

Informatica Deel III

Hoofdstuk 2

De Von Neumann-architectuur

2.1. Organisatie.

De overgrote meerderheid der digitale computers zijn georganiseerd zoals weergegeven in fig. 2.1. Deze organisatie wordt dikwijls "Von Neumann" architectuur genoemd, naar de naam van de Amerikaanse wiskundige die nauw betrokken was bij het bouwen van één van de eerste computers, en deze uitvoerig in wetenschappelijke verslagen heeft beschreven.

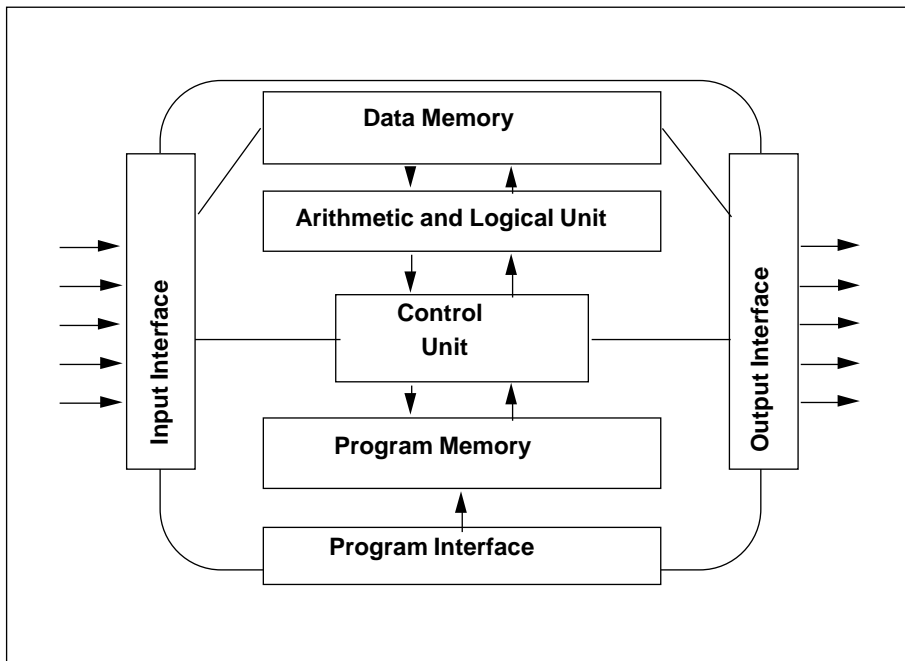


Fig. 2.1. Functional diagram of a Von-Neumann Computer.

Deze architectuur bepaalt de logische opdeling van de *centrale verwerkingseenheid*: de "Central Processing Unit" of CPU. De CPU is het hart van de computer en is verantwoordelijk voor de controle van de interne werking van de machine.

2.1.1. De in- en uitvoer omzetters (input and output interfaces).

De interactie van de CPU met de buitenwereld bestaat uit informatie die de processor binnenkomt (input) en informatie die de processor verlaat (output).

De invoer-, uitvoer- en programma-omzetters (in het Engels: Input - Output equipment) zijn apparaten die de nodige omzettingen verrichten tussen het formaat dat binnen in de machine (getallen) en er buiten wordt gebruikt om informatie voor te stellen. De aard van deze omzetters hangt natuurlijk af van de toepassing. Bij administratieve en wetenschappelijke toepassingen gebeurde vroeger de invoer op indirecte wijze, van toetsenbord naar ponskaart, en dan via een kaartlezer naar de computer. De uitvoer gebeurde exclusief via afdrukeenheden.

Vandaag gebeurt de invoer rechtstreeks via toetsenborden, "muizen", optische "bar code" lezers, enz. Voor de uitvoer van informatie heeft het beeldscherm een belangrijke plaats ingenomen naast de afdrukeenheden. De banksector heeft aan deze randapparatuur bankkaartkassa's voor grootwarenhuizen en benzinestations alsook bankbiljetautomaten toegevoegd. De in- en uitvoeromzetters eigen aan de industriële wereld, noemt men *sensoren* en *actuators*.

Sensoren laten toe quasi alle fysische grootheden binnen het meetbereik van computers te brengen, zij bevatten mechanische schakelaars, fotocellen, temperatuurgevoelige weerstanden, enz. Nieuwe sensoren zijn bvb de bewegingssensor van de Wii of iPad of de dieptesensor van Microsoft's Kinect. Deze laatste meet de afstand van elk object tot de camera. Terwijl een gewone camera een 2D-projectie geeft, krijg je hier een beeld met de afstand van elke pixel tot de camera. Ons eigen spinoff-bedrijf SoftKinetic heeft een eigen dieptesensor gemaakt (de 'DepthSense camera') die vele voordelen heeft t.o.v. de Kinect. Dit was het resultaat van onderzoek binnen ons electronicadepartement.

Actuators geven aan computers de mogelijkheid rechtstreeks hun fysische omgeving te beïnvloeden, zij maken gebruik van elektrische lampen, motoren, elektromagneten, verwarmingselementen, enz.

2.1.2. Het gegevensgeheugen. (data memory)

Het gegevensgeheugen wordt gebruikt

- om informatie -afkomstig van de invoeromzetter- te bewaren tot wanneer die

- informatie verwerkt kan worden,
- om resultaten te bewaren tot ze naar de uitvoeromzetter gezonden kunnen worden
 - om eventuele tussenresultaten van bewerkingen bij te houden.

Principieel is het geheugen opgebouwd uit genummerde cellen die elk een binair getal met een vast aantal binaire cijfers kunnen bevatten. Om historische redenen is de basiseenheid 1 byte (8 bits). Een geheugencel noemt men dikwijls "*woord*" terwijl het nummer van de cel het "*adres*" genoemd wordt (fig. 2.2).

0000 11000101	0001 10010010	0010 11111111	0011 11100010
0100 00010111	0101 11000101	0110 10000001	0111 00010001
1000 11111111	1001 00011100	1010 10101010	1011 11110101
1100 00001111	1101 11000001	1110 00000000	1111 11001100

Fig. 2.2. Central Memory Organisation.

Zoals we zagen gebruiken integers en floats 4 bytes, dus 4 geheugencellen.

In beschrijvingen van programma's worden woorden uit het gegevensgeheugen dikwijls *variabelen* genoemd, en in plaats van naar de variabelen te verwijzen door middel van hun adres geeft men hen namen die meestal een verwijzing naar de betekenis van de variabele vormen (in plaats van '*het getal op adres 1001100011*' zegt men bv. '*maandwedde*').

2.1.3. De rekeneenheid. (Arithmetic and Logical Unit)

De ALU is het apparaat dat instaat voor de feitelijke verwerking van de informatie. Het is in staat rekenkundige bewerkingen uit te voeren, getallen te vergelijken, enz. De ALU krijgt de operanden (de te bewerken gegevens) uit het gegevensgeheugen en plaatst de resultaten van de bewerkingen er terug in.

2.1.4. Het programmeergeheugen. (program memory)

Het programmeergeheugen bevat al de instructies die één voor één zullen uitgevoerd worden. Deze instructies worden op voorhand in het programma geheugen geladen via de programma omzetter. Het programmeergeheugen is georganiseerd zoals het geheugen voor gegevens.

In veel computers vindt men maar één enkel geheugen: het programma- en het gegevensgeheugen zijn fysisch samengesmolten. Beide gaan immers om het bijhouden van bytes, een programma is ook een collectie bytes. Deze samensmelting laat toe om met éénzelfde type van computer zowel grote programma's met maar weinig gegevens, als kleine programma's met veel gegevens te verwerken. Vanzelfsprekend blijven beide geheugens *logisch* gescheiden: de ene wordt enkel gebruikt voor data, de andere enkel voor het opslaan van het programma. Er wordt voor gezorgd dat er geen data in het programmeerdeelte geschreven kan worden en omgekeerd. Dit zou namelijk de bestaande inhoud (verkeerdelijk) overschrijven.

Oorspronkelijk werd het programma via de programma-interface binnen gehaald (zie schema), bijvoorbeeld toen het programma van ponskaarten werd gelezen. Nu dat het programma ook maar een file is, hoeft er geen aparte interface voor het programma zijn.

2.1.5. Stuureenheid. (control unit)

De stuureenheid leest één voor één instructies uit het programmeergeheugen, decodeert ze en zendt de nodige stuursignalen naar de verschillende onderdelen van de computer om de uitvoering van de instructies te verzekeren. Zo wordt de operatie samen met de adressen van de operanden en het resultaat aan de ALU doorgegeven. De ALU leest de data uit het geheugen (adrv de opgegeven adressen), voert de berekening uit en schrijft het resultaat terug in het geheugen.

De kloksnelheid bepaalt in grote mate de snelheid waarmee de instructies in de processor worden uitgevoerd. De klokfrequentie wordt geregeld door een chip op het moederbord, de klokgenerator, waarin zich een kwartskristal bevindt. Die wordt door middel van een elektronische schakeling aan het trillen gebracht. Elke trilling betekent een impuls voor een volgende stap in de verwerking binnen de processor. De huidige chips halen een kloksnelheid van 4GHz, d.w.z. 4 miljard pulsen per seconde.

2.2 Instructies

Principieel kan men twee soorten instructies onderscheiden: *informatieverwerkende instructies* en *stuurinstructies* (fig. 2.3.).

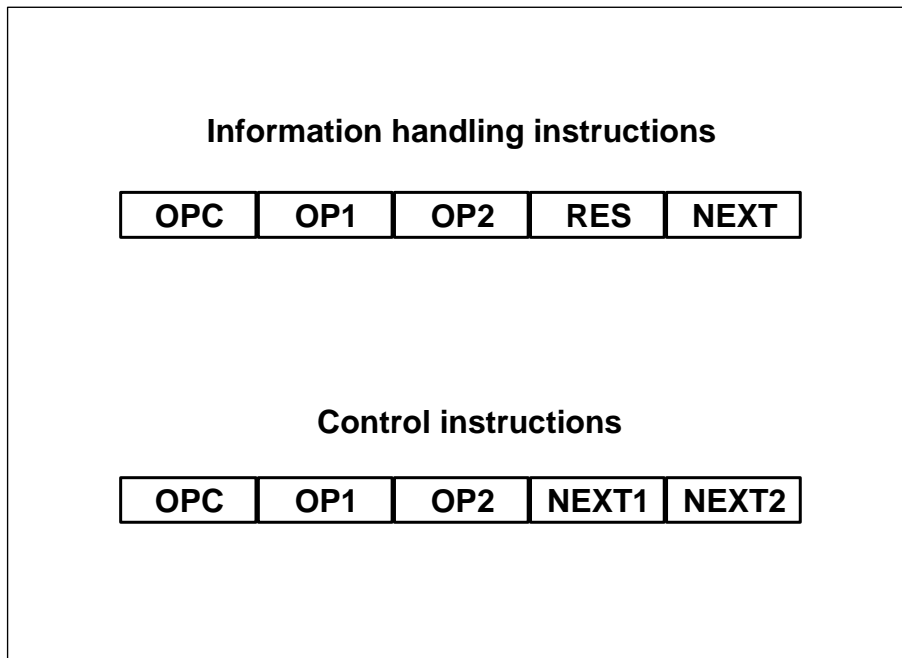


Fig.2.3. Instruction formats.

Een *informatieverwerkende instructie* bevat vijf verschillende elementen:

- OPeration Code: de aard van de bewerking die door de ALU moet verricht worden (bv. 'tel op').
- OPerand addresses: de adressen van de twee woorden in het gehevensgeheugen waaruit de operanden moeten gelezen worden.
- RESult address: het adres in het gehevensgeheugen van het woord waarin het resultaat van de bewerking moet geplaatst worden.
- NEXT instruction: het adres in het programmageheugen van het woord waaruit de volgende instructie moet opgehaald worden.

Een *stuurinstructie* laat toe het verloop van het programma te beïnvloeden aan de hand van de waarde van een bepaalde variabele. Dit zal het bijvoorbeeld mogelijk maken een betalingsopdracht al of niet uit te voeren naargelang het saldo van de rekening van de opdrachtgever positief of negatief is.

Stuurinstructies bevatten de adressen van twee variabelen die moeten vergeleken

worden en van twee programma-adressen waartussen, in functie van het resultaat van de vergelijking, zal gekozen worden.

Een voorbeeld van een stuurinstructie is een *if*. Een *while* resulteert ook in een stuurinstructie: afhankelijk van de test wordt de lus nogmaals uitgevoerd of beëindigd. Een *for* bevat ook een conditionele test.

In bovenstaand schema houdt elke instructie het adres van de volgende instructie bij (zoals bij een linked list). Maar dit is in feite een verkwisting want de instructies staan in volgorde van uitvoering in het geheugen. Om deze verkwisting te vermijden wordt het adres van de instructie die uitgevoerd wordt in de stuureenheid onthouden. Hiervoor gebruikt men een één woord geheugen dat men het *P-register* of de *Program Counter* noemt. De instructies worden één voor één uit het programmeergeheugen opgehaald aan de hand van de inhoud van het *P-register*. Bij het uitvoeren van informatie verwerkende instructies wordt de inhoud van het *P-register* automatisch verhoogd terwijl stuurinstructies de mogelijkheid hebben een nieuwe waarde in het *P-register* te plaatsen wanneer een sprong doorheen het programma moet gemaakt worden. Figuur 2.5. toont de resulterende instructieformaten,

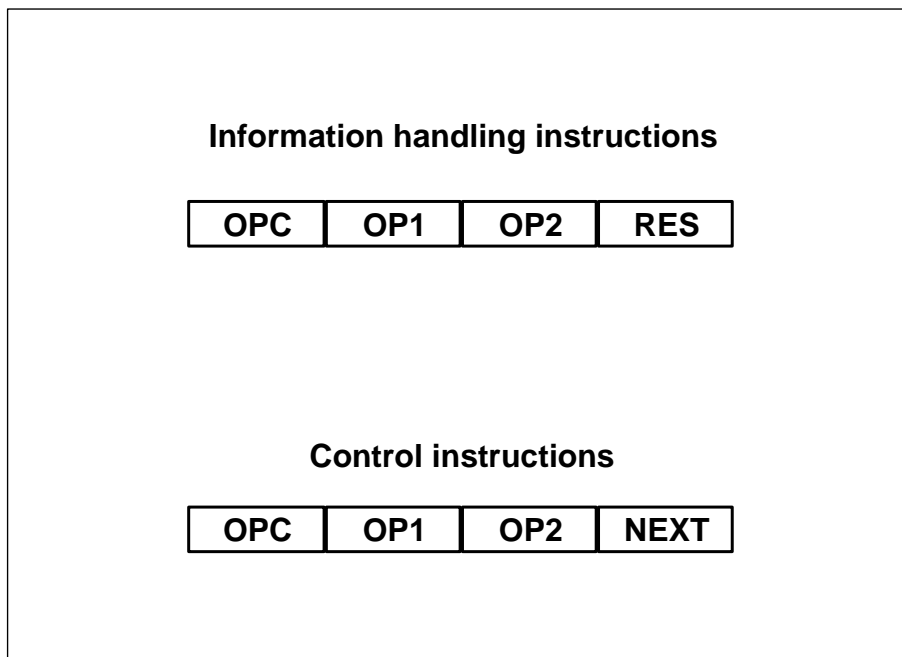


Fig. 2.5. Instruction formats with a P Register.

2.3. De geheugens.

Ondanks het feit dat in de von Neumann architectuur het computergeheugen als één enkele component wordt afgebeeld, gebruiken moderne computers een combinatie van geheugentypes, elk met z'n eigen karakteristieke prestatie en kostprijs. Het geheugen dat de snelste transfer van bits van en naar de CPU toelaat is dat type van geheugen waarbij elektrische schakelingen gebruikt worden, waarbij 1 of 0 wordt voorgesteld als de aan- of afwezigheid van spanning over een geleider.

Voorbeelden van dit type van geheugen zijn de registers, het cache geheugen en het RAM geheugen.

Registers zijn snelle geheugens (toegangstijd < 30 ns; 1 ns = 10^{-9} s) die zich in de stuureenheid en in de ALU bevinden, in het hart van de processor, zie Figuur 2.6. Zij worden gerechtvaardigd door het feit dat het resultaat van een bewerking dikwijls alleen maar als operand voor een volgende bewerking moet dienen, en het dus niet altijd nodig is dat resultaat in het geheugen te schrijven. Gelijkaardige snelle registers worden ook dikwijls aangewend om de adressen van frequent gebruikte gegevens bij te houden.

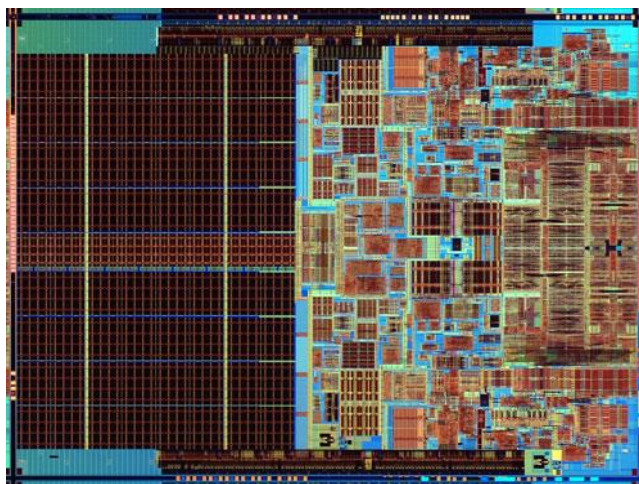
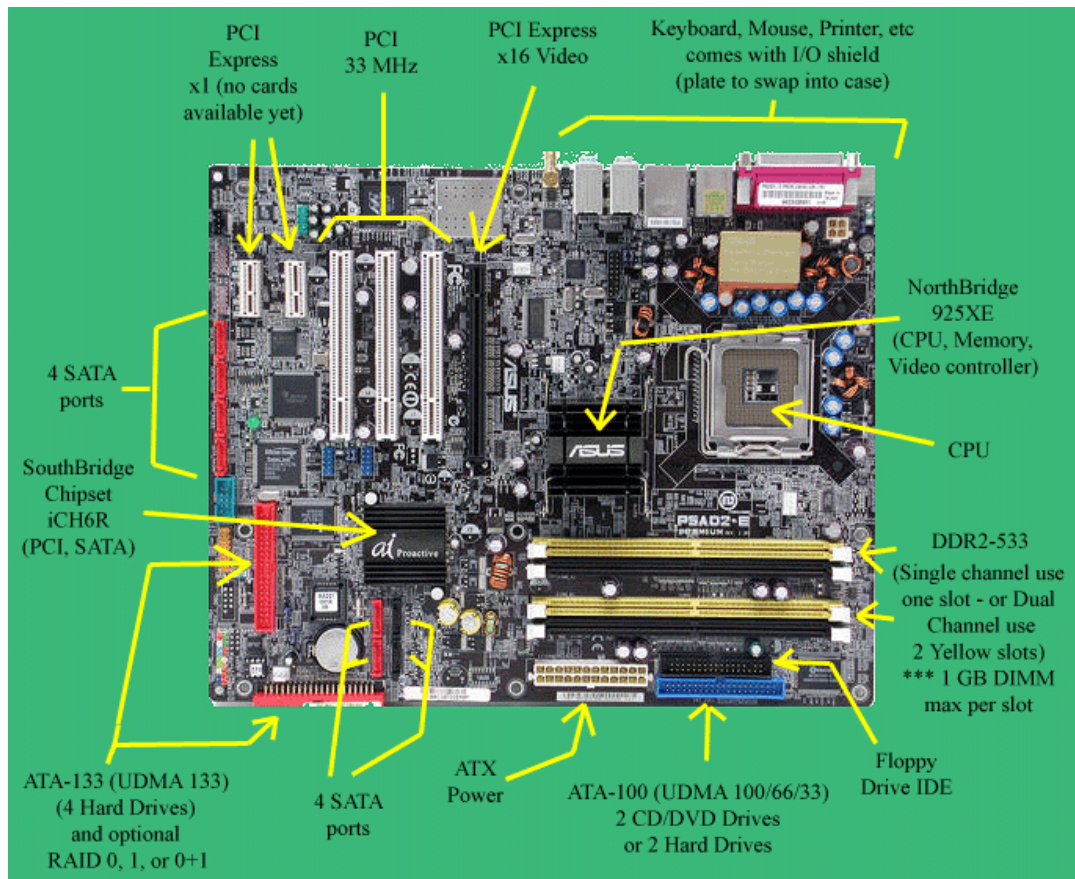


Fig. 2.6 Dual core CPU chip met aanduiding in blauw van registers. Het cache geheugen bevindt zich in het donkere linkergedeelte.

Het *cache geheugen* is typisch in de CPU chip zelf ingebouwd en gebruikt hoge snelheidsconnecties zodanig dat de data erin opgeslagen extreem snel kan aangesproken worden. Het *RAM geheugen* (afkorting van *Random Access Memory*) daarentegen is op afzonderlijke chips ondergebracht en communiceert met de CPU tegen een lagere snelheid. De data aanwezig in het RAM geheugen kan dus minder snel

aangesproken worden dan de data in het cache geheugen, maar het RAM geheugen is veel goedkoper. Men zal daarom kostprijs tegenover prestatie afwegen en een klein gedeelte (meestal uitgedrukt in kilobyte) cache geheugen voorzien voor de opslag van essentiële gegevens, terwijl men voor de overige gegevens RAM geheugen zal voorzien (meestal uitgedrukt in megabyte of gigabyte).



Figuur 2.7: Moederbord met aanduiding van de verschillende onderdelen.

In Figuur 2.7 zien we midden rechts de positie van de processor (CPU) en daaronder de DDR2- 'slots' waarin de RAM-geheugenchips komen. Deze chips zijn te zien in onderstaande figuur. Verder zien we aansluitingen voor keyboard, muis, printer en de



Figuur 2.8: Chips met RAM-geheugen die in te pluggen zijn in de DDR-slots van het moederbord.

PCI Express-slots. Deze laatste worden gebruikt om extra apparaten aan te sluiten zoals een harde schijf of grafische kaart (Graphics Processing Unit of GPU).

Er is naast het RAM geheugen ook nog een beperkt gedeelte ROM (afkorting voor *Read-only Memory*) aanwezig. Read-Only-Memory (ROM) is geheugen waarvan de inhoud permanent is, het kan enkel gelezen worden en niet overschreven. Bovendien verdwijnt de inhoud van dit geheugen niet wanneer de stroom komt weg te vallen. Typisch wordt het ROM geheugen gebruikt om permanente programma's in te stockeren zoals deze die bv. nodig zijn wanneer de computer wordt aangezet.

Het cache- en RAM geheugen vormen samen het **primaire** of **werkgeheugen** (ook wel het centrale geheugen genoemd) van de computer, waarin alle data ondergebracht is of zal worden die de CPU actief gebruikt. De data die erin opgeslagen zit kan snel door de CPU aangesproken worden. In vergelijking met andere opslagmanieren is dit geheugen echter duur. Tevens is het centrale geheugen 'vluchtig', wat wil zeggen dat een constante stroom van elektriciteit nodig is om de informatie die erin opgeslagen is te behouden. Dit wil zeggen dat als je je computer uitzet, dat alle gegevens die in het cache- en het RAM geheugen opgeslagen waren verloren zijn. Computers vereisen dus ook nog secundair geheugen, wat goedkoper is en voor permanente opslag van gegevens kan zorgen. *Secundair of perifeer geheugen* is voorhanden in verschillende vormen. Een harde schijf ("hard disk") is een metalen plaatje dat bits opslaat als gemagnetiseerde en niet-gemagnetiseerde gebieden. Deze schijf draait onder een sensor die deze gebieden detecteert en interpreteert als bits. Ondanks het feit dat dit type van geheugen een stuk trager is dan het cache en RAM geheugen, is het een veel goedkoper en blijft de data behouden zelfs als de stroom naar het apparaat uitgeschakeld wordt. Een harde schijf kan zeer grote hoeveelheden informatie permanent opslaan (meestal uitgedrukt in GigaByte of recentelijk TeraBytes); deze gegevens kunnen, indien nodig, naar het werkgeheugen overgebracht worden.

De trade-off tussen performantie en kost is afgebeeld in Fig. 2.9. Aangezien de prijs van geheugens sterk afhankelijk is van de toegangstijd (dat is de tijd nodig om een gememoriseerde informatie terug te vinden) wordt het geheugen meestal fysisch opgesplitst in delen met verschillende eigenschappen: duur, zeer snel geheugen voor wat snel bereikbaar moet zijn, en traag, goedkoper geheugen voor wat niet onmiddellijk nodig is. De snelle technologieën worden gebruikt voor registers en het werkgeheugen,

terwijl de goedkope technologieën aangewend worden in periferische geheugens.

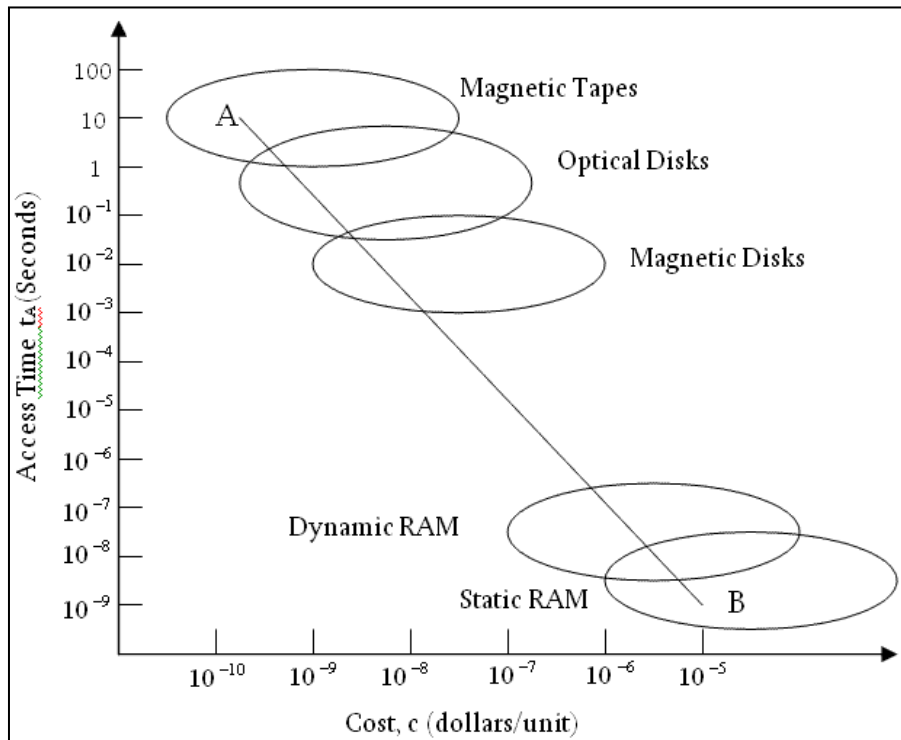
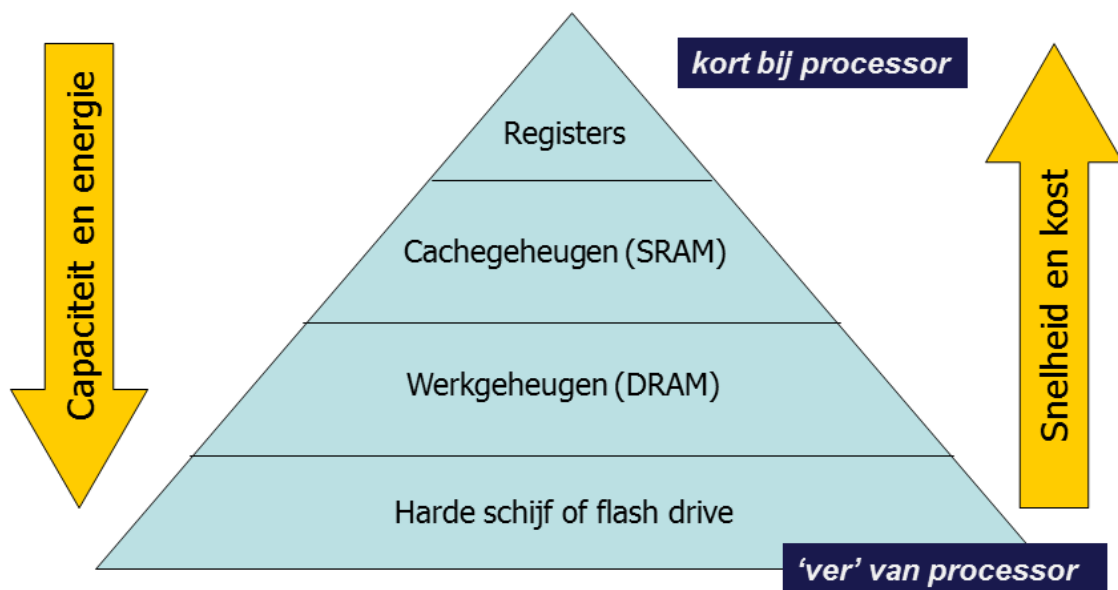


Fig. 2.9. Memory Access-Time vs. Cost

Omdat we van de duurdere geheugens minder plaatsen in de computer, kunnen we deze voorstellen als een piramide, die de hiërarchie aangeeft.



Merk op dat de snelle geheugens kort bij de processor staan, terwijl de trage verder mogen staan. Het cache geheugen staat op de processorchip (Figuur 2.6 – donkere linkergedeelte) terwijl het werkgeheugen als een apart chip (Figuur 2.8) op het moederbord wordt toegevoegd (Figuur 2.7).