Demystifying GPU architecture for AI processing

Blog 1 – Introduction to topic & understand basics of compute machines.

AI is no longer a futuristic concept but a transformative force in our daily lives. It has already revolutionized various aspects such as autonomous driving, voice-activated technology, home automation, language processing tools and office productivity solutions. Next time when you will be shopping for a laptop, you will encounter term like AIPC (AI Personal Computer), reflecting AI's integration into mainstream technology.

In discussions about AI, terms such as TOPs (Tera Operations Per Second), training, inferencing, and GPUs frequently arise. GPUs have seen a surge in demand recently. What sets GPUs apart and makes them so compelling for processing AI data models? Could GPUs evolve to become the primary computing platforms for AI?

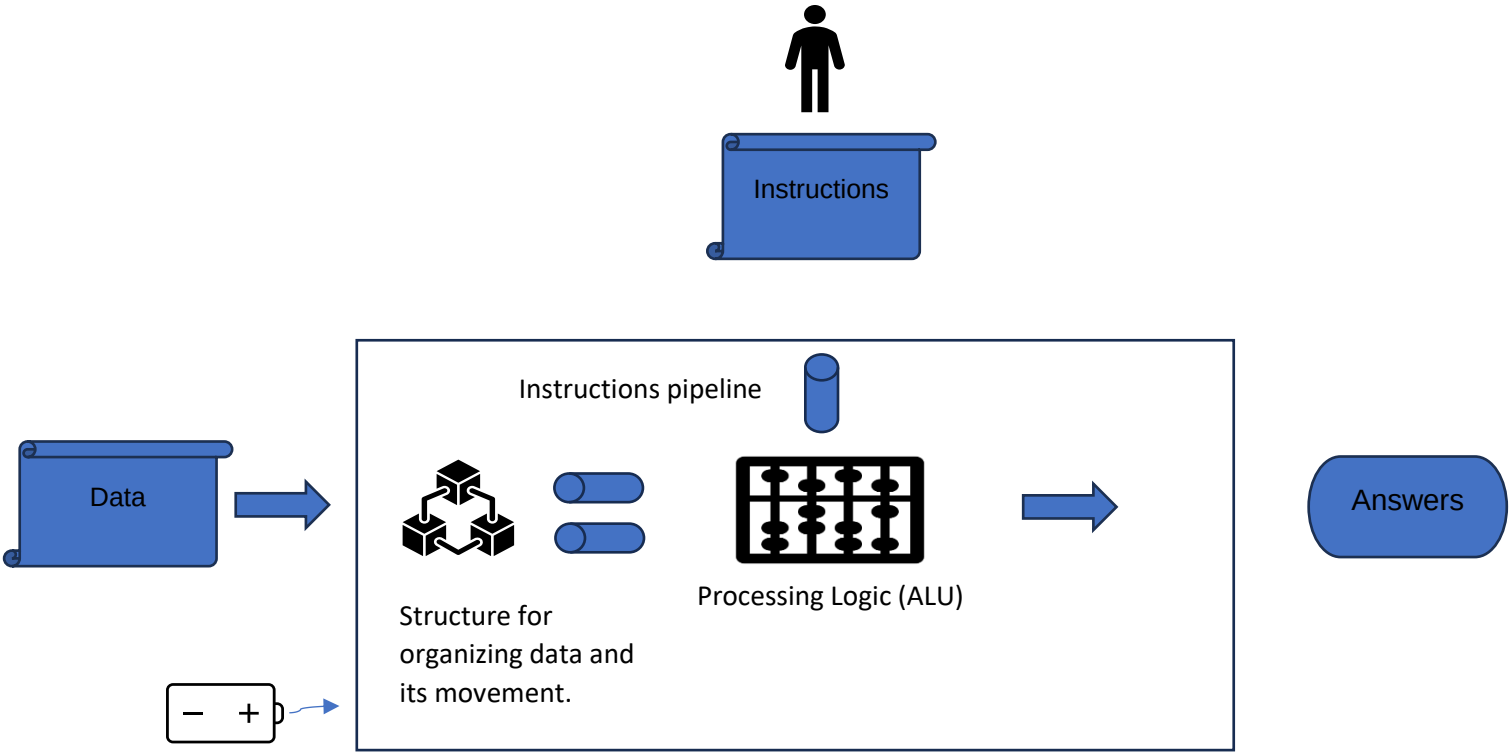
To grasp the answers to these questions, it's essential to understand the fundamental processing requirements of AI algorithms and how GPU architecture brings effective solutions for AI computations. Over the next few blogs, I'll endeavour to explain these architectural aspects in a simplified manner.

I will explain current machines and their architectural characteristics, AI data models, and algorithm requirements, as well as GPUs' key characteristics and why they are the optimal solution for current AI requirements. If you are interested in learning more about this topic, I hope the next 4-5 blogs will help you quickly grasp the concepts.

Current Compute Machines & their limitations:

When I refer to a "compute machine," I'm talking about a specific hardware structure made up of digital building blocks. This includes a logic processing engine (ALUs), an assembly pipeline for managing data flow to and from these engines, algorithms for fetching instructions and data,

memory, cache organization, extensive data movement networks (Network on Chip), power units, and more. The choice of the optimal machine architecture depends on the size and type of data payloads and instructions that need to be processed.



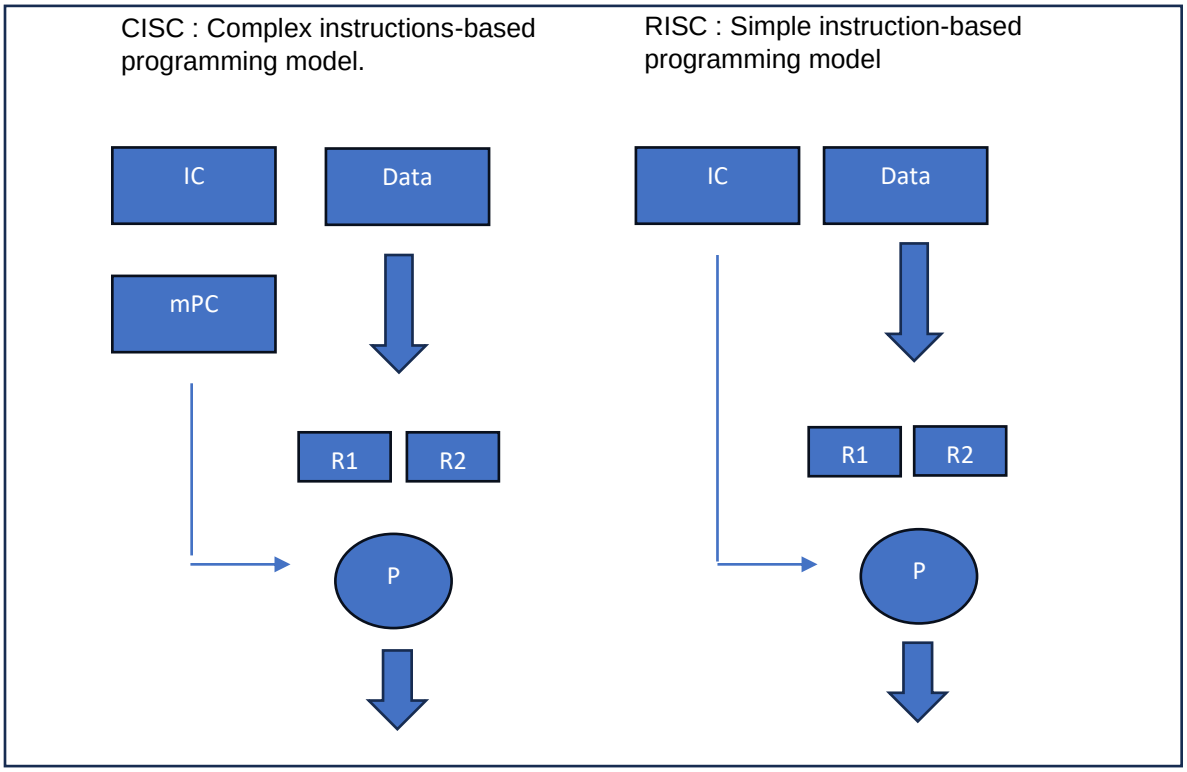
Simple model of a compute Machine

Last four decades are dominated by CISC and RISC architecture-based compute machines.

Over the past few decades, two primary architectures have profoundly influenced the computing landscape: CISC (Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer). These architectures differ in their support for type of instructions and the associated pipelines used for processing. Fundamentally, both types execute instructions sequentially, each typically handling one data element at a time—a process known as scalar processing.

Throughout their evolution, numerous variants of these fundamental architectures have emerged and been adopted based on varying power and performance requirements. Examples are Mobile

phones, embedded system processors, PCs, data servers etc. These architectures have effectively met the processing needs of data & programming models needed in the past four decades.



Scalar processing: CISC & RISC machines executes one instruction at a time on two operands.

With AI applications, there is an emerging need for specific data structures and algorithm processing capabilities. The question arises whether CISC and RISC architectures are suitable for processing these requirements, or if different machines are necessary. I will delve into this topic in the next blog post.