

Verständnis der CPU

Die CPU, auch zentrale Verarbeitungseinheit genannt, ist eine Kernkomponente und durchaus vergleichbar mit dem menschlichen Gehirn. Sie ist verantwortlich für die Ausführung von Befehlen und Verarbeitung von Daten oder ggf. Eingaben des Nutzers.

Die Architektur einer CPU lässt sich in folgende Punkte unterteilen:

- Steuerwerk oder auch Control Unit
- Rechenwerk oder auch Arithmetic Logic Unit, kurz ALU
- Register
- Busse
- Taktgeber oder auch Clock
- Cache-Speicher

Control Unit

Die Control Unit steuert quasi die Prozesse, die innerhalb der CPU verarbeitet werden. Sie liest, dekodiert und steuert die Ausführung von Befehlen aus dem RAM. Kurz: Wir starten ein Programm, Daten und Befehle werden im RAM geladen und von der Control Unit, so wie beschrieben, gelesen, dekodiert und gesteuert.

Ähnlich wie bei einem Dirigenten, der den Ablauf und die Koordination der Musiker leitet, leitet und koordiniert diese Komponente die verschiedenen Prozesse innerhalb der CPU.

Arithmetic Logic Unit

Die ALU ist für die arithmetischen und logischen Prozesse der Befehle verantwortlich. Grundsätzlich arbeitet sie mit binären Daten in der Form von Bit-Muster. Darauf einzugehen wäre zu komplex und würde den Rahmen nur noch weiter ausreizen. Dennoch hier ein Beispiel für ein 8-Bit-Muster: 00101011

Grob gesagt verwendet die ALU solche Bit-Muster, um am Ende Zahlen, Zeichen, Befehle und andere Informationen darzustellen.

Die Verarbeitung dieser Bit-Muster erfolgt durch Addition, Subtraktion, Multiplikation, Division, AND, OR, NOT und Shift Left/Right.

Auch hier als Analogie: Die ALU ist quasi der Taschenrechner der CPU. Wir müssen erst etwas eingeben, erst dann erfolgt die Berechnung. Auch ist ein Taschenrechner kein vollwertiger Computer, genauso ist die ALU nur ein Teil der CPU.

Ablauf

1. Fetch oder auch Abrufen

Beim Öffnen und Bedienen eines Programms werden Daten erhoben, welche grundsätzlich im RAM gespeichert werden. Die Control Unit prüft erst den Cache der CPU, ob die benötigten Daten schon abrufbar sind. Sind sie abrufbar, ist es der sogenannte Cache-Hit und es muss nicht erst auf den RAM gewartet werden. Dies resultiert in einer kürzeren Verarbeitungszeit.

Sollte aber der Fall eintreffen, dass die Daten nicht im Cache gefunden werden, muss auf den RAM zugegriffen und die benötigten Daten dort abgerufen werden. Dies ist dann der Cache-Miss.

Innerhalb dieses Prozesses findet auch der Datenbus Verwendung. Dieser transportiert die Daten/Befehle zwischen dem RAM und der CPU.

2. Decode oder Dekodieren

Die Control Unit dekodiert die Daten/Befehle in Steuersignale - also Bit-Muster - legt fest, um welche Art Befehl es sich handelt und welche Operation durchgeführt werden soll.

Hier findet auch der Adressbus Verwendung, da dieser Adressen/Bit-Muster von folgenden Befehlen angeben kann.

3. Execute oder Ausführen

Nachdem die Steuersignale dekodiert und festgelegt wurde, welche Operation verwendet werden muss, leitet die Control Unit diese an die ALU weiter. Hier führt die ALU wie oben beschrieben Berechnungen durch.

4. Store oder Speichern

Folglich werden die Ergebnisse in den Registern der CPU und von dort mit den Datenbussen im RAM gespeichert

Interpretation der Taktrate, Kerne und Cache

Dieser Ablauf wird natürlich auch durch die Frequenz der CPU beeinflusst, heißt: Umso höher die Frequenz/Mhz, umso schneller ist der Ablauf zwischen den jeweiligen Operationen. Sobald auch Daten im Cache temporär gespeichert sind, verringert sich die Latenz, sodass nicht auf die Daten im RAM zugegriffen werden muss.

Und: Je mehr Kerne eine CPU hat, desto mehr Operationen oder Befehle können gleichzeitig bearbeitet werden.

Interpretation der Interaktion zwischen CPU und GPU

Wie oben beschrieben ist die Control Unit der Dirigent, heißt sie analysiert den Befehl oder die Daten, die verarbeitet werden müssen, stellt fest, dass es sich um einen Prozess handelt, der einer Grafik-Aufgabe entspricht und leitet die Daten an die Grafikkarte weiter, die für den Prozess notwendig sind.

Interpretation der Architekturen x86, ARM und MIPS

x86 ist eine CISC-Architektur (Complex Instruction Set Computing). Hier werden eine große Anzahl an komplexen Befehlen unterstützt, welche primär ihren Einsatz in Desktop- und/oder Server-Systemen finden.

ARM ist eine RISC-Architektur (Reduced Instruction Set Computing). Wie der Name schon vermuten lässt, handelt es sich hier um eine abgespeckte Ausführung einer Architektur, welche auf einen kleineren Satz von einfachen, aber effizienten Befehlen aufbaut. Diese Architektur wird häufig in Mobilgeräten, Servern und Rechenzentren mit energieeffizienten Anforderungen eingesetzt.

MIPS zählt wie ARM zur RISC-Architektur, die für klare Befehlssätze bekannt ist. Ähnlich wie bei ARM kommt MIPS bei Servern, Workstations zum Einsatz, aber auch eingebettete Systeme greifen auf diesen Architekturtypen zurück. Beispiele für eingebettete Systeme sind: Steuerungssysteme in Autos(ABS), Herzschrittmacher, Mikrowellen, Waschmaschinen, Fernseher, usw.

Verständnis Speichertechnologien

HDD (Hard Disk Drive)

Dieses Speichergerät ist die kostengünstigere Alternative und wird häufig in Systemen verwendet, wo größere Datenmengen abgelegt, aber nicht so häufig verwendet werden müssen.

Ein Merkmal der HDD ist die Verwendung magnetischer Speicherplatten, die auf einem sich drehenden Metallteller montiert sind. Ein Lese-/Schreibkopf schreibt oder liest die Daten von den jeweiligen Oberflächen dieser Platten.

SSD (Solid State Drive)

Dieses Medium krönt das preisliche und leistungsfähige Mittelfeld. Im Gegensatz zur HDD sind hier keine mechanischen Bauteile in Verwendung. Durch die Verwendung von Flash-Speicherchips erfolgt die Speicherung und der Zugriff von Daten hier über elektronische Schaltungen. Sie werden in Laptops und Desktop-Computern verwendet und gewährleisten schnellere Ladezeiten.

NVMe (Non-Volatile Memory Express)

Hierbei handelt es sich um eine Protokollspezifikation, die für die direkte Kommunikation zwischen dem Speichergerät und dem Systemprozessor entwickelt wurde. NVMe-Laufwerke können in der Regel SSDs oder andere Flash-Speichergeräte sein.

Diese Speichertechnologie ist von der Leistung wesentlich schneller, da die Latenz wesentlich geringer ist. Grund dafür sind die oben beschriebenen Eigenschaften und werden aus diesem Grund in Hochleistungsrechnern, Gaming-PCs und Workstations verwendet. Preislich befinden sie sich über SSDs.

Interpretation der Dateisysteme FAT32, NTFS und EXT

FAT32 unterstützt Dateigrößen bis zu 2TB und kann von vielen Betriebssystemen gelesen und beschrieben werden. Keine besonderen Vor- oder Nachteile.

NTFS unterstützt sehr große Dateigrößen, theoretisch bis zu 16 Exabyte.

Darüber hinaus bietet dieses Dateisystem Verschlüsselung, Kompression, Zugriffskontrollen und Fehlerkorrekturen. Bei Verwendung von Nicht-Windows-Systemen und alten Computern kann es hier zu Einschränkungen kommen.

EXT ist das Standard-Dateisystem für viele Linux-Systeme und unterstützt ähnlich wie bei NTFS große Dateigrößen. Dieser Typ bietet auf Linux-Systeme Journaling für verbesserte Datensicherheit und effiziente Verwaltung von Speicherplatz.

Verständnis RAM

Wie die CPU ist auch der RAM eine Kernkomponente des Computers und fungiert als temporärer Speicher von Daten und Programmcode, die von der CPU verwendet werden. Anders als bei einer SSD, wo der Zugriff auf Daten nur sequenziell gewährleistet werden kann, ermöglicht der RAM sofortigen Zugriff auf beliebige Speicherzellen/Adressen. Kurz gesagt: Der enthält alle temporären Daten, die vom Betriebssystem und laufenden Anwendungen generiert werden. Die Daten im RAM dienen der CPU als Zwischenspeicher, um ihr den sofortigen Zugriff zu benötigten Daten zu gewährleisten.

Der Virtual Memory ist lediglich ein Mechanismus, der den physischen RAM erweitert, indem Teile der Festplatte als temporäre Erweiterung des RAMs freigibt.

Vorteile einer Cloud

Eine Cloud ist eine Dienstleistung, bei der Ressourcen für bestimmte Aufgaben bereitgestellt werden. Ressourcen könnten sein: Rechenleistung, Speicher, Anwendungen, usw.

Vorteile sind:

- verzicht auf lokaler Hard- und Software
- Skalierbarkeit(Bedarfsorientiert)
- Flexibilität(Anwendungsorientiert)
- Kostenersparnis
- leicht Zugänglich
- Automatisierung