

SOCS EXAM 2022-2023

Prof B. De Decker

Liam Luys

KU Leuven

Veel succes met het examen!

Ik heb geprobeerd een samenvatting te maken die alle info bevat.
Geen garantie dat alles er in staat en/of juist is, doe zeker de boeken ook open.

DEEL 2: COMPUTERAPPARATUUR

2. Computergeheugens en Randapparatuur

2.2 Digitale Logica

Logica

- Booleaanse waarde
 - o Waar of onwaar
 - o 1 of 0
 - o Elektrisch: 5V voor "waar", 0V voor "onwaar"
 - o ...
- Booleaanse Bewerkingen
 - o Niet, En, Of, EOF (exclusieve of)...
 - o Waarheidstabellen kunnen maken

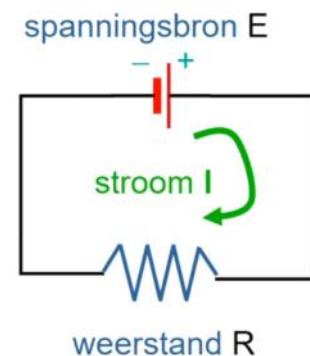
Begrippen uit elektronica

- Spanningsbron E (Volt)
- Weerstand R (Ohm)
- Stroom I (Ampère)

Elektrische stroom

- Transport van elektrische ladingen doorheen medium
- $I = E / R$ (Wet van Ohm)

Stroom omgekeerde richting van elektronen!



Elektrische Spanning

- Potentiaalverschil tussen twee punten
- Hoeveelheid arbeid nodig om eenheidslading van het ene naar het andere punt te brengen

Weerstand R afhankelijk van

- Materiaal
- Temperatuur
- Geometrische Vorm

Atomen en Kristallen

Silicium = Grondstof voor alle chips

- 14 elektronen
- 4 valentie-elektronen

Rooster vormen

- Streven naar 8 valentie-elektronen
- Door gemeenschappelijk stellen van elektronenparen

2.2.2 Transistoren

Klassen in Materialen

- Isolatoren
 - o Elektronen zeer vast in de structuur, niet geleidend
 - o Rubber, ...
- Geleiders
 - o Elektronen los in de structuur, mogelijkheid om te bewegen
 - o Koper, staal, aluminium

- Halfgeleiders
 - o In normale omstandigheden niet-geleiders
 - o Geleidende eigenschappen door toevoeging vreemde atomen (*Doperen van halfgeleiders*)
 - o 2 Mogelijke gevolgen voor structuur
 - n-type: Plaats voor extra e-
 - p-type: Overschot van extra e-
 - o Combinatie van beide types
 - Diode (p-type + n-type)
 - Stroom van p- naar n-type
 - Worden nog zelden gebruikt
 - Transistor (p-type + n- type + p-type)
 - Sandwichstructuur
 - o Silicium, germanium, ...

Diode

- Elektrisch schema voor basisbewerkingen

Wat als p-type en n-type samenkomen?

- Elektronenoverdracht waardoor ladingen ontstaan.
 - o p-type: negatief geladen
 - o n-type: positief geladen

Transistor

- Gebruikt als schakelaar in elektrische schakelingen
- Samenvoeging van gedopeerde halfgeleiders
- 3 Aansluitingen / Connectoren
 - o Collector C
 - o Emitter E
 - o Basis B
- Elektrische schakelingen

Schakeltijd = Tijd om van ene toestand naar de andere te gaan.

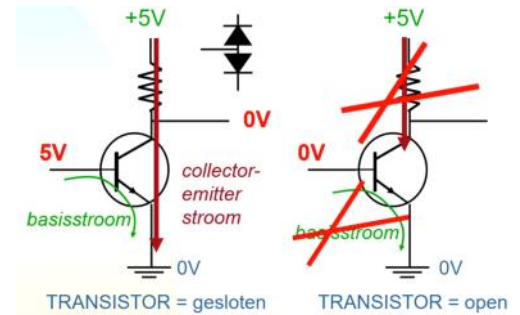
Poort = Schakeling met één ingang en één/meerdere uitgangen

Soorten Transistoren

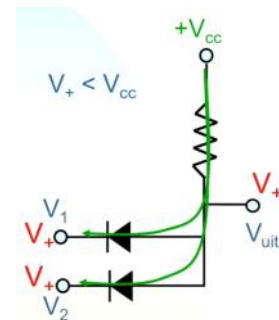
- Bipolaire Transistoren
 - o Beide polariteiten spelen een rol, zowel negatieve deeltjes, elektronen als positieve gaten.
 - o p-n-p en n-p-n transistoren zijn bipolair
 - o Voorbeelden bipolaire transistoren
 - TTL: Transisto-Transistor Logic
 - ECL: Emitter Coupled Logic
- MOS Transistoren
 - o Metal Oxide Semiconductor
 - o 3 Varianten
 - NMOS: Transistor sluit als basis positieve spanning heeft
 - PMOS: Transistor sluit als basis geaard wordt
 - CMOS: Energiebesparende combo van NMOS en PMOS

Verschillen Bipolair en MOS?

- MOS nemen minder plaats in
- MOS verbruiken minder energie
- MOS minder storend op naburige transistoren
- MOS zijn trager dan bipolaire
- MOS zijn eenvoudiger en goedkoper



NIET - Operatie via transistor



EN - Operatie via diode

Schakelingen Transistoren



2.3.3 Schakelalgebra

Basisbewerkingen

- Niet
- En
- Of

Aantal functies met n variabelen?

- Totaal aantal functies voorgesteld door 2^{2^n}
- Niet gemakkelijk om nog voor te stellen in waarheidstabellen

Oplossing?

- Voorstelling door formules ipv tabel.
- Formules vereenvoudigen dmv wetten in de algebra

Wet	EN-vorm	OF-vorm
Identiteit	$1.A = A$	$0+A=A$
Nulwet	$0.A = 0$	$1+A=1$
Idempotentie	$A.A = A$	$A+A=A$
Inversie	$A.\overline{A} = 0$	$A+\overline{A} = 1$
Commutatieve	$A.B = B.A$	$A+B = B+A$
Distributieve	$A+B.C = (A+B).(A+C)$	$A.(B+C) = A.B+A.C$
Absorptie	$A.(A+B) = A$	$A+A.B = A$
De Morgan	$\overline{A.B} = \overline{A} + \overline{B}$	$\overline{A+B} = \overline{A} . \overline{B}$

2.2.4 Geïntegreerde Schakelingen

Silicium --> Belangrijkste materiaal in schakelingen!

Er wordt van silicium door verschillende stappen een chip gemaakt.

- Zeer veel stappen om te fabriceren
- Bestaat uit vele schakelingen/Poorten die gecombineerd zijn en voorzien van invoer/uitvoer via pinnetjes

Afhankelijk van het aantal poorten in een de chip/processor een andere chip.

Nieuwste chips: Meer dan 57.000.000.000 poorten/transistoren in één processor

2.2.5 Wet van Moore

Voorspelling dat aantal transistoren op één chip zal om de 18 maanden verdubbelen

=> Aantal bits in één processor zal dus ook verdubbelen

Experten voorspellen

- Tegen 2025 zal deze wet niet meer gelden
- Nog kleiner en meer transistoren blijft niet gaan

Nagekeken en vervolledigd op 26-12-2022

Deel 2: RAM en ROM Geheugens

2.3 Geheugens met Onmiddellijke Toegang

Geheugens met onmiddellijke toegang

- Random Acces Memory - RAM
- Soorten RAM
 - o DRAM --> Dynamische RAM
 - o SRAM --> Statische RAM

2.3.1 Verleden: Ferrietkernegeheugens

Ferrietkernegeheugens

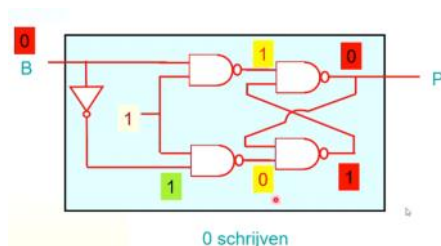
- Eerste geheugens
- Gebouwd uit magnetiseerbare ringetjes in matrixstructuur
- Voordelen tov RAM
 - o Niet Vluchtig
 - Indien spanning computer uitviel, bleef de inhoud bewaard
 - o Niet gevoelig voor stralingen
 - o Destructief gelezen
- Nadelen
 - o Niet compact
 - o Moeilijk productie
 - o Vrij traag
- Schrijven in geheugen
 - o Stroom door de 2 geleiders (in 2 richtingen)
 - o Ring wordt gemagnetiseerd, omgezet van 0 naar 1 of omgekeerd
- Lezen in geheugen
 - o '0' schrijven op de bit en spanning meten die wijzigt
 - Als er een 0 stond is er geen spanning
 - Als er een 1 stond wijzigt magnetisch veld en ontstaat spanning
 - o Terug herschrijven indien er een 1 stond

Oude geheugens: zeer groot qua oppervlakte (1MB in 1m³)

- Magnetische velden
- Bipolaire Flip-Flop
- Matrixstructuur
 - o Individuele bits
 - o Adressen

Bipolaire Flip-Flop

- Schakeling met 2 stabiele toestanden (1 en 0)
- Gemaakt met 2 NEN-poorten die verbonden worden
- Lezen
 - o Spanning meten
- Schrijven
 - o Een 1 schrijven?
 - Bovenaan een 0 aanleggen
 - o Een 0 schrijven?
 - Onderaan een 0 aanleggen



Blauwe kader = 1 BIT die de waarde '0' of '1' kan hebben
Geheugen = Matrix van zeer veel bits aan elkaar gekoppeld.

2.3.2 Statisch Geheugen - SRAM

SRAM

- Static RAM
- Bouwstenen: Bipolaire Flip-Flops
- Gebruik
 - o CMOS SRAM (batterijvoering: weinig energie. Vb voor de klok die doorloopt ook zonder stroom)
 - NOVRAM --> Niet vluchtig RAM
 - Wordt onder constante batterij-spanning gehouden om te blijven werken: BIOS-instellingen, datum en tijd, randapparatuur instellingen, ...
 - o Voorgeheugens (caches: snelheid)
- Eigenschappen
 - o Zeer snel
 - o Groot
 - o Zeer dure productie

2.3.3 Dynamische Geheugens - DRAM

DRAM

- Dynamic RAM
- Bouwstenen: condensatoren (*zeer klein*) met FET-transistoren
 - o Geladen: 1
 - o Ontladen: 0
- Voordelen tov SRAM
 - o Bit neemt 4 keer minder plaats in
 - o Veel lagere energieconsumptie
 - o Kleiner in oppervlakte, compacter
 - o Veel goedkoper te maken
- Nadelen
 - o Extra schakelingen voor opfrissing van geheugencellen
 - o Lezen is destructief (herschrijven als je leest)
 - o Trager, vanwege opfrissing en herschrijven
 - o Lekstroom in condensatoren, isolator zo dun dat het niet perfect werkt.
 - Nood aan opfrissing, herschrijven van info omdat lekstroom info verliest
 - Elke 20 ms de bit lezen en terug herschrijven
 - o Omgevingsgevoeliger voor straling, ...

2.3.4 Geheugenchips, -modules, -bestuurder

Geheugen = Matrix van bits

Bits in geheugen

- selecteren adhv rij en kolom
- Verdubbelen om de 18 maanden in chips (*Wet van Moore*)

Geheugenmodules

- Organisatie van DRAM in één module
- Kan geplugd worden in moederbord
- Soorten modules
 - o SIMM (*Single Inline Memory Module*)
 - Printplaat: meestal 8 of 16 chips
 - Printplaat aan één zijde contacten
 - 72 contacten, max 32 bits tegelijk lezen/schrijven
 - Onder hoek van 30° in moederbord
 - o DIMM (*Dual Inline Memory Module*)
 - Printplaat: meestal 8 of 16 chips
 - Printplaat aan beide zijdes contacten
 - 168 contacten, max 64 bits tegelijk lezen/schrijven (*nieuwere versies tot 240 contacten*)
 - Loodrecht in moederbord

Deze modules dienen als geheugen en plug je in aan de computer zelf

Geheugenbestuurder

- Verbinding tussen Geheugenmodules en het moederbord
- Gelegen in een chip van het moederbord
- Taken
 - o Ontvangen van lees- en schrijfoopdrachten van processor
 - o Ontleden van adressen
 - o Aansturing geheugenmodules met specifieke instructies

2.3.5 Eigenschappen van geheugens

Capaciteit (max aantal gegevenbits)

- Kan lager zijn dan aantal bit-posities voorzien

Snelheid (Snelheid van het geheugen)

- Technologie
- Soort RAM-geheugens
- Geheugenbestuurder

Toegangstijd (Tijd tussen lezen en resultaat)

- Door decodering adressen, schakeltijd van poorten
- SRAM: 10 ns
- DRAM: 35 ns

Geheugencyclustijd (Tijd tussen volgende opdrachten)

- Herschrijven, opfrissen, uitdoven van resterende signalen = parasiet signalen.
- SRAM: 12 ns
- DRAM 70 ns

1 ns = 1 miljardste van seconde

Optimalisaties voor snelheid

- Geheugenspreiding
 - o 2 Geheugenmodules die je tegelijk kan oproepen
 - Even adressen
 - Oneven adressen

2.3.6 Verbeterde DRAM

- EDO (*niet op examen*)
- SDRAM (*niet op examen*)

--> Beide technieken om meer te kunnen lezen/schrijven in dezelfde tijd

2.3.7 Nieuwe Ontwikkelingen

MRAM = Magneto-Resistieve RAM

- Informatie wordt magnetisch opgeslagen in het geheugen
- Bit bestaat uit twee gescheiden magnetiseerbare lagen
 - o Vaste onderlaag
 - o Veranderbare bovenlaag --> Bepaald de bit
 - Beide lagen gelijk gemagnetiseerd --> Geen weerstand --> 0 - bit
 - Beide lagen anders gemagnetiseerd --> Wel weerstand --> 1 - bit
- Bits in matrixstructuur verbonden door
 - o Bitlijnen
 - o Wordlijnen
- Lezen in MRAM
 - o Stroom door bit sturen en vergelijken
- Schrijven in MRAM
 - o Stroom door bit- en woordlijn te sturen
 - o Stroom zal bovenste laag van de bit magnetiseren in de andere richting

Eigenschappen MRAM

- Compactheid van DRAM
- Snelheid van SRAM
- Niet-vluchtigheid van Flash-geheugens --> Ook zonder spanning behoud van geheugen
- Zeer hoge kostprijs

Toepassingen MRAM

- IoT-apparaten
 - o Internet Of Things
 - o Kleine hoeveelheid geheugen nodig
 - o Spaarzaam met energie

2.4 Niet Vluchtige geheugens - Leesgeheugens

Leesgeheugens (ROM)

- ROM = Read Only Memory
- Verschillende varianten
 - o ROM --> Bij constructie gevuld
 - o PROM --> Eénmaal beschrijfbaar
 - o EPROM --> Kan gewist worden via Uv-licht
 - o EEPROM --> Ter plaatse wissen, of FLASH (in grotere hoeveelheid wissen) en herschrijven

2.4.1 Read-Only Memory - ROM

- 1 en 0 voorgesteld door het aanwezig/afwezig zijn van een transistor
 - o Bij meten van de spanning kan men bepalen welke bit het is
- Bestaan dus uit 1 transistor per geheugenbit
 - o Compacter dan DRAM-geheugens
- Lezen is niet destructief
 - o Kortere cyclustijd omdat ze niet moeten herschrijven
 - o ROM moet nooit opgefrist worden

Alles wat in ROM staat is vast, blijft staan en kan nooit worden aangepast

Waarvoor ROM gebruiken?

- Gegevens die altijd aanwezig moeten zijn in PC
 - o Boot program --> Opstarten van computer

2.4.2 Programmable ROM - PROM

- Eénmaal schrijfbaar met speciaal apparaat
- Elke geheugenbit bevat een zekering die kan opgeblazen worden door programmeur
 - o Opgeblazen wordt een 0-bit, omdat er geen spanning meer zal zijn
 - o Niet-opgeblazen zekeringen blijven een 1-bit

Waarom PROM gebruiken?

- Voor bedrijven die eigen ROM willen maken met eigen software

2.4.3 Erasable PROM - EPROM

- Geheugen kan gewist en herschreven worden
- Wissen door uv-licht in apparaat
 - o Chip moet dus uit de computer worden gehaald!
- Schrijven vergelijkbaar met PROM
- Hoe meer herschrijven, hoe slechter de chip (max 100 keer!)

2.4.4 Electrically Erasable PROM - EEPROM

- Meest flexibel leesgeheugen
- Kunnen ter plaatse gewist en herschreven worden
 - o Chip moet niet meer uit de computer!

Nadelen EEPROM

- Half zo snel als de EPROM
- Kleinere capaciteit

Waarom EEPROM gebruiken?

- BIOS in de computer, kan geüpdatet worden met speciale programma's

Soorten EEPROM

- Full Featured EEPROM
- Flash EEPROM / Flash Memory / Flashgeheugen

2.4.5 Flashgeheugen

- Opgedeeld in geheugenblokken, wissen en schrijven op blokniveau
 - o Telkens een volledige blok wissen (wissen = bit op 1 zetten)
- Max 10.000 tot 100.000 herschrijven mogelijk

Waarom Flash Memory gebruiken?

- Veel elektrische apparaten
- Compact opslagmedium
 - o USB, Flash Disk,

2.4.6 Samenvatting Geheugens

	SRAM	DRAM	Flash	MRAM	FeRAM	PCRAM
Lezen	Snel	Medium	Medium	Snel	Snel	Snel
Schrijven	Snel	Medium	Traag	Snel	Medium	Snel
Vluchtig?	Ja	Ja	Neen	Neen	Neen	Neen
Opfrissen?	Neen	Ja	Neen	Neen	Neen	Neen
Grootte	Groot	Klein	Klein	Klein	Klein	Klein
Lage spanning	Ja	Beperkt	Beperkt	Ja	Beperkt	Neen
Duurzaam	Ja	Ja	Neen	Ja	Neen	Neen

Deel 3: Hulpgeheugens

2.5 Magneetschijven

2.5.1 Fysisch Principe

- Gegevens bijgehouden in Magnetiseerbare Laag

Schrijfmechanisme

- Stroom door geleider --> Elektrisch veld --> Veld magnetiseert materiaal in bepaalde richting
- Door stroom te zetten schrijven/veranderen we bits

Leesmechanisme

- Geleider beweegt tov magnetisch veld --> Spanning ontstaat
- Door de spanning te meten lezen we bits

2.5.2 Harde Schijf

Magneetschijfstation

- Bestaat uit meerdere Platen, kunnen opgebouwd zijn uit
 - o Aluminium platen
 - o Glazen platen
 - o Platen uit composietmateriaal
- Platen zelf zijn dus niet magnetiseerbaar!
 - o Platen zijn aan beide zijden bedekt met magnetiseerbare laag
 - "Dunne Film"
 - Ijzerlegering

Eigenschappen platen

- Platen moeten uiterst glad zijn
- Diameter plaat: 2,5 - 12 cm
- Zijn zelf niet magnetiseerbaar
 - o Beide zijden bedekt met "Dunne Film"
- Verticaal gestapeld en vast gemonteerd op een as
- Draaien met constante snelheid rond (4200 - 15000 toeren/min)
- Elke plaatzijde heeft zijn Lees- en schrijfkop
 - o Gemonteerd op arm

Logische Opdeling Platen

- Oppervlakte elke plaat verdeeld in sporen
 - o Ringen die onder de kop draaien voor bepaalde stand van de kop
 - o Buitenste sporen > Binnenste sporen
- Elk spoor onderverdeeld in Sectors

Harde Magneetschijf



13 / 29

Harde Magneetschijf



- Ringen die onder de kop draaien voor bepaalde stand van de kop
- Buitenste sporen > Binnenste sporen

- Elk spoor onderverdeeld in Sectors
 - Kleinste hoeveelheid die in één keer gelezen/geschreven kan worden
 - Tussen sectoren ongebruikte tussenruimte
- Schijf opgedeeld in zones (nieuwe techniek)
 - Elke zone evenveel sectoren
 - Vergroot capaciteit schijf

Cilinder

- Alle sporen die boven elkaar liggen
- Binnen één cilinder lezen --> Koppen niet verplaatsen

Opbouw Sectoren

- Hoofding HD
 - Nummer van de sector
- Data
 - Informatie op de schijf
- Fout corrigerende Code ECC
 - Kan kleine fouten verbeteren

Formatteren Schijf

- Moet gebeuren alvorens er info op te zetten
- Kan op 2 manieren
 - Laag-Niveau
 - Hoofding van elke sector wordt geschreven
 - Controle of sector correct gebruikt kan worden
 - Hoog-Niveau
 - Op de schijf een (lege) tabel wegschrijven
 - Wordt gebruikt door besturingsprogramma om info over schijfbestanden bij te houden

Lees- en Schrijfkoppen

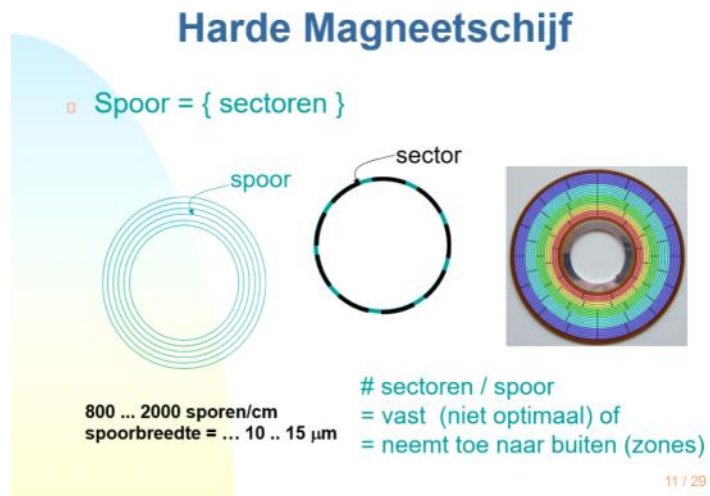
- Longitudinaal Schrijven
 - Zie cursus pagina 52
- Loodrechte Opname
 - Nieuwe techniek (2006)
 - Bits verticaal in magnetiseerbare laag opgeslagen
 - Verticaal schrijven ipv horizontaal
- Sterke evolutie doorheen jaren
 - Vroeger: Ferrietkoppen
 - Later: Dunne-Filmkoppen
 - Recent: Magneto-Resistieve Koppen (MR)
 - Veel kleiner geworden, dichter bij de plaat, meer bits, grotere capaciteit!
- Als schrijf- en leeskop niet in gebruik
 - Rusten op de ladingszone
 - Zone zonder data in het midden van de schijf
 - Rusten op helling (*nieuwere techniek*)
 - Weg van de platen
 - Veiliger

Behuizing

- Zeer belangrijk dat geen enkel stofdeeltje in platen komen
 - Koppen zweven direct boven platen, stof zou alles kapotmaken en blokkeren --> **HEAD CRASH**
- Nood aan sterke en (lucht)dichte behuizing
- Behuizing is luchtdicht (niet luchtleedig!)
 - Winchester-techniek

Snelheid Van Schijfstation

- Hangt af van verschillende factoren



- Zoektijd
 - Tijd om kop boven juiste spoor te brengen
- Rotatiewachttijd
 - Wachten tot juiste sector onder kop komt
- Transporttijd
 - Tijd om te lezen of schrijven
 - Dichtheid van de bits
 - Rotatiesnelheid schijf
 - Grootte van de sector
- Opstarttijd
 - Tijd om vanuit rust terug op snelheid brengen

Vroeger schijven veel te warm door rotatie --> Zette uit door hitte --> Moesten kalibreren --> Rampzalig voor snelheid

Schijfbestuurder (Controller)

- Chip die het schijfstation bestuurt
- Stuurt signalen van processor door naar de schijf
- Zorgt voor geheugentransport van en naar geheugen
- Heeft intern geheugen
 - Slaat sectoren van een spoor tijdelijk op
- Kan meer dan één schijf aansturen

2.5.3 Soepele Schijf / Diskette

- Manier om gemakkelijk gegevens te verspreiden

Eigenschappen

- Gelijkaardig aan de harde magneetschijf
- Gemaakt van buigzaam plastic
- Zit in een harde hoes
 - Onderaan opening voor lees- en schrijfkoppen
 - Afsluiting als schijf uit station wordt gehaald
- Koppen glijden over het oppervlak
 - Warmte ontstaat
 - Schijf draait altijd trager
 - Zal niet draaien wanneer er niet gelezen/geschreven wordt
 - Ontstaat sleet na bepaalde tijd
- Veel trager dan harde magneetschijf
- 2 Formaten van soepele schijven (beide in 2 groottes)
 - LD-schijven --> Lage densiteit/dichtheid
 - HD-schijven --> Hoge densiteit/dichtheid

Door gebrek aan apparaten die diskettes kunnen lezen gaat de informatie verloren!

2.5.4 RAID

Hoe de snelheid van magneetschijven opdrijven?

- Fysisch
 - Bits dichter bij elkaar,
 - Helpt beetje, maar niet veel!

Hoe schijf betrouwbaar maken?

- Bij kleine fout is alle data van de schijf verloren...
- Maken van Backups op andere media
 - Vergt veel tijd

RAID = Verzameling van manieren om betrouwbaarheid en snelheid te verhogen

- Spiegeling
 - Alle data op 2 schijven zetten
 - Capaciteit halveert!
- Ontdubbeling
 - Zowel schijven als bestuurder verdubbelen
 - Data blijft toegankelijk als bestuurder uitvalt

- Gebruik van stroken
 - o Data opdelen in kleinere stroken
 - o Maakt lezen en schrijven sneller
- Toevoegen van pariteitsstrook
 - o Berekenen van pariteitsstrook laat toe een kapotte schijf te recupereren
 - o Zie cursus pagina 59

RAID-Niveaus

- Alle eigenschappen onderverdeelt in verschillende niveaus
- Gebruiken fabrikanten om te produceren

2.5.5 Solid State Disk - SSD

- **Solid State = Geen bewegende delen**
- Schijf opgebouwd uit geheugenchips (*Flash Memory EEPROM*)
- Bevat bestuurder die de schijf aanstuurt
 - o Kan meerdere tegelijk lezen / Schrijven

DRAM SSD

- Nog sneller dan Flash SSD
- Gebruikt om snel en tijdelijk data op te slaan
- Verliezen data als spanning wegvalt (*RAM-eigenschap*)

Voordelen SSD

- Snel
- Leessnelheid overal gelijk
- Toegangstijd klein
- Betrouwbaar, geen bewegende delen die stuk kunnen
- Weinig energie, weinig warmte
- Zeer laag gewicht --> Geschikt voor laptops, ...

Beperkingen SSD

- Kostprijs
- Capaciteit (vroeger, nu niet meer)
- Flashgeheugen --> Verslijten na 5000-10000 overschrijvingen
- Bestuurder en firmware vrij complex

2.5.6 Varianten

Varianten op magneetschijven?

- ZIP
- HiFD
- JAZ
- ...

Niet verder kennen

2.6 Optische Schijven

- Oorspronkelijk voor film- en programma's
- Tegenwoordig voor muziek en opslag

Compact Disk - CD

- 1980 door Sony en Philips
- Vervanging van Vinylplaten
- Rode Boekje
- Diameter 12 cm

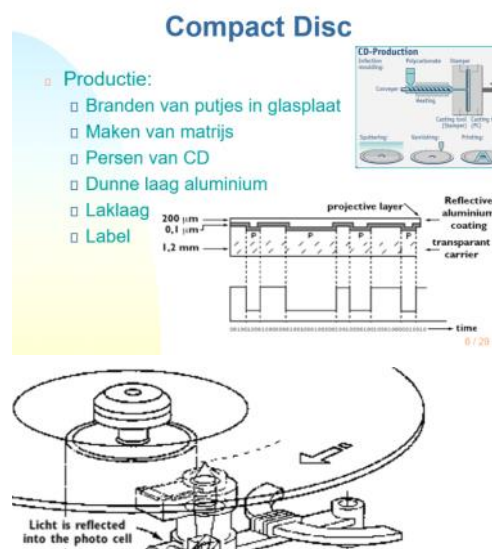
Maken van een CD

- Gaatjes gebrand in glazen plaat door infraroodlaser
- Via matrijs wordt de glazen plaat 'gekopieerd' op hars
- Kleine aluminium laag op hars met bescherming laag

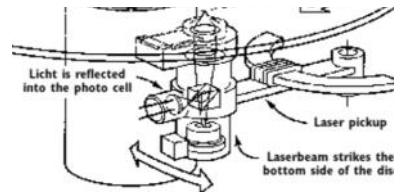
Bovenkant van CD is gevoeliger dan onderkant!

Afspelen van CD

- Infraroodlaser en reflectie van licht op de puntjes
- Door lichtbreking ontstaat verschil in reflectie bij punten
- Bits ontstaan
 - o 0 bij afwezigheid puntje
 - o 1 bij aanwezigheid puntje



- Door lichtbreking ontstaat verschil in reflectie bij punten
- Bits ontstaan
 - o 0 bij afwezigheid puntje
 - o 1 bij aanwezigheid puntje
- CD is een spiraal van puntjes, van binnen naar buiten
 - o Als je spiraal zou recht plooiën: 5,5 kilometer!
- Belangrijk dat snelheid constant blijft om muziek te spelen
 - o Constante lineaire snelheid behouden
 - Vertragen naarmate naar buiten gaan op CD



CLV = Constante Lineaire snelheid van de leeskop met verschil van binnen naar buiten

CAV = Constante Hoeksnelheid (*Tegenhanger CLV*)

2.6.1 CD-ROM

- Gele Boekje

CD-ROM = Compact Disk - Read Only Memory

- Zelfde fysieke kenmerken als audio-cd
- Moet veel betrouwbaarder zijn, mag geen fouten bevatten (*indien programma's worden bijgehouden*)

Enkele maatregelen

- Elke byte voorstellen als 14-bit symbool
 - o Fouten aan een bit kunnen hersteld worden
- Symbolen die instaan voor foutencorrectie van gegevens, besturing en synchronisatie
- Frames gegroepeerd in de sectoren
 - o Sector begint met
 - HD: 16 bits (*Hoofding*)
 - Modus (2 soorten)
 - Modus met fout-corrigerende bytes achter de data
 - ◆ 28% van de bits bevat werkelijke data
 - Modus zonder fout-corrigerende bytes (meer plaats voor de data)
 - ◆ 32% van de bits bevat werkelijke data
- In totaal dus 3 fout-corrigerende Systemen in CD-ROM
 - o Binnen symbool
 - Verbetering 1-bit fouten
 - o Binnen Frame
 - Verbetering opeenvolgende bits
 - o Binnen Sector
 - Overige fouten behandelen
- Standaardsnelheid CD
 - o 75 sectoren per seconde
 - o Data kan sneller worden gelezen door CAV en CLV te combineren
 - Niet zo makkelijk

2.6.2 CD-R

Compact Disk - Recordable

- WORM (*Write Once Read Many*)
 - o 1 keer beschrijfbaar
 - o Onbeperkt keer leesbaar
- Oranje Boekje
- Bevat extra groef om laser te helpen bij het schrijven
- Bits voorgesteld door doorgebrande puntjes in een kleurstoflaag
- Moet in één keer doorlopend beschreven worden

2.6.3 CD-RW

Compact Disk - Rewritable

- Herschrijfbaar
- Legering van ijzer ipv kleurstoflaag
 - o Kristallijn --> Goede reflectie

- Amorf --> Slechte reflectie

- Om CD te schrijven een speciale laser met 3 vermogens
- Faseverandering om te schrijven
- Duurder dan CD-R

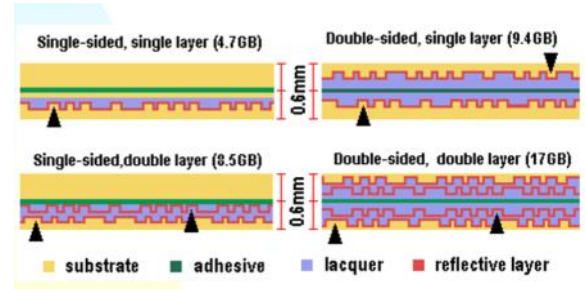
2.6.4 DVD

Digital Versatile Disc

- Meer capaciteit dan de CD's (7 keer meer)
- Belangrijke verschillen
 - Kleinere puntjes
 - Veel dichtere spiraal
 - Rode laser om schijf te lezen

Verskillende formaten

- Enkelzijdig één laag
- Enkelzijdig twee lagen
- Dubbelzijdig één laag
- Dubbelzijdig twee lagen



2.6.5 Uitbreiding DVD

Beschrijfbaar

- DVD-R --> WORM, kunststoflaag
- DVD+R

Herschrijfbaar

- DVD-RAM
- DVD-RW
- DVD+RW

2.6.6 HD DVD en Blue-Ray

High Definition DVD

- Toshiba en NEC
- 3 keer grotere capaciteit (15GB/laag)

Blue-Ray Disk

- Sony
- Grote capaciteit (25GB/laag)

Oorlog tussen beide, na lang won in 2008 de Blue-Ray

2.6.7 Magneto-Optische Schijven

- Combinatie van magnetische en optische technologie
- Laser bij het schrijven, bits magnetisch bijgehouden

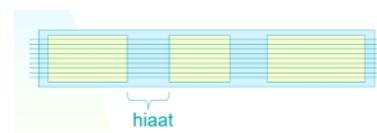
Verschillen met Harde Magneetschijven

- Magnetisatie verticaal
- Bits gelezen met laser
- Stabiele magnetiseerbare laag --> Wijzigen dmv curietemperatuur te bereiken met laser

2.7 Magneetbanden en Magneetcassettes

2.7.1 Magneetbanden

- Oudste geheugenmedia
- Plastieken lint bedekt met magnetiseerbare laag
- Bits gerangschikt in sporen, bits uit 1 kolom worden tegelijk gelezen.
- Informatie opgeslagen in blokken van variabele lengte
 - Tussen blokken een hiaat
 - Afremmen en terug op snelheid brengen van de band
- Sequentiële geheugens
 - Moeten in volgorde gelezen en geschreven worden



Gebruik Magneetbanden

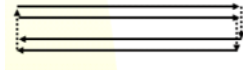
- Eerste computers, voor opslag van programma's of bewaren van resultaten

LTO Cartridge

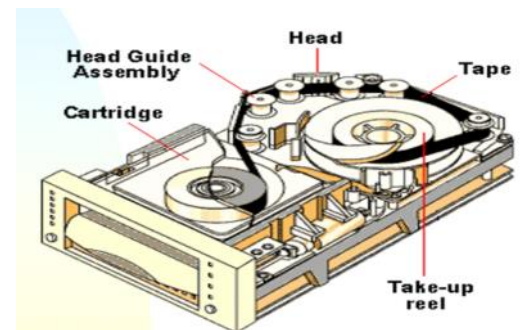
- Opslagmedium gebaseerd op de magneetband
- Recent, terugkeer van de magneetband dus!
- Zeer hoge capaciteiten
- Air-Gapped Security

2.7.2 Magneetcassettes

- Vervanging traditionele magneetbanden
- Gesloten behuizing, moeten behuizing niet verlaten
- Veel meer sporen dan magneetband
- Slangvormig schrijven



- Verschillende formaten / Types



2.8 Massageheugens

Geheugenmedia met zeer grote capaciteiten

- Trager toegankelijk
- Opgebouwd uit groot aantal verwijderbare media (*cd, cassettes, LTO-banden, ...*)
- Wordt nu nog uitzonderlijk gebruikt als back-up materiaal.

Nagekeken en vervolledigd op 29-12-2022

Deel 4: Communicatieverbindingen

2.9 Gegevenstransport

Uitwisseling van gegevens in een computer

- Instructies overbrengen van geheugen naar processor
- Resultaten van processor terug naar geheugen
- ...

Soorten Gegevenstransport

- Serieel
 - o Bits één voor één overbrengen
- Parallel
 - o Meerdere bits tegelijk overbrengen
 - o Moet meer dan één verbinding zijn
 - Is duurder
- Punt-tot-puntverbinding
 - o 2 Componenten met elkaar verbinden
- Busverbinding
 - o Gemeenschappelijk gebruik door meer dan twee componenten
 - o Goedkoper
 - o Beperking qua transport

2.10 Busverbinding

Bus = Gemeenschappelijk pad tussen meerdere componenten

Soorten verbindingen / draden in bus

- Adresbussen
- Gegevensbussen / Databussen
- Controlebussen
- Gemultiplexte Bussen

2.10.1 Busbreedte

Busbreedte = Totaal aantal verbindingen

- Hoe breder, hoe duurder

Breedte van de adresbus

- Bepaalt adresbereik, hoeveel adressen mogelijk zijn

Databus

- Gewoonlijk 8, 16 of 32 lijnen
- Hoe meer lijnen, hoe sneller transport

Breedte van Controlebus

- Hangt af van aantal opdrachten die bus moet overbrengen

2.10.2 Busprotocol

Busprotocol = Gebruiksregels van een bepaalde bus

Participanten (deelnemers aan een bus)

- Meester = Apparaat dat communicatie initieert
 - o CPU, DMA, ...
- Slaaf = Andere partij

- Geheugen, ...
- Hoe toegang tot de bus? Hoe meester worden?
- Welke signalen via welke lijn?
- Timing van signalen? Hoe lang en wanneer?
- Hoe bus vrijgeven?

Buscyclus

- Tijdens communicatie wordt de bus nooit voor iets anders gebruikt.

2.10.3 Synchrone of Asynchrone Bussen

Werking Bussen

- Synchroon
 - Op cadans van een klok
 - Elke toegang duurt even lang
- Asynchroon
 - Bus toegang slechts hoe lang het nodig is
 - Niet verder relevant in cursus

Bekijk cicly in cursus onderaan pagina 86

2.10.4 Busarbitrage

Arbitrage = Conflict

Participanten (deelnemers aan een bus)

- Meester = Apparaat dat communicatie initieert
 - CPU, DMA, ...
- Slaaf = Andere partij
 - Geheugen, ...

Enkel de meester kan een busoperatie starten!

Wat als meerdere componenten tegelijk de bus willen gebruiken?

- Nood aan een regeling, een mechanisme voor voorkomen busarbitrage

Mechanisme (2 soorten)

- Gecentraliseerd
 - Schakeling in CPU, arbiter chip
 - De chip bepaalt wie de bus gebruikt als "scheidsrechter"
 - Vraaglijnen
 - Vergunningslijnen
- Gedecentraliseerd
 - Componenten bepalen onderling wie bus eerst gebruikt

2.10.5 Busoperaties

Bus Cycli

- Lees cyclus
- Schrijf cyclus
- Block Transport cyclus
- Lees/Wijzig/Schrijf cyclus
- Programmaonderbreking cyclus
-

Deel 5: Opdrijven van de Snelheid

2.11 Voorgeheugens

2.11.1 Het Principe

RAM geheugen is relatief traag tov de CPU.

Cache = Veel sneller geheugen dat een kopie bevat van bepaalde info van het RAM-geheugen.

Hoe snelheid groot opdrijven?

- Combinatie van verschillende geheugens
 - o Snel maar duur geheugen samen met groot trager geheugen
 - o Voordelen van beide geheugens combineren

Werkingsprincipe Voorgeheugen

- Indien processor de inhoud nodig heeft van een geheugen, kijkt die eerst of de inhoud in het voorgeheugen zit
- Is dat het geval? *TREFFER*
 - o Dan kan voorgeheugen snel ter beschikking worden gesteld.
- Is dat niet het geval? *MISSER*
 - o Moet het hoofdgeheugen aangesproken worden, gevraagde wordt ook in voorgeheugen gebracht met omliggende geheugenregisters

Lokaliteitsprincipes

- Lokaliteit op tijd
 - o Als je iets op tijdstip t nodig hebt, zal je het ook nodig hebben op een volgend tijdstip $t+dt$
 - o Vb: Lussen, tussenresultaten
- Lokaliteit in ruimte
 - o Als je iets (x) op tijdstip t nodig hebt, zal je later iets in de buurt ($x + dx$) nodig hebben op tijdstip $t+dt$
 - o Vb: Programma's, rijen, velden van records,
 - o Steeds hele regel overbrengen naar cache om zeker hetgeen 'in de buurt' te hebben

Zelf computer ontwerpen

- Hoe groot cache?
- Welke regellengte?
-

2.11.2 Organisatievormen

Voor de lokaliteitsprincipes wordt het geheugen en het voorgeheugen opgedeeld in regels

Label nodig om te kijken of een regel in voorgeheugen zit

Organisatie van voorgeheugen

- Direct afgebeeld voorgeheugen
 - o Elke regel heeft vaste plaats in de cache
 - o Plaats krijgt label, groep, nummer
- Associatief voorgeheugen
 - o Elke regel kan op elke plaats in de cache staan
 - o Elke regel uniek label
- Set-Associatief voorgeheugen
 - o Combinatie van vorige twee
 - o Elke regel kan op enkele plaatsen in de cache staan
 - o Deelverzamelingen waarin label zit

2.11.3 Schrijven in Voorgeheugen

Voorgeheugen = Kopie van een deel van het hoofdgeheugen

- Enkel schrijven in het voorgeheugen zou dat tegenwerken!

Verschillende Mogelijkheden

- Schrijven doorheen het voorgeheugen
 - o Zowel voorgeheugen als hoofdgeheugen aanpassen
- Uitgesteld wegschrijven
 - o Enkel in voorgeheugen aanpassen, later ook de hele regel aanpassen in hoofdgeheugen

2.11.4 Vervangingsstrategieën

Wat als voorgeheugen vol zit en er moet een nieuwe regel in?

- Plaats maken noodzakelijk
- Keuze is afhankelijk van situatie
 - o Soms ligt het vast
 - o Vaak de regel die langst niet meer werd gebruikt
 - o Regel die al langst in voorgeheugen zit
 - o Willekeurige regel
 - o ...

2.11.5 Ontwerpparameters

Parameters bij ontwerpen van voorgeheugen?

- Grootte van het voorgeheugen
- Regellengte
- Organisatie van voorgeheugen
- Gezamenlijk of gesplitst voor data en instructies?
- Aantal voorgeheugens

2.11.6 Prestatieverbetering

Trefkans = Hoe vaak wordt info gevonden in het voorgeheugen?

- Hangt van vele factoren af

Resultaat: Toegangstijd 7 keer sneller dan zonder voorgeheugen!

2.12 Pipelining

Pipelining = Techniek die snelheid processor kan opdrijven zonder aanpassing kloksnelheid

Verschillende Stappen bevelencyclus

- Ophalen van instructie en ophogen BT
- Analyse van de instructie
- Berekenen adres operand
- Ophalen operand uit geheugen
- Uitvoeren van de instructie

Soort lopende band waarin verschillende stations de verschillende opdrachten afzonderlijk uitvoeren.

2.12.1 Het Principe

Voorstellen als lopende band met verschillende stations die de stappen uit de bevelencycli uitvoeren

2.12.2 Prestatieverbetering

Zonder Pipelining

- Elke instructie duurt even lang

Met Pipelining

- Overlappenden tussen ophalingen en acties
- Processor werd 5 keer sneller

2.12.3 De Complexe Werkelijkheid

Wat maakt het complex?

- Instructies kunnen afhankelijk zijn van voorgaande instructies
- Na elk sprongbevel staan onnodige instructie

Afhankelijkheden tussen instructies

- Mogelijke Oplossingen?
 - o Detectie van afhankelijkheid door hardware
 - o Afhankelijkheden niet toestaan

Sprongbevelen

- Mogelijke Oplossingen?
 - o Alle bevelen na sprongbevel uitvoeren
 - o Zodra sprongbevel komt geen enkel bevel meer uitvoeren
- Voorwaardelijke Sprongen
 - o Nog moeilijker
 - 2e Lopende Band
 - Voorspellen wat gaat komen

2.13 Instructiesetarchitectuur

Processor sneller maken

- Kloksnelheid opdrijven
 - o Instructies worden sneller uitgevoerd
 - o Kan tot bepaald punt
 - Schakelingen hebben minimale tijd om resultaat te bekomen
 - o Meer elektriciteitsverbruik
- Processor Eenvoudiger maken
 - o Minder stappen nodig voor een instructie --> Snellere instructies
 - o Aantal schakelingen verminderen
 - o Instructies die weinig worden gebruikt weglaten uit programma
 - Verschil in processoren

Soorten Processoren

Complex Instruction Set Computer - CISC

- Zeer uitgebreide instructieset
- Vele adresseringsopties: (in)direct, operand, ...
- Uitvoering heeft veel stappen nodig
- Compatibel met vorige versies van programma's
- Dichter bij HPT

Reduced Instruction Set Computer - RISC

- Beperkte instructieset
- Klein aantal adresseringsopties
- Instructies geheugentoegang gescheiden van de berekeningen
- Compiler complexer
- Kortere instructies, minder stappen
- 80% vd bevelen in programma's = 20% instructies bevelenset (Compilatorenbouwers)
- Soorten RISC
 - o MIPS
 - o SPRAC
 - o ARM

2.14 Multi-Processoren

Hogere kloksnelheid --> Hoger verbruik + Meer warmte

Wat als we nog sneller willen gaan?

- Kloksnelheid kan niet blijven stijgen
- Problemen met warmteafvoer

Oplossing = Multi-processoren

- Meerdere processoren voeren samen de taken uit
- Meerdere programma's echt op zelfde moment uitgevoerd

Verschil monoprocesor?

- Kan ook programma's 'tegelijk' uitvoeren
 - o Wisselt continu in processor af van programma's
- Multiprocessor kan echt meerdere programma's gelijktijdig uitvoeren

Multi-Core Processoren

- Andere naam voor meerdere processoren op één chip
- Wordt steeds populairder

Problemen bij meerdere processoren?

- Verbruik en warmteafvoer
 - o Zeer veel warmteproductie uit processoren
 - o Speciaal component die warmte beperkt
 - Ongebruikte processoren stilleggen
- Gebruik van voorgeheugens
 - o Elke processor heeft eigen voorgeheugen
 - o Wat met de kopies uit hoofdgeheugen?
 - Indien processoren over elkaar werken met zelfde geheugen kunnen problemen ontstaan
 - o Er is controle op "geldigheid" van kopie na elke schrijf- operatie --> Snooping
- Benutten van de beschikbare rekenkracht
 - o Er moet een verdeling kunnen gebeuren voor de processoren
 - Aanmaken van "Threads" die uitvoeringsspoor bepalen

2.15 Hardware Hyper-Threading

Bij hyper-threading-technologie gedraagt een fysieke processorkern zich als twee logische processors die de cache en externe interfaces delen. In de meeste gevallen wordt het hierdoor mogelijk dat de processorkern twee threads tegelijk uitvoert.

Deel van de processor ontdubbelen ipv dupliceren

- Alles met de "Threads" weghalen
- Na elke klokcyclus van Thread wisselen

Besturingsprogramma moet aangepast zijn om hyper-threading te kunnen gebruiken!

- Moet onderscheid kunnen maken tussen logische en fysische processoren

2.16 Computerprestaties

Hoe geven we computerprestaties weer?

2.16.1 MIPS, MFLOPS, SPEC

MIPS

- Million Instructions Per Seconds
- Niet objectief
 - o Niet alle instructies duren even lang of worden frequent gebruikt
 - o Niet alle processoren zelfde instructieset

MFLOPS

- Million Floating Point Instructions Per Seconds
- Niet ideaal
 - o Meeste programma's gebruiken bewegende kommagewerking veel meer
 - o Bevelenrepertoria kunnen verschillen
 - o Snelheid van voorgeheugen en randapparaten wordt niet meegerekend

SPEC

- Standard Performance Evaluation Corporation
- Bevat een standaardverzameling aan testprogramma's
 - o SPEC int --> Getallen en lijsten
 - o SPEC fp --> Zware rekenwerk

Experimenteel Testen

- Standaard Testen
 - o Whetstone --> Technische wetenschappelijke berekeningen
 - o Dhrystone --> Geen bewegende komma
 - o SPEC

2.16.2 Gebruikersstandpunt**Gebruikersstandpunt, waar let gebruiker op?**

- Responstijd = eerste tijd van resultaat
- Doorlooptijd = Tijd tussen opstart programma en uitvoering beëindigd
- Doorvoer = Aantal afgewerkte programma's per tijdseenheid

Nagekeken en vervolledigd op 29-12-2022

3. Basisprogrammatuur

3.1 De Vertaler

Vertaler / Assembleerprogramma = Maakt van lagere programmeertaal machinetaal
Zet bronprogramma om in uitvoerbaar programma

Werking van Vertaler

- Elke lijn inlezen
- Commentaar weglaten van elke lijn
- Wat blijft over?
 - o Niets
 - o Decimaal getal --> 10-complement maken
 - o Instructie --> Omzetten naar machinetaal
 - o EINDPR

Lege lijn in programma

- Niet lezen, wel meetellen (voor foutcodes en aanduidingen)

3.1.1 DRAMA-Machine-instructies

Register = Plaats voor 10 decimale getallen

- Kan een instructie voorstellen
- Kan een getal voorstellen

Programmaonderbreking --> Wanneer DRAMA vertaler een getal als instructie probeert te lezen en dit niet kan...

Machine-instructies

- Opbouw instructie

Opcode (0,1)	Modus (2,3)	Acc (4)	Idx (5)	Operand (6,7,8,9)
-----------------	----------------	------------	------------	----------------------

- **Opcode / Operatiecode**
 - o Twee decimale cijfers die bewerking aangeven HIA, BIG, VGL, ...
- **Modus**
 - o Geeft aan hoe operand berekend/gebruikt moet worden
 - o 1e Moduscijfer: Extra als .i, .w,... die aangeven hoe de operand gebruikt moet worden
 - o 2e Moduscijfer: Mogelijke indexatie (en in/decrement)
- **Accumulatorveld**
 - o 1 cijfer dat aangeeft op welk rekenregister de instructie van toepassing is (waarde tussen 0 en 9)
 - o Een voorwaarde bij de VSP, een onderbrekingsnummer
 - o Bij SPR is veld overbodig, standaard een 9 gebruikt
- **Indexregisterveld**
 - o 1 cijfer geeft aan welk rekenregister als indexregister gebruikt wordt indien nodig. (2e modusveld geeft dit aan!)
 - o Geen indexregister mogelijk --> Waarde 9 (2e moduscijfer ook 9)
 - Bij instructies als LEZ, STP, ...
 - o Geen indexregister gebruikt --> Waarde 0 (2e moduscijfer dan 1)
- **Operand**
 - o 4 cijfers die een adres of een getal voorstellen. Dat getal kan door indexatie gewijzigd worden...

Verdeling in klassen

- Voor sommige instructies zijn bepaalde velden nutteloos, hiervoor verdelen we de instructies in klassen, met elk hun eigen gebruikte velden.
- In DRAMA 7 verschillende klassen K1 tot K7.

Zie pagina 4 voorbeelden

3.1.2 De Mini-DRAMA vertaler

MINI-DRAMA

- Enkel gebruik van R0
- Enkel mnemotechnische functiecodes
- Bij VSP enkel NNEG
- Wel constanten en commentaar
- Geen RESGR, symbolische adressen, indexatie, interpretaties (.w, .a, ...), reserveringen, ...

Werking MINI-DRAMA Vertaler

- Lijn per lijn inlezen
- Voor elke lijn
 - o Commentaar wegdoen
 - o Wat overblijft
 - Niets --> Niets doen, wel meetellen ivm foutboodschappen
 - Decimaal getal --> Voorstellen met 10-complement
 - Instructie --> Mapping via machine-instructietabel.
 - Via tabel de instructie vertalen naar machinetaal
 - EINDPR --> Vertaling klaar, stoppen
- Nakijken op fouten
 - o Foutboodschappen afprinten
 - (ADRES MAG MAX 9999 en MIN 0000 ZIJN!)
 - Gebruik van onbestaande functiecodes
 - Constanten die niet met 10 cijfers voorgesteld kunnen worden
 - Opgeven operand bij functie zonder operand

Elke instructie van MINI-DRAMA kan direct worden vertaald naar de machinetaal.

3.1.3 De DRAMA-vertaler

DRAMA - Vertaler

- Invoerlijnen
 - o Commentaar
 - Worden overgeslagen door vertaler
 - o Constanten
 - Kunnen symbolisch adres hebben (met label ervoor)
 - Voorgesteld in 10-complement
 - Mogen ook een naam hebben
 - o Vertaler-directieven
 - EINDPR, RESGR
 - o Instructies
 - Registraanduiding van R0 tem R9 (= Stapelregister)
 - Interpretaties, indexatie --> Invloed op modusveld
 - Register-operand, speciale operand (poortnummer), ... Operand moet in context passen!
 - Kan symbolisch adres hebben (met label)
 - Maken het vertalen zeer moeilijk en complexer

Symbolische Adressen

- Vertaalprogramma kan vaak niet onmiddellijk vertaling uitvoeren door symbolische adressen
 - Voorwaartse Referentie --> Er wordt verwezen voor het gedeclareerd is
 - Achterwaartse Referentie --> Verwijzen nadat er een waarde aan gekoppeld is
 - Geeft geen problemen

Werking Vertalen van DRAMA - Tweestapsvertaling

Door mogelijke voorwaartse referentie is het nodig om twee keer te vertalen, voor gemak splitsen we dit op in 2 fases

0. GEGEVENSSTRUCTUUR INITIALISEREN

- Lijnteller op 1
- Programmateller op 0000
- Symbolentabel leeg

1. SYMBOLENTABEL OPSTELLEN

- Lijn per lijn inlezen
- Commentaar en lege lijnen negeren
- Voor elk label de naam en het adres opslaan in de symbolentabel
 - Lijnteller --> telt lijnen van invoer
 - Programmateller PT --> houdt adres bij van bevel dat vertaald wordt
 - Symbooltabel --> Symbolische adressen en overeenkomstige stand van de PT
- Bij toevoeging aan symbolentabel nakijken of adres niet eerder was gedefinieerd, anders foutmelding

Label wordt ook per letter in een tabel omgezet naar code (het is dus niet dat in de computers de letters zitten)

2. MACHINECODE GENEREREN

- Lege lijn --> Niets mee doen, enkel meetellen voor foutboodschappen
- LAADGR-adres
 - PT krijgt waarde van dat adres
- Vertaler-directieven
 - EINDPR --> Vertaling afgelopen
 - RESGR n --> Zorgen dat n aantal registers in geheugen vrij zijn
 - Reservatie door register vullen met n 0-constanten
 - Kan symbolisch adres hebben
- Constanten
- Instructies
 - Veel complexer
 - Verschillende tabellen
 - Met symbolische adressen mogelijk (symbolentabel gebruiken)
- Relocatietable maken voor relocerende lader

Mogelijke Fouten in vertaling

- Onbestaande mnemotechnische code gebruikt
- Syntax foutief; operand te veel meegeven, ...
- Zinloze operand bij instructie zonder operand meegeven
- Referentie naar symbolisch adres dat niet voorkomt in symbolentabel

Symbolentabel wordt ook bij de machinetaal toegevoegd in de code. Dit kan handig zijn voor

- Disassembler (*zet machinetaal naar LPT*)
- Debugger/Speurprogramma (*zoekt fouten*)
- Binder (*Voegt losse delen code aan elkaar*)

3.1.4 Vertaler voor hogere programmeertaal

Compiler = vertaler voor HPT naar LPT

Waarom niet direct naar machinetaal?

- Verschillende processoren kunnen zelfde assembleertaal
- Nakijken op fouten is mogelijk in LPT
- We hebben al een assembleerprogramma, die moeten we daarna toepassen.

Taken Compiler

- Lexicale analyse --> Zoekt woorden die herkenbaar zijn in de taal
 - o "Tokens" worden herkend zoals "if", "45", ";", ...
- Syntactische analyse --> Maakt van de woorden een zin
 - o Herkennen van individuele opdrachten met tokens
 - Toekenning (naam variabelen, =, expressie)
 - Expressie (naam van variabele, andere expressie,)
- Semantische analyse --> Probeert de zin te 'lezen' en begrijpen, de betekenis van de opdrachten
- Code Generatie --> Zet de zin om naar code (LPT of intermediaire taal)
 - Eenvoudig uit te testen en na te kijken dan machinetaal
 - Compiler kan gebruikt worden voor familie van machines
- In hogere / zeer hoge programmeertaal

Na compiler start het assembleerprogramma om van LPT naar Machinetaal te gaan

C-Compiler voor DRAMA?

- Vroeger: geschreven in LPT
- Vandaag: geschreven in HPT, taal van zeer hoog niveau

(Zie ppt voor interpretatie van de vertaler en compiler)

Talen in de compiler

- DRAMA (Assembleertaal)
- Drama (Machinetaal)
- Intel (Intel Machinetaal)
- C (Programmeertaal)

Kruiscompiler = Compiler die code genereerd voor een andere machine dan waar die op draait.

Veel verschillende compilatoren nodig die een taal omzetten in een andere taal!

Nagekeken en vervolledigd op 15/10/2022

3.2 De MACRO-Voorvertaler

Macro's = Hulpmiddel bij het programmeren

Voorvertaler

- Werkt alle macro's uit voordat de vertaler begint
- Voert volledige taak uit voor de vertaler begint
- Verwijdert de macro-definitie in de code
- Macro-Expansie --> vervangt de macro-oproepen door overeenkomstige code
- **Symbolmanipulatie**
 - o Kopiëren van tekens of vervangen van tekens door een sequentie van andere tekens

Implementatie

- Waarom Aparte voorvertaler?
 - o Macro-verwerking is geen vertaling, wel symbolmanipulatie (symbolen vervangen)
 - o In andere context gebruik maken van de macro's kan ook
- Ingebouwd in de vertaler
 - o Betere foutboodschappen krijgen
 - o Gebonden aan taal

3.2.1 Enkel MACRO- en MCREINDE-directieven

Taken voorvertaler

- Iets wat niets met macro's te maken heeft --> Laten staan
- Macro-definitie --> Onthouden en verwijderen
- Macro-oproep --> Expanderen / Vervangen door de definitie

Voorvertaler herkent geen commentaarlijnen!

Toestanden van voorvertaler

- Kopieermodus
 - o Lijn per lijn inlezen en kopiëren
 - o Bij invoer "MACRO" --> Overgaan naar definitiemodus
 - o Bij invoer: "xyz" (indien xyz een macro is)
 - Argumententabel opstellen en expansiemodus starten
 - Tabel waar formele parameters aan actuele macro-parameters wordt gekoppeld (soort "oproeprooster")
 - o Bij andere invoer --> Kopieermodus blijven
- Definitiemodus
 - o Eerste lijn = Hoofding
 - o Volgende lijnen (lichaam)
 - Allemaal kopiëren naar definitie en in definitiemodus blijven
 - MCREINDE --> Einde definitie, terug naar kopieermodus
 - o Wordt bijgehouden uit hoeveel lijnen definitie bestaat
 - o Na definitie is naam van macro een leutelwoord voor de voorvertaler
- Expansiemodus
 - o Lichaam definitie lezen
 - o Elke lijn

- Substituties <var> vervangen via argumententabel
 - Als niet in argumententabel --> Foutmelding
- \$ in lokaal symbolisch adres omzetten naar een teller _1, _2, In globaal symbolisch adres
- Genestelde oproep?
 - Huidige expansie onderbreken
 - Nieuwe argumententabel en terug nieuwe expansie
- Andere gevallen
 - Gewijzigde lijnen naar uitvoerbestand

3.2.2 Extra Macrodirectieven

Extra macrodirectieven als MEVA, MNTS, MFOUT, MSPR, MVSP, MVGL ook toevoegen in het hoofdprogramma buiten macrodefinities?

- Verhoogt het programmeercomfort
- Gebruik van globale voorvertalervariabelen
- Gebruik van globale voorvertaleretiketten

Globale macrovariabelen en globale voorvertalervariabelen

- MEVA-directief
 - Kent nieuwe waarde toe aan macrovariabelen of aan een formule van macroparameters
 - Kan ook globaal gebruikt worden (*voorvertaler*), heeft voordelen
 - Lokaal houdt op na de macro-oproep, globaal blijft bestaan en is nuttig voor tellers of doorlopende wijzers
 - Globale variabele kan door meerdere macro's gebruikt worden
 - Globale variabele kan ook buiten macro gebruikt worden als constante

Onderscheid naamgeving

- Globale voorvertalervariabele (*VV-Variabelen*)
 - Bestaat enkel als macro-directief, over heel de voorvertaling
 - Handig als je een constante gemakkelijk wil aanpassen en meerdere keren gebruikt
 - vb: Teller voor de etiketten \$ omzet naar _1, _2, ...
- Lokale macrovariabele
 - Bestaat enkel als macro-directief, binnen een bepaalde macro
- Globale variabele
 - Bestaat over heel het programma

Indien binnen macro een lokale macrovariabele dezelfde naam heeft als een globale VV-variabele, kan je de globale niet gebruiken binnen die macro!

Lokale Macro-etiketten en globale voorvertalersetiketten

- Voorvertalersetiketten (*VV-etiketten*)
 - Mogen slechts één keer per macro-definitie gedefinieerd worden!
 - Expliciet gedefinieerd, geen verwarring mogelijk
 - Binnen macro = Lokaal
 - Buiten macro = Globaal
- Beperking voor MVSP- en MSPR- directief!
 - **Nooit springen 'in of uit een macrodefinitie', is verboden!**
- Globale VV-etiketten
 - Nuttig om bepaalde stukken code over te slaan
 - Mogen slechts één keer gedefinieerd worden!
 - vb om testen in programma te verbergen

- **Bemerkingen**
 - o Als je over een macro-definitie springt wordt de macro later niet herkent!
 - o Als je door MSPR of MVSP een lus maakt waarin een macro-definitie zit zal dit fout geven
 - Macro wordt dan meerdere keren gedefinieerd --> Mag niet
 - o Om onnodig etiketten te creëren plaats je achter etiketten best MNTS

Gegevensstructuren VV-variabelen

- Globale VV-Variabelentabel GVT
 - o Tabel met alle globale vv-variabelen en hun waarde
- Argumenten-en-lokale-variabelentabel ALVT
 - o 1 Per macro-Oproep
 - o Tabel met argumententabel en alle lokale macro-variabelen en hun waardes
- Globale etikettentabel GET
 - o Bevat globale etiketten met hun bijhorende lijnnummer
- Lokale-Macro-Etikettentabel = LET
 - o 1 Per Macro-Oproep
 - o Tabel met etiketten en bijhorende lijnnummers
- Macro Conditiecode MCC
 - o Wordt bewaart in GVT
 - o Wijzigt bij bevelen als MEVA of MVGL
- Teller (Telt macro-oproepen)
 - o Wordt bewaart in GVT

Aanpassingen Voorvertaler voor alle directieven?

Nog steeds 3 mogelijke toestanden van de voorvertaler!

- **Kopieermodus**
 - o Bij toevoeging globaal macro-etiket, toevoegen aan GET
 - \$etiket --> Toevoegen aan GET
 - o Symbolen vervangen door waarde uit GVT
 - <naam> --> Opzoeken in GVT
 - o Verwerken van extra macro-directieven
 - MEVA --> Toevoegen GVT
 - MNTS --> Niets doen
 - MFOUT --> vertaling stopt, fout afdrukken
 - MSPR --> Lijnnummer etiket zoeken in GET
 - MVGL --> Waarde geven aan MCC
 - MVSP --> Lijnnummer etiket zoeken in GET, sprong als MCC klopt

Indien etiket niet wordt herkend, lijnen overslaan tot je etiket gevonden hebt.

- **Definitiemodus**
 - o LET Opstellen tijdens kopiëren van lichaam
- **Expansiemodus**
 - o <naam> --> Opzoeken in ALVT, indien niet gevonden zoeken in GVT
 - o Alle extra directieven herkennen en verwerken

- MEVA --> ALVT of GVT
 - Eerst kijken of var in ALVT staat, indien niet kijken of het in GVT staat, indien beide niet een nieuwe aanmaken in ALVT
 - Nieuwe waarde aan MCC
- MNTS --> Niets doen
- MFOUT --> Voorvertaling stopt, fout afdrukken
- MSPR --> Lijnnummer zoeken in LET
- MVGL --> Waarde veranderen van MCC
- MVSP --> Lijnnummer zoeken in LET, sprong indien MCC klopt

Van Bronprogramma tot Uitvoerbaar programma

(!! slide 44/48)

Nagekeken en vervolledigd op 05-01-2022

3.3 De Lader

Lader

- Programma in geheugen om machinetaal in het geheugen te brengen
- Nodig om programma uit te voeren
- **Tegenwoordig is de lader een onderdeel van het besturingsprogramma (zie opstarten pc)**

Waarom is lader nodig? Vertaler kan ook rechtstreeks in geheugen vertalen?

- Vertalen kost meer tijd en moet dan telkens opnieuw worden uitgevoerd
- Vertaler is (veel) groter dan een lader, minder geheugen is beter.

Vertaler heb je maar één keer nodig, wanneer je programma klaar is en dit naar schijf wordt geschreven

Soorten Lader-algoritmes / Laderschema's

- Absolute Laders
 - Zet programma ongewijzigd in het geheugen
- Relocerende Laders
 - Kan het programma aanpassen aan het geheugen

Dynamische adresvertaling = Aanpassing in hardware die de lader vergemakkelijkt.

3.3.1 De Absolute Lader

Absolute Lader

- Uitvoerbaar programma ongewijzigd in geheugen
- Meest elementaire geval
 - Programma steeds van adres 0000
 - Programma in invoertaal
 - Eerste getal is lengte van het programma (aantal instructies/constants)
 - Lader geladen vanaf adres 9000
- Algoritme / Uit te voeren acties
 - Lengte inlezen
 - Evt laadadres inlezen
 - Getallen inlezen --> in opeenvolgende geheugenplaatsen zetten vanaf 0000
 - Start uitvoering (springen naar adres 0000)
- Kan gebruikt worden als eenvoudig besturingsprogramma
- Simulator lukt niet, oplossingen nodig
 - Zelf lengte-aanduiding toevoegen
 - Invoer uit bestanden, niet zelf moeten ingeven op toetsenbord...
 - Lader overschrijft zichzelf! (uitvoering begint normaal standaard vanaf 0000)
 - Oplossing: lader starten vanaf 9000. (Reserveer ervoor 8999 geheugenplaatsen, spring daarna naar lader op adres 9000)
- Klein besturingssysteem?
 - Programma's inladen
 - Werking
 - In elk programma na het uitvoeren terug springen naar lader (SPR 9000)
 - Programma's vertalen
 - In elk vertaald programma "lengte" inlassen en symbooltabel weglaten
 - Alle programma's na elkaar zetten in 1 bestand

Verfijning Absolute Lader

- Programma niet altijd vanaf adres 0000 laden
 - Anders kunnen we steeds maar één programma gelijktijdig inladen

Problemen?

- Adressen kloppen niet langer
 - Vertaler gaat ervan uit dat programma op adres 0000 zou starten

Oplossing?

- Lader kan zelf programma niet wijzigen
- Vertaler moet correcte adressen genereren zodat lader weet waar plaats is
- Programmeur moet meegeven vanaf welk adres geladen moet worden
- LAADGR adres
 - "Te laden vanaf xxx"

Opbouw van uitvoerbaar programma

- Hoofding
 - Lengte programma
 - Laadadres voor absolute lader
- Programmacode
 - Data van het programma
- Symbooltabel

Aanpassing aan vertaler

- PT Programmateller krijgt in begin de waarde van LAADGR
- Vertaler geeft mee hoe lang het programma is en vanaf welk adres

--> Niet praktisch

- Programmeur moet zelf al weten op welk exact adres te laden

Alternatief: Relocerende lader

Nagekeken en vervuldigd op 15/10/2022

3.3.2 De Relocerende Lader

Nadelen absolute lader

- Programmeur moet voor vertaling laadadres kiezen
- Vertaalde programma kan maar op 1 plaats in geheugen

Oplossing?

- Lader zelf adressen laten kiezen
 - o Relocatietable
 - Tabel waarmee vertaler meegeeft welke dingen door de lader aangepast moeten worden
 - Bevindt zich achteraan het programma, achter symbolentabel
 - o Als laadadres geeft de vertaler '-1' (999999999) mee

Programma is Reloceerbaar als het een relocatietable bevat

Vertaler maakt in 2^e stap (codegeneratie) de relocatietable en voegt deze toe aan programma

Werking Relocerende Lader

1. Uitvoerbaar programma letterlijk in geheugen laden
2. Relocatiestap: code aanpassen m.b.v. de relocatietable

Absolute lader

- Programma ongewijzigd in geheugen
- Eenvoudig
- Nadeel: zelf kiezen welk laadadres op ogenblik van vertaling

Relocerende lader

- Lader die in staat is machinecode aan te passen
- Uitvoerbaar met relocatielabel
- 2 Stappen: laden en relocatiestap
- Voordeel: Last valt weg bij programmeur, veel efficiënter en sneller

3.3.3 Dynamische Relocatie

Statische Relocatie

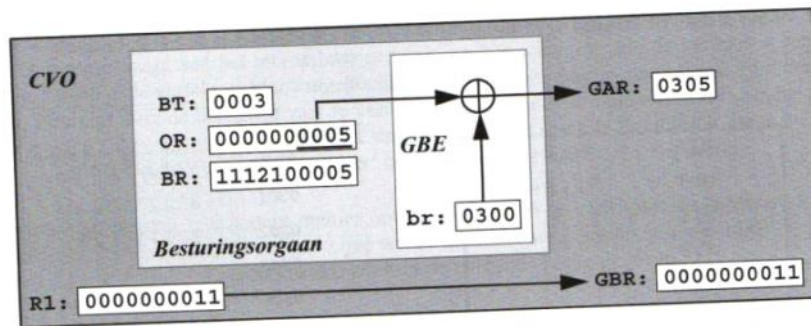
- Éénmalige relocatie voor uitvoering programma

Hardware Aanpassen

- Extra eenheid in processor: Geheugenbeheereenheid GBE
Memory Management Unit MMU

Dynamische Relocatie

- Relocatie mogelijk tijdens de uitvoering programma
- Extra hardware (CBE) nodig om te kunnen werken!
- Eenvoudigere software
- Alle adressen aanpassen TIJDENS de uitvoering
(enkel de adressen die naar geheugen gestuurd worden)
- Inhoud basisregister blijft constant
 - o Behoort tot CVO (centraal verwerkingsorgaan)
 - o Besturingsprogramma initialiseren het op het beginadres van het geladen programma
Zie voorbeeld pagina 59-60
- **Adresvertaling transparant (programmeur ziet niet dat er iets zal aangepast worden)**



BT = Bevelenteller
 BR = Bevelenregister
 OR = Operandregister
 CVO = Centrale Verwerkingsorgaan

GAR = Geheugen Adres Register
 GBR = Geheugen Buffer Register
 br = basisregister
 GBE = Geheugenbeheereenheid

Dynamische Relocatie vs Indexering

- Belangrijke Verschillen
 - o Inhoud basisregister blijft constant, een indexregister verandert veel
 - o Inhoud basisregister opgeteld bij adres juist voor het in het geheugenadresregister geplaatst wordt
 - o Voor het programma dat uitgevoerd wordt lijkt het alsof het vanaf 0000 wordt geladen

Geheugenbeheereenheid GBE

Nieuwe instructies nodig die basisregister invullen, in-uitschakelen van GBE,
GBE enkel actief bij uitvoering ingeladen programma!

- Nieuwe Bijhorende Machine instructies
 - o HIB: "Haal in basisregister"
 - o SGI: "Spring en schakel GBE in"
 - Vanaf dan wordt er dynamisch gereloceed!
 - Eerstvolgende bevel komt van 'adres' + 'br'
 - o SGU: "Spring en schakel GBE uit"
 - Vanaf dan geen dynamische relocatie meer, alle adressen ongewijzigd in GAR
 - Eerstvolgende bevel komt van 'adres'

Gevolgen voor vertaler

- **Zeer veel vereenvoudigd!**
- Mag steeds veronderstellen dat programma geladen wordt vanaf 0000
- Geen relocatietabel nodig voor lader
 - o Andere programma's hebben deze miss wel nog nodig (binder,...)

Gevolgen voor lader

- Invullen basisregister nodig
- GBE inschakelen juist voor uitvoering programma

Nagekeken en vervolledigd op 05-01-2023

3.4 De Binder

Grote programma's vaak opgedeeld in kleinere '**modules**'

Modules

- Elk in afzonderlijk bestand
- Elk afzonderlijk vertaald

Taak van binder = Afzonderlijk vertaalde modules terug samenvoegen tot één uitvoerbaar programma

3.4.1 Waarom Binder / Koppelprogramma?

Nadelen zonder binder

- Samenvoeging is niet praktisch, symbolische adressen vormen problemen
 - o Elk symbolisch adres moet uniek zijn
- Hoe groter programma, hoe meer tijd voor hervertaling na bv aanpassingen
- Manueel samenvoegen is niet gemakkelijk
 - o Bij vertaler krijg je meestal al wat subroutines --> Fabrikant wil broncode ook niet vrijgeven

=> Het zou handiger zijn indien we modules afzonderlijk kunnen vertalen en later laten samenvoegen tot een speciaal programma

3.4.2 Modules

Module

- Bevat deel van een programma
- Bevat soms verwijzingen naar symbolische adressen buiten de module zelf en in een andere module
 - o Vertaler zou dan fout geven
 - o Nieuw vertalingsdirectief is de oplossing: EXTERN
 - Een lijst van externe symbolische adressen als operand
 - o Om onderscheid te maken ook nieuw directief: GLOBAAL
 - Een lijst van globale symbolische adressen als operand

Soorten verwijzingen

- o EXTERN
 - Verwijzing naar symbolisch adres buiten de module
- o GLOBAAL
 - Definitie symbolisch adres waarnaar verwezen mag worden uit andere modules

Beide directieven worden door programmeur ingesteld en via de vertaler aan de binder gegeven

3.4.3 Bestanden

Elke module bewaard in afzonderlijk **bestand**

Bestand

- Bevat een module
- Bevat eventueel andere dingen als tekst, foto, geluid, andere (binaire) gegevens

Opsplitsen programma in modules

- Lage programmeertaal
 - o Elk bevel moet tot (max) één module behoren

- In elke module EXTERN en GLOBAAL voorzien
- Hogere programmeertaal
 - Procedure of functie mag niet gesplitst worden over modules
 - Hoofdprogramma en functies splitsen mag wel in verschillende modules
 - In elke module EXTERN en GLOBAAL voorzien

3.4.4 Hoofdprogramma

Moeilijk om te weten waar je moet beginnen na de samenvoeging door de binder!

- Oplossing = Extra directief :{
 - STARTPR
 - Door programmeur opgegeven
 - Geeft aan in welke module het hoofdprogramma zit
 - Argument het adres van eerste instructie
 - Sprong zal uitgevoerd worden naar het start-adres

Werking vertaling

- Afzonderlijke modules worden vertaald naar
 - Objectmodules
 - Nog niet volledig-vertaalde module, sommige info moet uit andere modules komen
 - Aangeduid door OBJECTMODULE (-1)
 - Uitvoerbare module
 - Indien nergens verwezen werd naar andere modules is na de vertaling de module uitvoerbaar
 - Aangeduid door UITVOERBAAR (-9)

We gaan er verder vanuit dat er enkel objectmodules komen uit vertaling

Binder zet de objectmodules samen om tot een uitvoerbaar programma

Informatie voor de binder

- Welke extra info heeft binder nodig van vertaler?
 - Elke objectmodule is reloceerbaar, dus bevat steeds een relocatietabel
 - Info nodig over extern symbolische adressen die de vertaler niet kent

Oplossing = Relocatie- en Bindingstabel

- Bevat alle gegevens uit relocatietabel
- Bevat info over externe symbolische adressen die de binder moet koppelen
- Vertaler voegt uitgebreide symbolentabel toe aan objectmodules
 - Symbolische naam met waarde en het type van elk symbolisch adres
 - Lokaal in module / extern in andere module gedefinieerd

Bekijk voorstelling van inhoud objectmodules in cursus pagina 72-73

3.4.5 De Taken van De Binder

1. Plaats voorzien voor verschillende modules
 - Objectmodules bevatten lengte van de module, zo is plaats voorzien niet moeilijk
2. Machinecode van de modules samenvoegen
3. Startadres voor uitvoering bepalen
 - Op basis van geldige start-aanduiding en plaats waar module in het uitvoerbaar bestand komt te staan

4. Code Reloceren
 - Buiten de eerste module wordt elke module opgeschoven en is relocatie nodig
 - Aan de hand van allocatietabellen
 - Houdt bij welke plaats voor welke module is voorzien
5. Externe Referenties in rekening brengen
 - Dit is het eigenlijke 'binden'
 - Aan de hand van globale symbolentabel
6. Relocatietabel opstellen
 - Indien het resultaat zelf ook nog relocerend moet zijn

Werking van de binder

1. ALLOCATIE- EN GLOBALE SYMBOLENTABEL

- Allocatiegedeelte
 - Binder leest modules en bepaalt beginadres voor elke module
 - Maakt allocatietabel waarin staat welk beginadres bij welke module hoort
- Symbolengedeelte
 - De Symbolensectie uit de code wordt gebruikt om symbooltabel op te stellen
 - Lokale en externe symbolische adressen genegeerd
 - Globale etiketten samen met waarde in symbooltabel toevoegen

2. SAMENVOEGEN, RELOCEREN EN BINDEN

- Code van alle modules achter elkaar plaatsen
- Reloceren en binden waar nodig
 - Reloceren door allocatietabel
 - Binden door symbolentabel (*globaal*)
- Symbolen sectie toevoegen aan uitvoerbaar programma
 - Nuttig voor andere programma's: speurprogramma,
- Relocatietabel toevoegen aan uitvoerbaar programma
 - Elke relocatieopdracht blijft behouden
 - Elke bindingsopdracht vervangen door een relocatieopdracht

Opmerkingen

- Welke invloed dynamische relocatie op binder?
 - Weinig, indien er hardware is moet binder geen relocatietabel toevoegen aan uitvoerbaar programma
- Vertaler is complexer om modules afzonderlijk te kunnen vertalen!
 - Moet externe symbolische adressen kunnen verwerken
 - Moet aan elke objectmodule een relocatietabel toevoegen
 - Omdat de binder doet aan statische relocatie
- Dynamisch binden (nieuwere techniek)
 - Binden uitstellen tot wanneer objectmodule nodig is
 - Voordelen
 - Als module niet wordt gebruikt moet ze niet in het geheugen komen

3.4.6 Programmabibliotheken

Bij aanschaffing vertaler --> Vaak subroutines meegeleverd in objectmodulevorm, samengevoegd in **programmabibliotheken**

Bibliotheek

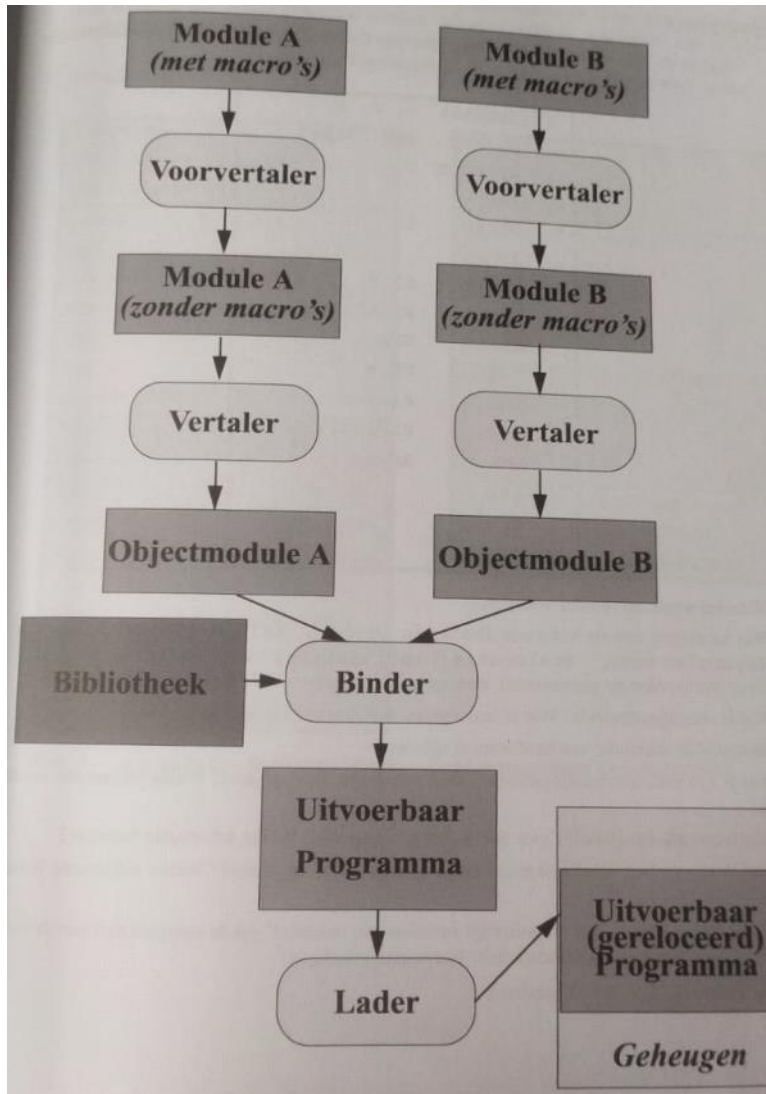
- Handiger dan afzonderlijke objectmodules
- Binder kan er mee werken
- Wanneer symbolisch adres extern is zal binder dit zoeken binnen een andere module in de bijhorende bibliotheek

3.4.7 Vertaler-, binder- en lader-directieven

Samenvatting van nieuw geïntroduceerde directieven in tabel pagina 78

3.4.8 Van bronprogramma tot uitvoering

Samenvatting stappen in uitvoering in schema pagina 79



Eénmaal programma correct werkt worden vertalingen en bindingen overgeslagen, enkel het uitvoerbaar bestand wordt nog opgeroepen en enkel de lader moet nog iets doen.

Nagekeken en vervolledigd op 05-01-2022

3.5 De Vertolker

Programma uitvoeren met vertaler?

- Vertaling programma --> Vertalingsfase
- Uitvoering door processor --> Uitvoeringsfase

Programma uitvoeren via Vertolker / Interpreter

- Voert beide fases tegelijk uit en zorgt voor directe uitvoer
 - o Vertaalt code
 - o Leest code
 - o Voert code uit

3.5.1 Werking van een vertolker

Eerste vertolker

- Lijn per lijn inlezen
 - o Vertalen
 - o Direct Uitvoeren
- Vertolker voert twee programma's tegelijk uit
 - o Vertolker zelf
 - o Het programma dat vertolkt werd

Opletten dat beide programma's niet interfereren met elkaar

Niet mogelijk bij lagere programmeertaal!

- Reeds vertaalde programma nodig, ook door symbolische adressen
- Rekenen met bevelen mogelijk
- Waardes in registers omgewisseld of fout gezet mogelijk

Wel bij Machinetaal (DRAMA, Java Bytecode)

- Vertolken en dus machinetaal omzetten in andere?

Waarom vertaald programma vertolken?

- Vertolken om uit te kunnen voeren op een ander apparaat.
 - o Doelcomputer: Computer waarvoor programma reeds vertaald is
 - o Gastcomputer: Computer waarvoor we vertolken

Hoe programma vertolken voor computer met andere machinetaal?

- Vertaalprogramma Ontwerpen
 - o Zet machinetaal om naar andere machinetaal
 - o Niet eenvoudig, soms geen tegenhanger voor bevel, andere verschillen,...
 - o Voorbeeld: JIT-compiler (Java Bytecode)
- Simulator Ontwerpen
 - o Een vertolker die het gedrag van de doelcomputer nabootst op de gastcomputer
 - o Voorbeeld: DRAMA-simulator
 - Doelcomputer (DRAMA-computer) bestaat niet

Vertolker wel mogelijk bij Hogere programmeertaal (Java, Python, ...)

- Niet mogelijk om te rekenen met bevelen
- Variabelen via toekenning eerste waarde, geen probleem met registers

3.5.2 Voordelen

Voordelen vertolker tov vertaler?

- Veel makkelijk te schrijven dan compiler
- Minder geheugenruimte nodig voor vertaalde programma
- Zinnige foutboodschappen afdrukken
- Machine-instructies moet vertolker niet kennen/begrijpen
 - o Vertolker is zelf in HPT

3.5.3 Nadelen vertolking

Langere uitvoering door vertolker ipv twee afzonderlijke fases

- Deel van het werk van de vertaler overdoen
- Vertolking vergt meer instructies
- Programmalus
 - o Telkens opnieuw analyseren van instructies binnen de lus

Optimalisaties

- Programma omzetten naar intermediaire code

3.5.4 Toepassingen van Vertolking

Toepassingen binnen de informatica

- Nieuwe programmeertaal maken?
 - o Eerst vertolker schrijven (is eenvoudiger), dan vertaler
- Talen van zeer hoog niveau
 - o Kunnen permanent door een vertolker behandeld worden
- Dynamische programmeertalen
 - o Kunnen zichzelf wijzigen tijdens uitvoering
- Simulator
 - o Voor programma in machinetaal
- Emulator
 - o Gastcomputer aanpassen zodat hij zich gedraagt als doelcomputer
 - o Simulator omzetten naar micro-code
 - o Invoer/Uitvoer = Moeilijk emuleren
- Computer waarvan machinecode HPT is

Nagekeken en vervolledigd op 05-01-2023

3.6 Het Speurprogramma

Speurprogramma

- Debugger die fouten uit programma kan halen/vinden
- Interactief
- Repertorium van opdrachten
 - o Toon code die uitgevoerd gaat worden
 - o Toon inhoud accumulatoren/registers
 - o Wijzig inhoud in accumulatoren/registers
 - o (Her)start uitvoering
 - o Verwijderen/Toevoegen van breekpunten
 - o Overzicht van breekpunten weergeven
 - o ...

Werking Speurprogramma

- Afwisselend uitvoeren 2 programma's
 - o Speurprogramma
 - o Uit te testen programma
- Gebruiker geeft leiding voor de uitvoering van deze programma's
- Breekpunten
 - o Plaats waar uitvoering onderbroken wordt (door gebruiker)
 - o Staan samen in de breekpuntentabel
 - o Mag niet overal gezet worden
 - Niet mogelijk bij instructies die CC niet wijzigen.
 - SPR, VSP, KTG, ...

Manier om programma te onderbreken

- Directief SBR *breekpunt*
- Oorspronkelijke functie bewaren om na de onderbreking verder te gaan

Speurprogramma's voor HPT

- Symbolisch Speurprogramma
- Moet je meedelen aan compiler
- Extra informatie in objectmodule

3.7 Het Opstarten van de Computer

Wat gebeurt er bij het opstarten van de computer?

- Centrale geheugen is leeg (*random waardes*)

Vroeger --> Besturingspaneel

- Stonden veel lampjes en drukknoppen op
- Computer kwam bij opstart in halttoestand (geen uitvoering instructies)
- Stappen nodig door programmeur
 - o m.b.v. schakelaars een adres instellen en in het GAR geladen door op knop te drukken
 - o Door schakelaars laadinstructies afzonderlijk in het geheugen zetten
 - o Tenslotte startadres instellen en uitvoering starten (knop)
 - Programma gaat dan naar uitvoeringsstand
- Veel kans op foute ingaven bij een instructie, kan alleen bij eenvoudige programma's

Bootstrap-Procedure

- Werkwijze waarbij we starten met zeer eenvoudig programma en hiermee steeds complexere programma's laden

Moderne computers

- Deel van geheugen is ROM-leesgeheugen, bij opstart computer begint deze uitvoering van dat leesgeheugen

- Programma test werking apparatuur
- Programma wordt vanuit vaste plaats ingelezen uit 'bootsector'
- Geladen programma wordt uitgevoerd, het zal zelf een besturingsprogramma inlezen

Voordelen ROM

- ROM is bestendig, programmeur moet niets meer ingeven
- ROM is beveiligd, je kan het niet per ongeluk overschrijven
- ROM is sneller dan RAM

Nagekeken en vervolledigd op 05-01-2023

4. Besturingssystemen

Besturingssysteem

- Werking regelen van de computer besturen/regelen
- In geheugen geladen en uitgevoerd bij opstart vd computer

Waarom besturingssysteem nodig?

- Anders niet mogelijk om met een computer te werken
- Veel eenvoudiger om op laag niveau taken uit te voeren
- Computer efficiënt te laten werken
 - o Meerdere programma's tegelijk, in- en uitvoer, ...
- Nood aan voldoende hardware ondersteuning om besturingssysteem te kunnen gebruiken!

4.1 Inleiding

Doelstellingen besturingssysteem

- Computer gebruiksvriendelijker maken
- Gebruik van de computer efficiënter maken

Laders = Voorbeeld van eenvoudige besturingsprogramma's

Zeer veel verschillende besturingssystemen

- Vaak met specifieke functies en anders dan de rest
 - o Aangepast aan de gebruiker
- Bevatten gemeenschappelijke elementen
 - o *Aanvullen*

4.2 Het Programma-Toestandswoord

Programmatoestandswoord PTW

- Register in het CVO (*Processor*)
- Bevat info over
 - o Toestand van programma in uitvoering
 - o Toestand van machine zelf
- 20 Decimale cijfers bij DRAMA-machine
- Verschillende velden
 - o Onderbrekingsniveau ONV
 - o Toestand van Processor H/U
 - 0 bij halttoestand
 - 1 bij uitvoeringstoestand
 - o Toestand van Processor S/P
 - 0 bij supervisietoestand
 - 1 bij probleemtoestand
 - o Neveneffect van uitvoering bevel
 - Conditiecode CC
 - Overloopindicator OVI (*Indien overloop*)
 - Stapeloverloop-indicator SOI (*Indien Stapeloverloop*)

Deze 2 indicatoren hebben enkel mogelijke waarde 1 en 0 met 1 'waar'
 - o BT (Bevelenteller), adres van volgende instructie
 - o Maskerwoord
 - Globaal Masker G
 - Globale Programma-fout GPF
 -

- Geen instructies om waarden van individuele velden weer te geven!
 - o Waarden van de velden indirect testen kan wel
 - VSP-instructie
 - Voorwaarden NUL, GEL, NEG, NNUL, testen op CC
 - Voorwaarde OVL ('overloop') en GOVL testen op OVI
 - Voorwaarde SO ('stapeloverloop') en GSO testen op SOI

4.3 Programma-Onderbrekingen

4.3.1 Inleiding

Overloop

- Resultaat kan niet voorgesteld worden
 - o Om overloop te testen zou je na elke berekening een VSP moeten zetten --> Niet overzichtelijk, vertraging!
 - o Beter = Automatische controle voor overloop door apparaten

Stapeloverloop

- Controle op over- en onderloop van de stapel

Programma-Onderbreking

- Denk aan thuis boek lezen als iemand aanbelt aan de deur.
- Programma heeft, buiten een vertraging, geen andere effecten
- Belangrijk voor
 - o Efficiënte werking computer
 - o Correcte werking programma's garanderen

Belangrijk dat dergelijke 'programma-wisselingen' goed kunnen gebeuren

Programma controle over processor --> Als CVO bezig is met instructies van dat programma

Oneindige lus

- CVO blijft instructies van zelfde programma uitvoeren
- Veelgemaakte programmeerfout
- Enkel uit geraken door computer afzetten
 - o Of door een programma-onderbrekingsmechanisme!

4.3.2 De Bevelencyclus

Uitvoering van instructies / Bevelencyclus

1. Instructie ophalen uit geheugenregister (adres in BT) en BT +1
2. Instructie analyseren
 - o Functiecode onderzoeken
 - o Operanden bekeken en evt. opgehaald uit geheugen
3. Uitvoering instructie

Sequentiële Besturing

- Tijdens opeenvolgende bevelencycli worden opeenvolgende instructies van een programma uitgevoerd
- Hoe onderbreken?
 - o SPR of VSP
 - o Programma-onderbreking

Verschillen tussen manieren SPR/VSP en Programma-onderbreking?

- Info van vorig programma gaat verloren bij een sprong-bevel
 - o Bevelenteller krijgt nieuwe waarden, adres van vorige stap is weg!

4.3.3 Het Programma-Onderbrekingsmechanisme

Programma-Onderbreking

- Mechanisme dat de volgordebesturing/sequentiële besturing kan doorbreken
- Is in staat uitvoering van programma te onderbreken en ander programma (verder) uit te voeren
- Aanwezig op elke Moderne computer
- Werkt met behulp van Programma-onderbrekingsvlaggen (**PO-vlaggen**)
 - o In DRAMA 9 verschillende soorten

- CVO zal enkel ingaan op PO-aanvragen net na de uitvoering van een instructie (niet tijdens!)
- Nood aan uitbreiding bevelencyclus!
 - 1) Instructie ophalen
 - 2) Instructie analyseren
 - 3) Uitvoering instructie
- 4) Indien PO-vlag staat en CVO wil hierop ingaan**
 - Toestand Bewaren
 - Bewaren van BT, CC, OVI, SOI, ... => **Hele PTW**
 - Bij oude computers op vaste plaats (werkt niet bij recursieve PO), nu op stapel
 - PTW op stapel bewaren --> Risico van overloop
 - Zet de PO-Vlag af
 - Ander Programma Uitvoeren
 - BT krijgt nieuwe waarde
 - ◆ Vaste waarde afhankelijk van onderbreking
 - ◆ Waarden vast bepaald of instelbaar
 - ◆ **Programma-onderbrekingsvectoren PO-vectoren**
 - ◇ Tabel waarin adressen staan die in de BT moet komen bij begin uitvoering PO-aanvraag

Illustratie pagina 113-114

4.3.4 Soorten Onderbrekingen

Klassen van Programma-Onderbrekingen

- Externe Oorzaken
 - Een klok, wekker, reset-toets indrukken, ...
- Randapparatuur
 - Bestuurder van invoer-uitvoerapparaat, hulpgeheugen, netwerkkaarten, ...
 - Klaar met aangevraagde bewerking, een fout, ...
- Machinefout
 - Elektrische spanning valt weg, geheugenfout, gegevensoverdracht-fout, ...
- Programmafout
 - Overloop, Stapeloverloop, Ongeldig adres, ongeldige instructie,
- Geprogrammeerde programma-onderbreking
 - Programma wil zelf onderbroken worden op bepaald moment

Zie tabel met soorten onderbrekingen en de oorzaken hiervan pagina 116

Synchrone oorzaken

- Veroorzaakt door het programma zelf
- "Traps", "Exceptions"

Asynchrone oorzaken

- Veroorzaakt door besturingssysteem
- "Interrupts"

4.3.5 Verbieden van programma-onderbreking

Soms is het nodig om onderbrekingen te verbieden

- Opschorten/Uitstellen onderbreking (*In DRAMA altijd het geval*)
- Negeren onderbreking

Methodes voor onderbreking

- Maskers
- Prioriteiten

Opschorten / uitstellen van programma-onderbreking

- Onderbreking (tijdelijk) onmogelijk maken
 - PO-vlag kan niet gezet worden

- Onderbreking tijdelijk opschorten

PO-Maskers (in DRAMA)

- Onderdeel van PTW
 - o Vormen maskerwoord in PTW
 - o Waarde 0 of 1
 - 0 --> "PO-aanvraag Toegelaten"
 - 1 --> "PO-aanvraag Afgekeurd/Uitgesteld"

Indien masker opstaat kan men nog steeds PO-vlaggen plaatsen, zodra masker wegvalt zal deze de aanvraag zien
"Je kan de telefoon laten rinkelen, zodra je de koptelefoon afzet hoor je het wel"

- o Bij alle maskers in DRAMA (buiten GPF)

Alternatief is om te verhinderen dat er nog PO-vlaggen geplaatst kunnen worden

"Je legt de hoorn langs de telefoon, niemand kan nog bellen"

- o Indien GPF opstaat
- Speciale Instructies voor plaatsen, verwijderen en testen van masker
 - o MKH (Masker Hoog)
 - Zet masker op 1
 - o MKL (Masker Laag)
 - Zet masker op 0
 - o TSM (Test Masker)
 - Wijzig CC (naar 0 of 1 van masker dat je test)
 - o TSO (Test onderbrekingsvlag)
 - Test waarde van onderbrekingsvlag, wijzig de CC
 - "Kijk of de telefoon aan het afgaan is"

Globale Maskers

- 2 Speciale maskers met grotere invloed
 - o G: Globale Masker
 - 1e cijferpositie maskerwoord in PTW
 - Indien op 1, geen enkele PO toegestaan
 - Bij instructie "MKH G" wordt deze op 1 gezet en komen geen PO meer binnen
 - o GPF: Globale Programma-fout
 - 2e cijferpositie maskerwoord in PTW
 - Verhinderd onderbrekingen ten gevolge van overloop en stapeloverloop
 - Indien op 1, OVL en SPL genegeerd
 - Onderbrekingsvlaggen voor stapeloverloop en overloop niet meer mogelijk
 - Wel nog plaatsing van OVI en SOI

Prioriteiten

Wat bij gelijktijdige PO-aanvragen door programma's?

- Prioriteitschema, gerangschikt op belangrijkheid programma
- Bij ingaan PO-aanvraag wordt Onderbrekingsniveau ONV gewijzigd naar huidige waarde prioriteit
 - o PO-aanvragen met lager ONV zullen niet uitgevoerd worden

Bekijk aanpassing aan bevelencyclus DRAMA-machine pagina 121

4.3.6 Programma-Onderbrekingsgroutines

Wat moet gebeuren als de PO-aanvraag aanvaard wordt?

PO-behandelingsroutine

- Apart programma voor elk soort programma-onderbreking
- Interrupt service routine / handler
- **Moet zorgen dat programma kan verdergezet worden na de PO**

- Programma-onderbreking mag geen registers wijzigen!
 - o Daarom PO-routine deze bewaren vóór uitvoering, en herstellen bij terugkeer naar oorspronkelijke programma
 - Niet op gewone stapel zetten --> Vergroot kans van stapeloverloop
 - Vaak op vaste plaats voorzien voor besturingssysteem --> Wordt toch niet gewijzigd
- KTO
 - o "Keer Terug na Onderbreking"
 - Eindigt de PO-Routine
 - BT Terug oorspronkelijke waarde (*gelijkenis met KTG*)
 - ONV terug oorspronkelijke
 - CC, OVI, ... herstellen

=> PTW wordt grotendeels herstelt!

4.3.7 Geprogrammeerde Programma-Onderbrekingen

Een PO-Vlag wordt geplaatst door een speciale instructie in programma: **Onderbrekingsinstructie OND**

OND - Onderbrekingsinstructie

- Plaatst PO₁-vlag
- Zorgt voor PO bij instructie
 - o Indien globaal masker afstaat en ONV dit toelaat
- Heeft geen argumenten/operand
- Belangrijk in context van besturingssystemen!
- Laat programma's toe om 'ordelijk' hulp te vragen aan het besturingssysteem bij moeilijkere taken

Systeemoproep

- Een OND gericht naar het besturingsprogramma voor een specifieke taak
- Roept procedure van besturingssysteem op
- Taak wordt gevraagd door het dienstnummer
 - o een argument / operand (*laatste vier cijfers van instructie specificeren*)
- vb: vragen om invoer van het toetsenbord,

Voor- en nadelen OND

Voordelen

- Gemakkelijk invoeren in bestaande programma-onderbrekingsmechanismen

Nadeel

- Veel geheugentoegangen nodig
 - o PTW op stapel zetten
 - o PO-vector ophalen
 - o Dienstnummer ophalen
 - o KTO top van stapel halen

Oplossing op modernere computers

- Nieuwe instructies SYSETER / SYSEXIT
- Adres van PO-routine in speciaal register, Terugkeeradres in ander register
- Bewaren geen PTW

Gevolg = Veel minder geheugentoegangen nodig

Initialisatie DRAMA-Machine

- Start standaard met ONV gelijk aan 0 en alle maskers in PTW de waarde 1
 - o Alle PO worden onderdrukt
- Meeste PO nodig voor in- en uitvoerapparaten

Nagekeken en vervolledigd op 06-01-2023

4.4 Invoer en Uitvoer

Besturing Invoer- en Uitvoerapparaten

- Besturingssysteem
 - o Opdrachten geven/omzetten, PO opvangen, fouten weergeven, ...
- Gebruikersprogramma
 - o Hoog niveau opdrachten geven die BS zal omzetten
- LEZ (*complexe operatie!*)
 - o Toetsaanslagen registreren
 - o Blanko's, enter/return, '-', backspace, ...
 - o Omzetten naar decimaal getal
 - o Tonen op scherm ("echo")

4.4.1 Randapparaten

Randapparaten

- Mechanisch gedeelte
- Elektronisch gedeelte
 - o Besturingseenheid / Bestuurder
 - o Insteekkaart
 - o Zoals SSD, ...

Processor communiceert met randapparaten via de **bestuurder**

- Bevat verschillende poorten / Speciale Registers waarin processor opdrachten kan schrijven of info kan lezen

Lezen en schrijven in poorten

- Met behulp van speciale in- en uitvoerinstructions
 - o INV --> kopieert inhoud van poort naar register en wijzigt CC
 - o UTV --> schrijft de inhoud van register in poort en wijzigt CC
- Door 'inpassing' van poorten in het centrale geheugen
 - o Deel van gewone adresbereik wordt gebruikt om registers aan te duiden
 - o Poorten worden dan in adressen gestoken die kunnen worden opgeroepen
 - o Lezen en schrijven door BIG en HIA
 - o Nadeel: Beschikbare geheugen wordt kleiner
 - In DRAMA gebruiken we daarom de speciale instructies INV / UTV

Poorten van Bestuurder

- Toestandspoort P_t
 - o Toestand van bestuurder / Randapparaten bijhouden
 - KLAAR --> Bestuurder klaar voor nieuwe opdracht
 - BEZIG --> Toestel bezig met uitvoering
 - GEGEVENS --> Gegevens van randapparaat beschikbaar voor gegevenspoort
 - FOUT --> Fout voorgedaan (*onbestaande opdracht, opdracht terwijl apparaat niet KLAAR was, vorige gegevens niet uitgelezen, ...*)

Zie schema pagina 132

- o Alleen uitlezen door INV
- Opdracht- en gegevenspoort P_{og}
 - o Hieruit worden gegevens van het randapparaat gelezen door de processor (INV)
 - o Opdrachten worden hierin geschreven d.m.v. UTV

Bepaalde dingen belangrijk om te melden

- Fouten die zich voordoen
- Indien het randapparaat klaar is
- Indien er resultaten beschikbaar zijn

4.4.2 Intermezzo: I/O en PO-Routines in C

Zie boek pagina 133-134

4.4.3 In- en Uitvoer Organisatievormen

Hoe geeft processor instructies aan de bestuurder van randapparaat?

- Volledig geprogrammeerd door processor
- Via PO (programma-onderbreking)
- Via Directe geheugentoegangen DGT
- Met speciale invoer/uitvoerprocessoren
- Door Satellietcomputers

Geprogrammeerde in- en uitvoer

Oudste methode --> Enkel nog op zeer eenvoudige apparaten

- Gegevenstransport gebeurt teken per teken
- Gegevenstransport onder controle van de processor (INV / UTV)
- Niet efficiënt opgebouwd, veel energie nutteloos

Kenmerken geprogrammeerde in- en uitvoer

- Alle gegevenstransport via de processor
 - o Door gegevens via UTV in gegevenspoort te plaatsen
 - o Door gegevens via INV uit gegevenspoort te lezen
- Processor synchroniseert met bestuurder randapparaat
 - o Voor een opdracht nakijken of bestuurder klaar is
 - o Nakijken door toestand te testen
 - Nieuwe opdracht pas starten bij "KLAAR"
 - Gegevens slechts uitlezen bij "GEGEVENS"
- ACTIEF WACHTEN
 - o Processor wacht tot de bestuurder klaar is, maar test ondertussen in een lus de toestand van de bestuurder
 - o Niet efficiënt

*Actief wachten duurt vaak langer dan het effectief uitvoeren in een programma.
Dat is niet effectief! Zolang je wacht kan er geen nieuwe instructie starten...*

0,6% van de computertijd nuttig gebruikt.

In- en uitvoer m.b.v. Programma-Onderbrekingen

Betere manier = Bestuurder brengt processor op de hoogte als die klaar is

Programma-Onderbreking

- CVO test zelf niet op toestandspoort --> Niet meer actief wachten
- Bestuurder van randapparaat
 - o Meldt aan CVO als er klaar is met opdracht
 - o Nieuwe opdracht klaar voor
 - o Fout voorgedaan --> bestuurder herinitialiseren door processor
 - o Gegevens beschikbaar --> Lezen mag uitgevoerd worden door de processor in de gegevenspoort
- Processor bepaald nog steeds wat er gebeurt, bestuurder geeft aan wanneer

98,7% van de computertijd nuttig gebruikt

Directe Geheugentoegang

Directe Geheugentoegang DGT

- Bestuurder plaatst zelf gegevens in het geheugen, zonder tussenkomst processor
- Bestuurder roept PO op na uitvoering van hele opdracht

- Niet mogelijk om één bus meerdere keren te gebruiken op hetzelfde moment op verschillende apparaten!

Voorbeelden pagina 147-152 bekijken en begrijpen

- **Grote tijdswinst afhankelijk van schijf**
 - o Rotatiewachttijd (wachten op juiste sector)
 - Hoe sneller schijf draait, hoe kleiner wachttijd
 - o Debiet (snelheid aflevering gegevens)
 - Hoe hoger debiet, hoe kleiner wachttijd

? - Geheugencycli Stelen

- o Schijfbestuurder steelt geheugencycli van processor

Kanaalbestuurders

Grote Computerinstallaties met veel randapparaten

- Aparte gegevenspaden met gespreid geheugen
- Geheugencyclus-diefstal beperken
- Aparte invoer/uitvoer processoren of kanaalbestuurders voorzien
 - o Nemen aantal taken van processor over
- Processor
 - o **Kanaalprogramma opstellen**
 - Lijst opstellen met uit te voeren opdrachten doorgeven naar kanaalbestuurder
- Kanaalbestuurder
 - o Voert kanaalprogramma uit zonder processor betrekken
 - Geven van besturingsopdrachten aan besturingseenheid
 - Transporteren van gegevens naar geheugen
 - Uitvoeren van testen op correctheid van centrale geheugen
 -
 - o Is een "gespecialiseerde computer"
 - o Eigen verwerkingsorgaan, geheugen en set bevelen (= kanaalopdrachten)
- Voordelen
 - o Veel minder PO's in de processor

Kanaalprogramma

- Eigen machinetaal
- Bestaat uit **kanaalopdrachten**
 - o Vaak lange bevelen met veel argumenten
 - o Nemen soms meerdere registers in beslag
 - o Moet zowel aangeven voor welke bestuurder als voor welke schijf een opdracht is (want kan meerdere apparaten besturen)

Satelliet-Computers

Satellietcomputers

- Geen gebruik maken van kanaalbestuurder, wel een 'gewone' computer om te versterken
- Front-end
 - o Computer verbonden met in/uitvoer apparaten
 - o Verbonden met 100-en terminals
 - o Invoer van toetsenborden
 - Inlezen
 - Lokaal editeren
 - Op scherm tonen
 - Doorsturen naar hoofdcomputer
- Back-end
 - o Vervult andere taken dan front-end
 - o Beheer van info op hulpgeheugen
 - o Computer verbonden met hulpgeheugens
- Voordelen om computer ipv kanaalbestuurder te gebruiken?
 - o Kanaalbestuurders zijn gespecialiseerd in bepaald randapparatuur, satellietcomputer kan alles.

- Satellietcomputer ook voor gewone berekeningen (kanaalbestuurder enkel in- en uitvoer)
- Eenvoudiger om programmeur van satellietcomputer te vervangen dan die van kanaalbestuurder
- Bij satellietcomputer niet afhankelijk van fabrikant
- Nadelen?
 - Betrouwbaarheid van computerinstallatie
 - Grotere kans op defecten van computer
- Oplossing voor defecten?
 - Reserve-Computers inschakelen
 - Verzekering inschakelen bij aankoop
 - Recht op onderhoud binnen bepaalde tijd

4.4.4 Randapparatuur-Besturingsroutines

Redelijk moeilijk om rechtstreeks met randapparatuur te werken

- Moet volledig op de hoogte zijn over hoe een randapparaat bestuurd moet worden
 - Welke bevelen beschikbaar?
 - Welke bevelenopmaak?
 - Welke volgorde bevelen?

Er wordt geprobeerd om standaard te maken voor randapparatuur, voorbeeld SCSI-Standard

- Vorm van buffering nodig bij PO-Routines
 - Goede boekhouding nodig en duidelijke structuur
- Tijdsafhankelijke Problemen
 - Indien je niet snel genoeg op PO-aanvraag ingaat
 - Overloop, fout-statussen, ...

Stuurprogramma

- Invoer/Uitvoer zijn complexe handelingen
 - Je moet op de hoogte zijn
 - Welke bevelen, volgorde, ...
 - Weinig systematiek
 - Soms standaardisatie maar veel standaarden?
 - Indien met PO
 - Buffering en boekhouding
 - Tijdafhankelijke problemen
 - Verloren gaan van gegevens

=> Voor elk soort randapparaat

- **Stuurprogramma**
- **PO Routine**

Randapparaat-Besturingsroutine uitgevoerd als

- Het gebruikersprogramma invoer wil krijgen of uitvoer wil sturen naar randapparaat
- Een PO vanwege het randapparaat is opgetreden, en PO-routine wil het randapparaat een nieuwe opdracht geven.

Nagekeken en vervolledigd op 07-01-2023

4.5 Processortoestanden

4.5.1 Halttoestand en Uitvoeringstoestand

Vroeger was halttoestand eerste stand bij aanzetten spanning computer.
Door ROM geheugens kan men direct in uitvoeringsstand starten

Waarom dan nog halttoestand op moderne computers?

- Er zullen periodes zijn waarop de processor niets te doen heeft
 - o Geen programma om uit te voeren
 - o Programma dat wordt uitgevoerd heeft gegevens van (hulp)geheugen nodig
 - processor moet 'wachten' tot gegevens zijn ingelezen
 - o Processor zal dan ACTIEF WACHTEN tot er wel werk is (doorgegeven via PO)
 - Dit kan een DGT-transport verstoren
 - Niet de beste manier!
 - Oplossing = Halttoestand

Uitvoeringstoestand

- Gewone bevelencyclus uitvoeren (bevelen ophalen, analyseren en uitvoeren)

Halttoestand

- Geen bevelen ophalen
- Wel letten op PO-vlaggen, soort "slaapstand"

Stand waarin processor zich bevindt staat in het PTW als tweede cijfer (H/U)

Overgang halt- en uitvoeringstoestand

- Door de uitvoering van een instructie
 - o Enkele instructies kunnen toestandsverandering aanvragen
 - STP --> Plaatst processor in halttoestand
 - KTO --> Indien processor in halttoestand was, zal dit terug gebeuren
 - PO --> Zet processor terug in uitvoeringstoestand
- Door het optreden van een Programma-Onderbreking

4.5.2 Probleemstand en Supervisiestand

In- en uitvoer zijn zeer complexe bewerkingen, overgelaten aan Besturingssysteem

Besturingssysteem

- Voor elk randapparaat besturingsroutines en bijhorende PO-routines

Hoe krijgen we correcte werking van programma?

- Verhinderen dat gewoon programma rechtstreeks bevelen geeft aan besturingseenheden van randapparaten
 - o Zou niet goed zijn en fouten zijn dan snel gemaakt!
- Alle opdrachten door het besturingssysteem laten doorgeven aan de besturingseenheden van randapparaten

Er moet dus een onderscheid gemaakt worden tussen het uitvoeren van het BS of uitvoeren van gewoon programma!

Supervisietoestand

- Geeft aan dat processor bezig is met de uitvoering van het besturingsprogramma
- Ook de geprivileerde bevelen uitvoerbaar

Probleemtoestand

- Geeft aan dat processor bezig is met de uitvoering van een gewoon programma
- Enkel de niet-geprivileerde bevelen

Stand waarin processor zich bevindt staat in PTW derde cijfer (S/P)

Overgang tussen halt-, probleem- en supervisietoestand

- Elke PO zal processor steeds in supervisietoestand
- Terugkeer van PO (KTO) zal processor in vorige toestand brengen
 - o De vorige waarde van PTW stond op stapel, daarin zitten ook de toestanden
- Uitvoering van bevelen verandert toestand niet, behalve bij STP (*naar halttoestand*) en KTO (*naar vorige toestand*)

Zie schema pagina 168 bovenaan voor mogelijke overgangen

4.5.3 Geprivilegieerde Bevelen

Verhinderen dat tijdens uitvoering rechtstreeks de randapparaten worden aangesproken, enkel BS mag dat (via INV, UTV)

Het besturingsprogramma is dus geprivileerd

Soorten Machinebevelen

- Niet-Geprivileerde Machinebevelen
- Geprivileerde Machinebevelen
 - o Enkel als processor in supervisiestand staat
 - o Niet bij probleemtoestand
 - o In DRAMA: alle bevelen die toestand processor of randapparaat beïnvloed
 - INV en UTV
 - MKL, MKH, TSM, TSO
 - KTO en STP (*beïnvloeden processortoestand*)

4.5.4 Supervisie-Oproep

Programma mag zelf niet lezen/schrijven in randapparaten, moet beroep doen op besturingssysteem!

Hoe?

- Niet met SBR --> Verandert de toestand niet, zou dan in probleemtoestand worden uitgevoerd (geen INV/UTV)
- Enkel mogelijk door PO in te zetten
 - o Kan programma zelf via instructie OND
 - Een "Supervisie-Oproep"
 - OND heeft operand het dienstnummer om taak te specificeren voor BS
 - Zet een PO₁-vlag op waardoor PO zal optreden

Supervisie-Oproep

- Besturingssysteem wordt zeer vaak opgeroepen
- Voordelen
 - o Programmeur minder werk
 - o Complexe laag-niveau taken worden correct geïmplementeerd
 - o BS kan veel optimalisaties voorzien, efficiëntie stijgt
 - Minder "Actief Wachten"
 - Betere doorstroom van gegevens door voorspellingen van geheugenbeheer en instructies

4.5.5 De Volledige Bevelencyclus

1. Indien PTW aangeeft dat halttoestand actief is --> Stap 4

Instructie wordt **opgehaald** op adres BT (Optellen BT)

2. **Analyse** Instructie

- Functiecode nakijken

- Geprivilegieerde instructie en processor in probleemtoestand --> PO-vlag geplaatst
- Ongeldige instructies --> PO-vlag geplaatst
- Operand berekend en eventueel uit geheugen opgehaald

3. **Uitvoering** van instructie

- STP brengt processor in 'halttoestand'

4. Processor checkt of er een **Programma-Onderbreking** moet ingaan

- Indien globaal masker --> Begin nieuwe bevelencyclus
- Onderzoek PO-vlaggen indien grotere prioriteit dan ONV
- Negeer PO-vlaggen met masker of waarvoor geen aanvraag is
 - Indien dan geen PO-vlaggen overblijven, begin nieuwe cyclus
- Kies PO-aanvraag met hoogste nummer
 - Plaats PTW op stapel
 - PO-vlag op 0 ("zet de bel uit")
 - Zet de PO-Vector juist
 - Zet CC, OVI, SOI op 0 in PTW

4.6 Multiprogrammatie

Processor zou lang moeten wachten bij sommige programma's

Door PO en DGT kan de processor iets anders doen terwijl de randapparaten bezig zijn.

4.6.1 Het Principe

Wat als er niets te doen is? Geen extra taken voor processor?

- Processor voert STP-bevel uit
 - Zet zichzelf in halttoestand
- Processor kan "actief wachten"
 - Verstoort DGT van randapparaat

Wat als er wel werk is? Meerdere programma's tegelijk?

- Programmawisselingen
 - Overgangen tussen programma's in de processor
 - Kan door PO's, KTO, STP,
- Gelijktijdig uitvoeren van meerdere programma's
- Processor is nooit werkloos
- Uitvoeringstijd programma's komt vrij goed overeen met hun uitvoeringstijd bij monoprogrammatie
 - Omdat het actief wachten vervangen wordt door werk in andere programma heeft geen enkel nadeel

4.6.2 De Doorvoer

Doorvoer = Aantal afgewerkte programma's per tijdseenheid

4.6.3 Het STP-Bevel

STP = Geprivilegieerd bevel

Gewoon programma mag processor niet in halttoestand brengen!

- Weet niet of er nog ander werk is voor processor
- Moet vragen om gestopt te worden aan BS

- BS zal dan programma uit geheugen halen en mogelijk STP-bevel uitvoeren

Als gewoon programma toch STP uitvoert --> Er wordt PO gemaakt en BS kan nog kiezen om te weigeren

4.6.4 Het Besturingsprogramma

Besturingsprogramma wordt complexer door de multiprogrammatie!

- Moet voor elk programma een gegevensstructuur bijhouden
 - Waarde van accumulatoren
 - Plaats waar de processor was (BT)
 - Waarde van CC, indicatoren, (PTW)
- Routines BS worden altijd opgeroepen via een PO-aanvraag
 - Behalve bij het opstarten van computer

Opstarten Computer

- Code die in ROM zit wordt uitgevoerd
- Besturingsprogramma wordt geladen
 - Eerst vanaf vaste plaats de "boot-sector" ingelezen
 - Boot-sector bevat lader die de rest van het besturingsprogramma zal inlezen
- Besturingsprogramma zoekt werk
 - Indien niets, processor in halttoestand

Vanaf dan wordt besturing enkel door PO-aanvragen geactiveerd!

4.7 Soorten Besturingssystemen

Besturingssystemen met Algemeen Nut

Computersystemen voor veel doeleinden

- **Niet-Interactieve** besturing
 - Stapelverwerking
 - Programma kan in 'wachtrij' komen en pas later worden uitgevoerd
 - Programmeur kan tijdens uitvoering niet rechtstreeks interageren met programma
- **Interactieve** besturing
 - Commando's geven mogelijk
 - Eén-gebruiker Systemen
 - Eén gebruiker tegelijkertijd werken
 - Multiprogrammatie
 - Meerdere-Gebruikers Systemen
 - Time-Sharing Systems
 - Meerdere gebruikers tegelijk werken
 - Processor voert taken alle gebruikers om de beurt uit

Besturingssystemen met specifiek doel

Systeem ontworpen voor specifieke taak

- Reële-Tijd Systemen
 - Moeten binnen vaste tijdslimiet reageren

vb: Ziekenhuis-apparatuur, vliegtuigonderdelen

Nagekeken en vervolledigd op 08-01-2023

4.8 Taken van een Besturingssysteem

Taken Besturingssysteem

- Geheugenbeheer --> Beslissen welk stuk geheugen aan welk programma
- Processorbeheer --> Welk programma aan processor en hoe lang?
- Beheer in- en uitvoer --> Besturing alle randapparaten
- Bestandsbeheer --> Op aanvraag beschikbaar stellen voor programma's
- Informatiebeheer --> Op aanvraag beschikbaar stellen voor programma's

4.8.1 Geheugenbeheer

Geheugenbeheer

- Waar welk programma?
 - o Aaneengesloten
 - o Opgedeeld in kleinere stukken
- Geheugenbeheer-eenheid (Dynamische Relocatie)
 - o Ondergrens (Adres)
 - o Bovengrens (Adres)
- Vermijden dat programma geheugen van ander programma kan overschrijven
 - o Geheugenbeschermingsregister --> Duid onder- en bovengrens van toegekend geheugen aan
- Verschillende Algoritmes
 - o Om meerdere programma's uit te voeren
 - o Verschillende mogelijkheden, afhankelijk van de hardware
 - Aaneengesloten programma's,
- Beveiligingsaspecten
 - o Programma's mogen niet aanpassen of schrijven
 - In andere programma's
 - Het besturingssysteem
 - o Anders een grote kans op fouten!

4.8.2 Processorbeheer

Processorbeheer

- Aan welk programma zal CVO werken?
- Keuze gebaseerd op
 - o Prioriteit
 - o Beurtrol / Timesharing
 - Tijd eerlijk verdelen
 - o **Enkel programma's in toestand "klaar" kunnen gekozen worden!**

Beurtrol / Timesharing

- o Meerdere gebruikers werken
- o Besturingssysteem bepaald wanneer welk programma uitgevoerd zal worden
- o Zet "wekker" (PO) om te bepalen wanneer CVO volgend programma zal uitvoeren
- o Responstijd --> Hoe lang voor antwoord?
 - Moet zo klein mogelijk zijn
- Toestanden van programma's bijhouden
 - o Passief --> Nog niet in geheugen gebracht

- Klaar --> Klaar om uitgevoerd te worden
- Wachten --> wacht op andere elementen
- Uitvoering --> Processor is er mee bezig
- Afgewerkt --> Alle bevelen uitgevoerd
- Wijziging in toestand
 - Programma doet supervisie-oproep
 - Randapparaat meld een PO dat het klaar is met opdracht

4.8.3 Beheer Randapparaten

Beheer in/uitvoer

- Besturen van alle randapparaten
- Toestand van alle randapparaten bijhouden
 - Klaar, bezig, defect
- Opdrachten bijhouden indien apparaat bezig is

4.8.4 Bestandsbeheer

Bestandsbeheer

- Bestanden met bijhorende gegevens bijhouden op (hulp)geheugens -> Snel toegang om lezen of schrijven
- Bestand = 'container' met informatie

4.8.5 Informatiebeheer

Informatiebeheer

- Eigen info bijhouden
 - Tijd, Datum, aantal programma's, schijfruimte
- Datum en tijd**
 - Via een klok (2 soorten)
 - Circuit dat om de tijd een PO aanvraagt
 - Oscillator die onder spanning periodisch signaal levert
 - Onder spanning door batterij
- Samenwerking programma's aan één taak
 - Informatie Uitwisselen
 - Communicatie

4.8.6 Boekhouding

Boekhouding

- Facturatie gegevens
 - Hoe lang aan programma gewerkt?
- Statistieken voor planning
 - Wanneer werkloos? Knelpunten? --> Systeembeheer moet analyseren

4.9 Kosten/Baten Aspecten

4.9.1 De Baten

- Gebruiksvriendelijkheid
- Correcte werking
- Efficiëntie besturing

4.9.2 De Kosten

Kosten

- Apparatuur
 - o CVO complexer, duurder
 - o Groot centraal geheugen
 - o Voldoende randapparaten nodig
- Programmatuur
 - o Veel jaren nodig om een besturingssysteem te maken/programmeren
- Overhead aan Tijd
 - o Verspilling extra instructies
 - o Wordt zeker gecompenseerd door grote tijdswinst van BP

4.9.3 Uiteindelijke Rendement

Rendement

- 80% Processortijd: Besturingsprogramma BP
- 20% Processortijd: Gebruikersprogramma GP
- 80% Apparatuur: Beheer van andere apparatuur
- 20% Apparatuur: Uitvoeren van bevelen
- 75% Programma: Organisatie programma
- 25% Programma: Rekenwerk

Conclusie: 1% totaal van de computer voor echt rekenwerk

Nagekeken en vervolledigd op 08-01-2023

5. Computernetwerken

5.1 Inleiding

5.1.1 Waarom Computernetwerken?

Waarom computernetwerken?

- Gemeenschappelijk gebruik hulpmiddelen
 - o Apparaten gemeenschappelijk gebruiken (printers, gegevensbanken, supercomputers,)
 - o Gegevens
- Betrouwbaarheid en beschikbaarheid verhogen
 - o Reserve computer in netwerk kan taken overnemen
 - o Bestanden opslaan op kopieën, ...
- Kostenreductie
 - o Goede prijs
 - Prijs/Prestatie zeer goed
 - Gebruik van servers: apparaat op netwerk levert diensten voor alle gebruikers
 - o Dure randapparatuur: printer, scherm, ...
 - o Uitbreiding eenvoudiger
- Communicatiekanaal
 - o "Elektronische post"
 - o Bestanden overzetten, versturen boodschappen

5.1.2 Componenten van een netwerk

Beeld van de Gebruiker

1. Communicatienetwerk

- Bekabeld netwerk
- Draadloos netwerk
 - o Bekabeling vanaf zendmast

2. Computernetwerk

- Communicatienetwerk
- Toelating nodig van beheerder
- ...

2.2 Gedistribueerd Systeem

- Verzameling onderling verbonden computersystemen
- Computernetwerk
 - o 1 Virtueel Systeem aangeboden aan gebruiker
- Vaak Transparante toegang tot bestanden, printer, apparaten
 - o Het lijkt dat bestanden op je computer staan, maar ze staan eigenlijk op een externe file server
 - o Voor iedereen binnen systeem beschikbaar

Opbouw van een Communicatienetwerk

- Lange afstandsnetwerken
 - o Opgebouwd uit schakelaars
 - o WAN (Wide Area Network)
 - o Bandbreedte
 - o Uitdaging = Routing
- Lokaal Netwerk
 - o LAN (Local Area Network)
 - o Binnen 1 gebouw (max 1km)
 - o Geen schakelaars of routing nodig
 - o Alles via kabel

5.1.3 Schakeltechnieken

Schakeltechnieken

- Lijnschakelen
 - o 3 Fases
 - Opzetten verbinding
 - Gebruik van verbinding
 - Kleine vertraging door snelheid over kabel
 - Verbreken verbinding
- Boordschapschakelen
 - o Verbinding gemeenschappelijk gebruikt, geen reservatie van kanalen
 - o Boordschap van schakelaar naar schakelaar
 - o Store and forward-techniek
 - Boordschap wordt met adressen van zender en ontvanger verzonden via schakelaars
 - o Grote vertraging
 - Boordschap moet volledig ontvangen en bewaard worden bij elke schakelaar
 - Ontstaan wachtrijen aan schakelaars
- Pakketschakelen
 - o Opdeling boordschap in kleinere pakketten
 - Vaak max grootte per pakket
 - o Pakketten zelf kunnen andere weg volgen
 - o Communicatielijnen nooit heel lang bezet

5.1.4 Netwerkachitectuur

Communicatie niet eenvoudig

- Moet betrouwbaar kunnen
- Routebepaling voor elk pakket
- Opsplitsen en correct samenvoegen

Systematische Aanpak Nodig

- Problemen afzonderlijk oplossen
- Gelaagde architectuur met protocols en diensten

Protocol

- Regels waar je je aan moet houden bij samenwerking
 - o Aard en samenstelling van boordschappen
 - o Hoe verloopt dialoog?

Proces = Programma in uitvoering

Vaak communicatie via verschillende lagen

- Voor elke laag een hoofding toevoegen

Dienst

- Aan gebruikers van "entiteit"
- vb: foutvrij transport van pakketten

5.1.5 OSI-Referentiemodel

Open Systems Interconnection

- Model voor opbouw connectie
- Definieert geen protocol! Beschrijft enkel functionaliteit lagen

Verschillende Lagen OSI

- Fysische laag
 - o Versturen van bits over fysische kabel
 - o In protocols over deze laag
 - Aard van kabel/medium
 - Kenmerken Connectoren

- Voorstelling bits op kabel
- Dataverbindingslaag
 - Creatie foutvrij communicatiekanaal
 - Verantwoordelijk voor
 - Herkennen pakketjes
 - Detectie van fouten
 - Controle op stroom van pakketjes "*Flow Control*"
- Netwerklaag
 - Verkeer van pakketjes door netwerk(en)
 - 2 Soorten diensten
 - Virtuele verbindingen (*Lijnschakelen*)
 - Datagram dienst (*Aflevering pakketten*)
 - Letten op
 - Routebepaling
 - Kostendoorrekening
 - Onderling verbinden van netwerken en routeplanning
- Transportlaag
 - Betrouwbaar communicatiekanaal tussen eindsystemen
 - Letten op
 - Efficiënt gebruik van netwerken
 - Opdelen en samenvoegen van pakketjes
 - Flow Control
- Sessiel laag
 - Organisatie van samenwerking
 - Hulpmiddelen
 - Organisatie dialoog (*altijd maar 1 iemand zender*)
 - Plaatsen van synchronisatiepunten (*dialoog hervatten na evt fout*)
- Presentatielaag
 - Behoud van betekenis van uitgewisselde informatie
 - Taken
 - Keuze voorstelling gegevens op de kabel
 - Encryptie van data
 - Eventueel compressie of verscijferen van data
- Applicatielaag
 - Hoogste laag in OSI-model
 - Protocols voor
 - Applicaties, geschreven door eindgebruikers
 - Standaardapplicaties
 - Deelproblemen

Kenmerken OSI

- Belangrijk referentiemodel
- Niet basis voor praktische protocols
- Definitie te lang geduurd
- Internet groeide snel, wachten kon niet
- Lagen voegen hoofding toe om nodige gegevens toe te voegen bij hun communicatie

5.1.6 Internetmodel

- Nooit formeel gedefinieerd of beschreven
- Zelfde structuur als OSI-model
- 5 Verschillende Lagen, geen Sessie- en Presentatielaag

5.2 De Fysische Laag

Bits over een kanaal / Kabel versturen

5.2.1 Basisbegrippen

Analoog vs Digitaal

- Analoog --> Continu kan veranderen
- Digitaal --> Eindig aantal punten zonder tussenliggende punten
- In computergeheugen digitale info opgeslagen

Periodieke Signalen

- Signaal volgens een periode
- Frequentie: Aantal periodes per seconde

Transmissiemedia

- Verschillende soorten media beschikbaar
 - o Gevlochten koperparen
 - Gebruikt in telefoonsystemen
 - o Coaxiale kabels (basisband, breedband)
 - Veel gebruikt!
 - Basisband --> Digitale communicatie
 - Breedband --> Televisiekabels, verdeeld in meerdere kanalen
 - o Glasvezel
 - o Draadloos

5.2.2 Digitale Communicatie

Digitale Communicatie

- Digitaal signaal 0-en en 1-en
- Sync tussen zender en ontvanger
 - o Ontvanger moet met zelfde frequentie lezen als het geschreven is
 - o Vaak beginnen met kleine bitstream voor synchronisatie
- Vervorming signaal tussen zender en ontvanger
 - o Verzwakt (*Weerstand kabel, afstand*)
 - o Vervormt (*Beperkte bandbreedte*)
 - o Vertraging (*Snelheid van kabel = Transmissiesnelheid*)
 - o Ruis (*Bronnen uit omgeving rond kabel*)

5.2.3 Modems

Modem

- Modulator - Demodulator
- Zet digitaal signaal om naar een analoog signaal
- Digitaal signaal over kanaal met beperkte bandbreedte
 - o Draaggolf (*periodieke functie*)
 - Amplitude
 - Frequentie
 - Fase
- Baudrate = Aantal mogelijke signaalveranderingen per seconde
- Aantal bits per seconde = Baudrate * n (bits)

ADSL

- Asynchronous Digital Subscriber Line
- Manier om snelheid op te drijven
- Telefoonkabel van centrale naar eindgebruiker
 - o Bandbreedte 1,1MHz
 - o Gefilterd voor stemgeluid
- Zonder filter
 - o Bandbreedte * 250!
 - o Opdeling in 256 kanalen
 - o 1e kanaal voor telefonie
- ADSL Modem
 - o Continu kwaliteit van kanalen opvolgen en optimale parameters bepalen
 - o Filter zelf instellen

Digitaal signaal <-> Analooog signaal

5.3 Dataverbindingslaag

Mogelijke fouten in transport herstellen, Maken van betrouwbaar kanaal

5.3.1 Taken

Taken van Dataverbindingslaag

- Afbaken van pakketjes
 - o Begin en einde herkennen
 - o Meegeven door bitpatroon
 - Speciaal gemaakt en zender verwerkt zodat bitpatroon steeds uniek is
- Detectie van fouten
 - o Extra controlebits toevoegen voor controle transport
 - Pariteitsbit --> Maakt totaal aantal 1-bits even
 - Complexere manieren
 - o Indien fout: opnieuw zenden door zender
 - o Duplicaten herkennen
 - Volnummers meegeven
- Stroombeheersing
 - o Flow Control
 - o Verhinderen dat snelle zender een trage ontvanger gaat overspoelen
 - o Ontvanger geeft toegang om pakket te versturen, zender wacht
- Pakketlayout
 - o Pakketten die Dataverbindingslaag stuurt hebben vaste vorm
 - Beginpatroon (01111110)
 - Hoofding (met volgnummer)
 - Data
 - Controlebits
 - Eindpatroon (01111110)

5.3.2 Algoritmen en Protocols

Algoritmes & Protocollen

- 1 zender en 1 ontvanger
- Communicatie gebeurt in één richting

5.3.3 Protocol 1: Geen fouten, geen stroombeheersing

Zender stuurt onbeperkte hoeveelheid pakketten naar ontvanger, opeenvolgend zonder wachttijd

--> Niet realistisch, ontvanger is hier oneindig snel

5.3.4 Protocol 2: Met stroombeheersing, geen fouten

Stroombeheersing

- Zorgen dat zender nooit de ontvanger kan overspoelen met pakketten
- Tweerichtingsverkeer tussen zender en ontvanger

Zender stuurt 'aanvraag' en wacht op de toelating van ontvanger om daarna het pakket te versturen

5.3.5 Protocol 3: Stroombeheersing, Fouten mogelijk

Mogelijke Fouten

- Verzonden pakket wordt niet ontvangen
- Verzonden pakket wordt als foutief herkend

Zender stuurt aanvraag, wacht op toelating en stuurt dan pakket.

Indien bevestiging van ontvangst niet ontvangen stuurt zender na tijd opnieuw.

Zie schema's in boek pagina 34

Volnummers = Om duplicaten te herkennen

- Afwisselend gebruik van 1 en 0 als volgnummer
- Ook meegeven in de toelating om fouten te vermijden

Bevestigingen = Reactie van ontvanger op een aanvraag

- *Zie cursus pagina 37 en begrijp wat er mis kan gaan bij oproepen*

5.3.6 Protocol 4: 1-Bit Vensterprotocol

Dataverkeer in beide richtingen ondersteund

Zie schema pagina 39

Zender stuurt enkel pakket indien bevestiging is ontvangen van vorige zending, anders terug dezelfde

5.3.7 Uitbreidingen

Zendvenster = Zender kan meerdere pakketten versturen zonder eerst bevestiging te ontvangen

Zie pagina 40-41 voorbeeld

5.4 Lokale Netwerken

5.4.1 Algemene Kenmerken

Lokaal Netwerk = Verbinding tussen computers in ruimte

- Eenvoud zeer belangrijk
 - o Geen afzonderlijke schakelaars, enkel in computers zelf
 - o Eenvoudige topologie zodat probleem van routebepaling vervalt
 - Bus- of ringtopologie
- Gemeenschappelijk gebruik van communicatiekanaal
 - o Ieder aangesloten computersysteem kan op willekeurig moment starten met zenden van pakketten
 - Onder leiding van een protocol (in Dataverbindingslaag) die toegang regelt
 - o Ieder aangesloten computersysteem ontvangt ieder pakket uitgezonden
 - Enkel de pakketten voor dat lokale systeem bestemd worden doorgegeven aan hogere lagen
 - Vrij eenvoudig om 'af te luisteren'
- Netwerklaag is meestal leeg in lokaal netwerk, tenzij dat met een ander netwerk verbindt
- Dataverbindings- en fysische laag worden verder opgedeeld in lokaal netwerk

- Logic Link Control LCC
 - Zorgt dat dezelfde dienst door alle lokale netwerken aan netwerklaag wordt aangeboden
 - Doet hetzelfde als de Dataverbindingslaag, **datagramdienst (hoge kwaliteit fysische kanalen)**
- Medium Access Control MAC
 - Typisch voor elk lokaal netwerk, transport van pakketten

5.4.2 Ethernet

Ethernet

- Naam voor lokaal netwerk gestandaardiseerd onder IEEE 802.3
- Bus-Netwerk
- Fysische laag veel alternatieven mogelijk

Fysische laag Ethernet

- Elke computer heeft een interfacebord met aansluiting voor coaxiale kabel
- Kabel verbind verschillende computersystemen

Botsing pakketten

- Indien meerdere computers tegelijk een pakket versturen over de kabel
- Signalen zullen interfereren en geen enkel pakket komt aan

Werkingsprincipe Ethernet

REGEL 1

Computersysteem mag slechts starten met uitzenden pakket indien kabel vrij is, indien het computersysteem geen signaal ontvangt

Er blijft een mogelijkheid op botsing, indien ze op gelijke momenten versturen en de ene nog niet aangekomen was.

REGEL 2

Tijdens uitzenden pakket blijft zendend computersysteem luisteren naar signaal op kabel. Uitzenden wordt gestopt zodra botsing wordt vastgesteld, kort stoorsignaal uitzenden

Daarna opnieuw poging tot uitzenden zelfde pakket (zie Regel 3)

REGEL 3

Na botsing wachten systemen willekeurige tijd voor ze het pakket opnieuw uitzenden. Algoritme zorgt dat wachttijd groter wordt na veel botsingen. Na 16 mislukte pogingen een faling gemeld.

- **Werking Omschreven door CSMA/CD**
 - MA --> Multiple Access
 - Alle computers hebben gelijktijdig toegang
 - CS --> Carrier Sense
 - Computer kan kijken of kabel in gebruik is of kunnen draaggolf detecteren
 - CD --> Collision Detect
 - Computer kan botsing detecteren
- **Voordelen Ethernet**
 - Bij lage belasting netwerk is vertraging klein
 - Bij hoge belasting wordt 90% van beschikbare transmissiecapaciteit gebruikt
- **Nadelen Ethernet**
 - Geen waarborg dat systeem dat pakket wil sturen dit ook kan doen
 - Het kan dat telkens een systeem merkt dat een ander systeem iets verzend
 - Minimale lengte voor uitgezonden pakket vereist om botsingen te herkennen
- **Pakketlayout Ethernet**
 - Iedere verbindingskaart uniek adres
 - 6 Bytes
 - Hoofding, adres van zender en ontvanger, **broadcast**-adres (om naar alle computers binnen netwerk te sturen),

- **Kan Broadcast in Lokaal Netwerk!**
- Diensten Ethernet
 - o MAC-laag --> Datagramdienst
- Bekabeling
 - o Tegenwoordig geen coax, maar gebruik van Koperparen in Hubs
 - o Soms Switches die hubs vervangen
 - Hebben een bufferruimte voor pakketten
- Fast Ethernet
 - o Nieuwe standaard, hogere snelheid zelfde minimale pakketgrote

5.4.3 Draadloze LAN (WiFi)

Wireless LAN (WiFi)

- Celvormige architectuur
 - o Elke cel gecontroleerd door Basisstation / BS / Toegangspunt
- Toegangspunten van cellen via ruggengraatnetwerk verbonden
- Werking omschreven door CSMA/CA
 - o Carrier Sense Multiple Access with Collision Avoidance
 - o **Botsingen proberen te vermijden in plaats van ze te detecteren**
- Zelfde regels als Ethernet
 - o Elk station kan ontvangen en zenden
 - o Niet zeker dat alle stations die zich in één cel bevinden elkaar kunnen horen
 - Botsing in buurt van ontvanger nog steeds mogelijk
- Protocol
 - o Station dat iets wil versturen zal nagaan of niemand anders pakket verstuurt, anders wachten
 - o Indien kanaal vrij is mag versturen, ontvangende station controleert de controlebits en stuurt bevestiging terug
 - o Indien zender bevestiging ontvangt was er geen botsing, indien geen bevestiging wordt pakket opnieuw verzonden

Hoe kunnen stations nagaan of kanaal vrij is?

- o Virtual Carrier Sense
 - Station dat wil versturen zal kort een aanvraag-pakket versturen (*Request To Send*)
 - Ontvanger zal antwoorden met controlepakket met gelijke inhoud (*Clear To Send*)
 - Alle stations die RTS- of CTS-pakket ontvangen zullen gedurende gegeven tijd het kanaal als 'bezet' beschouwen

Nagekeken en vervolledigd op 08-01-2023

5.5 Netwerklaag

Moet computersysteem mogelijkheid bieden pakketten uit te wisselen met ander computersysteem

5.5.1 Diensten

Diensten door Netwerklaag

- Virtuele Verbinding
 - o Alles correct in volgorde
Niet besproken in cursus
- Datagramdienst
 - o Geen garantie aflevering
 - o Volgorde aflevering kan verschillen van de verzending

Aan elke computer in netwerk een adres toegekend

5.5.2 Protocol

Protocol voor netwerklaag met enkel datagramdienst

- Pakket kunnen versturen
- Nodige info in hoofding
 - o Netwerkadres van ontvanger
 - o Netwerkadres van zender
 - o Protocol van transportlaag, waaraan data moet worden doorgegeven

Netwerklaag vraagt pakket over bepaalde verbinding te zenden

5.5.3 Routing

Belangrijke taak = Zorgen voor transport van pakketten van zender naar ontvanger

- Iedere schakelaar heeft een routingstabel
 - o Bevat per mogelijke bestemming de uitgaande lijn die moet gebruikt worden

Statische Routing

- Tabellen worden door netwerkbeheerder op verschillende schakelaars geplaatst
- Voor iedere bestemming heeft schakelaar 1 lijn
- Alle pakketten naar een bestemming volgen dus dezelfde route --> **Belangrijk nadeel**

Routeringsprotocol

- Schakelaars wisselen onderling info uit over aangesloten computersystemen
- Op basis van die info de routes berekenen

5.6 Netwerken Verbinden: een internet

Tot nu toe altijd één netwerk beschouwd, niet logisch

- Bedrijven willen volledige controle over eigen computernetwerk
- Aansluitingen op andere netwerken is wel gewenst
- Binnen organisatie verschillende netwerken; voor elk departement een netwerk

Internet = Netwerk opgebouwd door verbinden van verschillende netwerken

Communicatie binnen netwerken: Routers!

5.6.1 Adressering

Communicatie tussen computersystemen van verschillende netwerken

- 1 Globaal, uniform adresseringsschema
- Internet
 - o Adres: 32 bits
 - o Verschillende soorten adressen
 - Lengte van adres kan verschillen
- Omzetting internetadres naar lokaal adres?
 - o Vertaaltabellen of vertaalprotocol

Opbouw Adres (32 bits)

- Gedeelte duidt netwerk aan
- Overige verwijzen naar computersysteem
- 2^{32} mogelijke combinaties voor IP-adressen
 - o Vroeger zeker voldoende, nu niet meer
 - o Andere manier nodig

Hoe meer apparaten een IP-adres geven dan er adressen zijn?

- Adressen dynamisch toekennen
 - o Tijdelijk een adres geven indien apparaat online
 - o Indien offline IP-adres terug vrijgeven
- Netwerkadresvertaling
 - o Alleen router die netwerk van bedrijf verbindt met Internet heeft een echt IP-adres
 - o Intern geeft bedrijf dan aan elk apparaat een eigen adres
 - o Router zal dan uitgaande IP-adres wijzigen naar globale IP-adres en vertaling bijhouden in vertalingstabel
 - o Router zal inkomende Pakketten toewijzen en het globale IP-adres vervangen door een lokaal IP-adres
 - o Voordeel --> Men weet van buitenkant niet hoeveel apparaten op netwerk zijn aangesloten
- Nieuwe versie van IP-Protocol (2017)
 - o 128-bits adres ipv 32-bits adres gebruiken
 - o Eerste 64 bits duiden netwerk aan, laatste 64 apparaat

5.6.2 Diensten en Protocol

Netwerklaag: alleen datagramdienst

Protocol

- IP = Internet Protocol
- Eén type pakket
 - o Hoofding
 - o Adres afzender en bestemming
 - o Lengte datagedeelte
 - o Transportprotocol
 - o Time-To-Live TTL (aantal routes waarlangs pakket mag passeren)
 - o Type-Of-Service (Soort dienst dat verwacht wordt)
 - o Opsplitsing IP-pakketten in meerdere pakketten

Indien TTL 0 wordt --> Foutmelding naar afzender door ander protocol

5.6.3 Routing

Elke router

- Een eigen (globaal) adres
- Een eigen routingstabel
 - o Adressen van andere routers in netwerk worden getoond

Routeringsprotocols

- Autonoom Systeem AS
 - o Tussen routers gebruik van protocols

- Kortste pad tussen 2 routers berekenen
- Beleid van netwerkbedrijven

5.6.4 Praktisch Voorbeeld

Zie cursus pagina 59

5.7 Transportlaag

Twee protocols en bijhorende diensten voor applicatielaag

- TCP Transmission Control Protocol
- UDP Universal Datagram Protocol

5.7.1 Adressering

Adres van communicatie-eindpunt voor processen

- IP-adres
- Poortnummer.
 - Poortnummer gegeven door besturingsprogramma
- Identificeert een computersysteem en het proces zelf
 - IP-adres gebruikt in IP-Protocol
 - Poortnummer mee opgenomen in hoofding van TCP- / UDP- Protocol

5.7.2 TCP Protocol & Diensten

Virtuele verbinding tussen eindpunten

- 3 Fases onderscheiden
 - Opzetten van verbinding (in een datagramdienst-netwerk niet zo eenvoudig)
 - Gebruik van verbinding met betrouwbaar transport van data
 - Protocols gebruikt zoals gezien in Dataverbindingslaag
 - Uitgezonden pakketten komen niet in volgorde van verzending toe, binnen netwerk alternatieve routes
 - Verbreken van verbinding

5.7.3 UDP Protocol & Diensten

- Datagram dienst tussen eindpunten
- Redelijk eenvoudig, geen toevoeging aan datagramdienst die IP reeds levert

5.8 Naamdiensten

IP-adressen niet gebruiksvriendelijk

- Leesbare namen zijn gemakkelijker

Oplossing

- DNS!
- Domeinnamen naar IP-Adressen

5.8.1 Domeinnamen

Domeinnamen

- Internet domeinnamen
- Hiërarchisch gestructureerd
 - Lezen van rechts naar links
- Onderdelen

- Landcode, verdere info
 - Meest rechtse deel = Top Level Domain
 - 3 Groepen Top Level Domains TLD
 - 2-letter landcodes
 - Gedefinieerd in ISO, groep bevat '.eu', '.be',....
 - Generieke namen
 - Namen die verwijzen naar organisaties (oorspronkelijk 4)
 - .com --> Commerciele organisaties
 - .org --> Niet-Commerciele organisaties
 - .int --> Internationale organisaties
 - .net --> Bedrijven die netwerkdiensten aanbieden
 - Namen verwijzen naar organisaties binnen VS
 - .edu --> Universiteiten
 - .gov --> Amerikaanse Overheid
 - .mil --> Amerikaanse Leger
 - Elke TLD
 - Beheerder voor toekennen van namen in domein
 - Legt beleid van naamkeuze vast
- Voor .be --> Lang gedaan door Departement Computerwetenschappen, nu door vzw DNS BE
- Elke organisatie met eigen domein
 - Zelf subdomeinen maken

Hiërarchische naamgeving met overdracht van bevoegdheden zorgt voor werkbare structuur

5.8.2 Omzetting Van Domeinnaam naar IP-Adres

Naambeheerder

- Computersysteem dat omzetting van domeinnamen naar IP-adres uitvoert
- Ook in een hiërarchische structuur!
- Voor Ieder Domein
 - Bevat verwijzing naar alle IP-adressen van subdomeinen (A)
 - Bevat verwijzing naar andere naambeheerders (NS)
- Voor TLD-Namen
 - Root-naambeheerder
 - Gekend IP-Adres
 - Enkel verwijzingen naar andere naambeheerders

Proces bij opzoeken web-adres

- Naam doorgegeven aan de root-naambeheerder
 - Stuurt verwijzing (IP-adres) van root-domein (.be)
- Naam doorgegeven aan naambeheerder van root-domein (.be)
 - Stuurt verwijzing (IP-adres) naar naambeheerder van domein (google.be)
- Naam doorgegeven aan naambeheerder van domein
 - Stuurt verwijzing (IP-adres) naar naambeheerder van subdomein (afbeeldingen.google.be)
- Laatste subdomein bereikt? Naambeheerder geeft IP-adres terug van webpagina

Optimalisaties?

- Naambeheerders onthouden lokaal opgezochte verwijzingen
- Voor elk domein meerdere naambeheerders voor de veiligheid!

=> Zeer stabiel systeem dat zelden op grote schaal uitvalt

5.9 Toepassingen

Toepassingen van het Internet

- WWW (World Wide Web)
 - o Is voor velen een synoniem voor Internet, maar is niet hetzelfde!

5.9.1 Elektronische Post (E-mail)

E-Mail

- Boodschap naar andere gebruiker versturen over internet
- Opgeslagen op de computer van ontvanger

E-mailadressen

- Binnen Internet wordt in e-mailadres een domein gebruikt (achter de @)

E-mailprogramma's

- Verzorgen grafische interface naar gebruiker van PC
- Bieden minimaal
 - o Lezen van binnenkomende berichten
 - o Verzenden en opstellen berichten
 - o Beantwoorden, doorsturen en organiseren berichten
 - o Bijhouden van e-mailadressen

E-mailboodschap

- Briefhoofd
 - o E-mail ontvanger/kopie
 - o Onderwerp
 - o Prioriteit boodschap
- Boodschap zelf

E-mail Server

- E-mailprogramma geeft berichten door aan de e-mail server
- Server zorgt voor transport naar server van de ontvanger(s)
- Verschillende protocollen
 - o SMTP (*Simple Mail Transfer Protocol*) --> Wordt steeds gebruikt
 - o POP (*Post Office Protocol*) --> Laat e-mailprogramma toe mail op te halen bij de server
 - o IMAP (*Internet Mail Access Protocol*) --> Communicatie e-mailprogramma en e-mailserver

Hoe vindt e-mailserver de server van ontvanger?

- Door de naambeheerder van het domein
- Voor domeinnaam kan IP-adres van de e-mailserver opgegeven worden

5.9.2 WWW

World Wide Web

- Bestaat uit pagina's met allerlei info
- Gestructureerd als Hypertekst (met links naar andere pagina's)

URL - Uniform Resource Locator

- Bestaande uit 3 delen
 - o Protocol --> Naam protocol dat gebruikt wordt om pagina te laden
 - HTTP(S) : Hypertekst Transfer Protocol
 - o Domeinnaam --> Domeinnaam van computersysteem waarop pagina wordt bewaard (www = webserver)
 - o Padnaam --> Naam van een bestand op de computer

Componenten

- Webbrowser nodig om pagina te bekijken
 - o Kan pagina ophalen bij webserver en tonen op het scherm
 - o TCP-verbinding tussen webbrowser en webserver (met protocol http)

Opbouw Pagina's

- Standaard HTML (HyperText Markup Language)
- Webbrowser beslist hoe de pagina wordt weergegeven

5.10 De Wolk (*The Cloud*)

De Cloud = Metafoor voor het internet

- Alle apparaten die aan de cloud verbonden zijn kunnen communiceren via netwerken in de cloud

Cloud Computing

- Vorm van werken met computers
- Apparaten en hulpmiddelen gemeenschappelijk gebruikt om een taak uit te voeren
- Voordelen voor bedrijven
 - o Niet langer dure investeringen of IT-kennis nodig
- Veel buzzwords die aangeven welk soort diensten aangeboden worden
 - o SaaS - Software as a Service
 - Je krijgt via abonnement of betaling per gebruik de software die je kan gebruiken
 - Geen complexe installaties nodig
 - o IaaS - Infrastructure as a Service
 - Klant krijgt gevraagde infrastructuur ter beschikking
 - Klant kan op infrastructuur zelf installaties uitvoeren
 - Is in werkelijkheid een 'virtuele machine' op krachtige server
 - o PaaS - Platform as a Service
 - Tussenvorm, klant krijgt platform waarop ze een toepassing kan ontwikkelen
 - Moeten zelf platform niet beheren of aankopen
 - Platform heeft reeds besturingssysteem en basissoftware
 - o ... *Nog veel varianten die nog specifiek de diensten weergeven*

Private Cloud

- Een 'wolk' die beperktere omvang heeft dan het volledige Internet
- Privé Wolk
 - o Binnen bedrijf, enkel toegankelijk voor mensen in dat bedrijf
- Publieke Wolk
 - o Toegankelijk via het Internet
- Hybride Wolk
 - o Combinatie van privé en publiek
 - o vb: Google

Voordelen Cloud Computing?

- Zeer beperkte kosten
- Beheer van infrastructuur is veel goedkoper
- Onderhoud moet niet meer lokaal, kan op hoger niveau
- Hard- en software wordt beter gebruikt
- Grote elasticiteit, gemakkelijk groeien of krimpen afhankelijk van de noden. Klein beginnen en opbouwen
- Overal toegankelijk
-

Nadelen Cloud Computing?

- Veiligheid van informatie
- Niet eenvoudig om te veranderen van aanbieder van de Cloud-dienst

5.11 RFID

RFID = Radio Frequenty Identification

- Werd gebruikt om vliegtuigen te detecteren
- RFID Tag
 - o Microchip met radiozender en -ontvanger
 - o Bevat info over voorwerp waarop tag is aangebracht

EPC = Electronic Product Code

- Langere code dan RFID
- Uitgelezen met ONS

Varianten RFID

- Passieve RFID
 - o Energie halen uit signaal dat door lezer is verstuurd
 - o Kunnen enkel sturen als ze signaal ontvangen
- Actieve RFID
 - o Hebben batterij en kunnen zelf iets versturen
 - o Uitgerust met extra functies
 - o Sommige zijn programmeerbaar
- Tussenvorm met kleinere batterij

Toepassingen RFID

- Selfscan in winkel
- Koelkast kan bijhouden wat er in zit
- Boeken in de bib
- *Zie pagina 77*

Communicatieprotocol

- Interactie tussen lezer en de tags moet snel gebeuren
- Via protocol zorgt dat goed gebeurt en chaos vermeden wordt

Hoe zit het met Privacy?

- Gaat in tegen sommige privacy-normen
- Uitlezen van tags gaat zonder eigenaar het weet
- Nog geen degelijke oplossing
 - o RFID tag uitschakelen aan de kassa wanneer het product verkocht werd
 - Speciaal KILL-commando
 - o Beveiligen door code waarvoor je enkel kan lezen met sleutel?

5.12 Het Internet Der Dingen

Internet Of Things

- Dingen die uitgerust zijn met processor of communicatieapparatuur
- Rechtstreeks met internet verbonden, of draadloos communiceren over bepaalde afstand

Toepassingen

- Elektrische thermostaat
- Auto voorverwarmen
- Sensoren hartslag in lichaam
- Chromecast
-

Nog in de kinderschoenen!

- Beveiliging
- Privacy
- Hoe organiseren? Energiebehoefte? Hoeveelheid data?
- E-waste --> Groeiende elektronische afvalberg

Nagekeken en vervolledigd op 08-01-2023

6. Binaire Voorstelling en Verwerking van Gegevens

Dit hoofdstuk zou ik grotendeels ook in het boek studeren, veel afbeeldingen en verduidelijkingen

6.1 Getallen en Talstelsels

Getal = Abstract begrip dat een hoeveelheid aanduidt

Getallenvoorstelling = Manier om naam te geven

Talstelsel = Voorstellingswijze van getallen met zekere systematiek

Berekeningen

- Altijd uitgevoerd met getallenvoorstelling
- Hebben rekenregels

r-tallig talstelsel

- Kannoniek additief-positioneel talstelsel
- Heeft een vast grondtal r
- Getallen voorgesteld door cijfers
 - o Elk cijfer een gewicht in functie van de positie

Voorbeelden r-tallige Talstelsels

- Decimaal ($r=10$)
 - o $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Binair ($r=2$)
 - o $\{0, 1\}$
- Octaal ($r=8$)
 - o $\{0, 1, 2, 3, 4, 5, 6, 7\}$
- Hexadecimaal ($r=16$)
 - o $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f\}$

Zie pagina 83 voor voorbeelden van voorstelling in getallenstelsels

6.2 Getallen in een Computer

Getallen in een computer

- Eindig aantal cijfers
 - o Ligt vast
 - o Soms verschillende formaten (16-bits / 32-bits)
- Enkel voorstelling van eindige-precisiegetallen
 - o Getallen die met beperkt aantal cijfer kunnen worden voorgesteld
- Gevolgen
 - o Niet alle getallen kunnen we voorstellen (er is beperkt aantal cijfers)
 - o Eigenschappen uit de getallenleer gelden niet altijd
 - Soms is optellen niet mogelijk
 - Associativiteit geldt niet altijd
 - Distributiviteit geldt niet altijd

6.3 Natuurlijke Binaire Getallen

Binair stelsel = 2-tallig talstelsel met cijfers 0 en 1, **bits** genoemd

- Eenvoudige rekenregels
- Nadeel --> 3 keer meer symbolen nodig om getal voor te stellen dan in decimale weergave

6.3.1 Omzetting tussen decimaal en binair stelsel

Omzetting van binair naar decimaal

- Via formule
 - o Is vrij lastig en wordt snel heel lang
- Via Horner
 - o Als exponent gebruik je 2 (van het 2-tallig stelsel)

Omzetting van decimaal naar binair

- Zo veel mogelijk machten van 2 aftrekken van het decimaal getal
 - o Grootste macht van 2 zoeken kleiner dan het getal
 - o Die macht van 2 aftrekken van het getal
 - o Proces blijven herhalen tot het verschil 0 wordt
 - o Die som kan je via de formule omzetten naar bits
- Even/Oneven
 - o Indien getal even, is de laatste bit 0
 - o Getal geheel delen door 2
 - o Indien getal even, voorlaatste bit 0
 - o Proces herhalen tot je 0 uitkomt

6.3.2 Aantal bits en aantal decimale cijfers

Hoeveel bits nodig om decimaal getal van n cijfers voor te stellen?

- Bij benadering!
 - o Aantal bits = Aantal decimale cijfers x 3,32

6.3.3 Het Octaal en Hexadecimaal Stelsel

Worden uitsluitend gebruikt als 'compacte versie' voor binaire getallen

Octaal Stelsel

- Cijfers 0, 1, 2, 3, 4, 5, 6, 7
- Voorgesteld door formule
- Begint steeds met inleidende 0
 - o Om onderscheid te maken tussen decimaal en octaal getal

Hexadecimaal Stelsel

- Gebruikt 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
 - o De letters a tot f stellen de getallen 10 tot 15 voor
- Onmogelijk om uit te spreken
- Begint steeds met inleidende 0 gevolgd door een 'x'.
 - o Om onderscheid te maken tussen decimaal en hexadecimaal

Omzetting van / naar Octaal Stelsel

- Vertrekken van binaire voorstelling
- Groepjes van 3 bits maken (vertrekkende van rechts)
- Elk groepje door 1 octaal-cijfer voorstellen, volgens decimale bit-waarde van het groepje

- Omgekeerd gelijke manier, groepjes van 3 maken die getal voorstellen van octaal cijfer

Zie cursus pagina 90 voorbeeld

Omzetting van / naar Hexadecimaal Stelsel

- Vertrekken van binaire voorstelling
- Groepjes van 4 bits maken (vertrekkende van rechts)
- Elk groepje door 1 hexadecimaal-cijfer voorstellen, volgens decimale bit-waarde van het groepje
- Omgekeerd gelijke manier, groepjes van 4 maken die getal voorstellen van hexadecimaal cijfer

Zie cursus pagina 91 voorbeeld

Aantal octale/hexadecimale cijfers nodig voor getal voorstelling?

- Octaal: Aantal decimale cijfers $\times 1,11$
- Hexadecimaal: Aantal decimale cijfers $\times 0,83$

6.3.4 De BCD-voorstelling (Binary Coded Decimal)

Computers rekenen met binaire getallen, maar krijgen veel te maken met decimale getallen (invoer, uitvoer,)

Binary Coded Decimal

- Manier om decimale getallen toch op te slaan in binaire vorm
- Elk cijfer wordt door bits afzonderlijk voorgesteld
 - o Bij de normale binaire voorstelling 1 grote string van bits, bij BCD afzonderlijke strings van 3 bits per cijfer in decimaal getal

Omzetting Binaire getallen en BCD

- Gebeurt meestal via extern programma (*soms in processor zelf*)
- Kan ook door gehele deling door 10 herhaaldelijk toe te passen
 - o De rest is telkens de voorstelling van elk afzonderlijk cijfer

Computers met veranderlijke woordlengte

- Meeste computers met vaste getallengte
 - o Is vaak beperkt, compacte voorstelling van getal is nodig
 - o Om te rekenen wordt binaire voorstelling gebruikt
 - o Getallen kunnen 1 tot 10 decimalen lang zijn
- Sommige computers kunnen rekenen met de BCD-voorstelling
 - o Getallen hebben geen vaste lengte, variëren van 1 tot 16 bytes
 - o Elke byte kan 2 BCD-cijfers bevatten, halve byte gebruikt om teken aan te duiden
 - o Getallen kunnen 1 tot 31 decimalen lang zijn

6.3.5 Bewerkingen met Binaire Structuren

Welke bewerkingen en op welke manier haalbaar?

OPTELLING

De Optelling

- Steunt op de optellingstabel van binaire cijfers 0 en 1
 - o $0+0=0$, $1+0=0+1=1$, $1+1=01$
- Getallen door optellingstabel en het 'meenemen' van cijfers
- Veel overdrachten nodig
 - o Kunnen ook in extra stap pas toegevoegd worden (*zie cursus pagina 96*)

Binaire Optellers

- Elementaire schakeling voor optellen van twee bits
- **Halve Opteller HOPT**
 - o Telt twee bits op en geeft de sombit en overdrachtsbit

- Ingangen --> Op te tellen bits getallen
- Uitgangen --> Som van de twee bits, overdrachtsbit bedoeld voor volgende opteller
- Kan gerealiseerd worden door combinatie van Exclusieve OF- en een EN-poort
- **Hele Opteller OPT**
 - Telt twee bits op, samen met de overdracht uit vorige positie
 - Ingangen --> Op te tellen bits, overdracht uit vorige positie
 - Uitgangen --> Som van de twee bits, overdrachtsbit bedoeld voor volgende opteller
 - Kan gerealiseerd worden door combinatie van 2 HOPT en één OF-poort

Optelregisters

- Combinatie van elementaire optellers om binaire getallen te kunnen optellen
- *In DRAMA Uiteinde optelregister verbonden met overloopindicator van PTW*
- Parallel-Optelschakeling
 - Alle x- en y-bits gelijktijdig aanwezig aan ingangen van de optellers
 - Bewerkingen gelijktijdig uitgevoerd
- Seriële-Optelschakeling
 - Som-bits één voor één berekend
 - Eenvoudigere schakeling, slechts 1 opteller nodig (operand bewaren in schuifregister)
 - Ook hier kijken op overloop

VERSCHIL

Vershil binaire getallen

- Enkel bepaald indien de uitkomst nog positief is (aftrekker < aftrektal)
- Steunt op de aftrekkingstabel
 - $0-0=1-1=0$, $1-0=10-1=1$, $0-1=/$
- Getallen vormen door aftrekkingstabel en cijfers 'meenemen' van hogere positie
- Zelfde eigenschappen als optelling

Binaire Aftrekkers

- Halve Aftrekker - HAFT
- Hele Aftrekker - AFT
 - Heeft een extra **leenbit** die mee in rekening wordt gebracht

PRODUCT

Product van binaire getallen

- Steunt op vermenigvuldigingstabel
 - $0*0=1*0=0*1=0$, $1*1=1$
- Getallen door partiele oplossingen en vermenigvuldigingstabel
- Bij vermenigvuldigen van n bit-getallen is het resultaat grootte van 2n-bits
 - Snel kans op overloop!

Schakeling voor product

- Bij ontwerp verschillende problemen
 - Er moet plaats zijn voor de partiele oplossingen
 - Elementaire opteller zou voor elke bitpositie heel veel ingangen hebben (om partiele oplossingen samen te tellen)
- Oplossing
 - Partiele oplossingen direct optellen indien ze gevormd zijn
Zie oplossingsalgoritme cursus pagina 105

Product met macht van twee

- Indien product met een macht van twee bestaat er een eenvoudiger algoritme
- Indien $X * 2^k$
 - o X verschuiven met k posities naar links (rechts 0 invoegen)

DELING

De Deling van binaire getallen

- Zelfde soort manier als product
- Werkt met serie aftrekkingen en verschuivingen
Zie oplossingsalgoritme cursus pagina 107

Deling met macht van twee

- Indien deling met een macht van twee bestaat er een eenvoudiger algoritme
- Indien $X / 2^k$
 - o X verschuiven met k posities naar rechts

6.4 Gehele Binaire Getallen

n bits kunnen 2^n getallen voorstellen

Gehele binaire getallen voorstellen?

- Voorteken
- 2-Complement
- Verhoogde Voorstelling
- 1-Complement

6.4.1 Met Voorteken

Voorteken voorstelling

- Eerste bit is een tekenbit
 - o Begint eerste bit met 0 --> positief, met 1 --> negatief
- Symmetrisch getallenbereik: evenveel positieve als negatieve
- Twee voorstellingswijze voor 0: -0 en +0
- Veranderen van teken is zeer eenvoudig
- Eenvoudig om teken na te gaan
- Nadelen die overheersen!
 - o Maakt de rekenregels redelijk complex
 - o Door complexiteit niet gebruikt

Bewerkingen op binaire getallen met voorteken

- *Zie cursus voor algoritmes*

6.4.2 2-Complementvoorstelling

2-Complementsvoorstelling

- Positieve getallen: gewone binaire voorstelling
- Negatieve getallen: complement van absolute waarde tov 2^n
- Eerste bit vaak een tekenbit, geen voorteken!

Eigenschappen

- Assymetrisch getallenbereik
- Teken eenvoudig bepaald (1e bit)
- Minder complex: maar 1 voorstelling voor 0
- Eenvoudige rekenregels

R-Complementvoorstelling

- Als negatieve getallen worden voorgesteld door complement van absolute waarde tov R^n

Omkeren van het teken (inverteren)

- Bits omkeren (NIET-bewerking per bit)
- Eén bijtellen (Optelregister)

Berekeningen op 2-Complementsgetallen

- Restklassen --> equivalentieklassen met getallen die dezelfde rest bekomen bij gehele deling (modulo)
Zie cursus pagina 117-120
- Inverteren: bits omdraaien +1
- Som: zelfde als bij natuurlijke getallen
 - o Overloop Mogelijk!
 - Indien 2 positieve getallen negatief resultaat geven of andersom
 - Kan makkelijk gecontroleerd worden
- Verschil: berekend in 2 stappen (indien $x-y$)
 - o Verander teken van y
 - o Tel $-y$ op bij x
 - o Overloop mogelijk als getallen van teken verschillen
- Product: zelfde als bij natuurlijke getallen
 - o Grote kans op overloop
 - Indien absolute waarde groter wordt dan 2^7

Aanpassen woordlengte

- Verlengen
 - o Tekenexpansie --> getal vooraan aangevuld met kopieën van tekenbit
- Verkorten
 - o Kan enkel als alle bits die wegvallen gelijk zijn aan de eerste die nog overblijft

Rekenkundige Verschuivingen

- Delen door macht van 2 --> opschuiven naar rechts
- Product met macht van 2 --> opschuiven naar links (aanvullen met 0)
- Hetgeen wat je "wegschuift" is de rest van de deling!

6.4.3 Verhoogde Voorstelling

Verhoogde Voorstelling

- Alle getallen worden verhoogd met getal N (vaak 2^{n-1})
- Resultaat als natuurlijk binair getal

Eigenschappen

- Asymmetrisch getallenbereik
- Eén voorstellingswijze voor 0
- Teken eenvoudig te bepalen (eerste bit is tekenbit, 1 is positief)
- Is heel gelijkaardig aan 2-complement! Enkel 1e bit inverteren

Bewerkingen Verhoogde Getallen

- Omzetten naar 2-complement (1e bit inverteren)
- Bewerking uitvoeren in 2-complement
- Terug omzetten naar verhoogde voorstelling (1e bit inverteren)

6.4.4 1-Complementvoorstelling

1-Complement

- Negatieve getallen door complement absolute waarde tov $2^n - 1$

Eigenschappen

- Symmetrisch
- Twee voorstellingswijzen voor 0
- Eenvoudiger veranderen van teken dan bij 2-complement
 - Enkel bits inverteren, niet meer optellen met 1
 - Verschil 2-Complement: eerste bit is omgekeerd
 - Voordeel: Positief getal in bits is groter dan negatief getal in bits (als je de bits als getal beschouwd)
 - Nadeel: Complexere bewerkingen

Nagekeken en vervolledigd op 13-01-2023

6.5 Gebroken Getallen

Heel vaak niet-reële getallen in een computer nodig

6.5.1 Klassieke Voorstelling van Niet-Reële Getallen

Gebroken getallen = Niet-reële getallen

Voorstelling

- Nederlands --> getal met komma
- Binair Talstelsel --> getal met decimaal punt

Omzetting Binair naar Decimaal

- Schema van Horner
 - o Werken van rechts naar links in getal
- Vermenigvuldigen gebroken gedeelte met 2^k tot een decimaal getal, dan k delen door uitkomst

Omzetting Decimaal naar Binair

- Gehele en gebroken deel steeds afzonderlijk
Zie werkwijze pagina 137

Optellen / Aftrekken gebroken getallen

- Zorgen dat komma's onder elkaar staan, daarna werken als gewone getallen

Vermenigvuldigen

- Komma wegwerken door product met 100 / 1000 /
- Uitrekenen als gewone getallen
- Aantal cijfers achter komma in resultaat = aantal gebroken cijfers factoren opgeteld

Delen

- Komma wegwerken door product met 100 / 1000 /
- Uitrekenen als gewone getallen
- Aantal cijfers achter komma in resultaat = aantal uit gewone oplossing - verschil aantal van gebroken cijfers deler-deeltal

Rekenregels binaire talstelsel

- Voor som en verschil de komma's aligneren
- Voor product heeft resultaat zoveel cijfers na de komma als de som van het aantal cijfers na de komma in de factoren
- Voor deling heeft het resultaat zoveel extra cijfers na de komma als het verschil tussen aantal cijfers na de komma in deeltal en deler

6.5.2 Gebroken Getallen in een Computer

Voorstelling in Computer

- Nooit oneindige getallen, steeds afgebroken
- Vaste Komma Voorstelling VK
- Bewegende Komma Voorstelling BK

Nagekeken en vervolledigd op 13-01-2023

6.6 De Vaste Kommavoorstelling

Probleem bij computer = Basiselement heeft maar 2 mogelijke waardes (0 of 1), hoe stellen we dan een komma voor?

Vaste Kommavoorstelling

- Je houdt vaste posities die 'voor de komma' staan, en de rest in de voorstelling staat 'achter de komma'
Zie pagina 141 voorbeeld tabel
- Hierdoor kunnen niet alle getallen voorgesteld worden!

6.6.1 Rekenen met de Vaste Kommavoorstelling

Optellen/Aftrekken

- Processor mag denken dat het met gehele getallen is

Vermenigvuldigen

- Niet zo eenvoudig!
- Er kan heel snel overflow ontstaan!
- Mogelijke oplossingen
 - o Uitkomst verschuiven naar rechts --> Veel verlies door overflow
 - o Getallen voor het vermenigvuldigen al verschuiven --> Uitkomst staat dan in juiste vorm

Delen

- Idem vermenigvuldigen
- Beter om al te verschuiven voor de deling
- Terug kans op overflow

Zie pagina 143 voorbeeld in DRAMA

6.6.2 Bereik en Resolutie

Getallenbereik

- Belangrijke karakteristiek van vaste kommavoorstelling
- Voorgesteld door een onder- en bovengrens

Resolutie

- Verschil tussen twee opeenvolgende voorstellingen
- Geeft weer wat kleinste verschil is tussen twee getallen in die getal-voorstelling
- Maximale absolute fout die kan optreden indien we een willekeurig getal willen voorstellen.

Zie figuur pagina 144, maakt begrippen duidelijk!

Relatieve Resolutie

- Quotiënt van de absolute resolutie en het getal dat we willen voorstellen
- Wordt kleiner naarmate getal groter wordt
- Hangt samen met maximale relatieve fout of nauwkeurigheid van de voorstelling

Negatieve vaste kommagetallen

- Indien ook negatieve getallen willen voorstellen kan dat door
 - o Getallen met voorteken
 - o R-Complement voorstelling
- Het bereik veranderd hierdoor!
- Bij verschuivingen steeds de rekenkundige verschuiving, en niet de logische

Afstand tussen vaste kommagetallen is constant!

6.6.3 Gedifferentieerde Vaste Kommavoorstelling

Elk getal mag op een andere plaats de komma hebben

Veel Complexere Rekenregels

- Optelling/Aftrekking
 - o Zorgen dat beide komma's gealigneerd worden (verschuiven van getallen, soms verlies nauwkeurigheid)
 - o Links verschuiven kan overloop veroorzaken!
- Vermenigvuldiging
 - o We weten waar komma in eindresultaat moet staan
 - o Eventuele verschuiving van resultaat
- Deling
 - o We weten waar komma in eindresultaat moet staan
 - o Eventuele verschuiving van resultaat of deler/deeltal

Programmeur kan plaats van komma laten variëren naargelang grootte van getal

- Getal voorstellen door record met
 - o Waarde
 - o Plaats van komma

6.7 De Bewegende Kommavoorstelling

Veel berekeningen is het bereik moeilijk voorspelbaar, veel getallen vallen buiten bereik van vaste kommavoorstelling

Bewegende Kommavoorstelling

- Voorstellingswijze waarbij het bereik van getallen onafhankelijk is van het aantal significante cijfers
- Computerversie van de wetenschappelijke notatie

6.7.1 De Wetenschappelijke Notatie

Veel grote of kleine getallen in wetenschappelijke notatie

$$x = m \cdot b^e \quad \text{met } m = \text{mantisce, } b = \text{basis, } e = \text{exponent}$$

Eigenschappen

- Bereik bepaald door aantal cijfers in exponent, nauwkeurigheid door aantal cijfers in de mantisse
- Voorstelling van een getal is niet uniek, je kan het op meerdere manieren voorstellen
 - o Zou heel ingewikkeld maken, we volgen een standaard

Genormaliseerde Voorstelling

- $1/b \leq |m| < 1$ of $m = 0$
- Basis is steeds 10

6.7.2 Reële Getallen en de Bewegende Kommavoorstelling

Nog niet mogelijk om alle reële getallen voor te stellen (zeer kleine of zeer grote getallen vallen buiten bereik)

Indien een resultaat van een berekening zo'n getal uitkomt met

- Zeer grote positieve/negatieve getallen --> Overloop
- Zeer kleine positieve/negatieve getallen --> Onderloop
- Vaak gebruik van afronding om getal nog te kunnen voorstellen

Afstand tussen bewegende kommagetallen is niet constant!

- Neemt toe naarmate exponent groter wordt
- Is de maximale fout op een getal (kan variëren tot heel groot!)
- Relatieve fout = Absolute fout / Getal

Niet-Genormaliseerde Getallen

- Sommige getallen zouden we wel kunnen voorstellen als we van die norm zouden afwijken

Aantal cijfers in voorstelling

- Door verdeling van cijfers in de voorstelling verschuift het bereik
- Meer cijfers voor mantisse --> Betere nauwkeurigheid
- Meer cijfers voor exponent --> Groter bereik en gelijke nauwkeurigheid
- Meer cijfers voor bereik --> Groter bereik maar kleinere nauwkeurigheid

Bewegende Kommavoorstelling in de Computer

- Getallen binair voorgesteld
- Basis is een macht van 2 (2,4,8 of 16)
- Genormaliseerde voorstelling

6.7.3 Kenmerken van Bewegende Kommasystemen

Ontwerpkeuzen

- Grootte van mantisse en exponent
 - o Meer bits voor mantisse --> Nauwkeuriger
 - o Meer bits voor exponent --> Groter bereik
 - o Programmeur moet afweging maken
- Waarde van de basis
 - o Programmeur afweging moeten maken
 - o Kleinere basis --> Nauwkeuriger
 - o Grotere basis --> Hoger bereik
- Plaats van de komma in mantisse
 - o Meestal helemaal vooraan (voor eerste cijfer)
 - o Kan variëren
- Voorstelling van negatieve mantissen
 - o Hangt af van plaats van komma
 - Is mantisse geheel getal? --> Werken met voortekenen, verhoogde notatie of 2-complement
 - Staat komma vooraan? --> Meestal gebruik van voortekenen of variatie van complement (niet belangrijk)
- Voorstelling negatieve exponenten
 - o Exponent is steeds geheel getal, bekende methodes mogelijk
 - Verhoogde notatie wordt meeste gebruikt
 - Voordeel dat binaire voorstelling van negatieve exponenten kleiner is dan die van positieve exponenten.
- Plaats van verschillende bits
 - o Welke volgorde moeten de bits staan?
 - o Niet zo belangrijk
 - Er kan toch niet rechtstreeks met de voorstelling gerekend worden
- Voorstelling speciale getallen
 - o Getal Nul
 - Niet mogelijk om genormaliseerd voor te stellen
 - Er is een standaard exponent die gebruikt wordt om 0 voor te stellen
 - o Oneindig
 - Voorgesteld door cijferpatroon
 - Rekenregels eenvoudig, resultaat blijft oneindig of onbepaald
 - o Onbepaald resultaat
 - Of NaN (Not a number)
 - Of speciaal cijferpatroon
 - Rekenen met onbepaalde getallen niet mogelijk

6.7.4 Bewerkingen met Bewegende Kommavoorstelling

Afbreken of Afronden

- Als getal uit meerdere cijfers bestaat dan er plaats voorzien is kunnen we het niet meer correct voorstellen
- We moeten het benaderen door een wel voorstelbaar getal
- 2 Manieren om te benaderen
 - o Afbreken
 - Getallen die er niet bij kunnen gewoon weglaten
 - Niet nauwkeurig
 - o Afronden
 - Getal dichtst bij oorspronkelijke getal voorstellen

- Maximale fout is helft kleiner dan bij afbreken!
- 2 Manieren
 - Klassiek afronden --> Getal dat op de grens ligt ook naar boven afronden
 - Statisch afronden --> Getal dat op de grens ligt wordt soms naar boven/beneden afgerond
 - ◆ Even afronden --> Laatste behouden cijfer grensgetal na afronding is even
 - ◆ Oneven afronden --> Laatste behouden cijfer grensgetal na afronding is oneven

Som en verschil

- Om op te tellen/ af te trekken moeten exponenten van getallen gelijk zijn!
- Stappenplan volgen
 - Exponenten gelijkmaken
 - Mantissen optellen/afrekken
 - Resultaat Normaliseren
 - Resultaat Afronden

Product

- Zonder problemen vermenigvuldigen
- Stappenplan volgen
 - Mantissen vermenigvuldigen
 - Exponenten Optellen (overloop mogelijk!)
 - Resultaat Normaliseren
 - Resultaat Afronden

Quotiënt

- Zonder problemen delen
- Stappenplan volgen
 - Mantissen delen
 - Exponenten aftrekken
 - Resultaat Normaliseren
 - Resultaat Afronden

6.7.5 Een Bewegende Kommavoorstelling in DRAMA

In DRAMA geen instructies voor Bewegende Komma, zelf een systeem ontwerpen noodzakelijk

Opdeling geheugenregister (10 cijfers totaal)

- Eerste cijfer is voorteken mantisse (0 voor positief, 9 voor negatief)
- Volgende twee cijfers voor exponent (verhoogde notatie)
- Basis is 100, niet 10
- Volgende 7 cijfers voor absolute waarde van de mantisse (genormaliseerd)

Speciale gevallen in DRAMA

- Oneindig en NaN
 - Eigen numerieke voorstelling

Bewerkingen

- 3 Afzonderlijke delen van een bewegende-kommagetal in aparte registers (teken, exponent, mantisse)
- Na bewerking de delen opnieuw samenbrengen tot bewegende kommavoorstelling
- Bij product moeten we mantisse afronden

★ ***Afzonderlijke bewerkingen uitgelegd met visualisatie pagina 171-181***

6.7.6 Het IEEE-Voorstel

Zeer veel verschillende bewegende kommasystemen BKS

Verschillen van elkaar, dat is een nadeel

- Programmeur moest rekening houden met BKS van computer waarvoor programma bedoeld is
- Programmeur afhankelijk van computertype, moeilijk overzetten naar andere computers
- Veel slordige en onnauwkeurige programma's

IEEE - Standaard

- Verschillende binaire formaten
 - Enkelvoudige nauwkeurigheid (32 bits)
 - Dubbele nauwkeurigheid (64 bits)
 - Vierdubbele nauwkeurigheid (128 bits)

- Gebruiken allemaal de basis 2 en de verhoogde notatie voor exponent
- Hoe meer bits, hoe meer plaats voor mantisse, hoe nauwkeuriger het kan zijn!
- Formaten
 - Beginnen met tekenbit
 - Plaats voor exponent in verhoogde notatie
 - Mantisse, komma steeds vooraan
 - Bit voor de basis (die hier steeds 2 is) wordt verondersteld en niet opgenomen in voorstelling
- Numerische types
 - ★ ○ **Zie cursus pagina 184 onderaan!**

6.8 Letters en Symbolen

Hoe werkt computers met letters en symbolen?

Inwendig Computer

- Alles wordt in binaire getal-vorm bewaard
- Afspraak welk symbool met welk getal overeenkomt = CODE

Communicatie tussen computers? Essentieel dat ze dezelfde taal gebruiken!

=> Codes zijn gestandaardiseerd

6.8.1 EBCDIC

Extended Binary Coded Decimal Interchange Code

- Tabel met voor elk symbool een waarde

6.8.2 ASCII

American Standards Code for Information Interchange

- Elk teken bestaande uit 7 bits
- Werkt met controlewoorden
 - SOH, STX, ETX en EOT bij uitwisseling tussen computers
- Later uitgebreid met een 8e bit
 - Maakte meer voorstellingen mogelijk

6.8.3 UNICODE

Ook nood aan letters met accenten of trema's,

UNICODE

- Aanvaard als internationale standaard
- Elke letter, symbool en teken een uniek 16-bitgetal
 - (16 bits = 65536 verschillende mogelijkheden)
- Eerste 256 waarden zijn gelijk aan de ASCII
 - Maakt overgang gemakkelijker
- Werd nog vaak uitgebreid met meer bits
 - Om nog meer symbolen te kunnen beschrijven

Nagekeken en vervolledigd op 14-01-2023

Test 2 - DRAMA

- 3 Vragen van elk 2 delen
 - o Vraag 1: Vertaling van 4 C-Opdrachten, Administratie + Vertaling (1 zonder en 1 met records/Structs)
 - o Vraag 2: Macro's
 - Macro-Expansie --> Zeggen wat de uitvoer zal zijn
 - Macro Schrijven --> Zeggen hoe je de macro kan maken
 - o Vraag 3: Grote Oefening
 - AR, Toewijzing en Oproeprooster invullen van C-Programma
 - Vertaling van een C-Programma waarbij je de administratie krijgt (12 punten)
 - Stapel kunnen tekenen op een bepaald moment (11 punten)

"Hoe vertaal je de volgende C-code, toewijzing gegeven"

"Schrijf macro die code genereert die iets doet"

Iets met <tekst>[] --> Wat zijn die haakjes in C? Nakijken, staat in een vraag

Test staat op 70 punten!

Tips Test 2

Wijzer in register --> Geen .i nodig om wijzers op te roepen!

Hoe best test maken?

- Begin met de vraag die je goed ligt
- Lees en wees nauwkeurig de opgave
- Let op invullen van tabellen en stapel, daar staan veel punten op!
- Nooit variabele wijzigen als dit niet toegelaten is, andere registers gebruiken dan!
- Index van record altijd vermenigvuldigen met de grootte van het record, denk aan de structuur van de stapel!
- Hoe stapel tekenen?
 - o Niet naar je code kijken, niet nodig. Wel nodig: AR en oproeprooster
 - o Vertrek van de gegeven AR
 - o Maak kopieën van lokale variabelen in registers
 - o Kijk welke functies opgeroepen worden
 - o Kijk oproeprooster na, moeten R1/R2 bewaart worden, ook op de stapel zetten?
 - o Duid de grenzen van het AR aan, altijd onder de parameters/resultaat, boven kopie van variabelen!
 - o Link oude R8 aan elkaar
 - o Duid R8 en R9 aan, R9 bovenaan, R8 naar de laatste R8 op de stapel
 - o Bekende waardes? Bij KTG, schrijf label op de plaats van TKA
- Kennen
 - o Functieresultaten via stapel
 - o Dynamisch geheugenbeheer
 - o ...

Geen punten weg voor fouten, enkel punten krijgen voor juiste dingen.

Examen

Helft van de punten op de testen, andere helft op het examen --> minstens op beide samen een 3/10

Leerstof

- Hoofdstuk 2 tot 6

Vragen

- 5 Vragen
 - o Vraag 1: Wat hoort niet in deze rij? Waarom niet? Vooral kunnen uitleggen!
 - o Open vragen (voorbeelden)
 - Hoe werkt een harde schijf? Hoe wordt informatie bijgehouden?
(Sporen en sectoren, lees-en schrijfkoppen die zweven boven oppervlak, gemiddelde toegangstijd, continue draaisnelheid, ...)
 - Exacte getallen niet heel belangrijk, wel een goede schatting!
 - Hoe werkt de DRAMA vertaler?
(Voorvertaler en vertaler, relocatietabel, ...)
 - Wat zijn programma-onderbrekingen? Waarom nuttig? Leg de werking uit
(Enige manier waarop Besturingsprogramma controle kan houden en efficiënt kan beheren, actief wachtten vermijden, manier om naar supervisietoestand over te gaan, ...)
 - Wat zijn taken van netwerklaag?
 - Stel bepaald getal voor in 2 complement, voer een som uit $01011011 + 11011010 = 11011011 = ?$

LAATSTE LES SOCS GEDAAN!