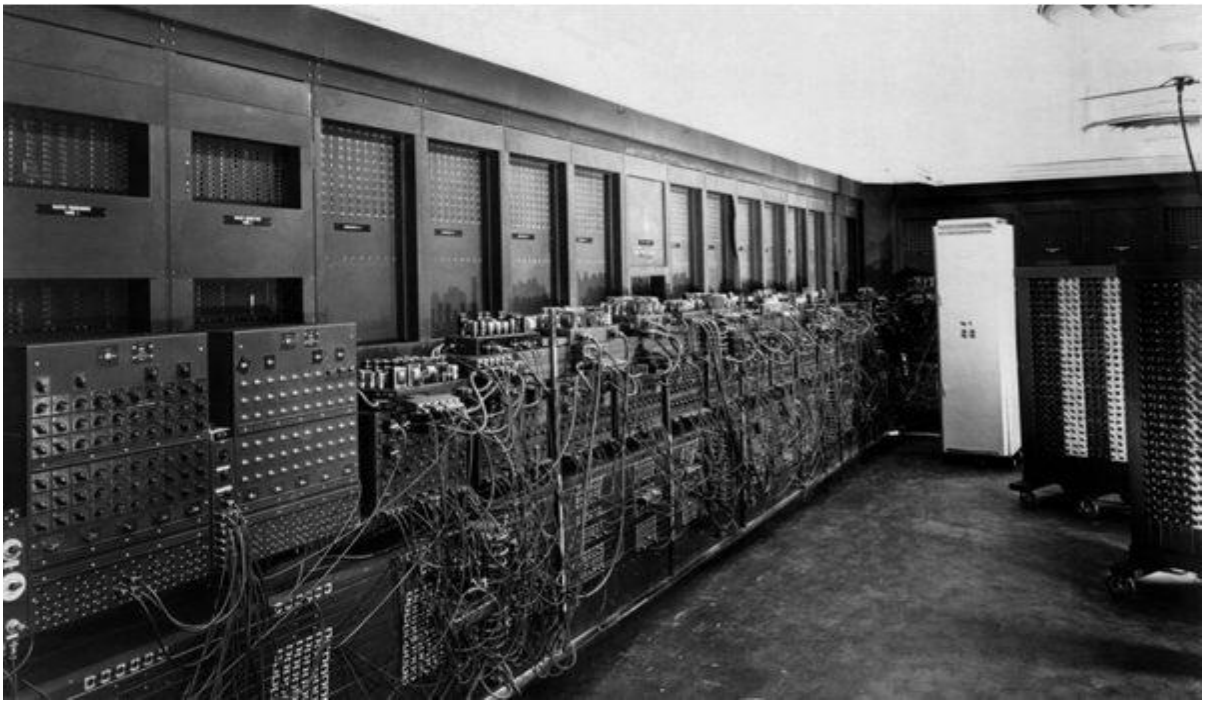


Figure 1.5. The world's first vacuum tube, as well as an electronic numerical computer and integrator

Source: https://www.hpcwire.com/2021/02/15/eniac-at-75-celebrating-the-worlds-first-supercomputer/

The 30-ton ENIAC, developed by John Mauchly and J. Presper Eckert at the Pennsylvania University between 1943 and 1946, needed used massive quantities of electricity and 18,000 vacuum tubes for their running time. This is because every decimal digit in the ENIAC needed ten vacuum tubes to express (Alon, 1990).

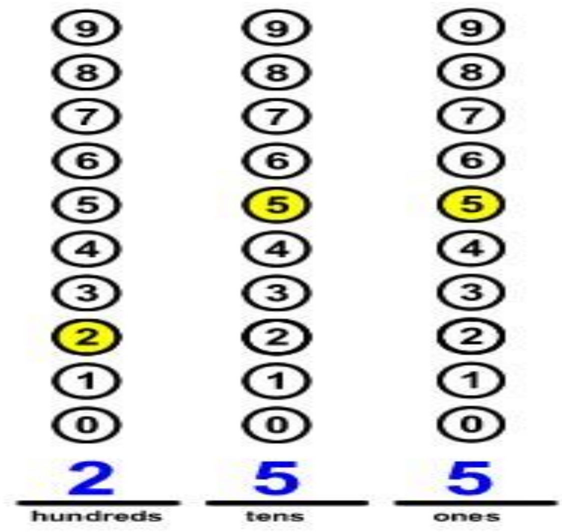


Figure 1.6. Base-2 numeral system.

Source: https://eduquestionbank.blogspot.com/2020/05/chapter-6-number-system.html

Contrarily, the 1st electrical digital computer, created by **Clifford Berry** and **John Atanasoff** at Iowa State University between 1937 and 1942, employed a binary – or Base-2–numeral system, as do all electronic digital computers nowadays (Amir et al., 2009).

Decimal digits are dependent upon powers of ten, with each number representing a different power of ten: ones (10^0), tens (10^1), hundreds (10^2), thousands (10^3), as well as so on. Therefore, the decimal number **two hundred & fifty-five** is expressed as 255, arithmetically equivalent to the sum of **2** hundred, **5** tens, and **5** ones. Therefore, ENIAC will only need to switch on 3 vacuum tubes to keep that number, but there will be a maximum of thirty vacuum tubes necessary to depict all of the possibilities of such 3 numbers (Amtoft et al., 2002).

On either side, **Binary digits** also referred to as **bits** are dependent upon powers of 2, with each number moving to the left representing a different power of 2: ones (2^0), twos (2^1), fours (10^2), eights (10^3), sixteen's (10^4), as well as so on. Therefore, the number 18 will be expressed in binary as **10010**, which is the total of **1** sixteen, **0** eights, **0** fours, **1** two, and **0** ones in Base-2:

Figure 1.7. Number 18 in the binary system

Source: https://cs.calvin.edu/activities/books/processing/text/01computing.pdf

To put it another way, the number **two hundred & fifty-five** will be represented in the binary numeric by the number **11111111**, which can be thought of arithmetically as the total of the **1** one, two, **1** four, **1** eight, **1** sixteen, **1** thirty-two, **1** sixty-four, and **1** one hundred twenty-eight:

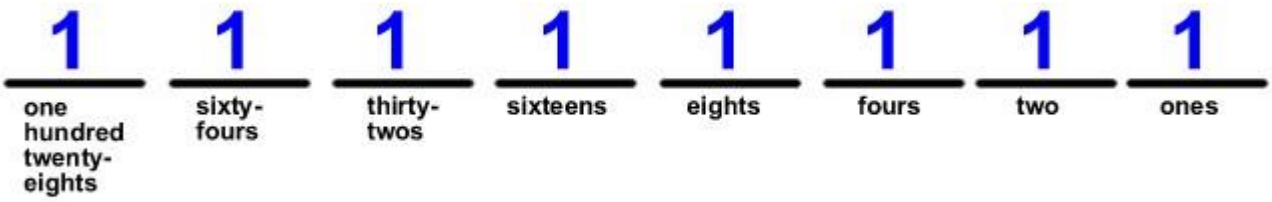


Figure 1.8. Number 255 in the binary system

Source: https://cs.calvin.edu/activities/books/processing/text/01computing.pdf

Why'd computer developers want to construct a system that performs arithmetic utilizing a binary, Base-2 numeral strategy, which is obscure and unfamiliar?

Every digit in a digital numeral system should be capable to count down to one lesser as compared to the base. As a result, under the Base-10 system, every decimal digit's counting order ranges between 0 and 9, subsequently reversing to zero. To depict a decimal-digit, we should be capable to account for all ten probabilities in the counting order, zero through nine, which necessitates the usage of either a device with ten feasible states, such as the Pascaline's ten-position gear, or ten separate types of equipment, such as the ENIAC's ten different vacuum tubes for every digit (Andersson, 1995).

The binary number system, on the other hand, is Base-2. Given that its digits must only be capable to count a maximum of one lesser as compared to the base, it implies that every binary digit's counting order runs between **0** and **1** and afterwards back to 0. In the other sense, although a decimal digit may include ten distinct integers ranging between 0 and 9, a binary digit may only contain a **0** or a **1**. Instead of needing to account for the ten potential states of a decimal digit, a binary digit may be represented using only one device with 2 feasible states. For instance, every binary digit may be represented by a simple off or on the switch, with the position of **ON** representing a **1** and the position of **OFF** representing a **0 (**Andersson, 1996**)**.

Similarly, every binary digit in the Atanasoff-Berry Computer might be replicated by a **single** vacuum tube. Therefore, rather than the twenty vacuum tubes needed by the ENIAC, the number **18** may be replicated with only five:

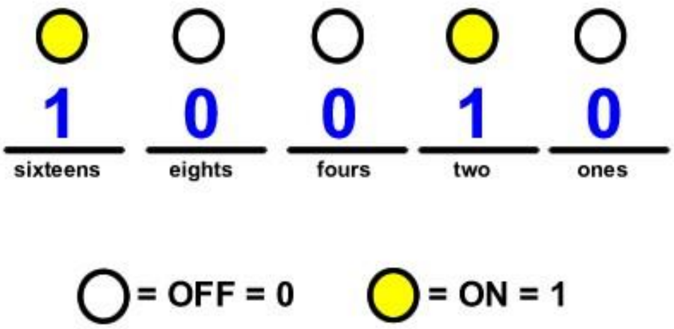


Figure 1.9. Number 18 depicted with only five vacuum tubes

Source: https://www.cuemath.com/numbers/18-in-binary/

Similarly, rather than the Thirty vacuum tubes used by ENIAC, just Eight vacuum tubes might be used to show the number **two hundred fifty-five**:

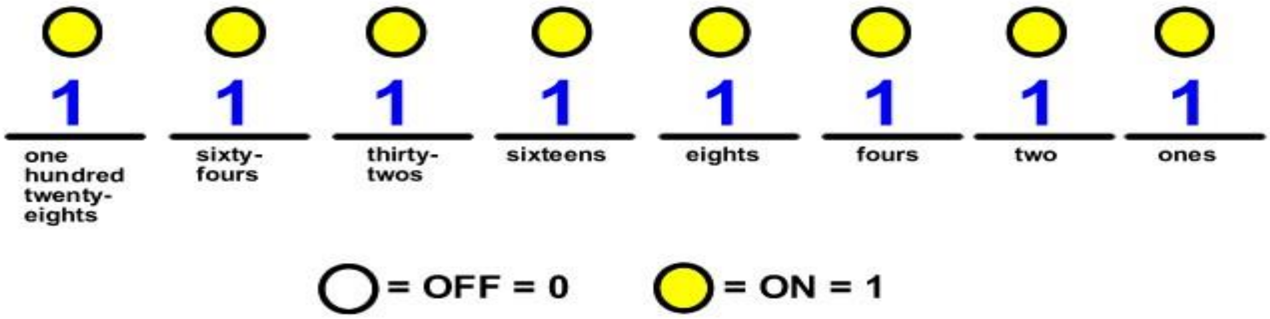


Figure 1.10. Number two hundred fifty-five depicted with just eight vacuum tubes

Source: https://www.sciencedirect.com/topics/computer-science/binary-to-decimal-conversion

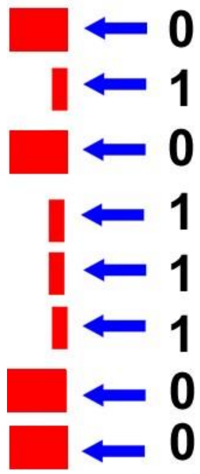
As a result, computer developers obtained an easy means to create electrical digital computers by the usage of 2-state electronic devices in return for the obscure unfamiliarity of binary representation.

This 1st generation of digital computers depending upon vacuum tubes gradually gave way to a 2nd generation to utilize the transistor-like an even quicker and much lesser and non-moving, off or on switch for depicting the 1/0 of a binary-digit, much like vacuum tube radios had been superseded through transistor radios beginning in the 1950s (Sjöberg et al., 2021).

1.4. Processors

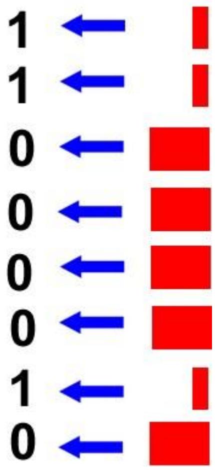
It's quite simple to grasp the basics of how a series of interlocking, 10-position gears may simulate decimal arithmetic processes. However, how a vacuum tubes arrangement or transistors utilization as electrical off or on switches replicates binary arithmetic processes are significantly less evident.

The analogy of a succession of dominoes may be quite effective. A domino programme on late-night television, for example, in which a domino champion constructs a complicated maze of dominoes, pushes one of them over and thereby initiates a lengthy chain reaction of dominoes falling. In the end, the succession of falling dominoes concludes with a dramatic flourish when the final set of dominoes falls over in a magnificent flourish. Think of a set of dominoes laid out on a table with a line of eight dominoes on one side and the second line of eight dominoes on the other side, divided by a maze of other dominoes in between. You may start a chain reaction of dominoes falling by going to the 8 dominoes on one side and knocking some or all of them over. The chain reaction will eventually come to a stop at another side when all or most of those 8 dominoes will be knocked over as a result of the chain reaction (Sandel et al., 2011).



Source: https://cs.calvin.edu/activities/books/processing/text/01computing.pdf

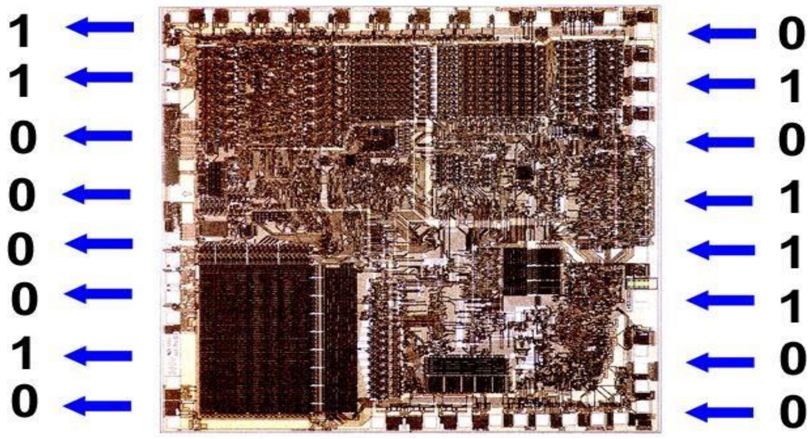
There are a few resemblances between this and how a processor operates. A domino, such as a transistor, is indeed a 2-state piece of equipment: it may be in either the on or off position and it may also be standing at the top or laying at the bottom. A domino or transistor, like every 2-state device, may represent the two alternatives for a binary digit: a 0 or a 1. Consider a domino that's also standing up like a 1 and a domino that is laying down like a 0 for instance. It's "inputting" an 8-digit binary number inside this domino "machine" by beating over part or every dominoes in the 1st row of 8 (Arora et al., 2001).



Source: https://cs.calvin.edu/activities/books/processing/text/01computing.pdf

As such, this binary number serves as a command to the machine, indicating the specific set of chain reactions between such 2-state equipment that must take place. The result is that after this chain reaction is finished, and a few of the entire of the 8 dominoes at either side have been thrown over, it seems as though this domino machine is "output" an 8-digit binary number.

Such domino similarity gives several resemblances to the method a processor chip composed of transistors functions in terms of functionality. In a processor, binary numbers that indicate the fundamental arithmetic operations addition, subtraction, multiplication, and division come into the processor in the format of electrical impulses that are either "low" or "high". As a result, a chain reaction occurs amongst the literally millions of tiny transistors as well as the off or on switches, that comprise the processor (Taylor-Robinson et al., 2011). A binary value indicating the outcome of the chain reaction is sent out on the wires heading away from the processor once it has been completed. The processor's maze of transistors is programmed in such a way that the outcome of the chain reaction is the 1 that corresponds to the "correct answer" to the arithmetic command that was provided as input. This was true because the Intel 8088 processor, which was utilized in the genuine IBM personal computer, had been an 8-bit processor. This meant that, during every cycle of instruction, an 8-digit binary number will be input, the processing (chain reaction) will occur, and the subsequently eight-digit binary number will be output (Kolliopoulos & Stein, 2004).



Source: https://cs.calvin.edu/activities/books/processing/text/01computing.pdf

Consequently now a day, at the core of its hardware, a contemporary electronic digital computer is indeed a machine that executes simple arithmetic operations. Furthermore, this has a computer machine that replicates or models the fact that numbers change as people perform simple arithmetic. When it comes to modelling arithmetic, the astounding speed with which today's computers do this task is something to behold. Today's microprocessors are generally 32-bits or high, which means that their commands have been made up of binary integers with 32 or more digits, as opposed to earlier generations. Command cycles for these kinds of computers machines are measured in "gigahertz," which means that they can do billions of command cycles per second, according to the specifications.

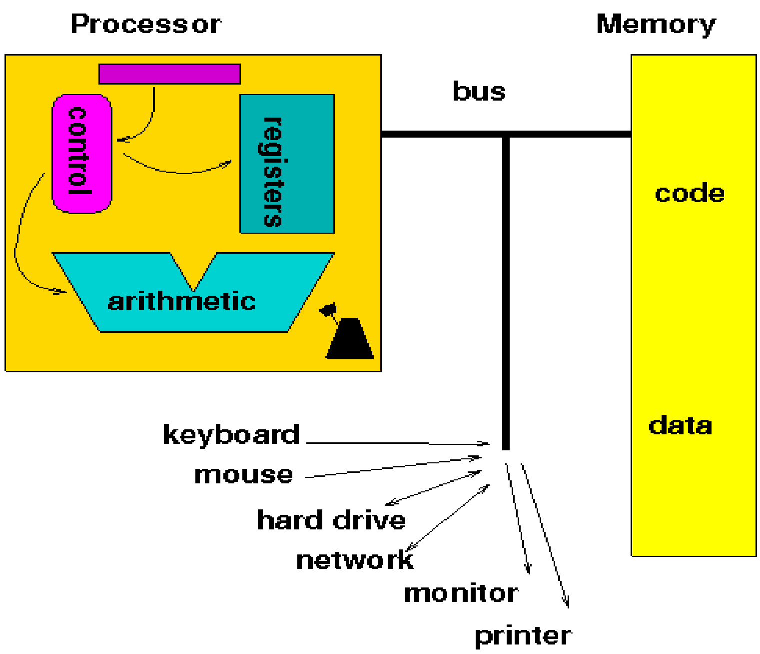


Figure 1.11. The processor's operating principle

Source: https://cs.wellesley.edu/~cs110/lectures/compModel/computer.html