

Hoofdstuk 5

Apparatuur

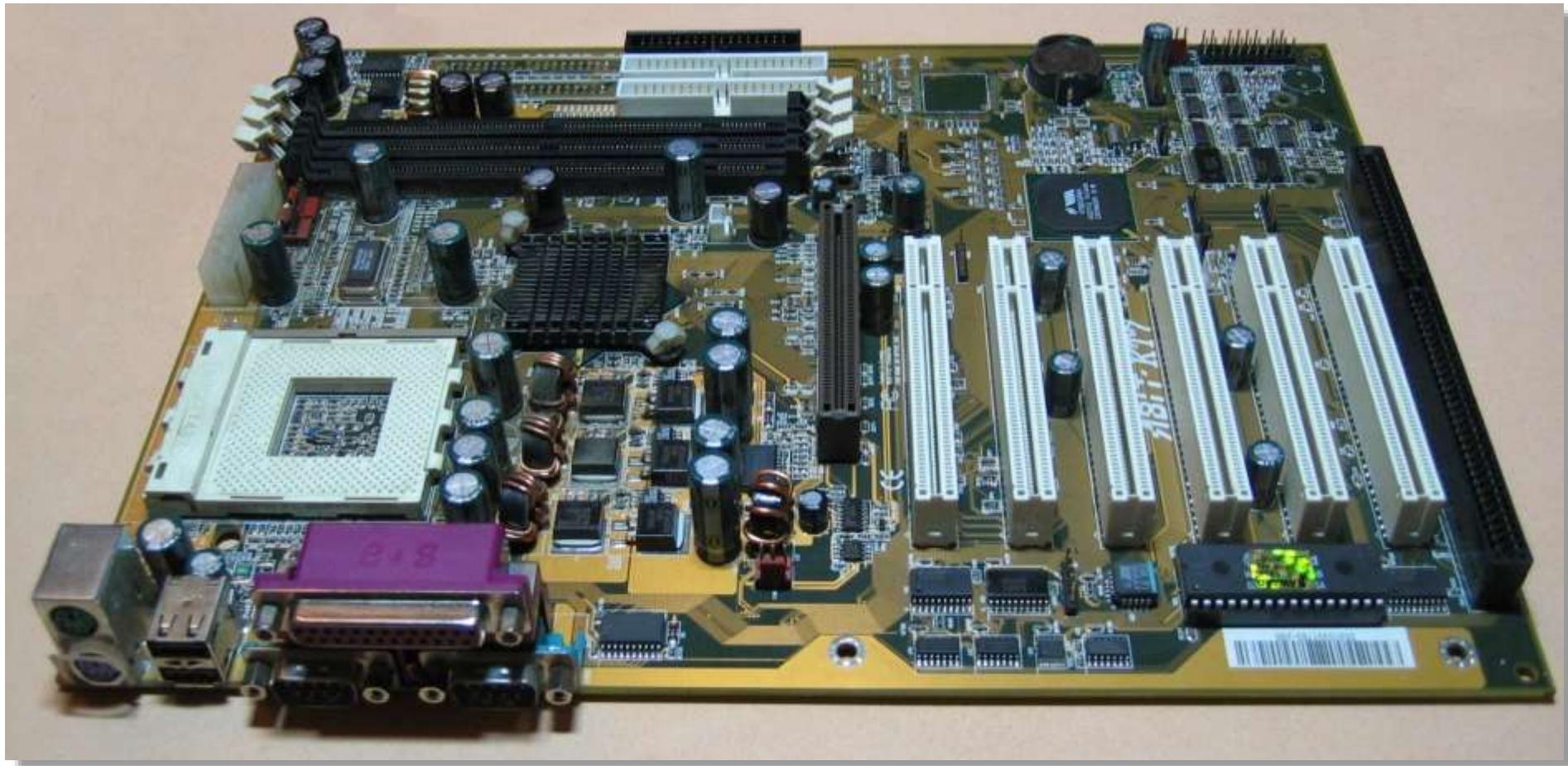


H5: Apparatuur

- Moederbord
- Chipset
- Bussen
- Processor
- Grafische Verwerkingseenheid
- Werkgeheugen
- Magneetschijven

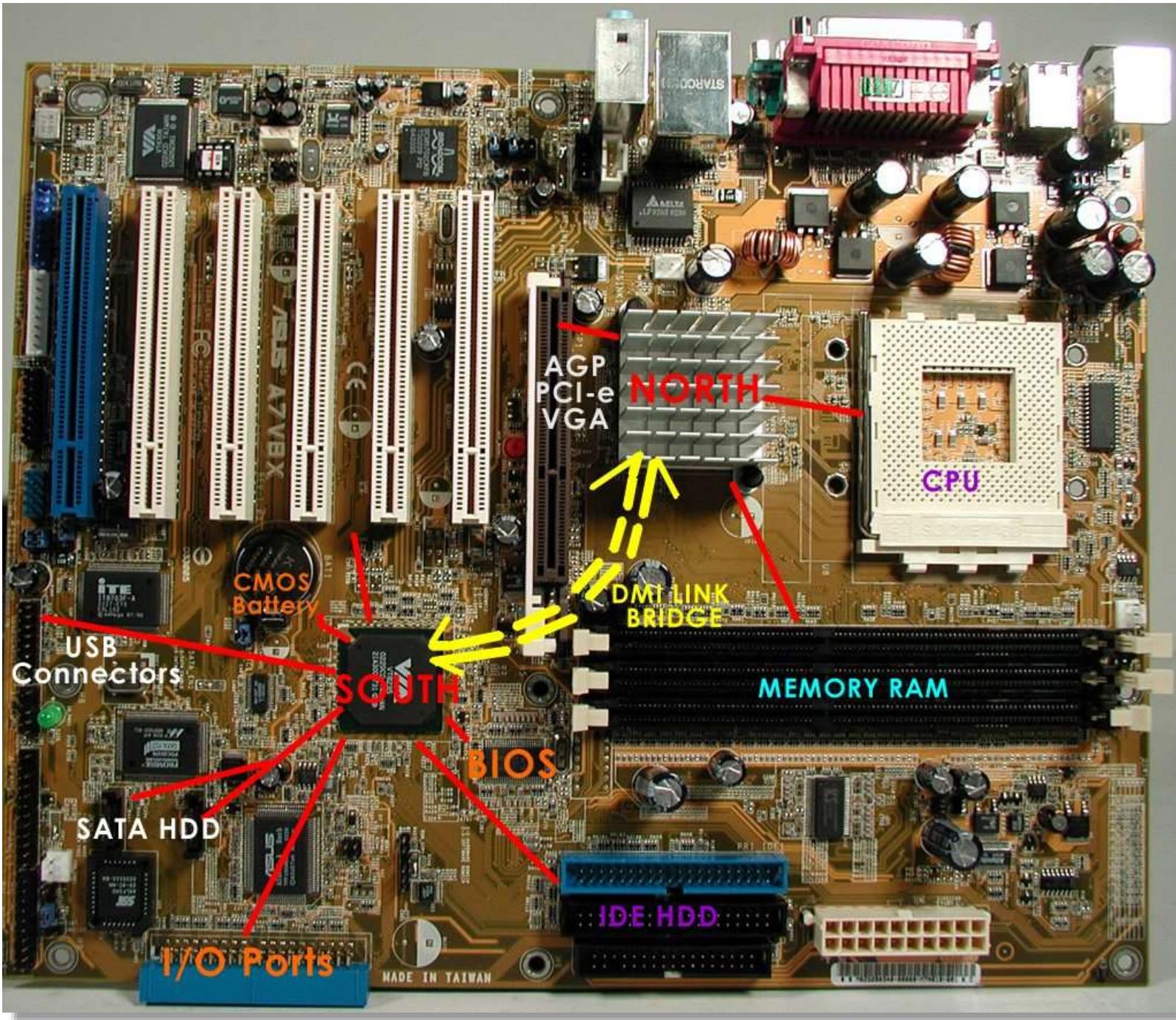
Wat staat waar op een moederbord?

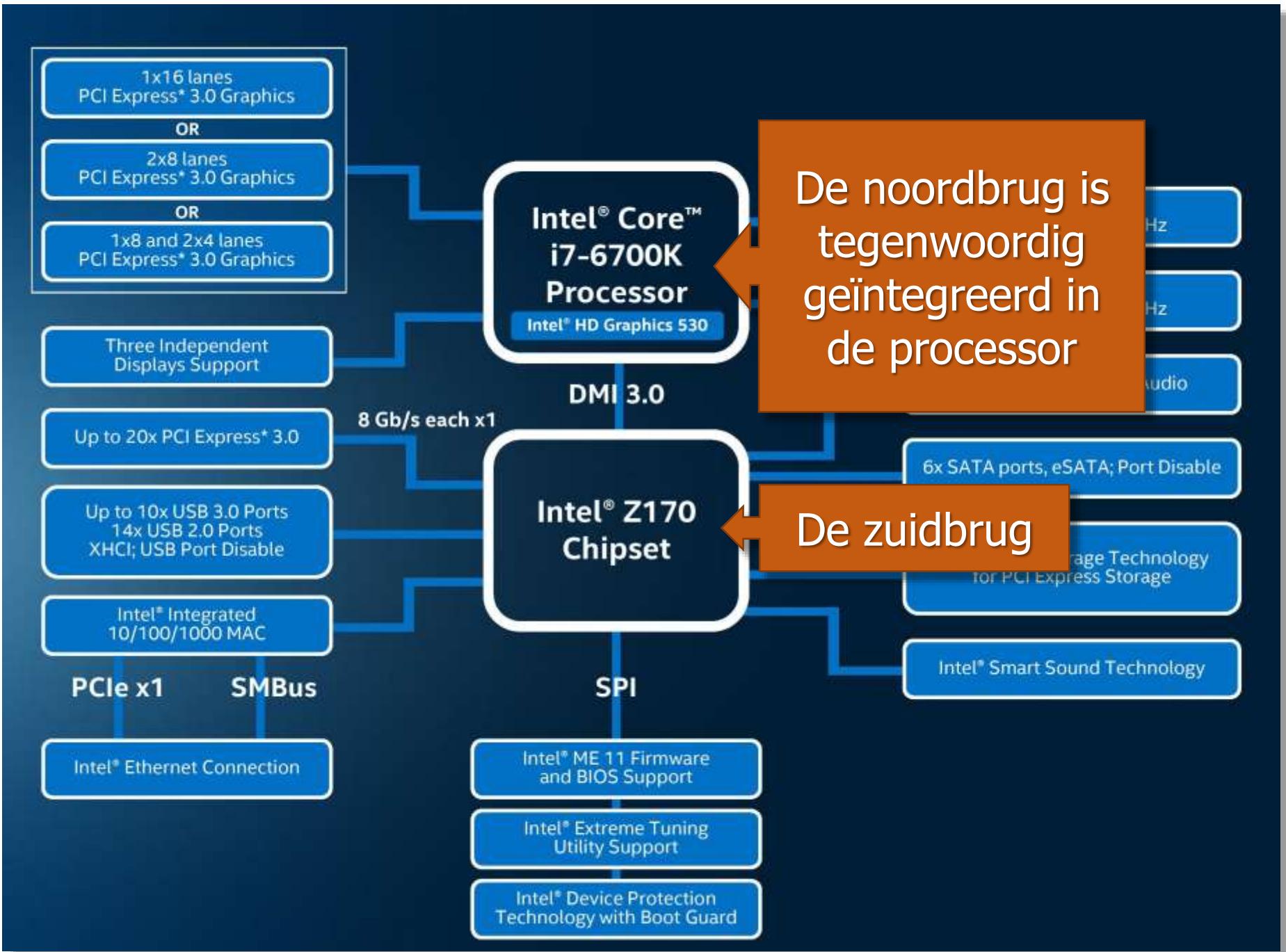
- Elektronische componenten (vb. weerstanden, condensatoren, ...)
- Connectoren, socketten, ...
- Elektrische verbindingen (achterkant)



- Onderdelen van een computer sturen bits naar elkaar
 - Cpu ↔ werkgeheugen
 - Werkgeheugen ↔ harde schijf
 - Netwerkkaart ↔ werkgeheugen
 - ...
- Lang geleden: via **busverbinding**, d.i.
 - alle onderdelen verbonden via één gemeenschappelijke verbinding
 - gebruik om beurten

- Lang geleden: via busverbinding
- Beter: via brug-ic's, de zgn. **chipset**
 - Noordbrug voor de snelle apparaten (processor, geheugen, grafische kaart)
 - Zuidbrug voor de tragere apparaten

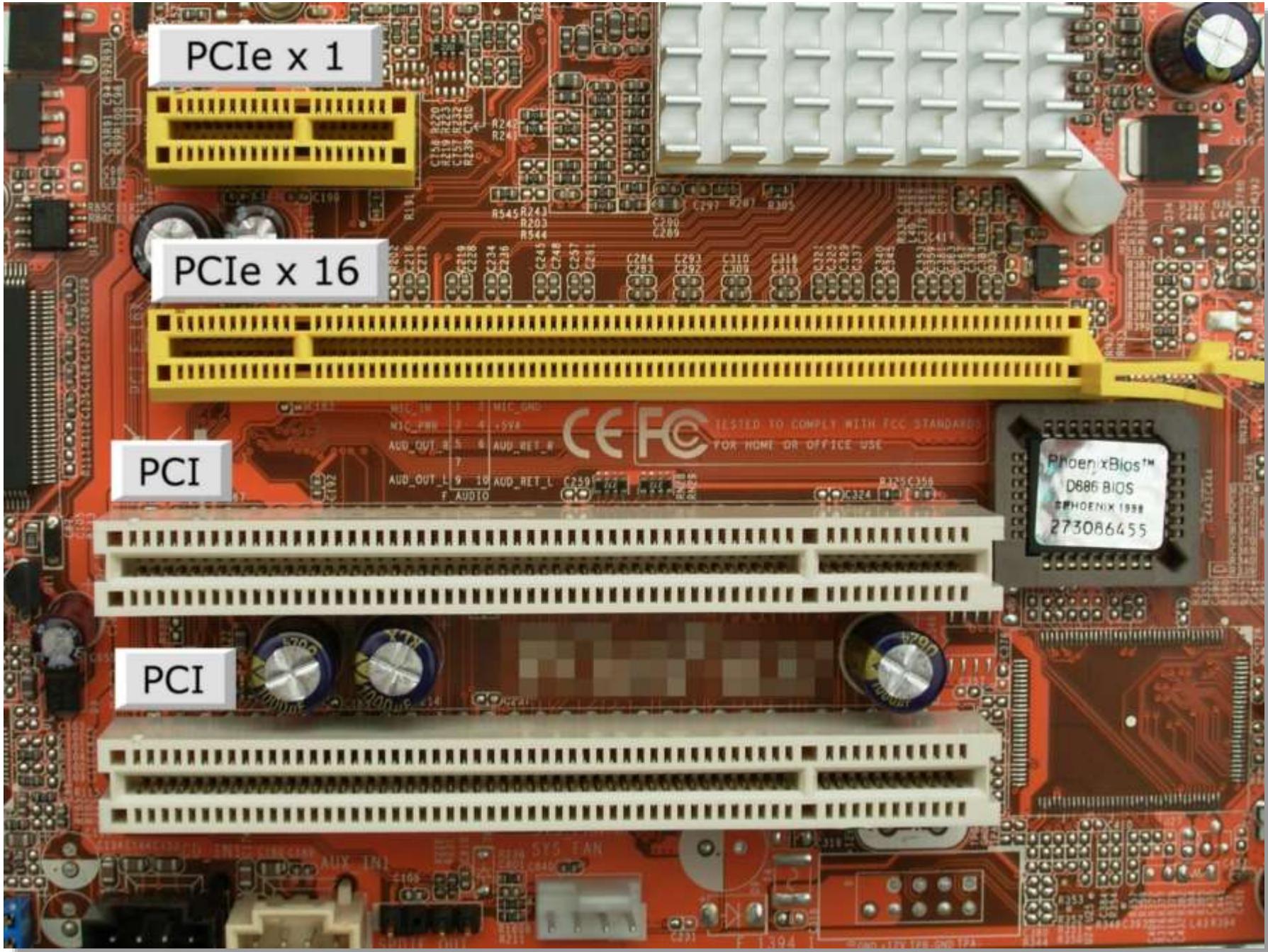




- Eenheden direct verbinden: snel maar duur
- **Bus** = meer dan 2 eenheden gebruiken zelfde transportmiddel: trager maar goedkoper.
 - Vroeger: alle verbindingen in pc via één bus

Opgelet: *Men blijft woord bus gebruiken, ook voor de directe verbinding tussen 2 apparaten*





Buseigenschappen

- Breedte
- Cyclus, frequentie
- MT/s
- Debiet, bandbreedte

Buseigenschappen

- **Breedte** is het aantal bits dat tegelijkertijd verstuurd wordt
 - 32, 64, 128, ...
 - NIET hetzelfde als **bandbreedte**
- **Breedte**
 - Cyclus, frequentie
 - MT/s
 - Debiet, bandbreedte

Buseigenschappen

- De buslijn moet enige tijd in dezelfde toestand blijven
 - een "0" (of een "1") op de bus plaatsen duurt enige tijd
- Een **klok** regelt de timing van de bussignalen
 - bvb. "een 1 op een buslijn"
= gedurende 1 klokcyclus 5V spanning op die lijn
- De bus heeft een **cyclus**: de tijd vereist om één bit te kopiëren
 - Breedte
 - **Cyclus**, frequentie
 - MT/s
 - Debiet, bandbreedte

Buseigenschappen

- **Frequentie** = omgekeerde van cyclus
 - Frequentie = 33 MHz
cyclus: $1/33000000$ sec
= 30 ns (nanoseconden).
- **Buscyclus** van bvb. 30 ns duurt langer dan C.V.E.-cyclus
 - Bus is de flessenhals: buscyclus kan veel langer zijn dan c.v.e.-cyclus.
 - Beetje oplossing: bloktransfer (zie later)
 - indien alles via dezelfde bus: nog trager.
- Breedte
- Cyclus, **frequentie**
- MT/s
- Debiet, bandbreedte

Buseigenschappen

- **MT/s** = Megatransfers per seconde
 - = frequentie (in MHz)
 - = aantal keer dat op één buslijn iets kan gekopieerd w. per seconde
- Breedte
- Cyclus, frequentie
- **MT/s**
- Debiet, bandbreedte

Buseigenschappen

- **Debit** = aantal (Giga)byte dat per seconde kan gekopieerd worden
= **bandbreedte**, bvb. 8 Gigabyte/s
 - Breedte
 - Cyclus, frequentie
 - MT/s
 - **Debit, bandbreedte**
- Debit = MT/s × busbreedte
- Vb.: breedte = 128 bit, MT/s = 667MT/s.
Wat is het debiet?
 - $128 \text{ bit} \times 667 \text{ MT/s} = 85376 \text{ Mbit/s} = 10672 \text{ Mbyte/s} = \sim 10.4 \text{ Gbyte/s}$

Voorbeeld: de PCI bus

- Peripheral Component Interconnect
- Verbonden met de zuidbrug
- Verschillende versies, bvb. PCI-X 533
 - frequentie : 533 MHz (PCI-X 533)
 - breedte : 64 bit (8 bytes per cyclus)
 - debiet : 4,3 Gigabyte/seconde

Parallel vs. Serieel

- PCI-X 533 stuurt 64 bits per keer **parallel** door
 - Parallelle communicatie is relatief traag
 - Per set van 64 bits dat je stuurt moet je een bepaalde tijd wachten eer je zeker bent dat alle bits aangekomen zijn
- Alternatief: **seriële** communicatie
 - slechts één lijn gebruiken aan veel hogere snelheid

Voorbeeld: de PCIe bus

- PCI Express (PCIe) is een bus met een aantal bidirectionele seriële kanalen (= lane)
 - Per kanaal kan een (aparte) datastream verstuurd worden
 - PCIe 3.0: 985 MB/s per kanaal en per richting
- Wat betekent PCIe x16 (of PCIe x4, of PCIe x8)?
 - 16 lanes; het debiet is dan $16 \times 985 \text{ MB/s} \times 2$ richtingen, d.i. ~32 GB/s



H5: Apparatuur

- Moederbord
- Chipset
- Bussen
- **Processor**
- Grafische Verwerkingseenheid
- Werkgeheugen
- Magneetschijven

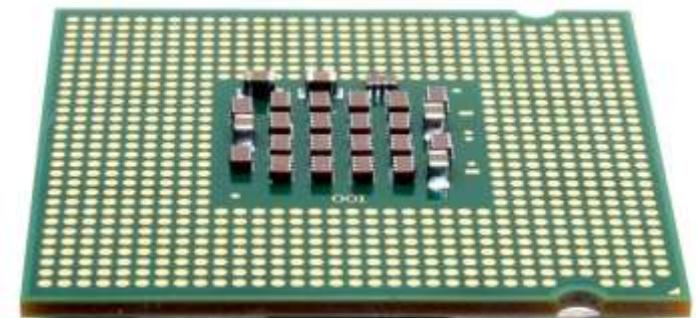
How a CPU is made



De Processor

- Wat is de functie van de processor? **Machinebevelen uitvoeren.**
- Processor is een IC (geïntegreerd circuit)
 - Bestaat uit transistoren
 - Is verbonden met de rest van de apparatuur via pinnen
 - I8086: 40
 - Nu: honderden

Tegenwoordig geen pinnen meer, maar 'pads'



De Processor

- Wat zegt de wet van Moore?
"Aantal transistoren op chip verdubbelt om de 18 à 24 maanden."
- Gevolg: processoren worden steeds complexer
 - Grotere woordlengte
 - Meer, krachtiger bevelen
 - Grottere cache op c.v.e.

De Processor

- IC heeft een matrix-structuur (rijen, kolommen)
 - Spoorbreedte: breedte van rij/kol.
 - Uitgedrukt in nanometer, bijv. 45nm, 32nm, 22nm, 14nm (anno 2015)
 - Verwachting: 5nm tegen 2021
- Transistoren van slechts enkele atomen

"In 2015 IMEC and Cadence had fabricated 5 nm test chips."

Woordlengte

- Wat is de woordlengte? **Aantal bits van de getallen waar de processor mee rekent.**
 - 8, 16, 32, 64, ...
 - Grootte van de:
 - Registers
 - Bussen in (c.v.e.) en tussen c.v.e. en werkgeheugen
 - ALU
 - ...
- Waarom is rekenen met getallen van 32 bits beter dan rekenen met getallen van 16 bits?

- Elk type van processor heeft een bevelenset
 - = al de machinebevelen die de processor kan uitvoeren
 - Hoe meer bevelen, hoe meer transistoren nodig
- Wat is het nut van meer bevelen in bevelenset?
 - Vb. 1: (IBM-mainframe) **mvc aha(25), oho**
 - Kopieer 25 bytes van adres "oho" naar adres "aha".
 - Hoe doe je dat bij Intel?

- Wat is het nut van meer bevelen in bevelenset?
- Vb. 2: rekenen met kommagetallen (floating points)
- $123,45 + 67,891 = ?$
 - **Oplossing 1:** subroutines en rekenen met integers :
 - $123,45 * 1000 + 67,891 * 1000 = \text{enz...}$
 - **Oplossing 2:** coprocessor
 - d.i. 2° processor, alleen voor reals
 - **Oplossing 3 (vanaf i486):** processor kan ook floating point bewerkingen

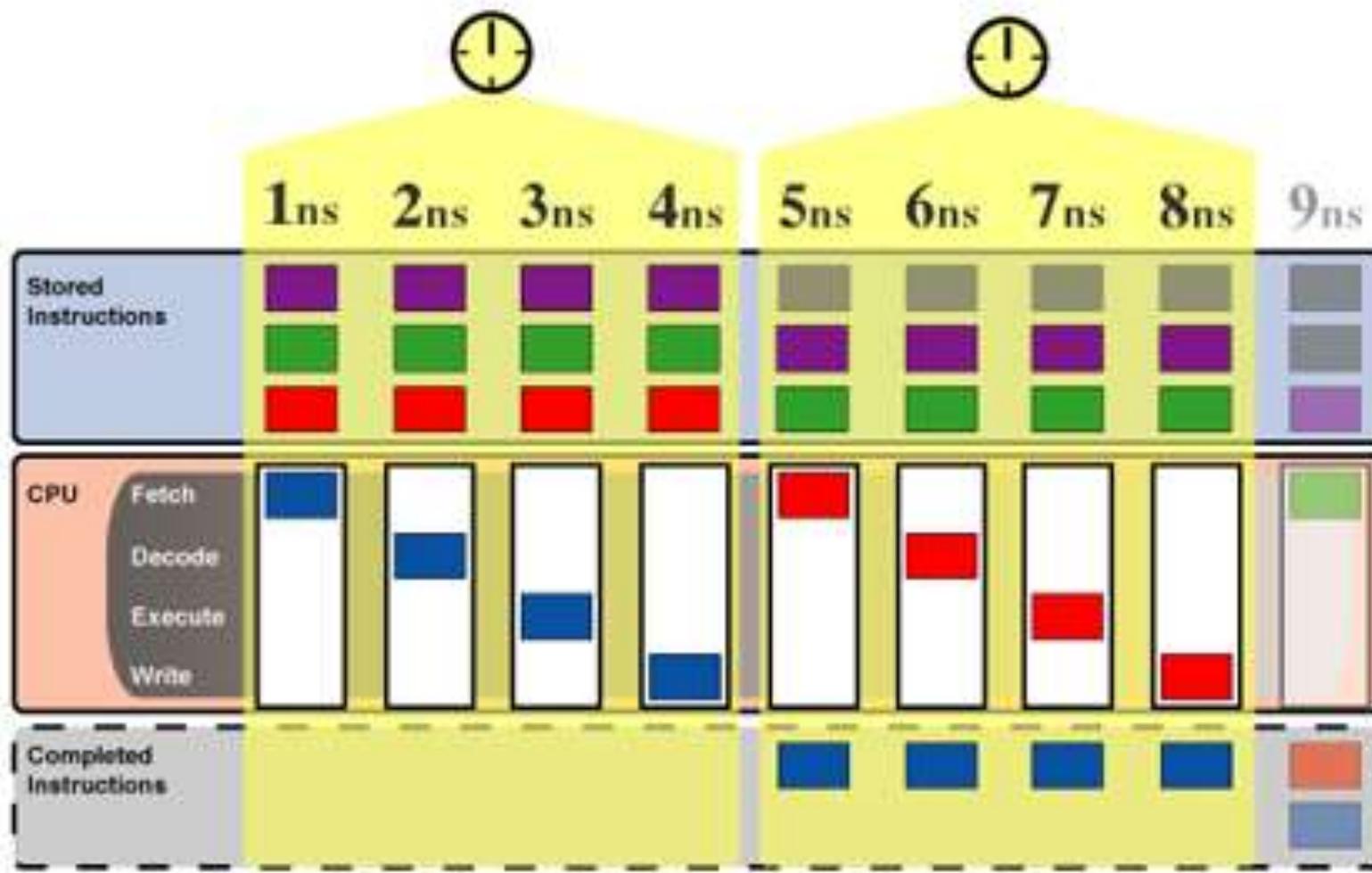
Hoe krachtig is de processor?

- Bevelen worden steeds krachtiger
- Klok gaat steeds sneller
- Hoe meten we de prestaties van de processor? **In Mips.**
 - Aantal (miljoenen) machinebevelen per seconde.
- Wat als processor A 4 bevelen nodig heeft, waar processor B er maar 1 nodig heeft?
 - **Megaflops:** miljoenen floating point operations

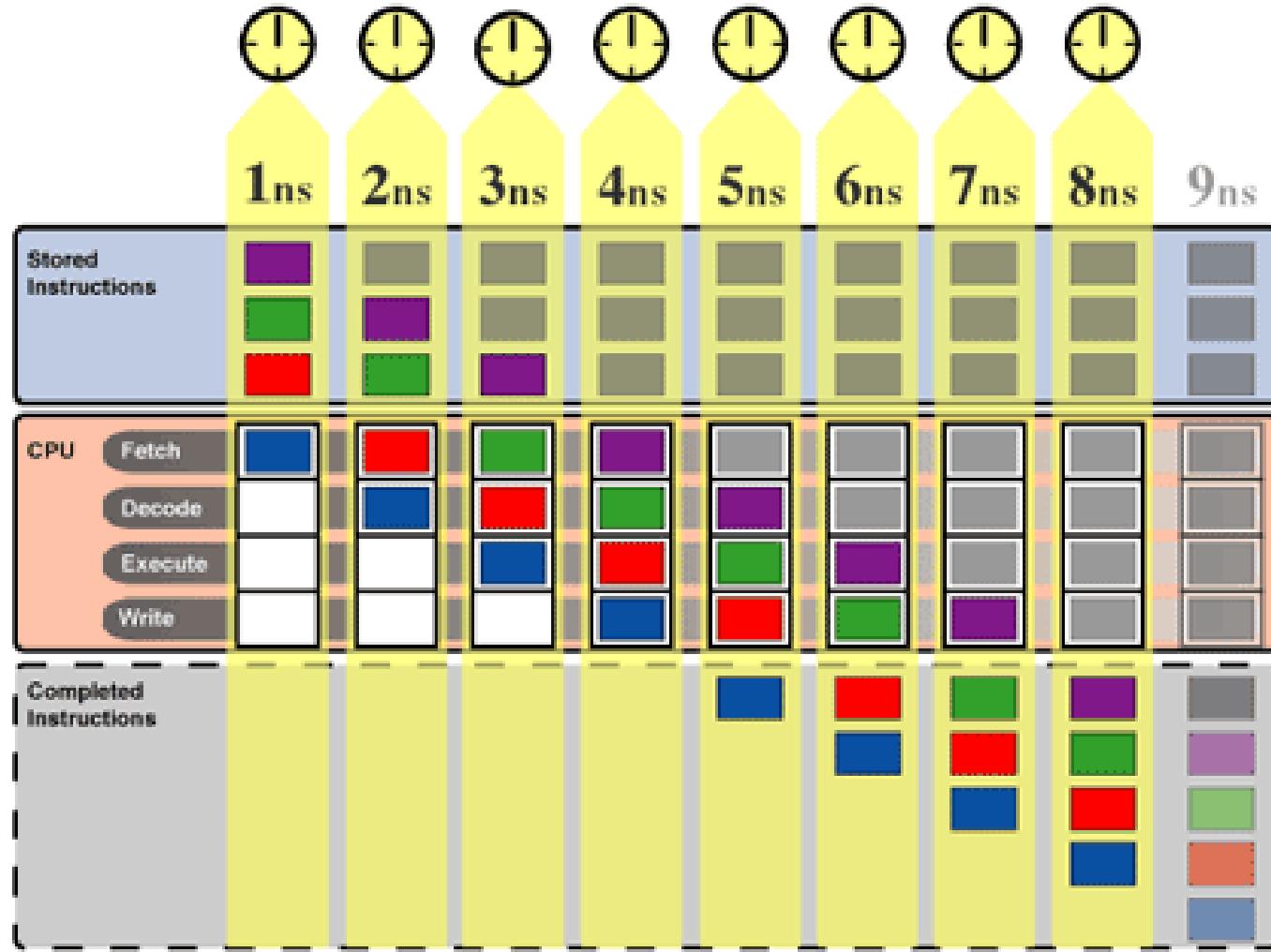
Optimalisaties

- **Bevelenwachtrij:** wat; nut?
 - Tijdens uitvoering van machinebevel : ophalen van volgend bevel
 - snelheidswinst
- **Pijplijn:** nog een stap verder
 - Uitvoering van een bevel in meerdere stappen

Geen pijplijn



Met pijplijn



Vb. MIPS pijplijn

- MIPS-pijplijn heeft 5 fases:
 1. **Instruction Fetch** – instructie ophalen uit geheugen
 2. **Instruction Decode** – instructie interpreteren
 3. **Execute** – instructie uitvoeren
 4. **Memory** – geheugentoegang (indien nodig)
 5. **Writeback** – resultaat wegschrijven
- Neem 20 bevelen. Na hoeveel cycli is het 1^e bevel afgewerkt? **Na 5 cycli.**
- Wat na cyclus 6? **Bevel 2 afgewerkt.**
- Wanneer zijn de 20 bevelen afgewerkt? **Na 24 cycli.**

Wat als een bevel het resultaat van een vorig bevel nodig heeft?

- Bvb.: `mov ebx, [aha]`
`add eax, ebx`
 1. Fetch 1
 2. Decode 1, Fetch 2
 3. Execute 1, Decode 2, ...
 4. Memory 1, **Wait 2**, ...
 5. Writeback 1, **Wait 2**, ...
 6. Execute 2, ...

Oplossing met wachtcycli werkt, maar dan is het voordeel gedeeltelijk weg.

Wat bij een voorwaardelijke sprong?

- Bvb.:
jg verder
sub eax, 13
verder: imul dword [een]

Is op te lossen door branch prediction

Wat bij een onderbreking?

- Bvb.: mov ebx, 0
 idiv ebx
 add eax, 10

Een bevel wordt volledig uitgevoerd of helemaal niet. Indien er toch iets misloopt tijdens een bevel wordt er een **roll back** uitgevoerd.

Wat bij korte/lange bevelen?

- Bvb.: `cld`
`movsb`

Niet elk bevel blijft even lang in de pijplijn. Bevelen kunnen elkaar ook inhalen in de pijplijn (**out-of-order execution**).

Besluit Pijplijn

- Vereiste voor pijplijn: alle fases "gelijkaardig"
 - Worden binnen 1 cyclus afgewerkt
- Lengte van de pijplijn varieert per processor



MIPS: 5 fases



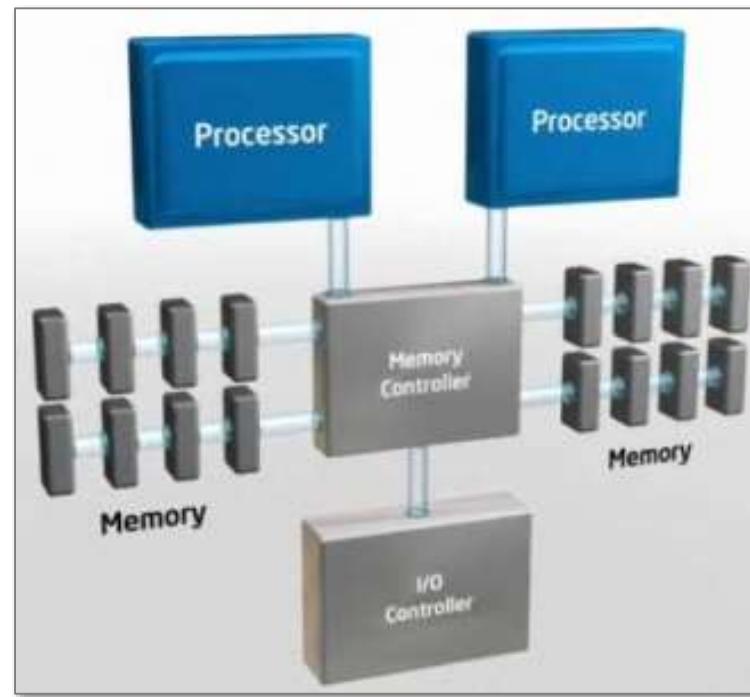
ARM7: 3 fases
ARM9: 5 fases
ARM10: 6 fases
ARM11: 8 fases

Penryn (2007): 14 fases
Nehalem (2008): 20-24 fases
Ivy Bridge (2012): 14-19 fases



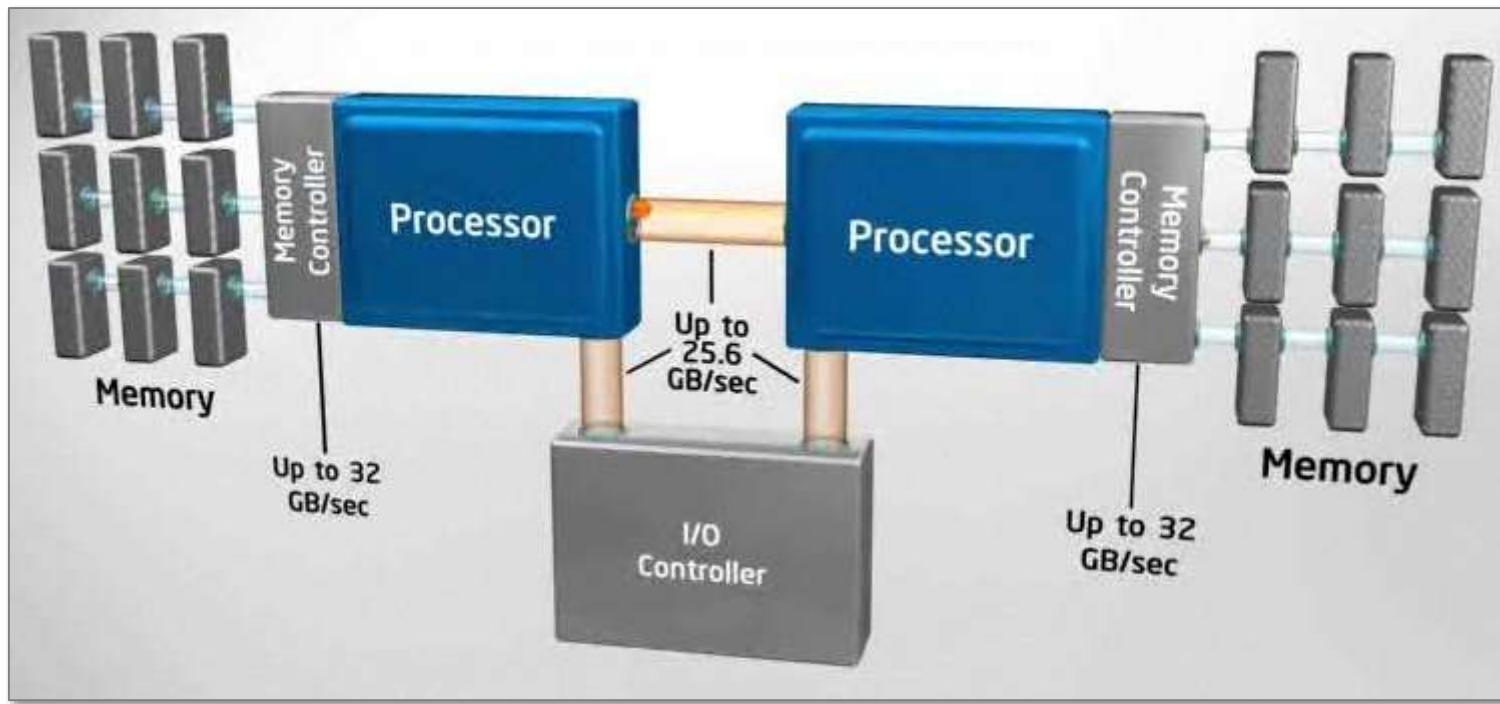
Meer dan één processor

- Hedendaagse server: meer dan één processor op het moederbord
 - Vroeger: verbonden via de Front Side Bus (FSB)



Meer dan één processor

- Hedendaagse server: meer dan één processor op het moederbord
 - Tegenwoordig: andere technologieën, bijv. Intel QuickPath



Multiprocessing

- **Multiprocessing:** "*het simultaan uitvoeren van twee of meerdere programma's op een computer met meer dan één CPU.*"
- Wat als we maar één CPU hebben? **Multiprogrammering.**
 - Het besturingssysteem laat de processor wisselen van programma (bijv. 100× per seconde)
 - Meerdere programma's kunnen dan stapsgewijs uitgevoerd worden

Nadeel Multiprogrammering

- Wisselen van programma's zorgt voor **overhead**
 - Registers e.d. moeten eerst weggeborgen worden in het geheugen, en hersteld worden vóór terugkeer
 - Op x64: 40+ registers
 - Er zijn oplossingen voor: **Hyperthreading** en **multicore**

Hyperthreading en Multicore

- **Hyperthreading**
 - Processor kan status van meerdere processen bijhouden
 - Er zijn meerdere sets van registers
- **Multicore**
 - Een stap verder dan hyperthreading
 - Er zijn meerdere sets van registers
 - Er zijn ook meerdere ALU's



H5: Apparatuur

- Moederbord
- Chipset
- Bussen
- Processor
- **Grafische Verwerkseenheid**
- Werkgeheugen
- Magneetschijven

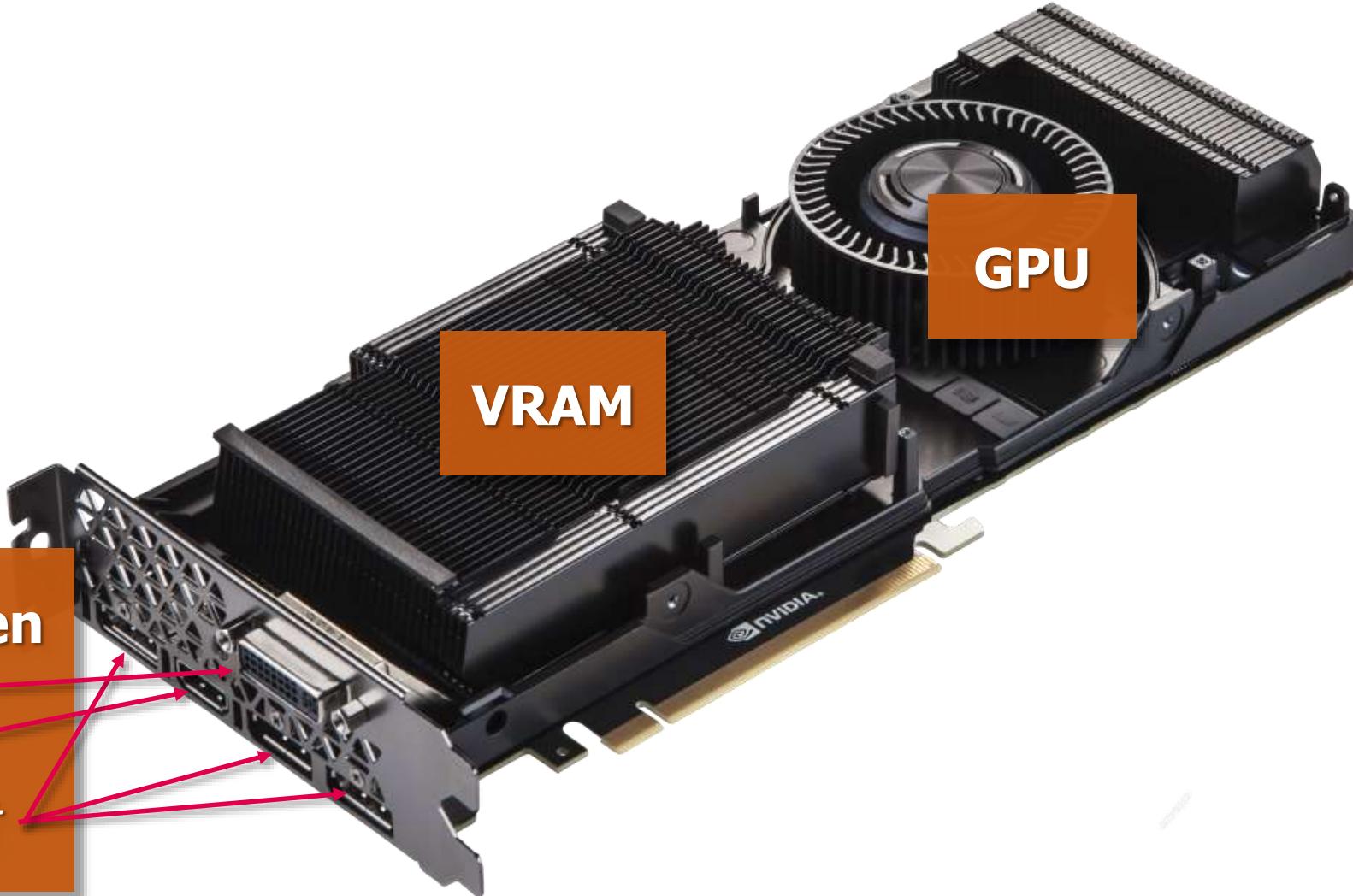
Beeldscherm

- Computerscherm bestaat uit **pixels**: puntjes met bepaalde kleur
 - Per pixel 3 bytes: **rood**, **groen** en **blauw**
 - Tegenwoordig ook 4^e byte: **opaciteit**
- Aantal pixels in de breedte en hoogte is de **resolutie**
 - Bijv. **VGA** (640×480), **SVGA** (800×600), **XGA** (1024×768), **HD Ready** (1280×720), **Full HD** (1920×1080), **4K** (4096×2160), ...

Grafische Verwerkseenheid

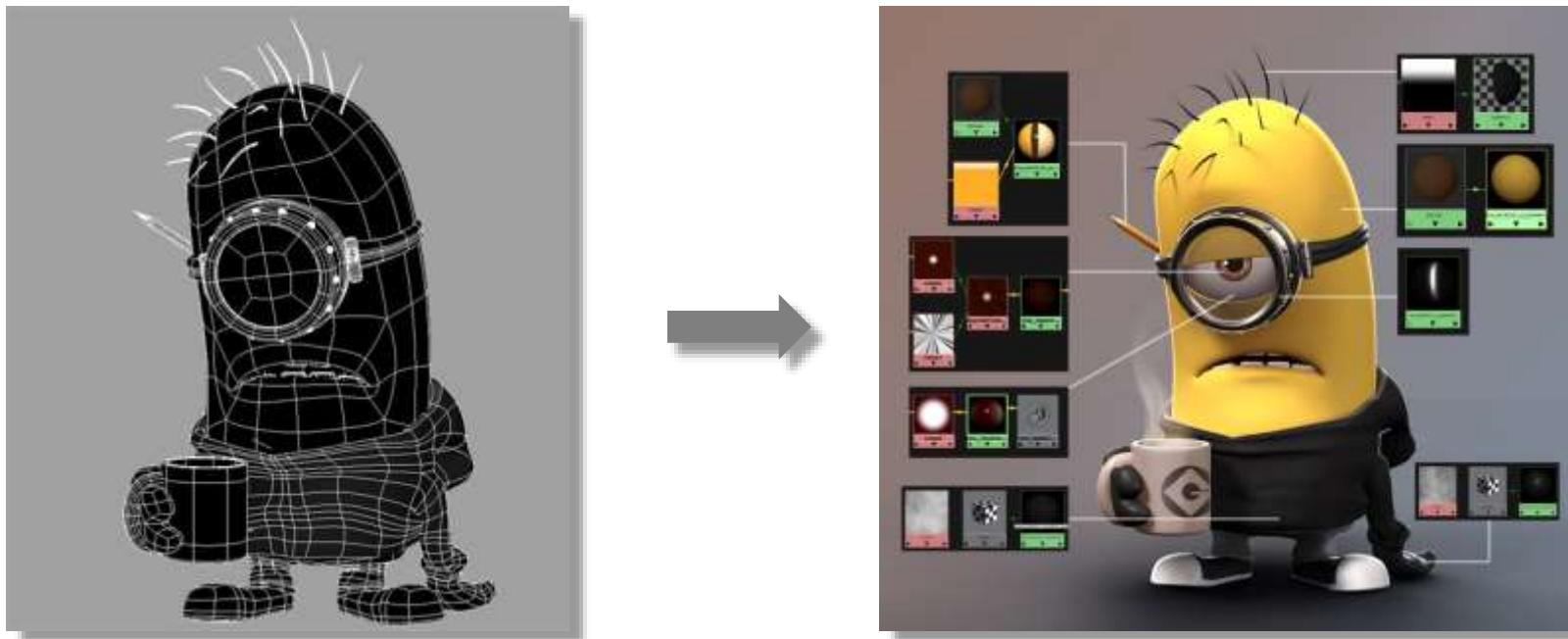
- Beeldscherm voorzien van data is de taak van de **grafische verwerkseenheid**
- Veel data:
 - Bewegende beelden minimum 25× per seconde verversen, maar typisch 60×
 - 4K-resolutie: $4096 \times 2160 \times 4 \text{ bytes} \times 60/\text{s} = 2\text{GB/s (!)}$
 - Via PCIe ×16 verbonden met de noordbrug

Grafische Verwerkseenheid



Grafische Verwerkingsseenheid

- Doet meer dan gewoon pixels tekenen
 - Video encoding/decoding
 - 3D-berekeningen
 - Texture mapping, geometrische berekeningen, polygonen renderen, ...



Grafische Verwerkseenheid

- Doet meer dan gewoon pixels tekenen
 - = Zeer veel werk
 - Daarom: zeer krachtige GPU
- Bijv. nVidia GeForce GTX Titan X
 - 3072 processor cores (!)
 - 12GB VRAM
 - 336.5 GB/s geheugen-bandbreedte



H5: Apparatuur

- Moederbord
- Chipset
- Bussen
- Processor
- Grafische Verwerkingseenheid
- Werkgeheugen
- Magneetschijven

Werkgeheugen

- Er wordt voortdurend gekopieerd van CPU naar het werkgeheugen en omgekeerd
 - Bevelen worden opgehaald uit het werkgeheugen
 - Resultaten worden weggeborgen in het werkgeheugen
 - Gegevens staan in het werkgeheugen
- CPU **leest** of **schrijft** het werkgeheugen

- Het werkgeheugen wordt bestuurd door de **geheugenbesturingseenheid (GBE of memory controller)**
 - Vroeger een chip tussen de bus en het werkgeheugen
 - Later geïntegreerd in de noordbrug
 - Nu zit noordbrug in de processor; GBE dus ook...
- GBE bevat twee registers
 - Geheugenadresregister (**GAR**)
 - Geheugenbufferregister (**GBR**)

Communicatie CPU↔WG

1. CPU berekent eerst het adres

- Bevel: inhoud EIP + basis;
- Data: basis + (verplaatsing in het bevel) (+ indexreg)

2. CPU plaatst adres in GAR

Indien lees:

- CPU geeft lees-bevel
- GBE plaatst data in GBR
- CPU kopieert naar eax, ebx, ALU, ...

Het initiatief gaat altijd uit van de CPU!

Communicatie CPU↔WG

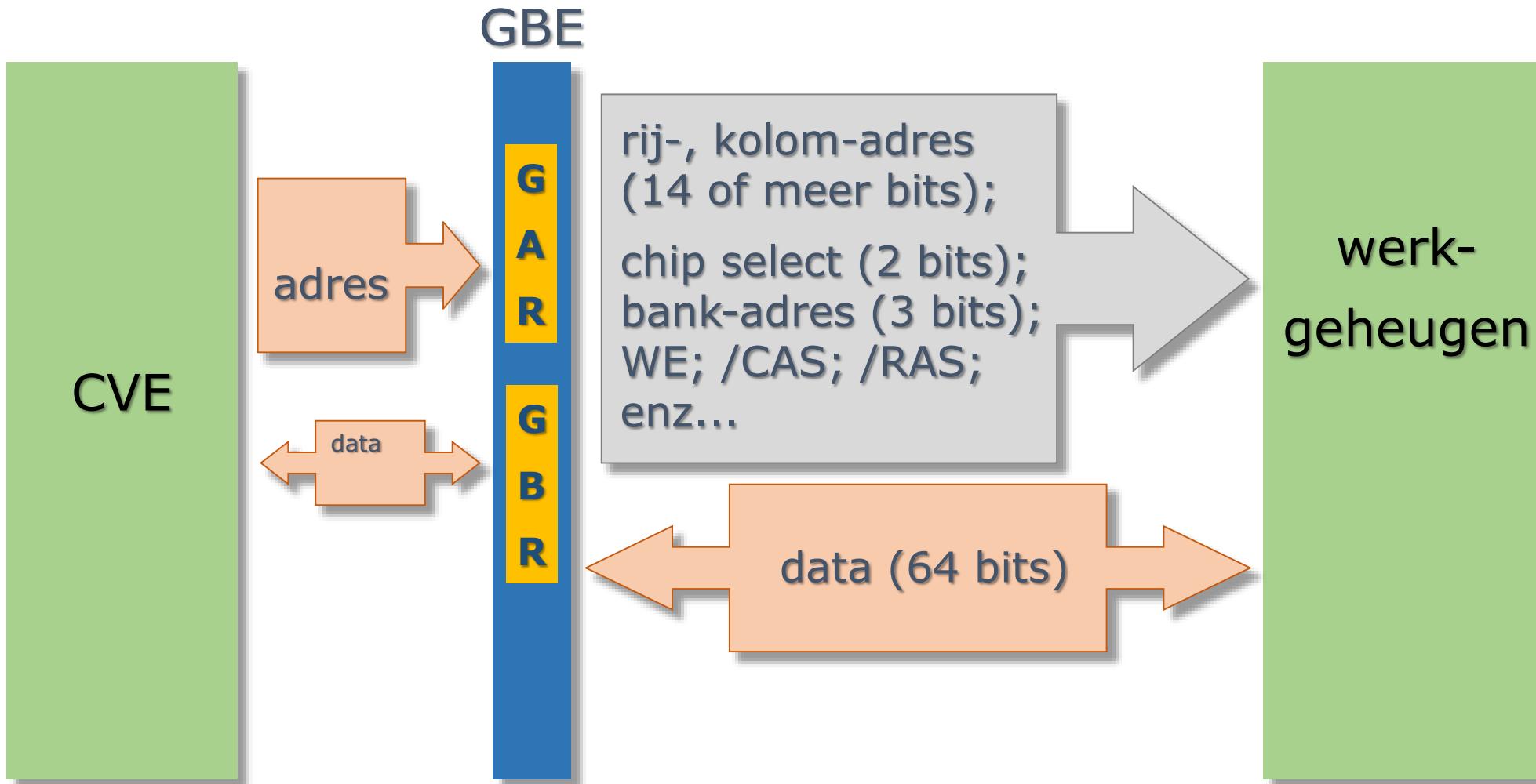
2. CPU plaatst adres in GAR

Indien schrijf:

- CPU plaatst data in GBR
 - CPU geeft schrijf-bevel
 - GBE slaat data op
-
- Per lees- of schrijfbewerking worden er altijd meerdere bytes gekopieerd (= een **woord**), bvb. 8 bytes

Ook indien maar één byte vereist is (bijv. lodsb) worden er 7 anderen meegekopieerd. Is dat goed?

Communicatie CPU↔WG



Synchroon vs asynchroon

Wanneer is lees- of schrijfbewerking klaar? 2 mogelijkheden:

Asynchroon: werkgeheugen geeft signaal indien klaar

Synchroon: na een aantal klokcycli (wordt tegenwoordig gebruikt)

Eigenschappen van WG

Toegangstijd is de tijd tussen een lees-signaal en het beschikbaar zijn van data in GBR

Cyclustijd is de duur tussen 2 bewerkingen op het geheugen (bvb. twee lees-bewerkingen)

Frequentie is het aantal klokcycli per seconde

- Bvb. frequentie = 200 MHz. Hoe lang duurt één cyclus? **5 ns.**

- S = synchroon
- DRAM = dynamic random access memory)
- Werkgeheugen is een aantal I.C.'s (chips), bestuurd door G.B.E.
 - CPU levert adres aan G.B.E. en lees-signaal
 - – of – CPU levert adres, data aan G.B.E. en schrijf-signaal
 - G.B.E. doet de rest.

- SDRAM bestaat uit modules, bvb. DIMM's (dual in line memory module)
- Geheugenmodule (DIMM) bestaat uit geheugen-I.C.'s
- I.C.'s bestaan uit transistoren

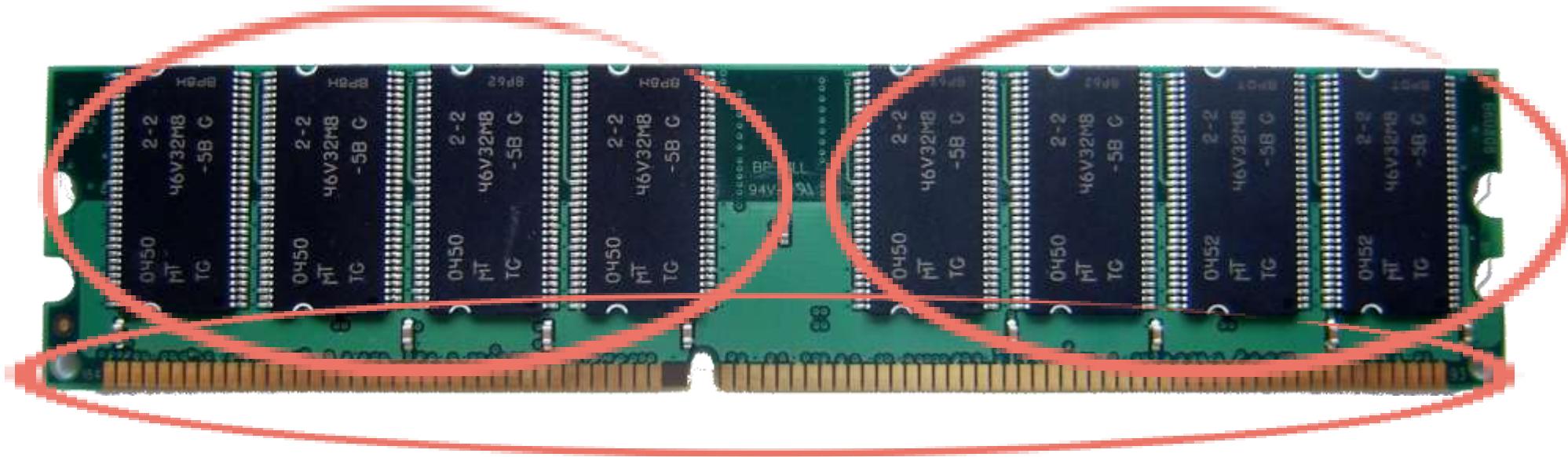


Serveropstelling



24 slots (=384GB RAM)

- Heeft typisch 8 of 16 IC's
- Dubbele rij contactpunten (voor en achter)
 - Daarom "double in line memory module" (t.o.v. SIMM)



- 168 tot 288 contactpunten
 - een aantal voor data overdracht dienen
 - Andere voor adres (zie verder)
 - Nog andere voor besturingssignalen

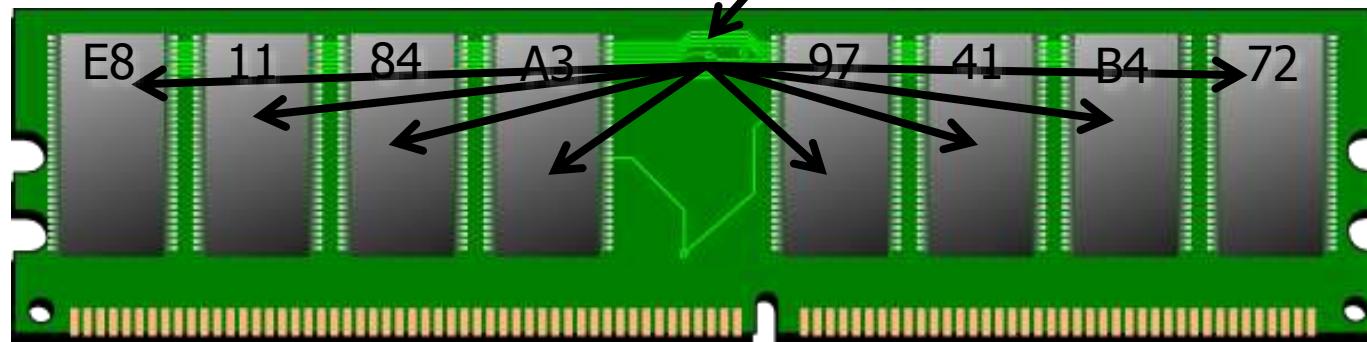
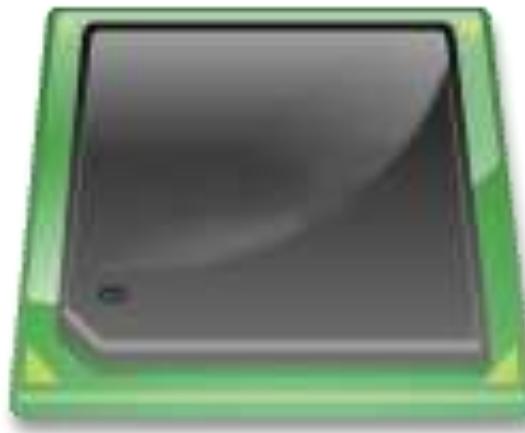


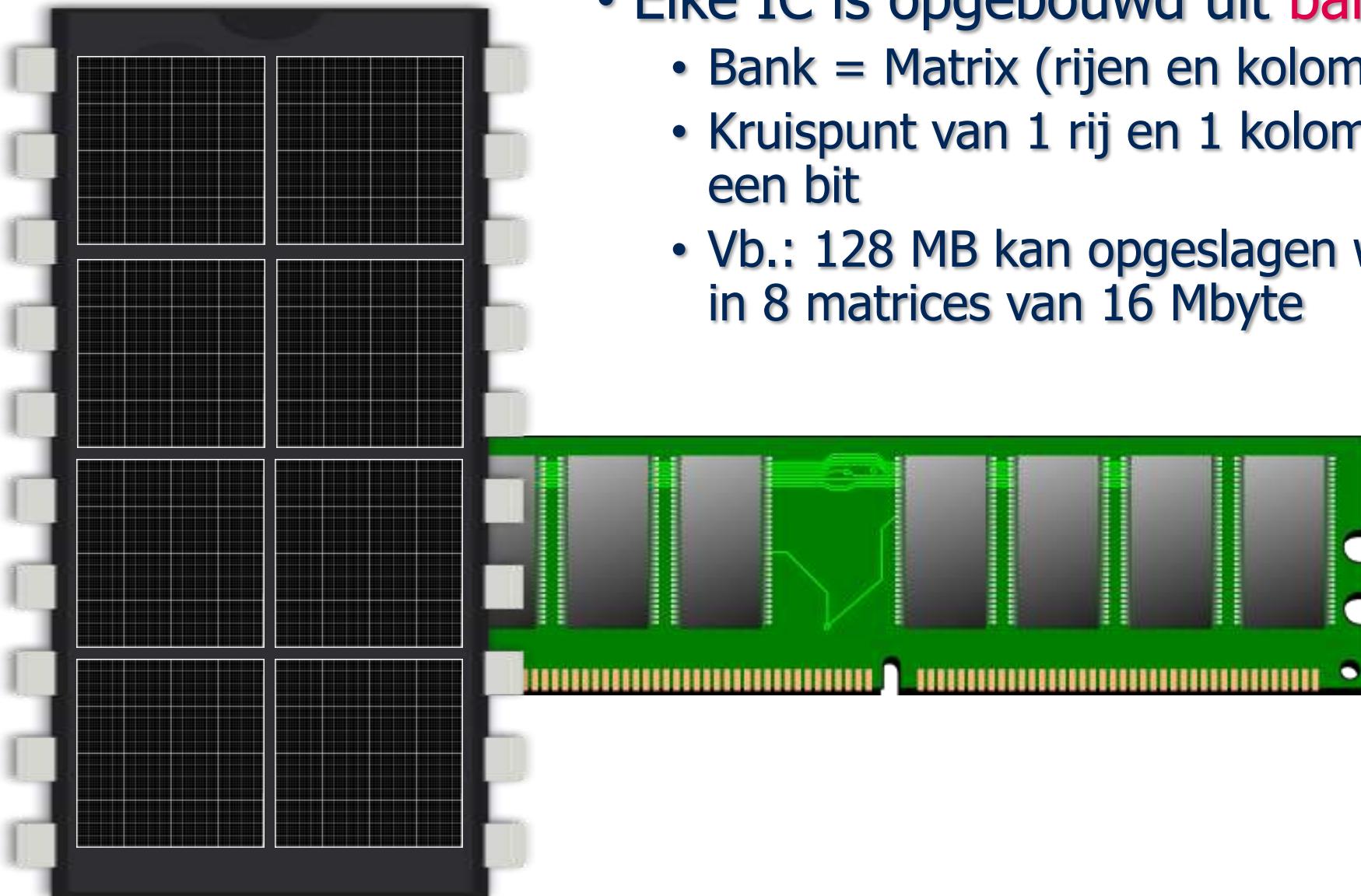
- Welke DIMM bevat de data? **Deel van het adres dient om DIMM te selecteren.**
 - 1,2,... bits
 - **chip-select lijn** (eigenlijk: DIMM select)



Vb.: module van 1 Gigabyte

- 8 SDRAM-IC's van 128MB
- In 1 cyclus wordt **per IC** één byte gelezen/geschreven.
 - Woordlengte = 64 bit
- Elke IC krijgt dezelfde signalen van de G.B.E.





- Elke IC is opgebouwd uit **banks**
 - Bank = Matrix (rijen en kolommen)
 - Kruispunt van 1 rij en 1 kolom bevat een bit
 - Vb.: 128 MB kan opgeslagen worden in 8 matrices van 16 Mbyte

Matrix van 16MB

- Bevat een aantal invoerlijnen
 - Besturingsdata
 - Besturingslijnen
- Bevat 8 I/O-lijnen
 - Bidirectioneel (afhankelijk van lees- of schrijfoperatie)
 - Per I/O-lijn één bit, dus 8 bits/cyclus



Matrix van 16MB

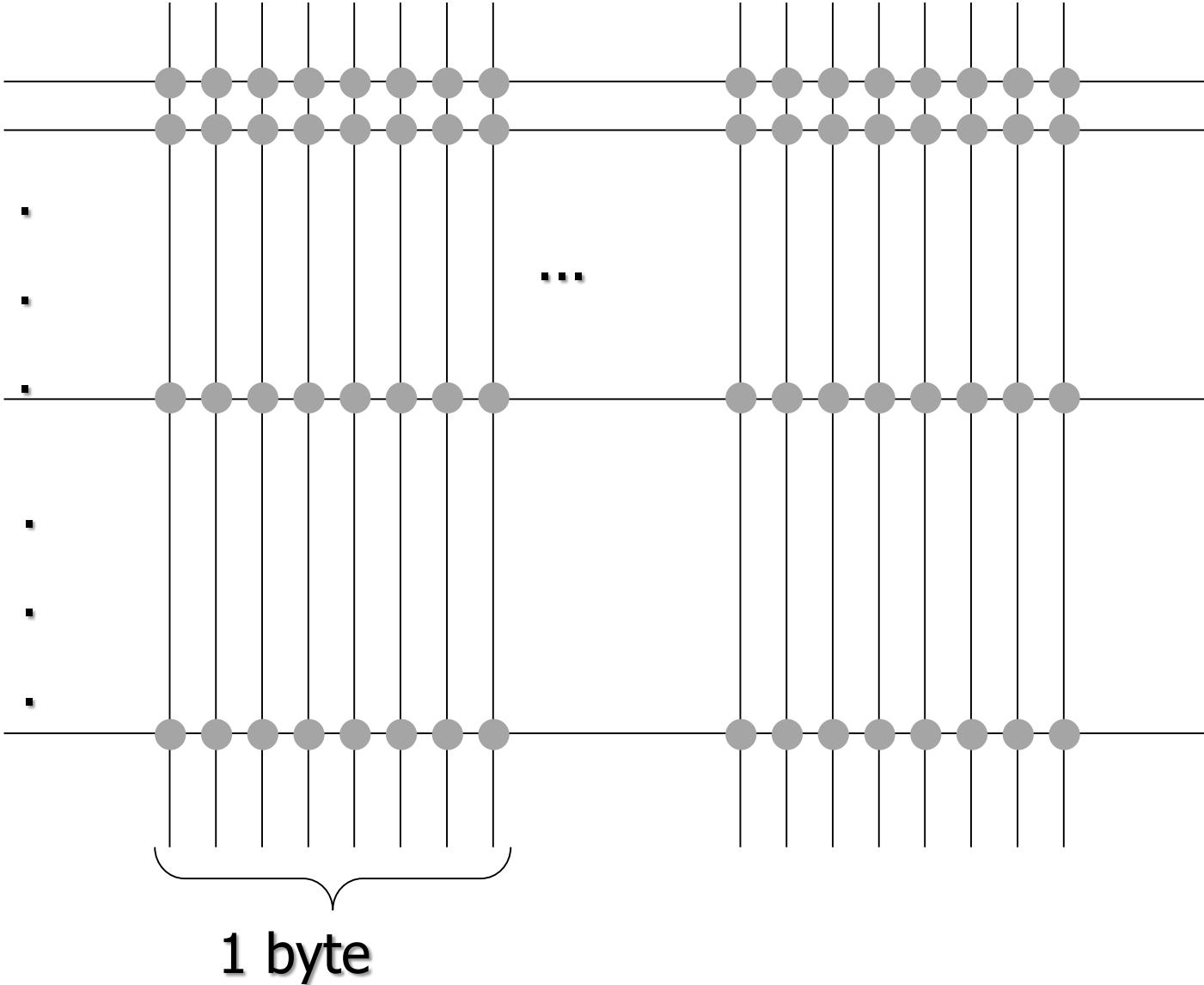
- Bevat 2^{27} bits (= $2^3 \times 2^4 \times 2^{20}$ bits)
 - 16384 (2^{14}) rijen en 8192 (2^{13}) kolommen
- Hoeveel bits bevat één rij?
8192 of 2^{13} .
- Hoeveel bytes bevat één rij?
1024 of 2^{10} .





$2^{10} = 1.024$ groepen van 8 kolommen

$2^{14} =$
16.384
rijen

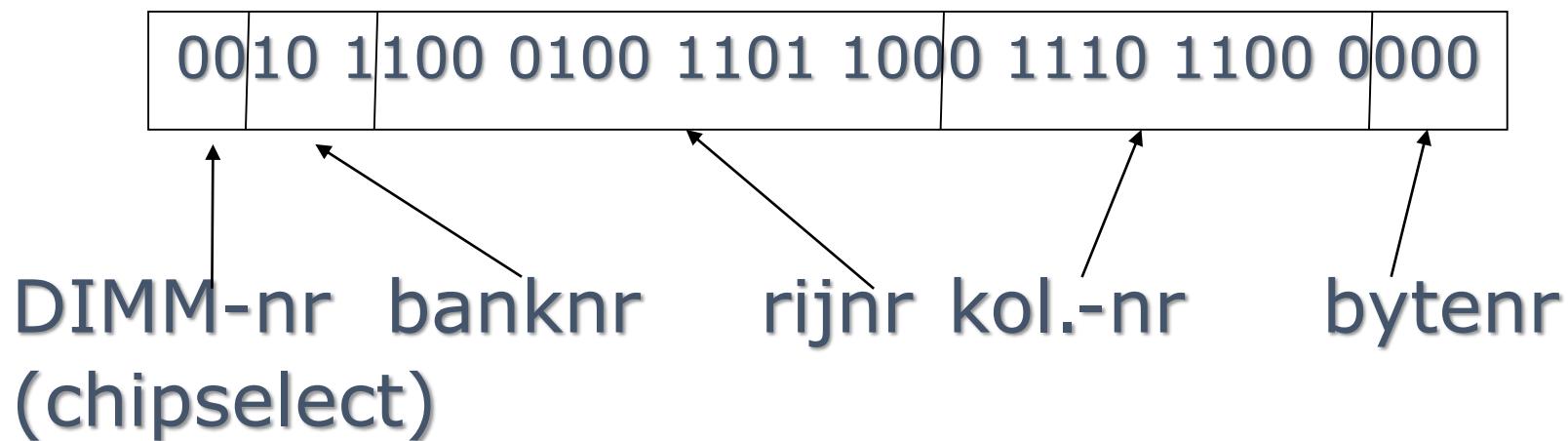


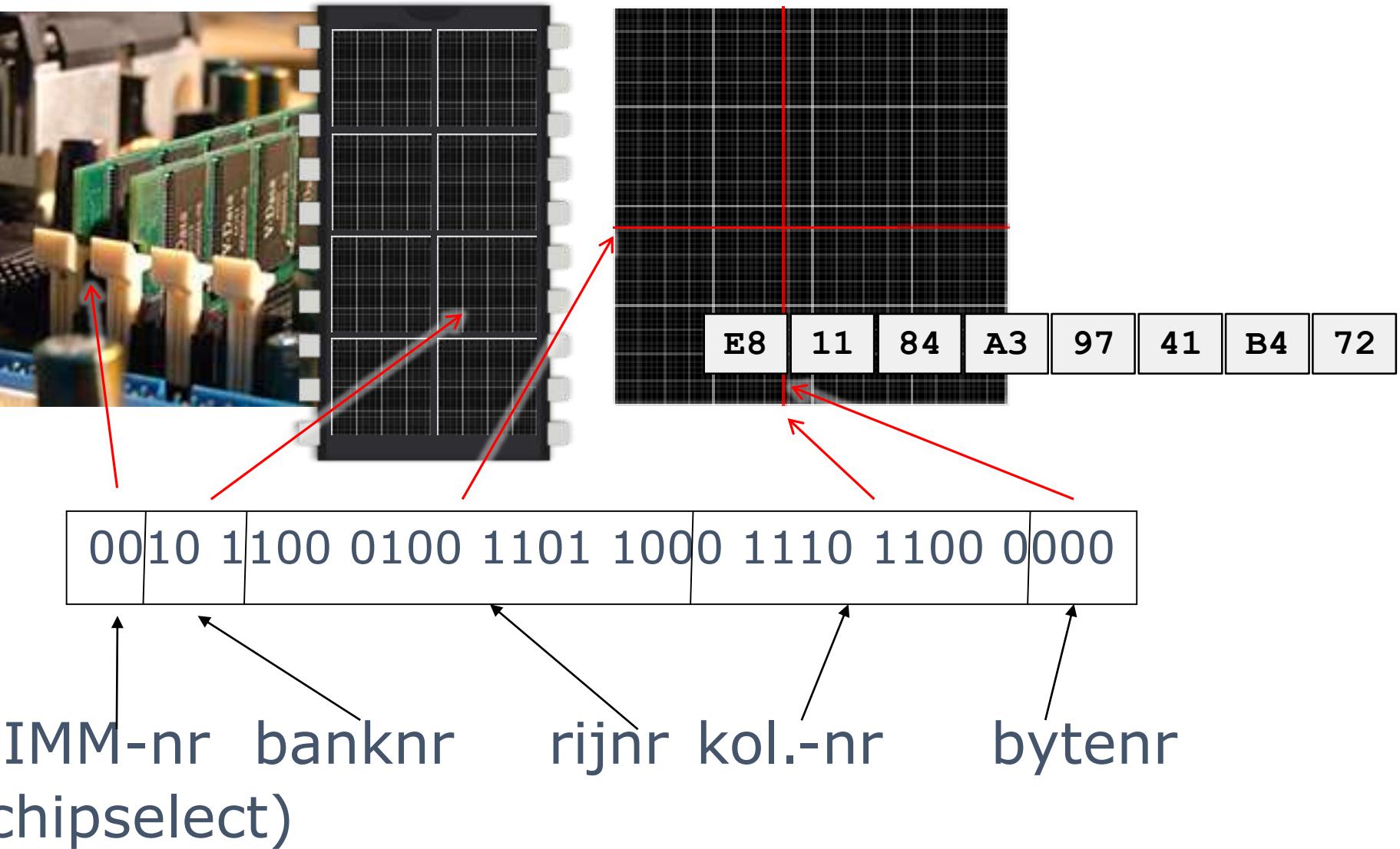
Adressering

- (1) : DIMM selecteren 2 bits
 - Elke IC van één DIMM : zelfde bewerking, 1 byte lezen/schrijven
- Welke byte? 3 bits, want 2^3 matrices
 - (2) welke matrix (bank)?
 - Welke byte in die matrix?
 - (3) welke rij? 14 bits, want 2^{14} rijen
 - (4) welke kolom? 10 bits, want 2^{10} groepen van 8 bits

Adressering voorbeeld

- Cve vraagt (4 bytes) vanaf adres 2C4D8EC0
- G.B.E. levert (altijd) 8 bytes vanaf adres (binair)
00101100010011011000111011000000
- Dit adres wordt als volgt opgedeeld:





Besturingslijnen

- WE = **write enable** (0 = lezen, 1 = schrijven)
- Rij- en kolomnummer: via dezelfde lijnen (na elkaar)
 - /RAS (**row address strobe**): het is het rij-nummer;
 - /CAS (**column a.s.**): het is het kolom(groep)-nummer.



*Waarom /RAS, /CAS?
 / = geïnverteerd, d.w.z.:
 spanning valt weg = signaal*

Werking: 1 byte lezen

- Gebeurt in 3 fases
- Fase 1: rij selecteren
 - GBE stuurt bank- en rijnummer
 - GBE geeft RAS-signaal
 - **Elke IC v.d. DIMM selecteert rij**



Werking: 1 byte lezen

- Gebeurt in 3 fases
- Fase 1: rij selecteren
 - GBE stuurt bank- en rijnummer
 - GBE geeft RAS-signaal
 - **Elke IC v.d. DIMM selecteert rij**



Werking: 1 byte lezen

- Fase 2: kolom selecteren en datatransfer
 - GBE deactiveert WE (= geen schrijf)
 - GBE stuurt kolomnummer
 - GBE geeft CAS-signaal
 - **Elke IC levert byte op i/o-lijnen**



Werking: 1 byte lezen

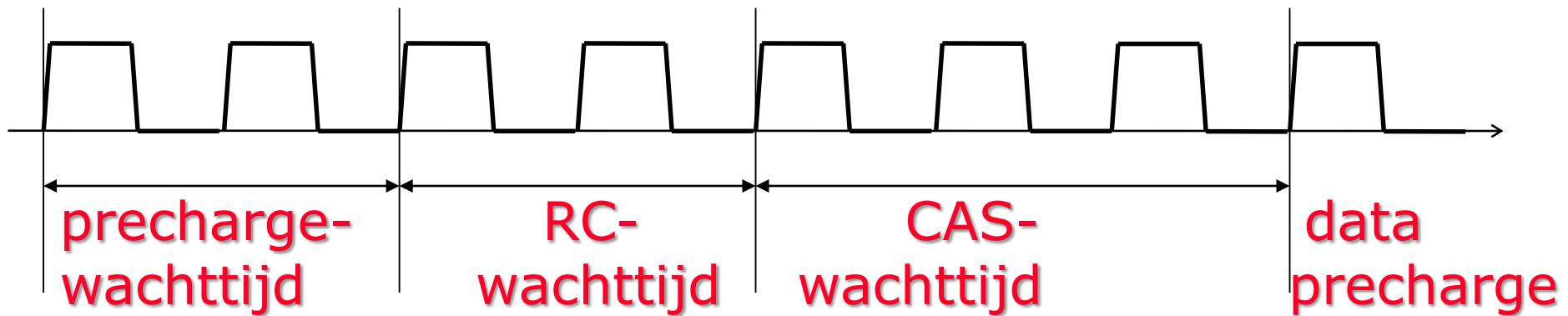
- Fase 3: precharge
 - GBE deactiveert RAS, CAS
 - IC maakt zich klaar voor volgende lees/schrijf-operatie



Wachttijden

- GBE stuurt rijnummer ... IC levert byte op i/o-lijnen
 - RAS-to-CAS-delay: **RC-wachttijd**
 - CAS-signaal ... IC levert byte op i/o-lijnen: **CAS-latency**
- GBE mag niet direct weer een rijnummer sturen: **precharge latency**

- Signalen komen volgens de timing van de bus, bvb.:



- Wachttijden vermeld als **CAS – RC – precharge** (– som)
 - Bvb.: 3-2-2 (of 3-2-2-7).

Performantie

- Vb.: SDRAM-module met wachttijden van 3-2-2

```
mov eax, [2C4D8EC0h]
mov ebx, [2C4D8EC4h]
mov ecx, [2C4D8EC8h]
mov edx, [2C4D8ECCh]
```

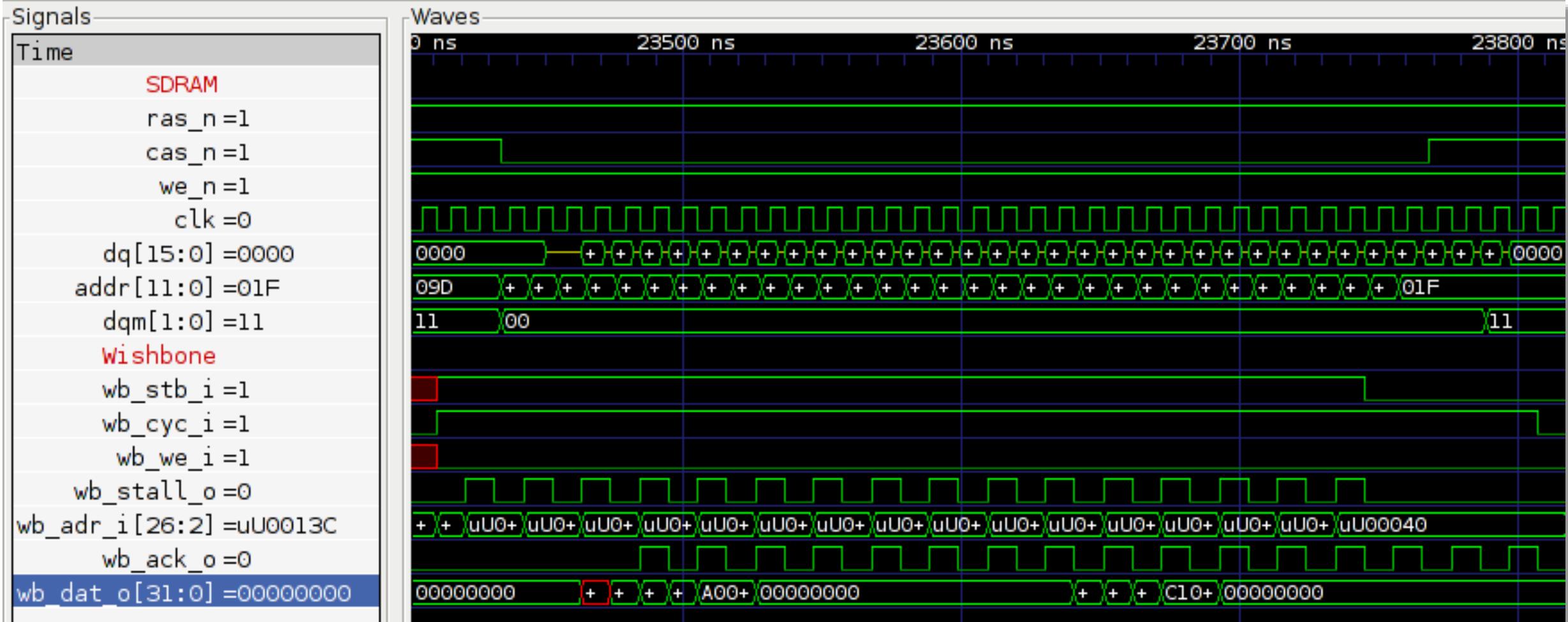
Performantie

- Voor G.B.E.:
geef de 8 bytes vanaf 2C4D8EC0
geef de 8 bytes vanaf 2C4D8EC8
- Deze adressen hebben hetzelfde DIMM-, bank- en rijnummer
 - 1° lees: 7 cycli
 - 2° lees: 3 cycli (alleen de CAS-latency)

Performantie

- Bloktransfer gaat verder:
 - bij uitlezen van een aantal 64-bits woorden met opeenvolgende adressen valt ook de CAS-latency weg
 - d.w.z. 1 64-bit woord / cyclus
- Waar komen ze terecht? **In de cve.**
 - Meer specifiek: in de cache (zie later)

Voorbeeld

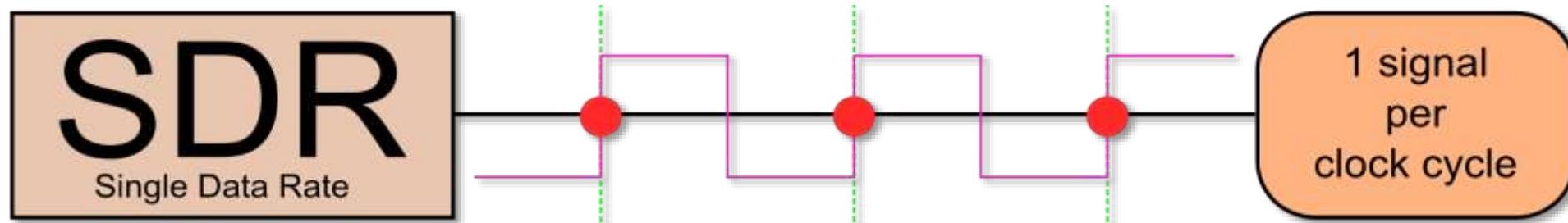


Uitbreiding mogelijk

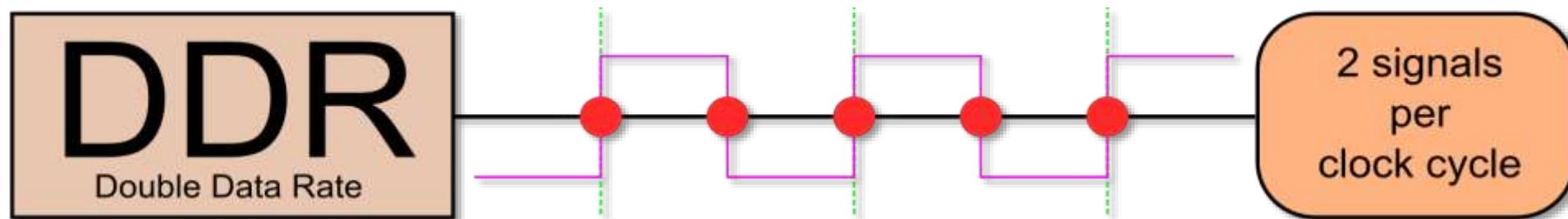
- **Wet van Moore:** meer transistoren op 1 IC, dus
 - Matrices met meer rijen en/of kolommen.\
 - Meer matrices
 - ECC (error correcting code): extra schakelingen om fouten op te sporen en te verbeteren

DDR SDRAM

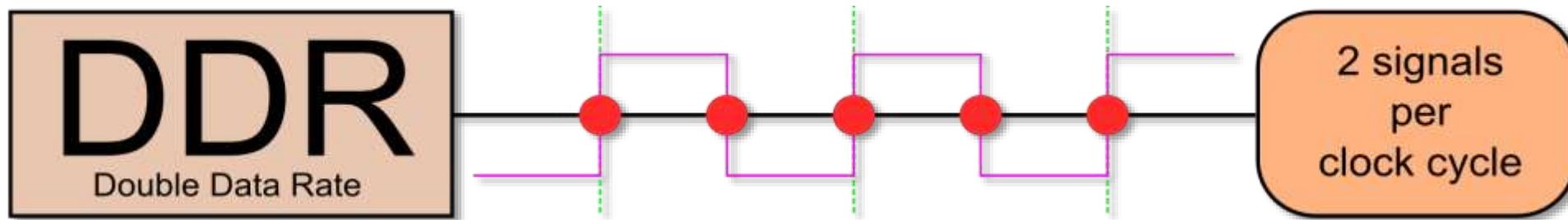
- Stijgende flank van de klok van "geen spanning" naar "spanning":



- DDR = double date rate
 - zowel bij stijgende als bij dalende flank worden bits afgegeven



DDR SDRAM



- DDR = double date rate
 - na sturen van rij- en kolomnummer worden 2 bits gelezen (per IO-lijn)
- DDR2: 4 bits/memory cycle
- DDR3: 8 bits/memory cycle

SDR/DDR snelheid

- Debiet bij bloktransfer aan 100 MHz?
 - SDR: elke klokpuls 1 byte, dus 100 Megatransfers per seconde (per IC)
 - Debiet: 800 MB/s
 - DDR: 2 bits per cyclus, dus 200 MT/s
 - Debiet: 1.6 GB/s
 - DDR2: 400 MT/s (of 3.2 GB/s)
 - DDR3: 800 MT/s (of 6.4 GB/s)



H5: Apparatuur

- Werkgeheugen
 - Samenwerking met CPU
 - Interne structuur SDRAM
 - Soorten transistorgeheugens
 - Voorgeheugen (cache)

Soorten transistorgeheugens

- Meerdere soorten transistoren
 - bipolaire
 - MOS = MOSFET (metal oxide ...)
 - NMOS, PMOS of combinatie : CMOS (complementary MOS)
- Nu wordt vooral CMOS gebruikt.

- Ook verschillende soorten geheugens: DRAM, SRAM, ROM, PROM, enz...

- Per bit: 1 MOS-transistor, 1 condensator
 - Klein, dus veel bits per mm²
- MOS is **dynamisch**: informatie lekt weg, daarom voortdurend **opfrissen**
- DRAM staat voor dynamisch RAM
 - **herschrijven** na lezen

- SRAM staat voor **statisch** RAM
 - Geen opfrissing nodig
- Per bit: 6 transistoren, 1 flipflop
 - **Sneller** dan DRAM
 - **Groter**, dus **minder bits** per mm²

Andere eigenschappen

- Energieverbruik, snelheid, aantal bits per oppervlakte, prijs per bit, ...
- CMOS: laag energieverbruik.
- SRAM: snelst, maar duur
- DRAM: trager, maar goedkoper
- Verstandige toepassing van SRAM? Cache.

ROM: read only memory

- Tijdens fabricage: opslaan van de informatie (dus niet meer te veranderen)
 - Toepassing: software om computer op te starten.
- **PROM** (programmable ROM): men kan één keer info opslaan. Daarna niet meer te veranderen.
- **EPROM** (erasable PROM): toch nog te veranderen (bestralen met uv-licht om te wissen)
- **EEPROM** (electrically EPROM): wissen gebeurt byte-per-byte
- **Flash-geheugen**: wissen per blok

Gebruik van het werkgeheugen

- Informatie liever bijhouden in een register dan in het werkgeheugen.
 - Helaas hebben we maar enkele registers (eax, ebx, ecx, edx, ...)
- Geniaal idee: snel, klein geheugen dat een kopie bevat van de meest gebruikte bytes.
 - Dit is het voorgeheugen (**cache**)

De programmeur beslist wat er in de registers komt. De hardware beslist zelf wat er in de cache komt (via een algoritme).

- In de cache-hardware is een algoritme ingebouwd om te beslissen wat in de cache komt
 - De bytes die de C.V.E. gebruikt, staan meestal in mekaar's buurt (d.i. **het principe van lokaliteit**). Dus voorspelbaar.

- **C.V.E.** geeft
 - adres + lees-bevel
 - adres + data + schrijf-bevel
- **G.B.E.** doet de rest:
 - Info uit echt werkgeheugen, of
 - info uit de cache.

Ophalen van een woord

- Vb. : woord met adres 22DC2A0C
- Eerst nagaan of het in de cache staat?
 - Ja: snel leveren
 - Nee:
 - Ophalen uit werkgeheugen (trager)
 - Alle woorden met adressen 22DC2Axx (64×4 bytes; bloktransfer)
 - Waarom? **Zullen later (waarschijnlijk) ook nodig zijn (lokaliteit).**

Inhoud voordeugeugen

0000BB: 00 00 2A C0 FF FF 0A 10 C0 ...

22DC2A: CC CC CC CC AB 00 FE 23 AB ...

- **Betekenis?**
 - In de 256 bytes waarvan de adressen beginnen met **0000BB** staat **00 00 2A C0 FF ...**

Het voordeugeugen is geen extra-opslagcapaciteit!
Het bevat enkel kopieën

Opslaan van een woord

- Vb.: kopieer inhoud van EAX naar 12ABC020
- Opslaan in cache en verder gaan met uitvoering volgende bevel.
- Probleem: **cache-inconsistentie**
 - Inhoud in cache is geen kopie van inhoud werkgeheugen;
 - Oplossingen?
 1. Onmiddellijk kopiëren naar werkgeheugen
 2. Wachten tot inhoud cache moet gewist worden

Gebruik van voorgeheugen

- Bij **hit**: G.B.E. levert info snel
- Bij **miss**: G.B.E. levert info later
 - Na miss: ook opslaan in het voorgeheugen
 - Later heeft C.V.E. misschien zelfde informatie weer nodig

Merk op: het voorgeheugen is veel kleiner dan het werkgeheugen.

Opslaan van een woord

- Komt altijd in het voorgeheugen
- Soms moeten we iets weglaten uit het voorgeheugen. **Wat?**
 - Het item weglaten dat al het langst niet meer gebruikt is
 - **Trefverhouding** tot meer dan 90%

Staat het in het voorgeheugen?

- CPU vraagt woord met adres 07600ADC
- Staat het in het voorgeheugen of niet?
 - Alle items nagaan? Dan is de winst weg.
 - Computergeheugen is als een magazijn (je moet weten waar alles staat)
- Oplossing? **Associatief geheugen.**

Associatief geheugen

- Vb.: wat gebeurde er in 1912 voor de kust van New Foundland?
 - Tip 1: film
 - Tip 2: ijsberg
- Computer heeft daar een programma voor nodig
 - Als je iets wilt, moet ge zeggen waar het staat
 - Computer herinnert zich nooit iets, werkt met adressen

Associatief geheugen

- Menselijk geheugen (bvb. van Piet Pienter):

Betty	02/2783219
Ellen	016/633119
Peter	03/6512121
Sofie	015/292997
Wendy	077/700700

- Naam geven (bvb. Peter) → geheugen levert direct het telefoonnummer
 - Men probeert dit na te bootsen in computers: alle namen tezelfdertijd vergelijken met opgegeven naam

Associatief geheugen

0000BB: 00 00 2A C0 FF FF 0A 10 C0 ...

22DC2A: CC CC CC CC AB 00 FE 23 AB ...

...

- CVE vraagt woord met adres 22DC2A04
 - 22DC2A wordt vergeleken met : 0000BB, 22DC2A, ... **terzelfdertijd**
 - 2° lijn: **hit**, AB 00 FE 23 is het gevraagde woord.

Associatief- en werkgeheugen

- **Werkgeheugen:** om info op te vragen moet je het adres leveren
- **Associatief geheugen:** deel van de inhoud opgeven; bvb. 22DC2A

0000BB: 00 00 2A C0 FF FF 0A ...

22DC2A: CC CC CC CC AB 00 FE ...

...

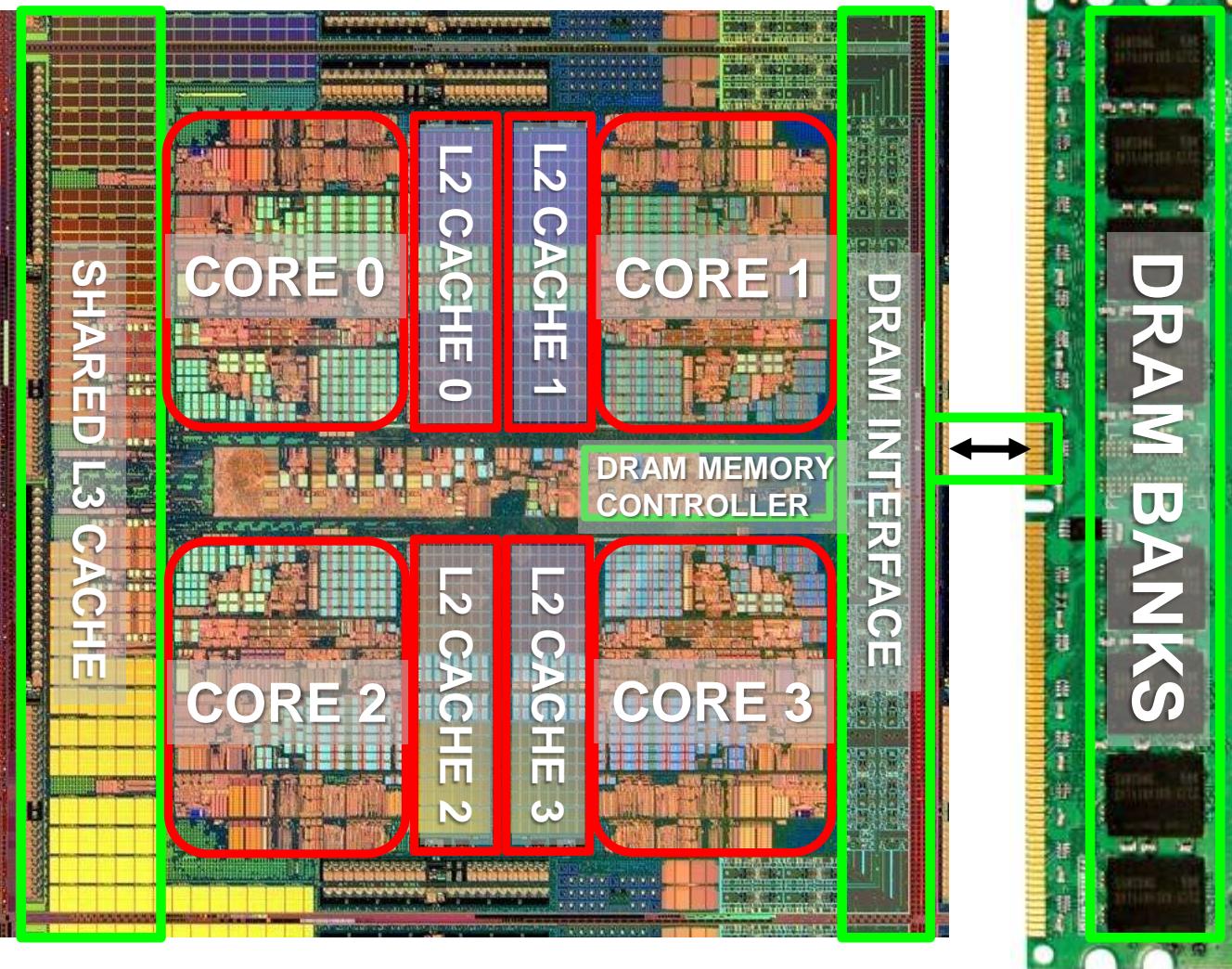
- **22DC2A is deel van de**
 - Er wordt niet gevraagd

Associatief geheugen wordt ook *content addressable memory (CAM)* genoemd.

- **Vroeger:** extra op moederbord
 - SRAM-chip;
 - eigen zgn. cache-controller;
- **Later:**
 - één cache op moederbord: L2
 - één cache op CPU: L1
- **Nu:**
 - Geen cache meer op moederbord
 - Alle caches in CPU: L1, L2, L3
 - Eventueel elke core haar eigen L1-cache.



Quad-Core
Chip



Hiërarchie



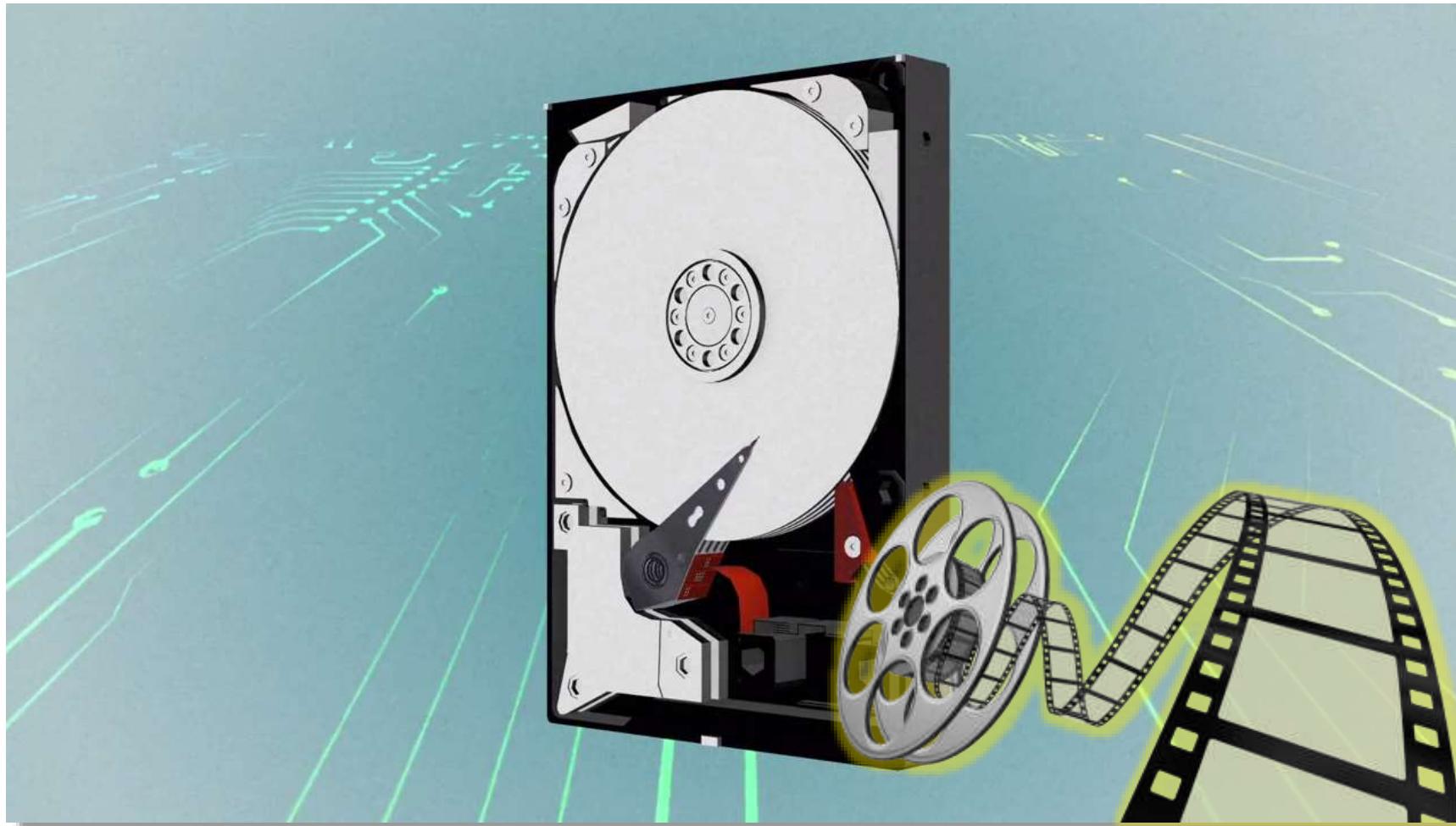
Snelheid, grootte en prijs varieert

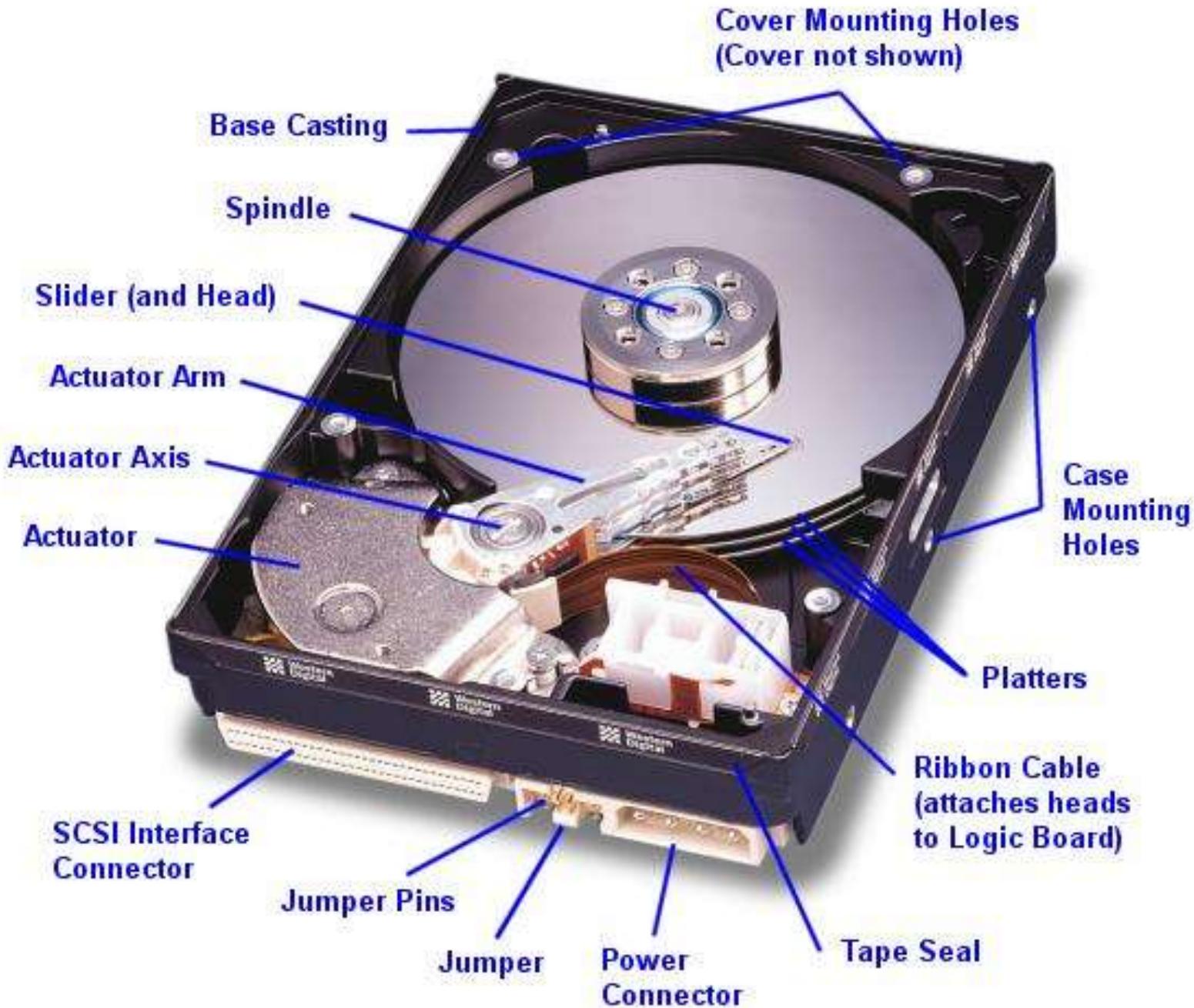


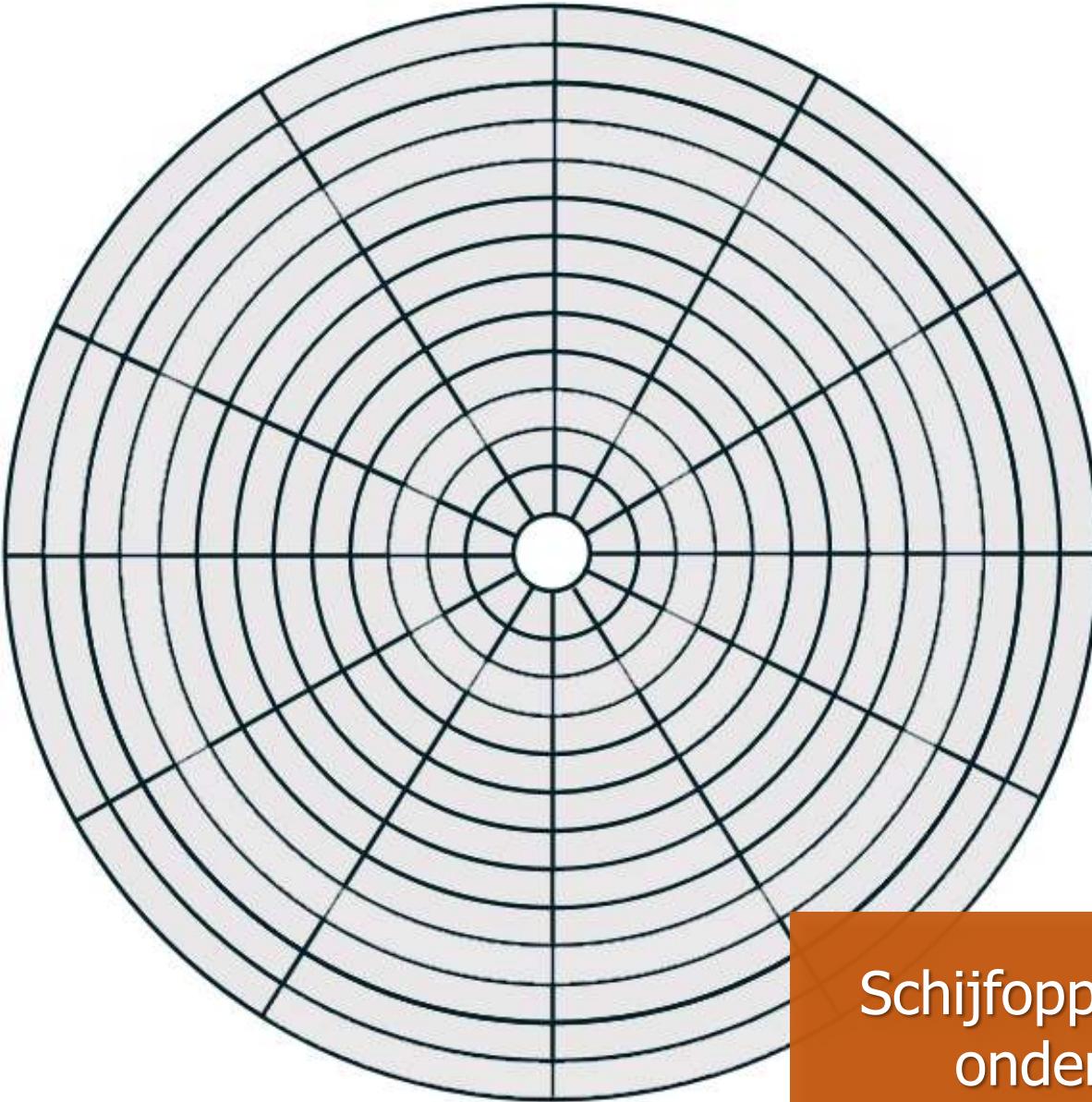
H5: Apparatuur

- Moederbord
- Chipset
- Bussen
- Processor
- Grafische Verwerkingseenheid
- Werkgeheugen
- Magneetschijven

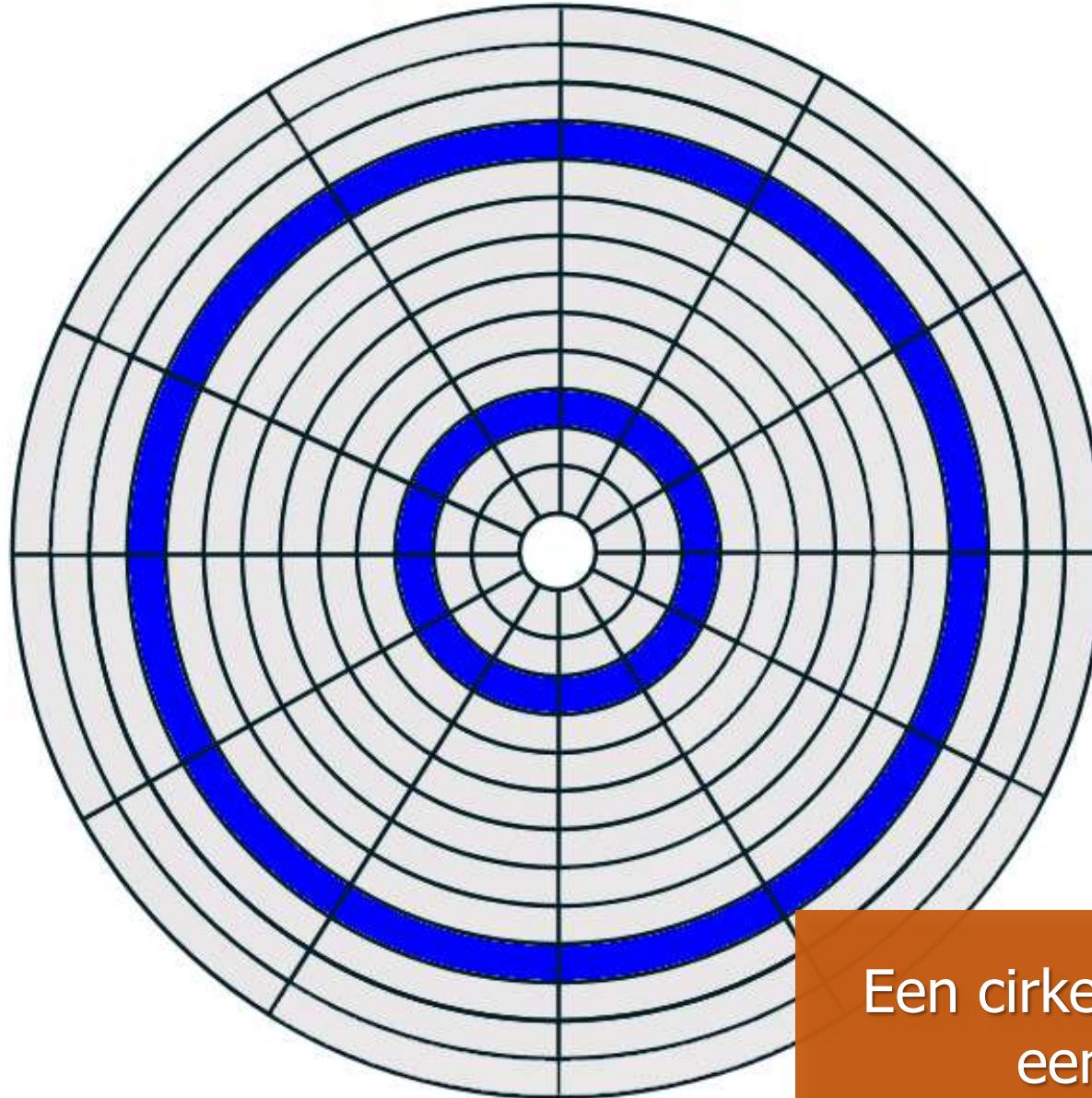
How do hard drives work



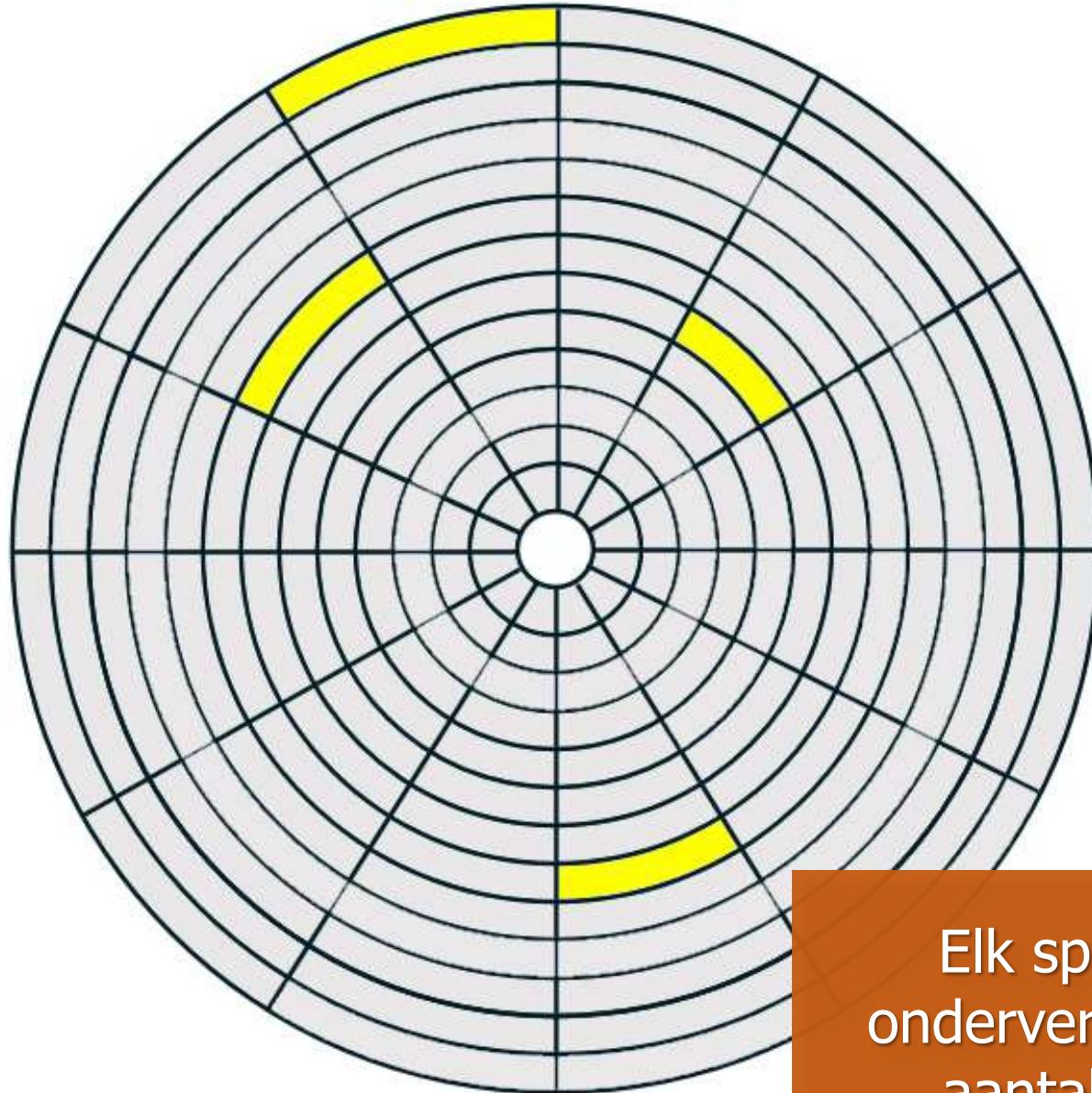




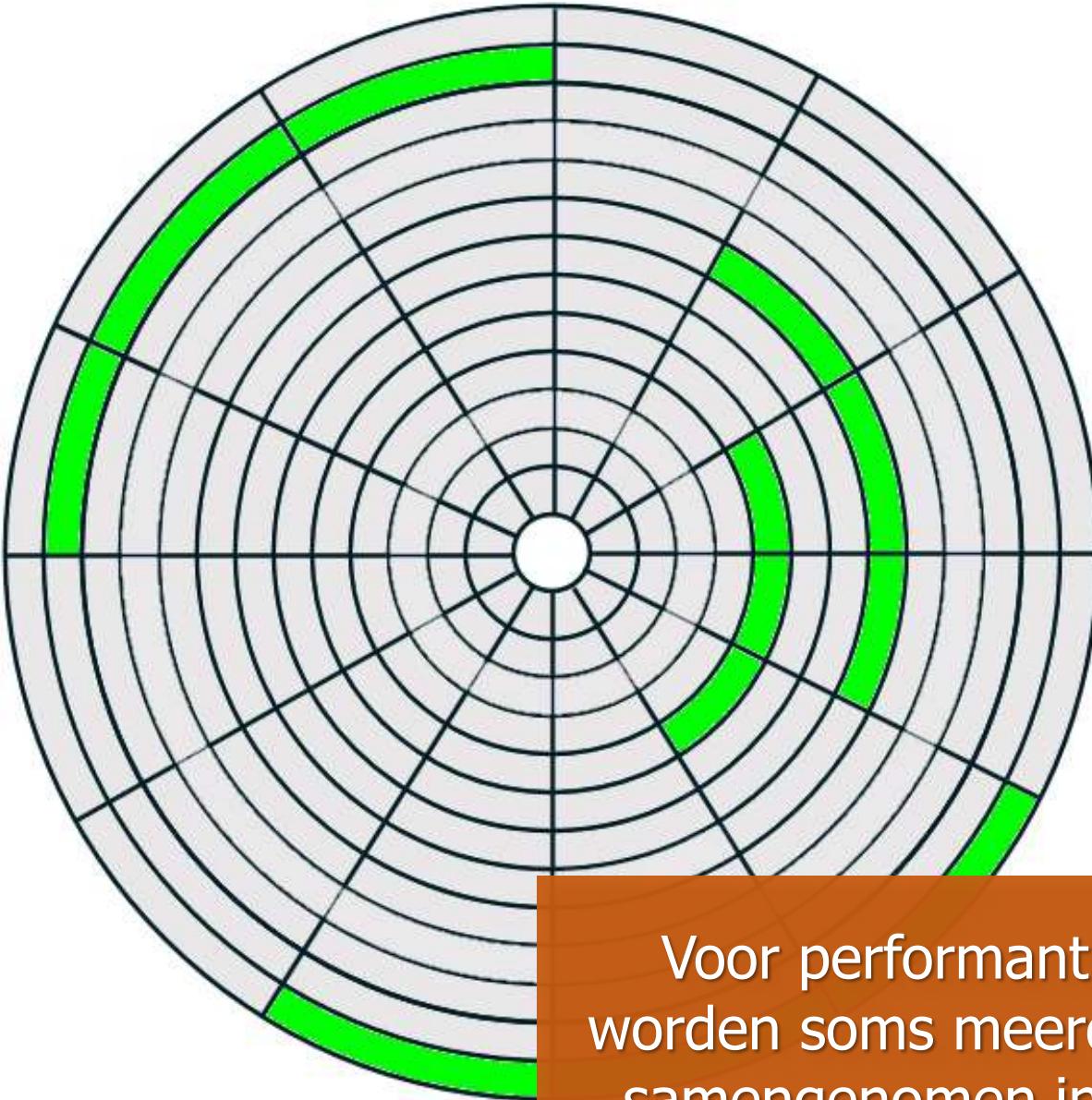
Schijfoppervlak wordt
onderverdeeld



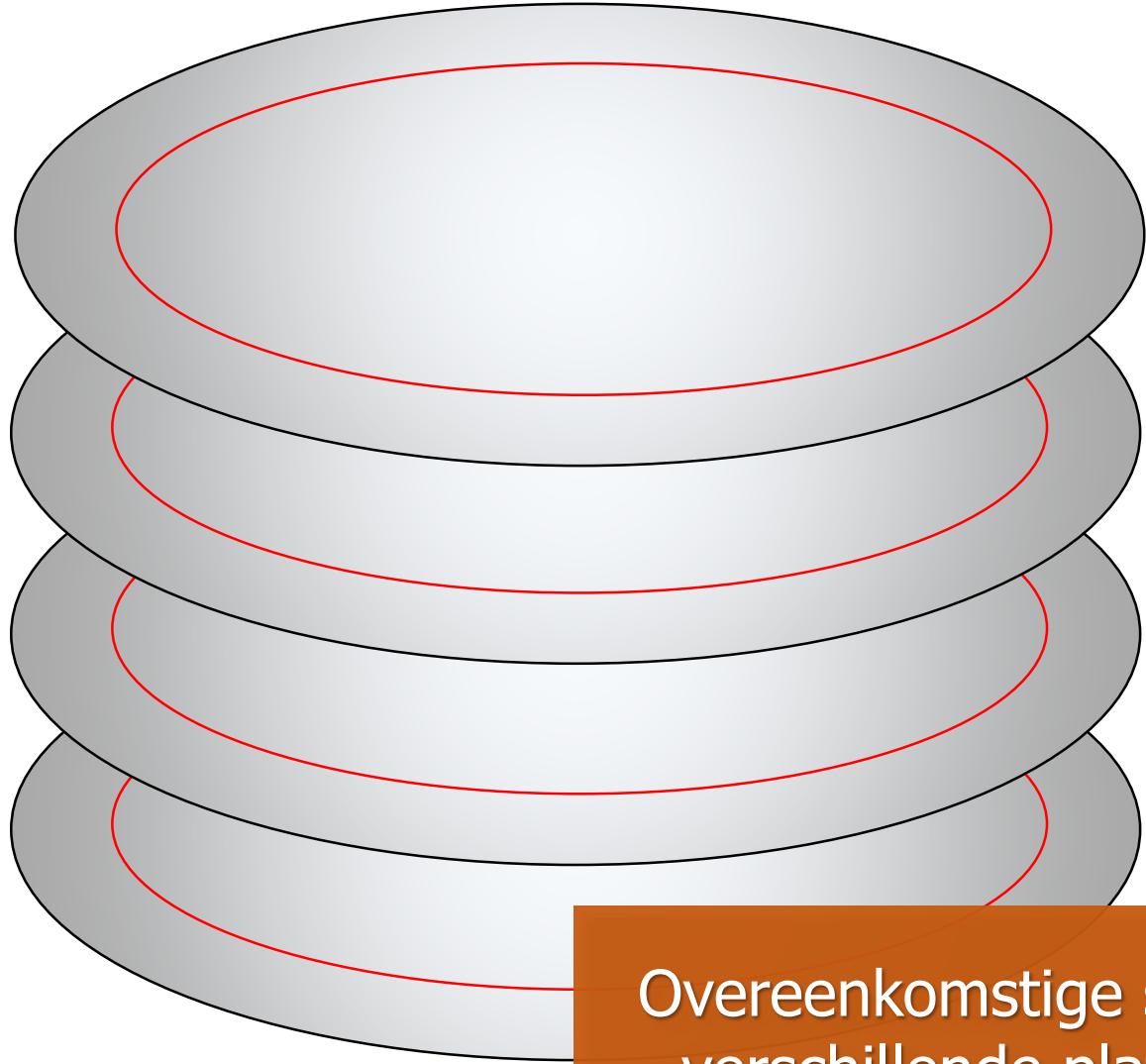
Een cirkel noemt men
een *spoor*



Elk spoor wordt
onderverdeeld in een
aantal *sectoren*

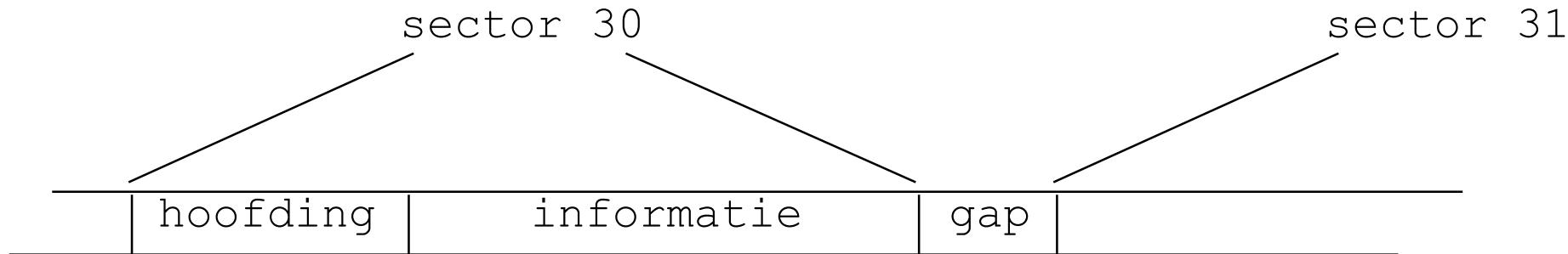


Voor performantie-redenen
worden soms meerdere sectoren
samengenomen in een *cluster*



Overeenkomstige sporen (op de verschillende platen) vormen samen een *cilinder*

Indeling van een spoor



- Naast de data zit in een sector ook een foutcorrigerende code: **pariteitsbits**.
 - Krijgen de waarde 0 of 1 al naar gelang het aantal enen in de byte even of oneven is.

0001 0101 → 1
0110 0110 → 0

- Om meerdere fouten te herkennen, wordt aan een groep van bytes een extra **pariteitsbyte** toegekend

0011 1001 0

1000 1001 1

...

...

...

1100 0001 1

0110 0011 0

0111 0001 0

1000 1001 1

...

...

...

1000 1001 1

0110 0011 0

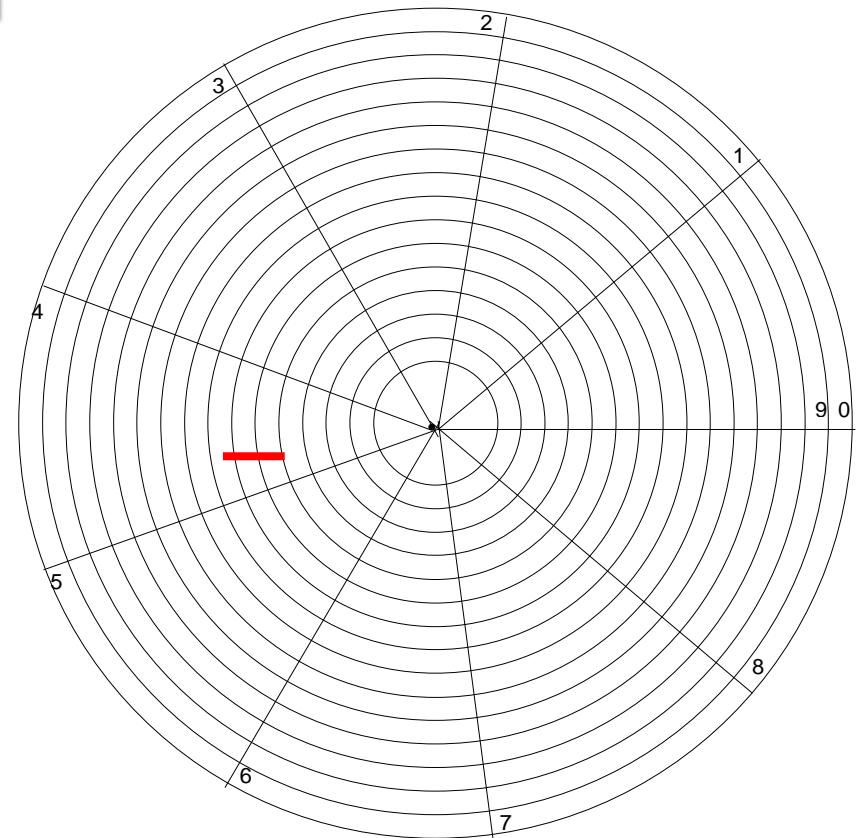
Welke fout wordt niet opgemerkt?

Magneetschijven: capaciteit

- Capaciteit is het aantal bytes dat beschikbaar is.
- **Netto-capaciteit:** het aantal bytes dat voor data kan gebruikt worden
- **Bruto-capaciteit:** netto-capaciteit, plus hoofdingen, controle bits en gaps

Informatiedichtheid

- **Radiale dichtheid:** volgens een straal
 - = aantal sporen per lengte-eenheid

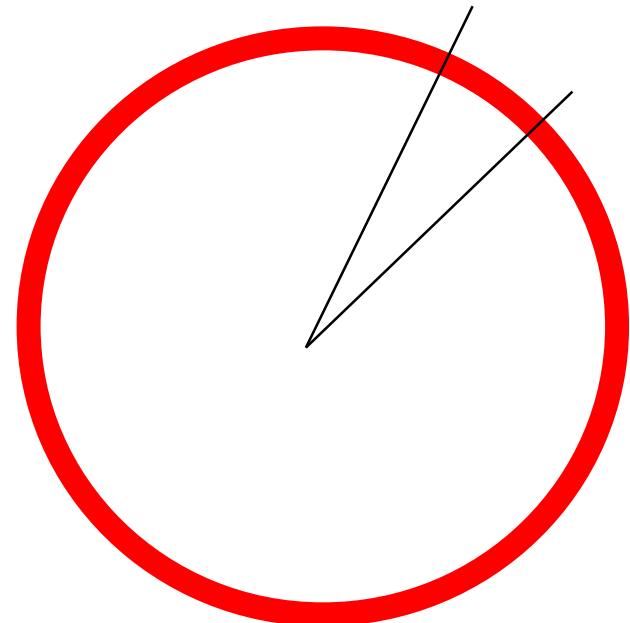


Informatiedichtheid

- **Lineaire dichtheid:** volgens een spoor

- **Vroeger:** aantal bits per sector constant
 - D.w.z. lineaire dichtheid in buitenste sporen is lager dan in binnenste sporen

- **Tegenwoordig:** lineaire dichtheid constant
 - D.w.z. meer sectoren in de buitenste sporen
 - Zie Zone Bit Recording (later)



Magneetschijven: toegang

- Stel: gebruiker wil een bestand lezen
 - Het besturingssysteem houdt bij in welke sectoren het bestand zit
- Het besturingssysteem stuurt lees-operaties voor de betreffende sectoren naar de **schijfbesturingseenheid (SBE)**

Magneetschijven: toegang

- Er wordt altijd een ganse sector in één keer gelezen of geschreven
- Welke sector?
 - Vroeger: doorgeven aan SBE van het cilinder-, kop- en sectornummer
 - Tegenwoordig: elke sector heeft een uniek nummer (= het LBA-nummer)

Lees-/schrijftoegang

- Eerst: arm (met de kop) verplaatst zich naar het goede spoor. Dit is de **zoektijd** (E: **seek time**).
 - Zoektijd zeer variabel
 - Startpositie niet gekend
 - Afstand niet gekend
 - Gemiddelde nemen van vele toegangen
 - gaat tegenwoordig van 4ms - 15ms

Lees-/schrijftoegang

- Na "zoeken": wachten tot juiste sector zich onder de kop bevindt. Dit is de **rotationele wachttijd** (E: **latency**)
 - Willekeurig tussen 0 en tijd van 1 toer
 - Gemiddeld: helft van tijd voor 1 toer

Lees-/schrijftoegang

- Wat is de gemiddelde rotationele wachttijd voor een schijf van 7200 tpm?

7200 toeren in 1 minuut

= 7200 toeren in 60 sec

= 120 toeren in 1 sec

= 1 'toer' in $1/120$ s = 0,008333 s

= 1 'toer' in 8,333 ms

Gem. rot. wachttijd = 8,333 ms / 2 = 4,17 ms

Lees-/schrijftoegang

- Toegangstijd = zoektijd + gem rot wt
 - typisch 5ms – 20ms
- Vergelijk toegangstijden
 - DVD ~ 150ms
 - Geheugen ~ 100 ns (SDRAM)
~ 10 ns (SRAM)

Lees-/schrijftoegang

- Stel: de gebruiker schrijft veel data, meer dan één spoor vol; in welk spoor wordt het beste het vervolg gezet?
 - Zelfde spoornummer op een ander oppervlak
 - Zo weinig mogelijk wisselen van cilinder

Lees-/schrijftoegang

- Stel: de gebruiker schrijft heel veel data, meer dan één cylinder vol; in welke cilinder wordt het beste het vervolg gezet?
 - De cilinder ernaast

Lees-/schrijftoegang

- Rest hangt af van snelheid van gegevensoverdracht
 - **Intern:** van schijf naar buffer
 - **Extern:** van buffer naar werkgeheugen
 - Eventueel herlezen indien fout (pariteit)

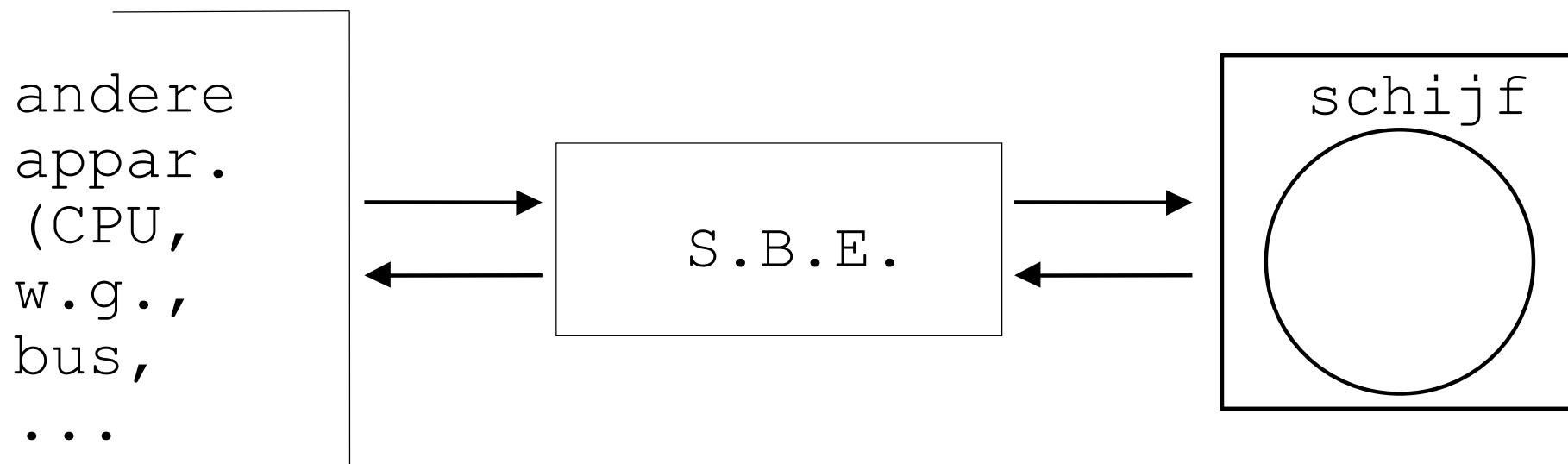


H5: Apparatuur

- Magneetschijven
 - Beschrijving, soorten, gebruik
 - Lees, schrijf
 - **Schijfbesturingseenheid**
 - Sectornummering
 - Zone bit recording
 - RAID

Schijfbesturingseenheid

- Schijf (incl. de aandrijfeenheid) communiceert met de rest van de apparatuur via de S.B.E.



Wat doet de SBE?

- Arm bewegen tot kop op het juiste spoor staat (**seek**)
- Van alle sectoren de hoofding lezen, tot juiste sector passeert (**search**)
- **Schrijven:**
 - Gegevens (v.b. 4 kbyte) aannemen van de CPU en opslaan in buffer (in SBE)
 - **Wegschrijven naar schijf**
 - elektrische informatie (0V/5V) → magnetische informatie (noord/zuid)

Wat doet de SBE?

- Arm bewegen tot kop op het juiste spoor staat (**seek**)
- Van alle sectoren de hoofding lezen, tot juiste sector passeert (**search**)
- Lezen
 - Gegevens **lezen van schijf** en opslaan in buffer (in SBE)
 - magnetische informatie (noord/zuid) → elektrische informatie (0V/5V)
 - Gegevens (4 kbyte) naar de CVE sturen

Wat doet de SBE?

- Pariteitsbits berekenen en toevoegen (bij schrijven)
- Pariteitsbits controleren (bij lezen), eventueel opnieuw lezen
- Clusters niet lezen in de volgorde van de vraag, maar met wachttijden zo klein als mogelijk = native command queuing (NCQ)

SBE: soorten

- Verschillende standaarden
 - ATA (IDE)
(Advanced Technology Attachment)

DISCONTINUED



SBE: soorten

- Verschillende standaarden
 - **SCSI**
 - Duurder dan ATA
 - Vooral bedoeld voor servers
 - Ook serial attached scsi : SAS



SBE: soorten

- Verschillende standaarden
 - S-ATA
 - Serieel ATA
 - 2 draden voor data, 2 voor besturing
 - Zeer populair voor hedendaagse low-end tot high-end schrijven



SBE: soorten

- Verschillende standaarden
 - M.2
 - Nieuwe standaard voor zéér snelle schijven
 - Kleine form-factor





H5: Apparatuur

- **Magneetschijven**
 - Beschrijving, soorten, gebruik
 - Lees, schrijf
 - Schijfbesturingseenheid
 - **Sectornummering**
 - **Zone bit recording**
 - RAID

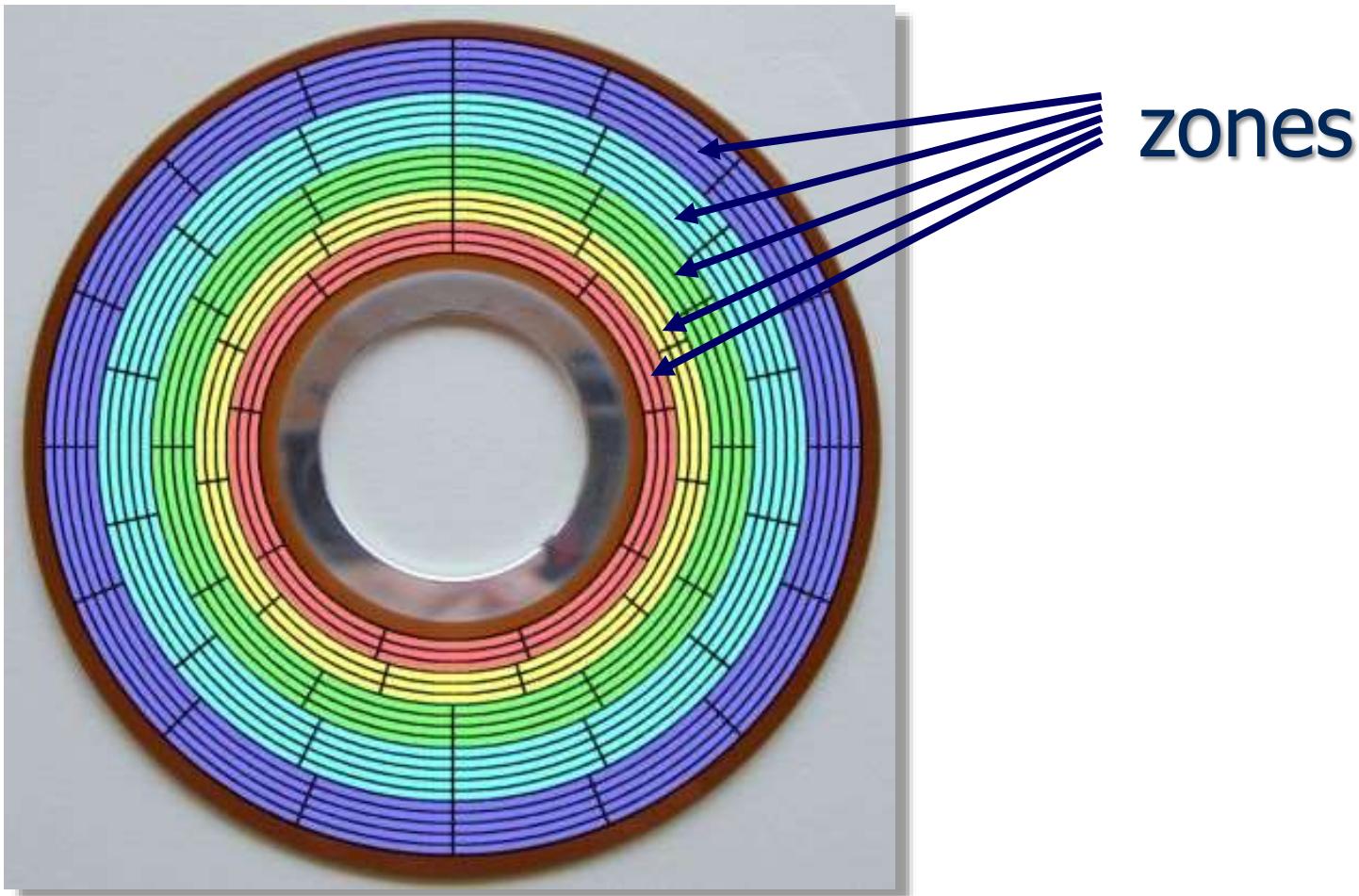
Sectornummering

- Hoe kan het besturingssysteem een bepaalde sector opvragen?
 - Vroeger moest het besturingssysteem rechtstreeks het cilinder-, hoofd- en sectornummer opgeven
- Nadelen?
 - Besturingssysteem moet op de hoogte zijn van de geometrie van de schijf (aantal cilinders, hoofden en sectoren per spoor)
 - Hedendaagse SSD-schijven zijn niet magnetisch; cilinders en hoofden zijn daar niet van toepassing.

Sectornummering

- Hoe kan het besturingssysteem een bepaalde sector opvragen?
 - Tegenwoordig heeft elke sector een uniek nummer, het **LBA-nummer**
- Eerst was dit een 28-bit getal. Hoe groot kon de schijf dan maximaal zijn?
 - 2^{28} sectoren van elk 512 bytes = **128GB**
- Tegenwoordig 48-bits
 - 2^{48} sectoren van elk 512 bytes = **128PB**

Zone Bit Recording



Zone Bit Recording

- In de buitenste sporen kunnen meer bits staan (want de sporen zijn langer)
 - ZBR: aantal sectoren per spoor is groter aan de buitenkant
- Vb. zonder ZBR:
 - schijf met 10000 cilinders
 - 50 sectoren in elk spoor
 - Per oppervlak: 10000×50 sectoren, d.i. 500000 sectoren

Zone Bit Recording

- Vb. met ZBR:
 - Buitenste sporen bevatten 99 sectoren
 - Binnenste sporen bevatten 50 sectoren
 - Per 200 sporen, 1 sector verschil
 - Sp 0 t/m 199: 99 sectoren
 - Sp 200 t/m 399: 98 sectoren
 - ...
 - Sp 9800 t/m 9999: 50 sectoren
 - Dus $(50+51+\dots+99) \times 200 = 745000$ sectoren
 - +49%

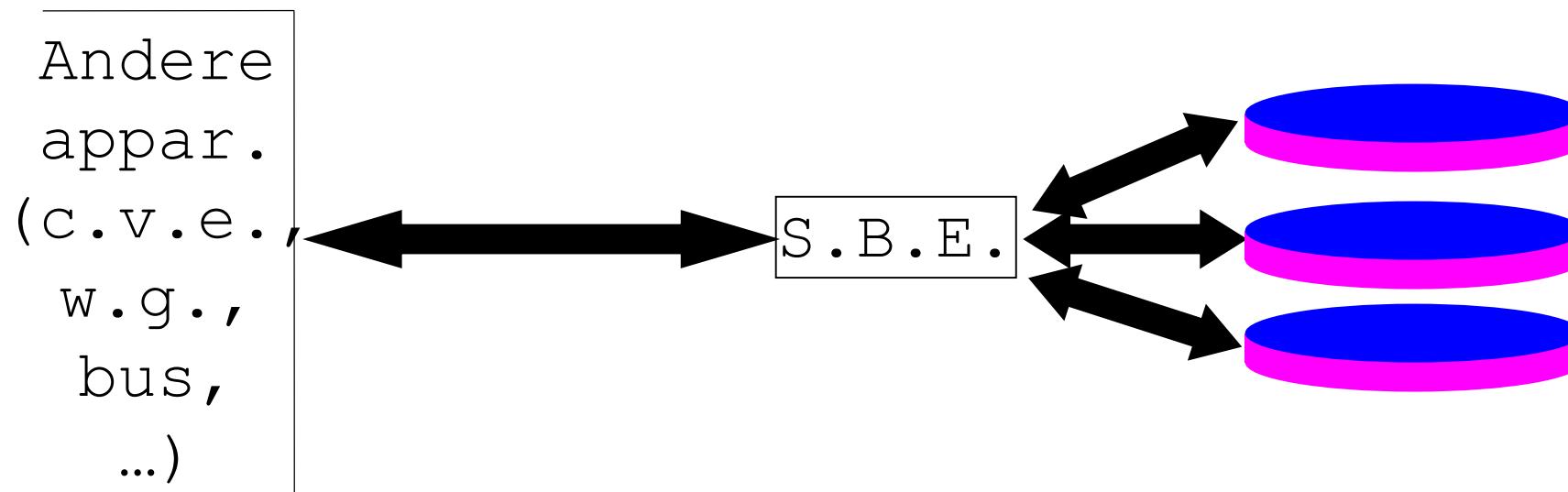


H5: Apparatuur

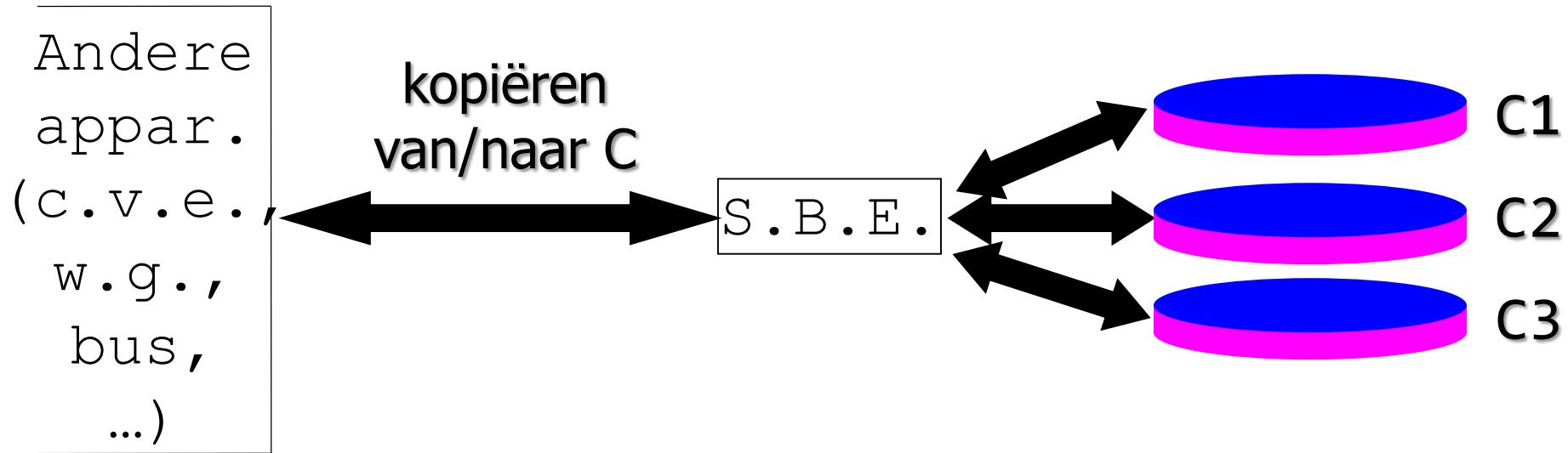
- **Magneetschijven**
 - Beschrijving, soorten, gebruik
 - Lees, schrijf
 - Schijfbesturingseenheid
 - Sectornummering
 - Zone bit recording
 - RAID

- Schijf moet **snel** zijn
 - Harde schijf is veel trager dan processor/RAM
- Schijf moet **betrouwbaar** zijn
 - Harde schijven *crashen* al wel eens
- Oplossing? **Redundant Array of Independent Disks (RAID)**.
 - SBE heeft meerdere schijven te zijner beschikking, CPU ziet alleen de SBE
 - SBE noemen we dan een **RAID-controller**

RAID Schema

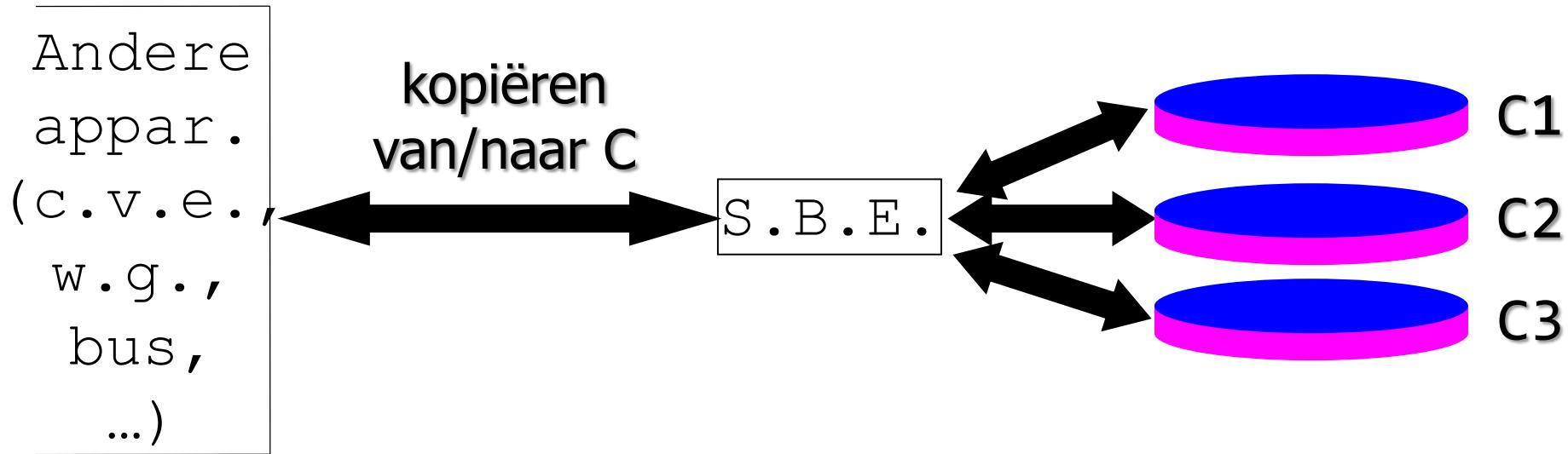


- **SBE heeft meerdere schijven te zijner beschikking, CPU ziet alleen de SBE**



- **CPU** gebruikt de harde schijf C en schrijft een bestand weg dat een aantal blokken groot is
- **SBE** verdeelt de blokken over bvb. 3 harde schijven C1, C2 en C3

RAID-0

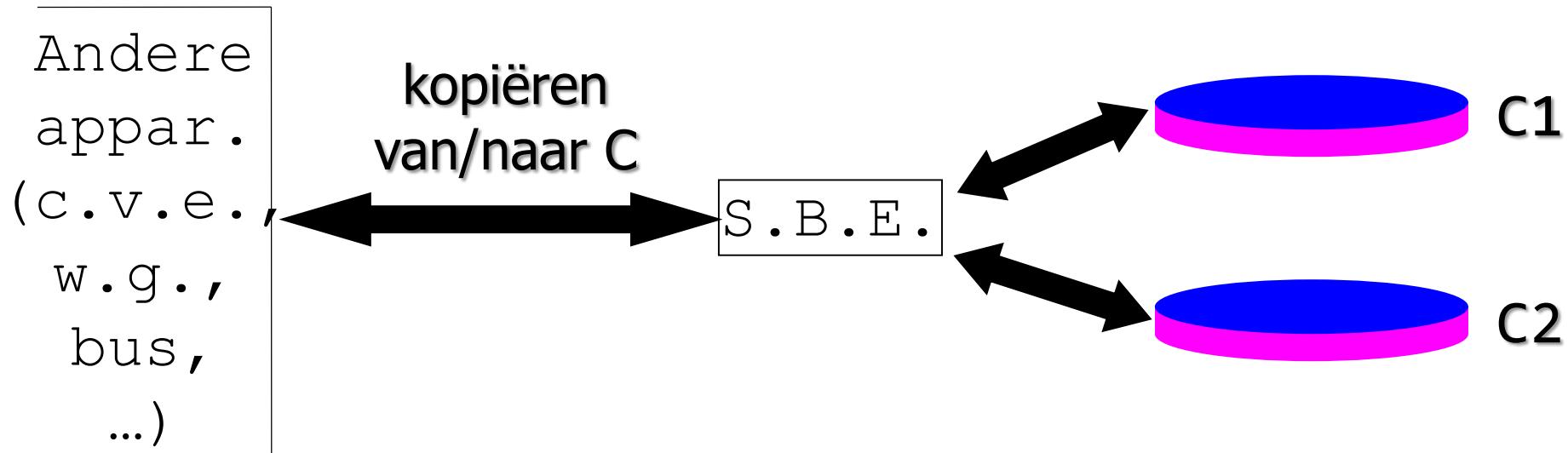


- 1^e blok op C1
- 2^e blok op C2
- 3^e blok op C3
- 4^e blok op C1
- ...

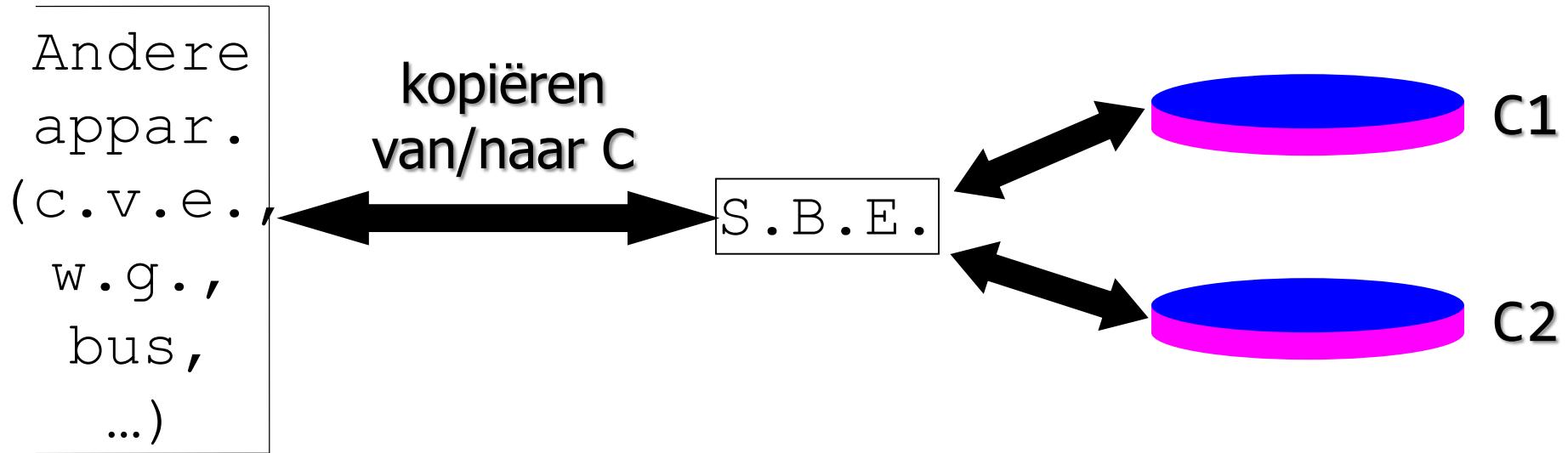
Resultaat?

- Lezen/schrijven gaat sneller
- Foutbestendigheid is kleiner

RAID-0 wordt ook *disk striping* genoemd



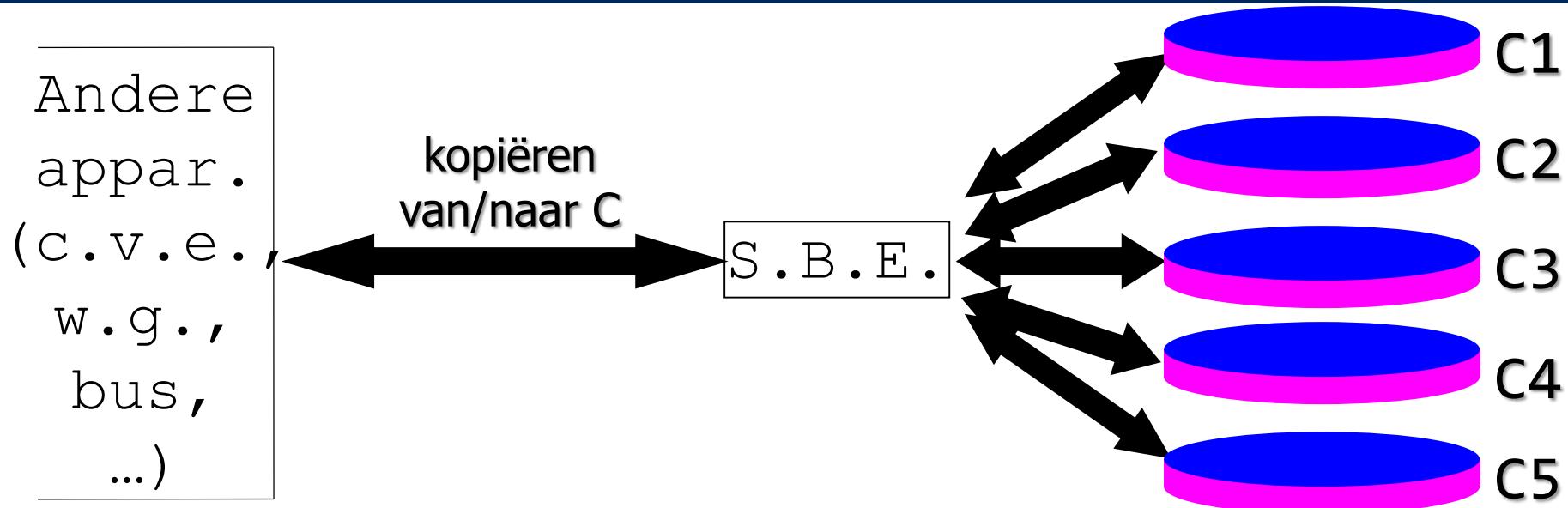
- **CPU** gebruikt de harde schijf C en schrijft een bestand weg dat een aantal blokken groot is
- **SBE** koptieert alle blokken naar bvb. 2 harde schijven C1 en C2
 - Resultaat? Beide schijven zijn identieke kopieën.



Resultaat?

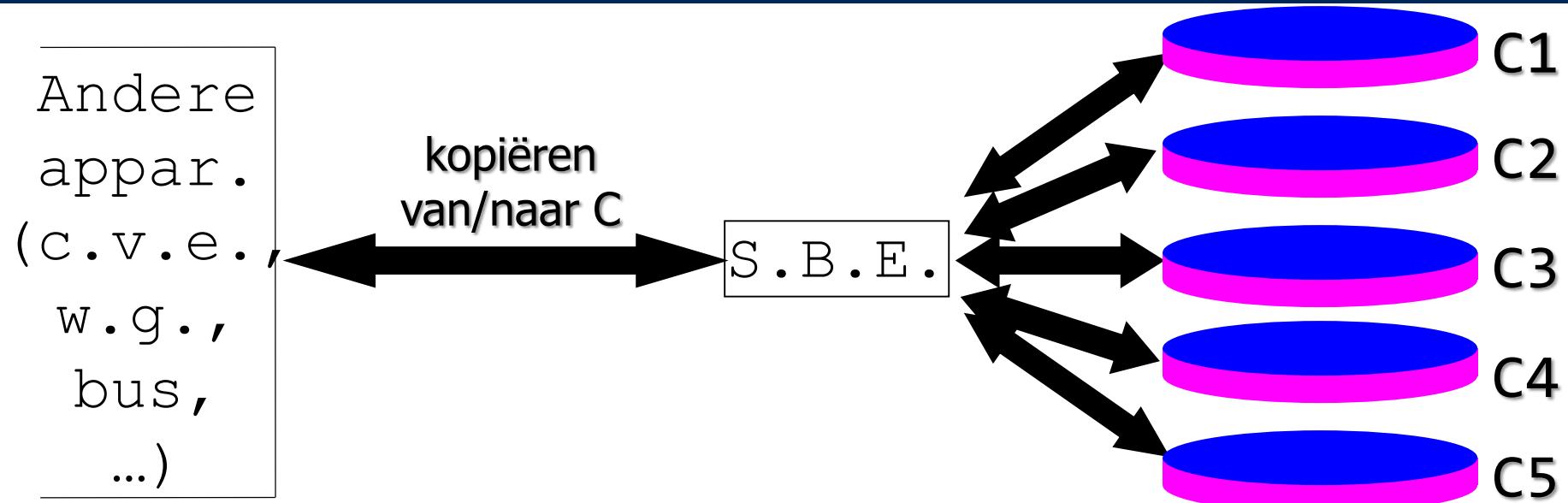
- Één schijf defect? Geen probleem...
- Schrijven is even snel
- Lezen gaat sneller

RAID-1 wordt ook *disk mirroring* genoemd



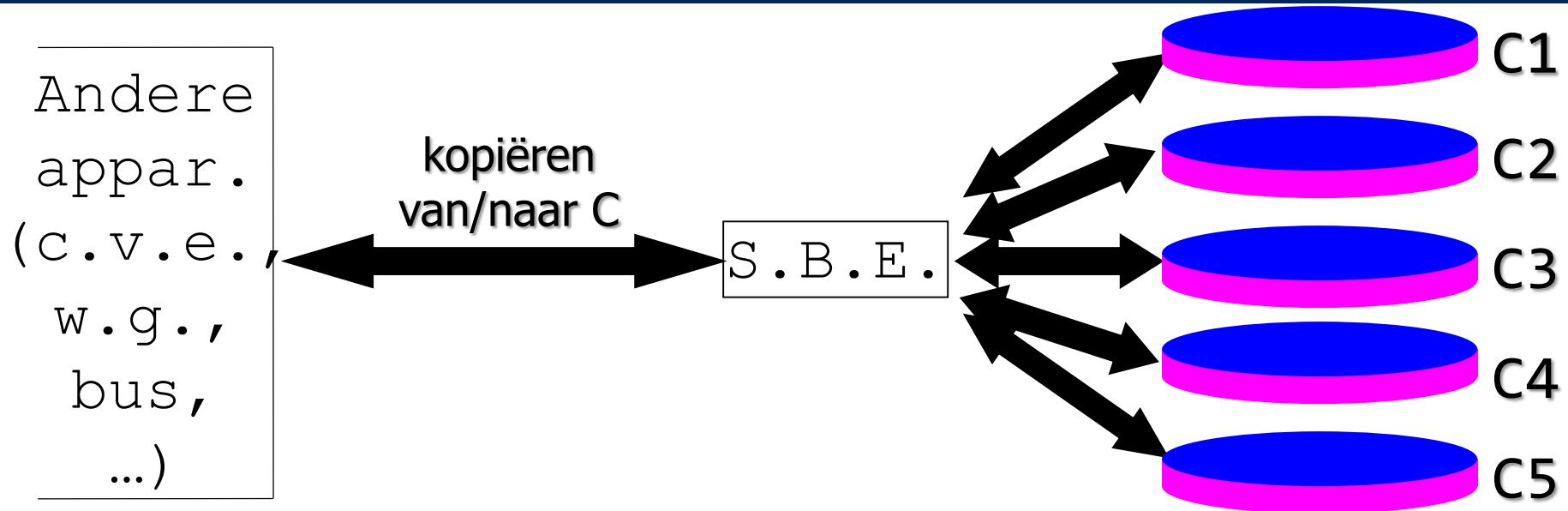
- **CPU** schrijft een bestand weg
- **SBE** koptieert alle blokken naar bvb. 5 harde schijven
 - Data wordt verdeeld in groepen van 4 blokken
 - Per 4 blokken wordt er een 5e **pariteitsblok** berekend

RAID-5



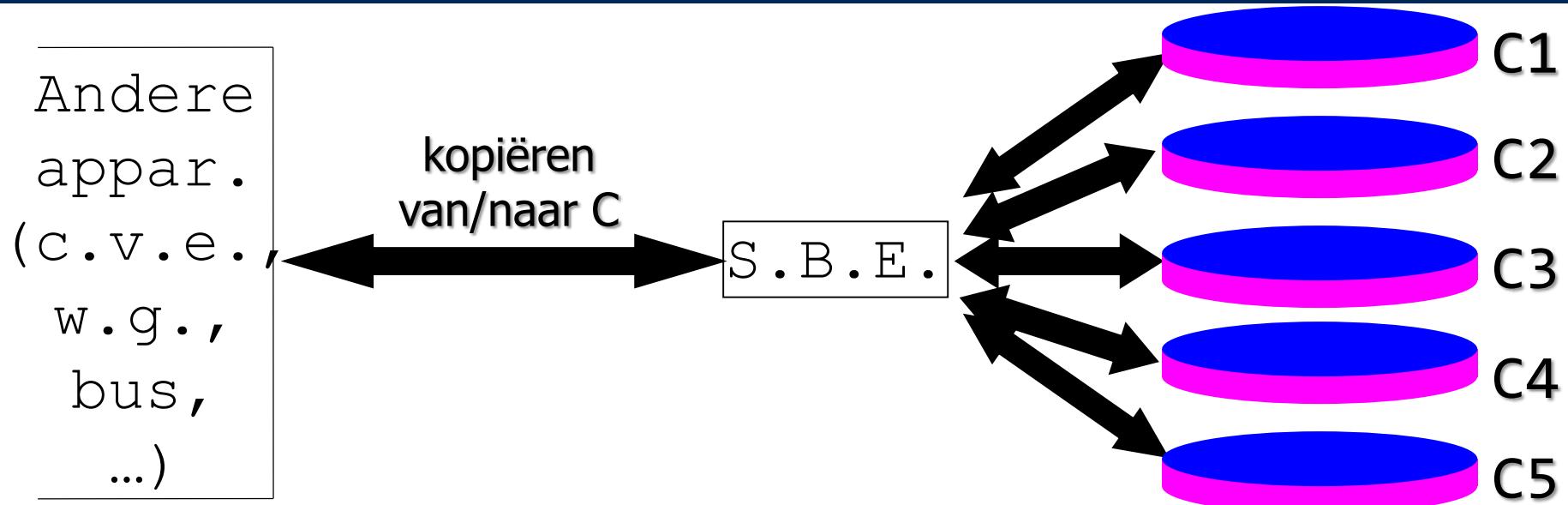
- $1^\circ \ 2^\circ \ 3^\circ \ 4^\circ \rightarrow \text{pariteit}$
 $1 \ 1 \ 1 \ 1 \rightarrow 0$
 $1 \ 0 \ 1 \ 1 \rightarrow 1$
 ...

RAID-5



- Pariteitsblok niet altijd op dezelfde schijf

1 ^e blok	2 ^e blok	3 ^e blok	4 ^e blok	P(1,2,3,4)
5 ^e blok	6 ^e blok	7 ^e blok	P(5,6,7,8)	8 ^e blok
9 ^e blok	10 ^e blok	P(9,10,11,12)	11 ^e blok	12 ^e blok
...



Resultaat?

- Lezen/schrijven gaat sneller
- Een schijf defect? Geen info verloren...

$$\begin{array}{cccccc} 1^\circ & 2^\circ & 3^\circ & 4^\circ & \rightarrow & \text{pariteit} \\ 1 & 1 & ? & 1 & \rightarrow & 0 \end{array}$$