Journal of Global Research in Computer Sciences e-ISSN: 2229-371X

Optimization and Programming of Computer Architecture

Rouf Ali*

Department of Computer Science, Government Degree College Boys Sopore, Jammu and Kashmir, India

Opinion Article

Received: 28-Nov-2022, Manuscript No. GRCS-22-83017; **Editor assigned:** 01-Dec-2022, Pre QC No. GRCS-22-83017 (PQ); **Reviewed:** 14-Dec-2022, QC No. GRCS-22-83017; **Revised:** 21-Dec-2022, Manuscript No. GRCS-22-83017 (A); **Published:** 30-Dec-2022, DOI: 10.4172/2229-371X.13.4.003

***For Correspondence**:
Rouf Ali, Department of Computer Science, Government Degree College Boys Sopore, Jammu and Kashmir, India
E-mail: alirou@gmail.com

DESCRIPTION

Computer architecture in computer engineering refers to the breakdown of a computer system into its basic components. Sometimes it can be a high-level explanation that eliminates implementation specifics. The description may go into greater detail and cover the design of the instruction set architecture, microarchitecture, logic design and implementation. The communication between Charles Babbage and Ada Lovelace detailing the analytical engine contains the earliest known examples of computer architecture. In two patent filings for his subsequent projects, Konrad Zuse said that machine instructions might be stored in the same storage employed for data while constructing the computer Z1.

The performance, efficiency, cost and dependability of a computer system must all be balanced according to computer architecture. The balance of these conflicting forces can be demonstrated using the example of instruction set architecture. Programmers can create more space-efficient programmes by using instruction sets with higher levels of complexity since a single instruction can encode a higher level of abstraction (such as the x86 Loop instruction). Longer and more complicated instructions, however, take longer for the processor to decode and can be more expensive to successfully implement. Large instruction sets also enhance complexity which increases the potential for unreliability when instructions combine in unanticipated ways.

Integrated circuit design, packaging, power and cooling are all part of the execution. Compiler, operating system, logic design and packaging knowledge are necessary for design optimization. The interface between the software and hardware of a computer is known as an Instruction Set Architecture (ISA), which is also known as the programmer's view of the hardware. High-level programming languages like Java, C++, and the majority of other languages are not understood by computers. Only instructions expressed in a numerical way typically as binary numbers are understood by processors. These high level languages are translated into instructions that the processor can understand by software tools like compilers. In addition to instructions, the ISA also specifies the components of the computer that a programme can access such as memory, registers, addressing modes and data types. Using memory addressing modes and register indexes (or names), instructions identify these items that are available. Performance-based products are optimised with the aid of computer organisation. Software engineers, for instance, need to be aware of CPU power. To get the best performance at the cheapest price, the programme might need to be optimised. An in-depth examination of the structure of the computer may be necessary for this. For instance, while designing an SD card, the designers may need to arrange the card in a way that allows the most data to be processed quickly.

The arrangement of a computer aids in planning the choice of a processor for a certain purpose. While virtual machines may require quick interruptions, multimedia projects may require very quick data access. Sometimes additional components are required for specific tasks. For instance, virtual memory hardware is required for a computer to run a virtual machine so that the memory of many virtual computers can be maintained distinct. Power usage and processor cost are also impacted by computer functionality and structure.

- A functional machine must be created after an instruction set and micro-architecture have been devised. The implementation of this design method is frequently used to refer in hardware design engineering rather than architectural design.
- The necessary circuits are designed at the logic-gate level by logic implementation.
- During circuit implementation, fundamental elements like gates, multiplexers and latches are designed at the transistor level. Some bigger blocks such as ALUs and caches may also be implemented at the logic-gate level or even at the physical level depending on the design.
- Physical circuits are drawn during implementation. The wires linking the various circuit components are made and they are then put in a chip floor plan or on a board.

Design validation tests the entire computer to verify if it functions properly in all circumstances and at all times.

A computer system's precise design is determined by its limitations and objectives. Standards, power versus performance, cost, memory capacity, latency (the amount of time it takes for information to move from one node to the source), and throughput are typically traded off in computer designs. Other aspects can also come into play, such as features, size, weight, dependability and expandability. The most popular design does a thorough power analysis to determine how to minimise power usage while keeping appropriate performance.