|  |  |  |
| --- | --- | --- |
|  | **Leerkracht: H. Maes**  **Vak: Hardware** |  |
| **Naam leerling: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** | | |

**Opdracht: de von Neumann-architectuur**

<https://www.youtube.com/watch?v=t8H6-anK0t4>

<https://www.youtube.com/watch?v=Z5JC9Ve1sfI>

Geef drie voorbeelden van fixed program computers vandaag de dag:

* Wasmachines
* Rekenmachines
* Microgolfovens

Een pc is geen fixed program computer, maar een **stored program computer** , waarbij zowel programma's als gegevens in het **geheugen** bewaard worden.

Het concept van de stored program computer, zoals dat vandaag algemeen in gebruik is, werd bedacht door **John von Neumann** in 1945, en heet daarom de **von Neumann-architectuur**.

Voordien werden in een computergeheugen enkel gegevens bewaard, geen programma's. Het programmeren van een computer gebeurde door de juiste verbindingen te maken via een massa kabeltjes.

Afbeelding met gebouw, kleding, schoeisel, buitenshuis

Automatisch gegenereerde beschrijving

Belangrijke componenten van de von Neumann-architectuur:

De **CPU** of centrale verwerkingseenheid, die bestaat uit de volgende componenten:

* De CU (voluit: **Control Unit**) die signalen zendt om het transport van gegevens regelen.
* De ALU (voluit: **Arithmetic Logic Unit**) voor berekeningen en logische beslissingen.
* De **cache**, die zorgt voor snelle toegang tot vaak gebruikte **gegevens** en **instructies**.
* 4 processor-registers (kleine maar snelle geheugens): namelijk:
  + PC (voluit: **Program Counter**) bevat het adres van de volgende instructie in het geheugen;
  + MAR (voluit: **Memory Address Register**) bevat het adres waar een gegeven moet worden opgehaald of bewaard.
  + MDR (voluit: **Memory Data Register**) bevat het gegeven dat opgehaald werd uit het geheugen, of bewaard dient te worden in het geheugen.
  + ACC (voluit: **Accumulator**) bevat het resultaat van berekeningen.

De Von Neumann-cyclus is het centrale mechanisme achter de werking een von Neumann-computer, en bestaat uit de voortdurende herhaling van drie opeenvolgende stappen:

\* stap 1: Instruction Fetch : Een instructie wordt als een rij bits ingelezen uit het werkgeheugen van de computer.

\* stap 2: Instruction Decode : De ingelezen instructie wordt gedecodeerd in de werkelijke processor-instructie en de eventuele parameters die aan de instructie meegegeven worden.

\* stap 3: Instruction Execute : De gedecodeerde instructie wordt uitgevoerd.

Een nadeel van de Van Neumann-architectuur is de zogenaamde von Neumann-flessenhals (von Neumann Bottleneck): doordat programma's en gegevens langs dezelfde systeembus tussen het **werkgeheugen** en de **processor** vervoerd worden, kunnen programma's en gegevens nooit tegelijkertijd tussen het geheugen en de processor vervoerd worden, wat voor een vertraging zorgt.

Er bestaat inmiddels een architectuur waarbij programma's en gegevens wel langs aparte bussen vervoerd worden, de zogenaamde **Harvard-architectuur**.