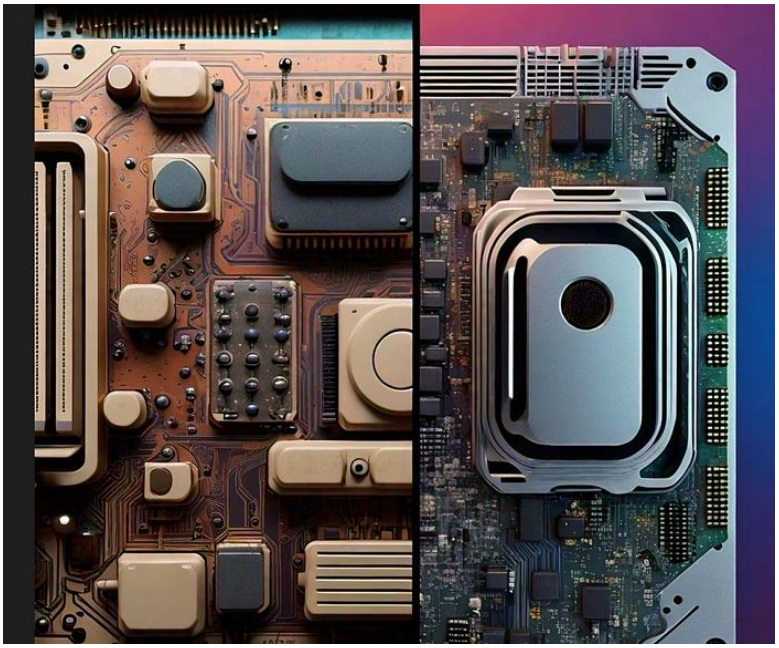
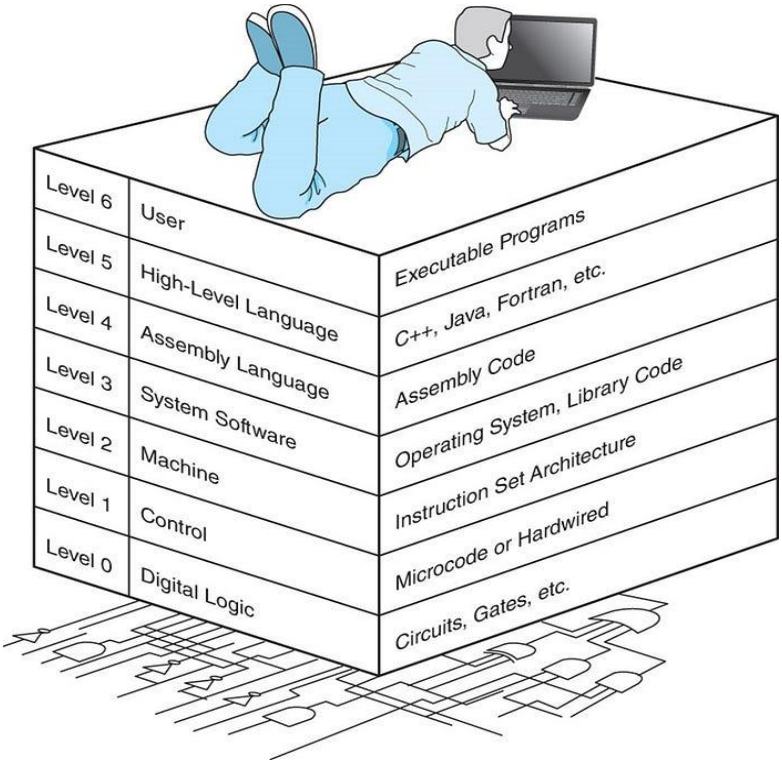
# Computer Architecture — History, Present and Future

Aaron Masuba
https://medium.com/
May 31, 2024



**Introduction**

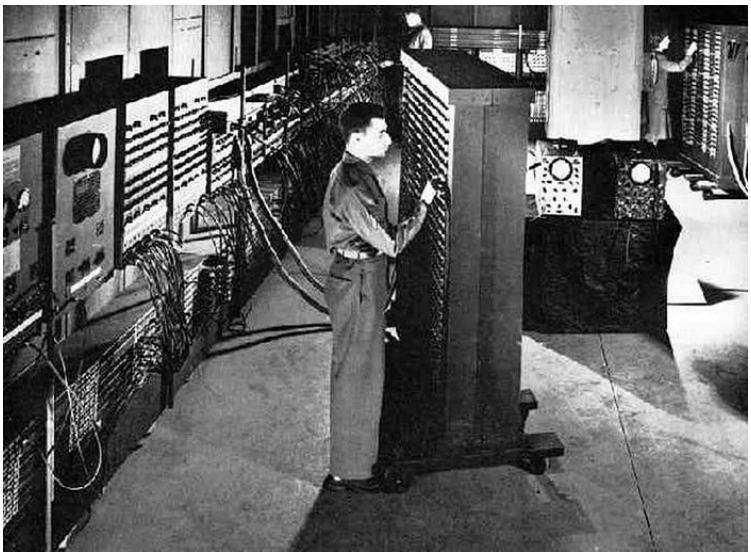
Computer architecture is the conceptual design and fundamental operational structure of a computer system. It is a blueprint and functional description of requirements and design implementations for the various parts of a computer, which, when combined, form the whole computer system. Understanding computer architecture is crucial for computer scientists, engineers, and programmers as it influences the efficiency, performance, and scalability of computational systems. This article delves into the history, present state, and future of computer architecture, guided by the visual representation of the various levels of abstraction in computer systems.

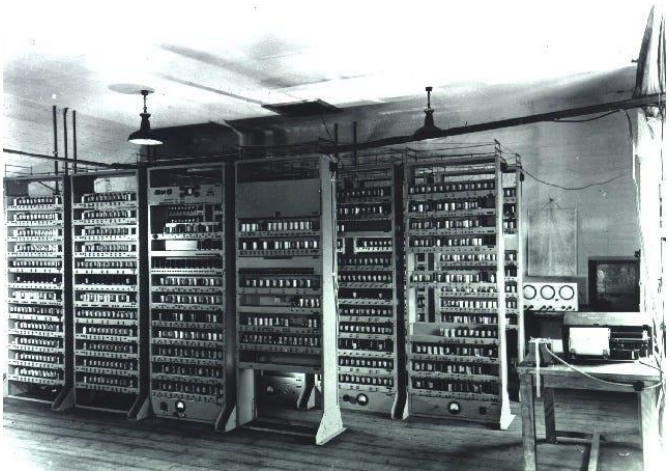


Abstraction of Computing

**Early Developments (1940s — 1950s)**

The inception of computer architecture can be traced back to the 1940s with the development of the first electronic computers. These early machines, such as the ENIAC, were programmed using machine code (Level 2 — Machine) and had rudimentary control logic (Level 1 — Control) implemented through hard-wired circuits (Level 0 — Digital Logic).


The Electronic Numerical Integrator and Computer (ENIAC), built between 1943–1946



The Electronic Delay Storage Automatic Calculator (EDSAC), built in 1949

**The Rise of Assembly Language (1950s — 1960s)**
In the 1950s, assembly language (Level 4 — Assembly Language) was introduced, allowing programmers to write code using symbolic representations rather than binary code. This advancement made programming more accessible and less error-prone compared to directly writing machine code.

```
; Example of IBM PC assembly language
; Accepts a number in register AX;
; subtracts 32 if it is in the range 97-122;
; otherwise leaves it unchanged.

SUB32  PROC         ; procedure begins here
       CMP  AX,97   ; compare AX to 97
       JL   DONE    ; if less, jump to DONE
       CMP  AX,122  ; compare AX to 122
       JG   DONE    ; if greater, jump to DONE
       SUB  AX,32   ; subtract 32 from AX
DONE:  RET          ; return to main program
SUB32  ENDP         ; procedure ends here
```

Figure showing Assembly Language

**The Advent of High-Level Languages (1960s — 1980s)**
High-level languages (Level 5 — High-Level Language) like FORTRAN, C, and later C++, emerged in the 1960s and 1970s. These languages abstracted many of the complexities of assembly language, enabling developers to write more complex and portable programs. This period also saw the development of compilers that translate high-level language code into assembly code, and then into machine code.

The Vintage Mainframe Computer Setup

**Present State of Computer Architecture**

**Multi-Level Abstraction**

Modern computer systems are characterized by multiple levels of abstraction, each building upon the lower levels. The image illustrates this hierarchy:

- Level 0: Digital Logic — The foundation, consisting of circuits and gates that perform basic operations.
- Level 1: Control — Implements control logic through microcode or hardwired mechanisms.
- Level 2: Machine — Represents the instruction set architecture (ISA) defining how software controls hardware.
- Level 3: System Software — Includes operating systems and library code that manage hardware resources and provide services to applications.
- Level 4: Assembly Language — A low-level programming language closely related to machine code.
- Level 5: High-Level Language — Languages like C++, Java, and Python, which provide powerful abstractions for software development.

- Level 6: User — The layer where end-users interact with executable programs and applications.

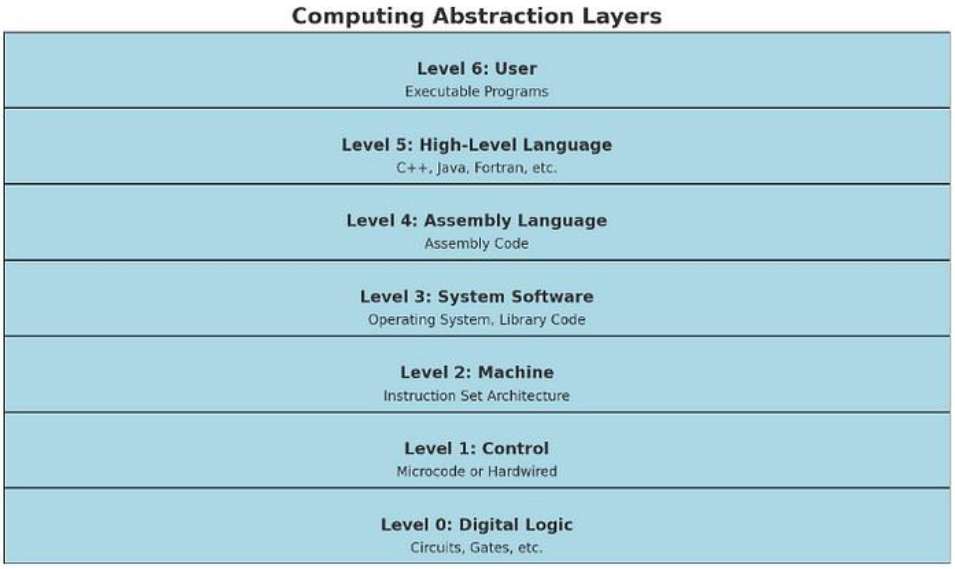
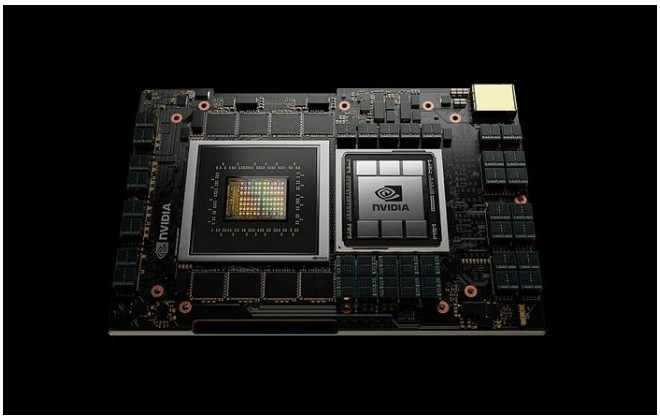


Image of Computing Abstraction Layers

**Advances in Processor Design**

Today's processors are immensely more powerful and efficient than their predecessors. Techniques such as pipelining, parallelism, and advanced branch prediction have significantly improved performance. Multi-core processors and specialized architectures like GPUs are now standard, catering to diverse computational needs from general-purpose computing to AI and graphics processing.



Nvidia's Arm-based Processor (Nvidia Press Release — April 12, 2021)

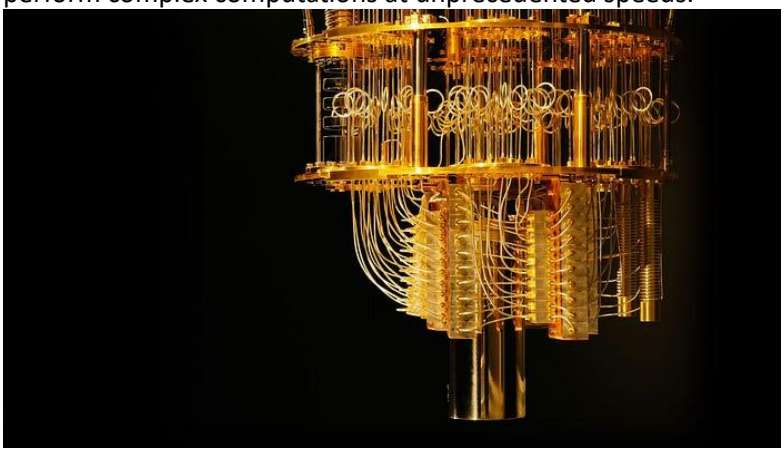
**Software and System Integration**

The integration of system software, such as modern operating systems, plays a crucial role in managing hardware complexity and providing a stable platform for applications. Virtualization and cloud computing are prominent trends, allowing efficient resource utilization and scalability.

AMD Versal AI Edge SoC FPGA is used in space applications

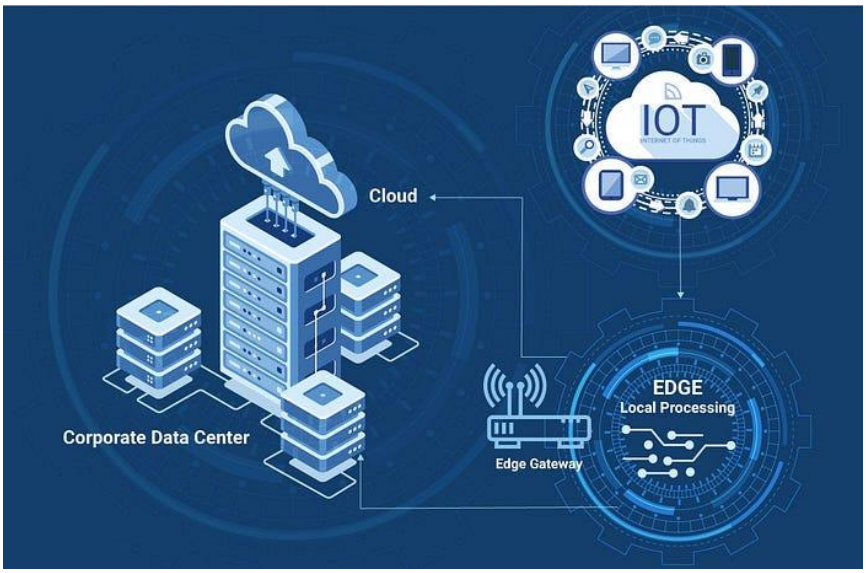
**Future of Computer Architecture**

**Quantum Computing**
Quantum computing represents a paradigm shift, potentially revolutionizing computer architecture. It leverages quantum bits (qubits) and quantum gates (analogous to circuits and gates in digital logic) to perform complex computations at unprecedented speeds.


IBM's System One — one of the first quantum computers to be developed

**Neuromorphic Computing**
Inspired by the human brain, neuromorphic computing aims to design hardware that mimics neural networks. This approach could lead to significant advancements in AI and machine learning, providing more efficient and powerful computation models.

**Edge Computing and IoT**
The proliferation of Internet of Things (IoT) devices demands new architectures that support edge computing. This decentralized approach processes data closer to the source, reducing latency and bandwidth usage, and enhancing real-time capabilities.




**Sustainable Computing**
Energy efficiency and sustainability are becoming crucial considerations. Future architectures will focus on reducing power consumption and utilizing renewable energy sources to minimize environmental impact.

**Conclusion**
Computer architecture has evolved significantly from its early days of hardwired circuits and machine code to the sophisticated multi-level abstractions of today. As technology continues to advance, the future of computer architecture promises exciting developments in quantum computing, neuromorphic computing, and sustainable designs. Understanding these concepts is essential for anyone involved in the design and implementation of modern computing systems.
This comprehensive view, spanning from the history to the potential future of computer architecture, underscores its dynamic nature and the continual innovation driving this field forward.