

Mogelijke examenvragen SOCS

Reinoud Berkein

20 januari 2018

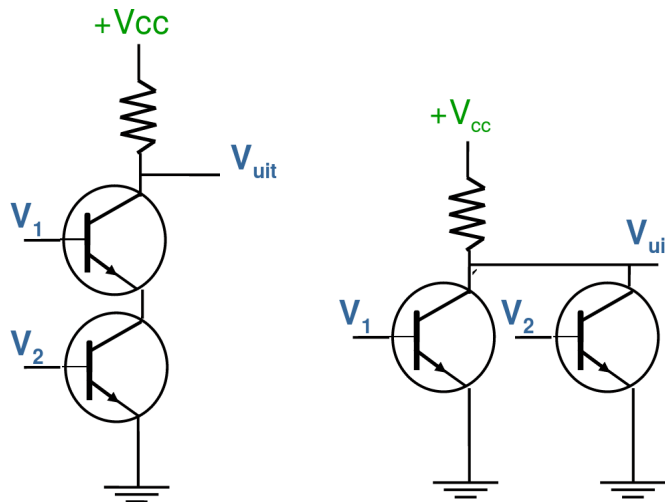
Samenvatting

De vragen zijn afkomstig van de vorige examens en van de leerstof die in de nieuwe druk is toegevoegd.

Hoofdstuk 2

Besprek NOF en NEN poort. Geef de gedragstabellen van de NEN- en de NOF-poort. Geef van de NOF-poort ook de schakeling (met transistors enzo). Waarom zijn deze 2 poorten zo belangrijk? Toon dat aan voor de NOF-poort.

De uit poorten van de NEN- en NOF-poort is de negatie van de EN- en OF-Poort.

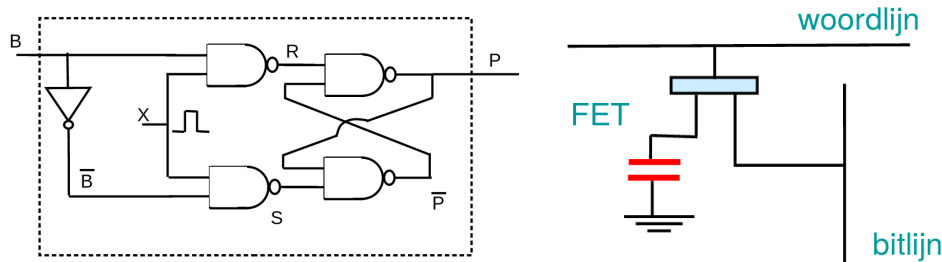


NEN			NOF		
IN1	IN2	OUT	IN1	IN2	OUT
0	0	1	0	0	1
0	1	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0

Elke logische schakeling kan met enkel NEN- of NOF-poorten gemaakt worden.

Wat betekent SRAM en DRAM? Wat zijn de belangrijkste verschillen? Hoe wordt DRAM en SRAM gerealiseerd? Geef ook de schakelingen. Waarom wordt de benaming statisch en dynamisch gebruikt? Waar wordt SRAM en DRAM in hedendaagse computers vooral gebruikt? Hoe wordt er gelezen en geschreven? Wat zijn de verschillen in toegangstijd?

SRAM staat voor Static Random Acces Memory en DRAM staat voor Dynamic Random Acces Memory.



SRAM: We kunnen de bit inlezen door P in te lezen. Schrijven doen we door de bit op B te zetten en daarna kort de kloklijn x een signaal te laten sturen. (R=1 en S=0 geeft P=0 en R=0 en S=1 geeft P=1)

DRAM: voor een 1 te schrijven wordt er een spanning aangelegd op de bitlijn en de woordlijn zodat de condensator geladen wordt. Om een 0 te schrijven wordt de bit lijn geaard en wordt er een spanning aangelegd op de woordlijn zodat de condensator zich kan ontladen. Om te lezen wordt er een spanning aangelegd op de woordlijn, de condensator zal zich ontladen over de bitlijn die kijkt of de spanning over een drempelspanning komt. Als dit het geval is wordt er een 1 uitgelezen en wordt de bit herschreven in het geheugen. Het DRAM-geheugen moet ook continu herschreven worden. Dit gebeurt in blokken. Een blok is tijdens het herschrijven niet toegankelijk om informatie uit te lezen. Het hele geheugen is opgefrist in een periode van (10 tot 100 ms)

Statisch verwijst naar het feit dat een spanning aanleggen genoeg is om de informatie te onthouden, bij een dynamisch geheugen moet het geheugen steeds worden verfrist en zal het altijd bezig zijn met het geheugen te herschrijven.

Voordelen van DRAM:

- Neemt minder plaats in (6transistors <-> 1 transistor en een condensator)
- Lagere energie consumptie
- Compacter
- Goedkoper

Nadelen DRAM:

- Er is een elektronische schakeling voor het opfrissen van het geheugen nodig
- Er wordt destructief gelezen
- Trager van wege het opfrissen en herschrijven
- Omgevingsgevoeliger (condensator)

Een SRAM geheugen heeft een gemiddelde toegangstijd van 10 ns en een cyclustijd van 12ns. DRAM geheugen heeft een gemiddelde toegangstijd van 35-50 ns en een cyclustijd van 70-100 ns.

Statische CMOS-geheugens hebben een hele lage spanning nodig en wordt gebruikt om de tijd en BIOS instellingen bij te houden. DRAM geheugen wordt in de vorm als DDR3 gebruikt als werkgeheugen voor een computer. SRAM geheugens worden vaak als voorgeheugens gebruikt.

Wat zijn MRAM geheugens? Hoe worden ze gerealiseerd? Wat zijn de voordelen en wat is het belangrijkste nadeel?

MRAM(Magneto-resistieve RAM): een bit bestaat uit twee magnetiseerbare lagen gescheiden door een dielectrische laag. De magnetisatie van de onderste laag ligt vast. De magnetisatie van de bovenste laag stelt de bit voor.(dezelfde richting = 0, tegenovergesteld richting = 1) de bits worden verbonden via gekruiste bit en woordlijnen.
Eigenschappen:

- Compactheid DRAM
- Snelheid SRAM(35ns)
- Niet vluchtigheid van flashgeheugens
- ! duur! 16 Mbit kost 20 euro.

Wat zijn ROM geheugens? Waarvoor wordt het gebruikt? Welke versies zijn er van?

ROM staat voor Read Only Memory en kan dus enkel gelezen worden. Ze worden vaak gebruikt in klein electro.

- PROM: Programmable ROM
- EPROM: Erasable PROM
- EEPROM: Electrically EPROM
- Flash geheugen

Magnetische schijven: beschrijf de fysische elementen. Hoe wordt de data wordt voorgesteld en hoe kan er data gelezen of geschreven worden. Bespreek de varianten?(hard zacht) Wat is geheugenspreiding? wat is hoge orde geheugenspreiding, wat is lage orde geheugenspreiding? Hoe lang duurt het vooraleer er data is ingelezen?

Een magneet schijf bestaat uit een as waarop 1 of meerdere platters bevestigd zijn en uit lees en schrijfkoppen. Ieder kant van een platter heeft een magnetiseerbare film en er is een lees en schrijfkop. De as draait rond tegen een vast toerental(4200-15000 toeren per minuut)

Opdeling van de schijf:

1. De plaat wordt verdeeld in cilindrische **sporen**(800/cm)
2. Ieder spoor wordt verdeeld in **sectoren** (verschillende lengte, worden langer als ze meer naar buiten liggen)
3. De bits op de buitenste sporen liggen verder uit elkaar.
4. Men kan ook met **zones** werken(10-30 per schijf). In elke zone hebben alle sporen hetzelfde aantal sectoren.

5. Alle sporen die boven elkaar liggen op de verschillende schijven vormen samen een **cilinder**. Gegevens worden in dezelfde cilinder geschreven en gelezen.

Een sector:

1. Tussen iedere sector is een lege ruimte
2. Ieder sector begint met **een hoofding**, daarna komt de **data** en tenslotte nog de **ECC**.
3. Een schijf moet **geformatteerd** zijn vooraleer je er iets kan op schrijven:
 - (a) **Het laag-niveauformatteren**: schijf opdelen in sectoren en ieder hoofding schrijven. Er wordt ook gecontroleerd of ieder sector beschreven en gelezen kan worden, anders worden ze als slecht beschouwd.
 - (b) **Het hoog-niveauformatteren**: er wordt een tabel bijgehouden over de bestanden op de schijf. Bij Windows zorgt **Format** voor Het hoog-niveauformattering

Door een stroom door de schrijfkop te sturen wordt de film gemagnetiseerd. Als je er met de leeskop overgaat, zal er een stroom worden opgewekt.

Er bestaat ook een soepele schijf: een plastieken schijf met langs beide kanten een magnetiseerbare laag. De koppen glijden over deze laag waardoor er sleet op treedt. De schijf kan daarom ook veel minder snel draaien waardoor lezen veel trager gaat. Er bestaan 2 formaten de 5.25" en de 3.5" beide zijn in een Low Density en High Density uitvoering. Diskettes worden niet meer gebruikt.

Wat is SSD? Hoe werkt het en wat zijn de voor en nadelen?

SSD bestaat uit meerde flash geheugens en een bestuurder. Er zijn ook SSD schijven die uit DRAM geheugens bestaan. Deze worden gebruikt om grote hoeveelheden tijdelijke data te verwerken. Ze zijn sneller maar vluchtig.

Voordelen:

- SSD zijn snel (3x Harde schijf)
- Leessnelheid is overal hetzelfde
- Kleiner toegangstijd(0.1ms)
- Zeer betrouwbaar omdat ze geen bewegende delen hebben
- Minder energie nodig en produceren minder warmte
- Compact en niet zwaar. Ideaal voor portables.
- Tegenwoordig hebben ze capaciteiten groter dan harde schijven.

Nadelen:

- 5 keer duurder dan een harde schijf
- Verslijten na 5000 tot 10000 keer herschrijven
- De besturingssystemen zijn afgestemd op harde schijven en niet op flash geheugens
- De bestuurder is zeer complex en bevat nog vaak fouten.

Wat is CD-R en CD-RW? Geef de voornaamste kenmerken? Hoe wordt de informatie bijgehouden? Zijn ze betrouwbaar?

CD-R staat voor Compact Disc – Recordable. Het is een plastieken schijf met een gouden laag. In deze laag is een groef ingebracht. De bits worden door donkervlekken voor gesteld die de lazer niet weerspiegelen. Een CD-R kan maar 1 keer beschreven worden(WORM).

Een CD-r moet in 1 keer beschreven worden, de schijf draait rond en kan niet worden stop gezet. Als de nodige bits er niet zijn zal de cd onbruikbaar zijn.

CD-RW staat voor CD- rewritable. Omdat de reflecterende laag uit een legering van zilver, indium en antimoon bestaat en tellurium als opnamelaag kan de cd herschreven worden.

Een lazer met hoog vermogen laat de laag smelten waardoor het puntje amorf wordt. Deze reflecteert het licht slecht en zal een 0 voorstellen

Een lazer met een gemiddeld vermogen zal het puntje opwarmen tot boven de kristallisatie temperatuur. Het kristal zal goed reflecteren en een 1 voorstellen.

Bespreek de magneto-optische schijven. Hoe worden gegevens geschreven, gelezen en bijgehouden?

De schijf bestaat uit een substraat van getemperd glas en een registratie laag die magnetiseerbaar is. De schijf kan met een hoog vermogen lezer beschreven worden. Het zal de richting van de magnetisatie veranderen. Initieel is een MO-schijf gemagnetiseerd in 1 richting. (0 of 1) er kan gelezen worden met een laserstraal, de weerkaatste straal zal in uurwerkzin of andersom gepolariseerd zijn en zal dus een 0 of 1 voorstellen. Bij kamertemperatuur is de magnetisatie stabiel.

Wat zijn bussen? Hoe werken bussen? Waarom is er geen chaos in het gebruik van bussen? Wat is busarbitrage?

In een computer moet heel wat data worden gedeeld tussen de verschillende componenten. i.p.v. ieder component onderling te verbinden is er een gemeenschappelijke bus die verbonden is met ieder component. Zo'n bus bestaat uit verschillende kabels die kunnen worden opgedeeld in een adresbus, gegevens bus en een controlebus.

Er is geen chaos in een bus omdat er een bus protocol is. Deze legt vast hoe een apparaat toegang tot de bus krijgt, welke signalen via welke lijn verstuurd moeten worden, de timing en hoe men de bus terug vrijgeeft. Ook de busarbitrage zorgt voor orde, het zorgt ervoor dat er geen 2 componenten tegelijk master kunnen zijn. Dit kan simpele gerealiseerd worden met een busaanvraaglijn.

Eerst wordt er een adres op de bus gezet en daarna worden gegevens verzonden. Bij synchrone bussen gebeurt dit op signaal van de klok.

Wat is het voorgeheugen? hoe werkt het? Over welke keuzemogelijkheden beschikt de ontwerper van een voorgeheugen ? Welke invloed heeft het voorgeheugen op de prestatie van het systeem? Wordt een computersysteem niet nodeloos duurder gemaakt? Toon aan.

Een snel, klein voorgeheugen gecombineerd met een groot, traag geheugen. De kostprijs is iets hoger dan bij een traag geheugen. Als de processor gegevens nodig heeft kijkt hij eerst in het voorgeheugen(treffer). Als het daar niet in zit(misser) kijkt hij in het geheugen en plaats hij de geheugenregisters en de naburige registers in het voorgeheugen. Het voorgeheugen is dus een kleine kopie van het hoofdgeheugen. Niet alles wat in het hoofdgeheugen staat zit in het voorgeheugen.

Prestaties ??

Toegangstijden: T_{RAM}, T_{VG}

Trefkans: t

Gemiddeld:

$$T = t \times T_{VG} + (1 - t) \times T_{RAM}$$

Voorbeeld

$$T_{RAM} = 50\text{ns}$$

$$T_{VG} = 5\text{ns}$$

$$t = 95\%$$

$$T_{gem} = 0.95 \times 5 + 0.05 \times 50 \\ = 4.75 + 2.5 = 7.25 \text{ ns}$$

Kostprijs ??

Capaciteiten:

Prijs:

Totaal:

C_{RAM}, C_{VG}

K_{RAM}, K_{VG}

$$K = C_{RAM} \times K_{RAM} + C_{VG} \times K_{VG}$$

Voorbeeld

$$C_{RAM} = 4\text{GB}$$

$$K_{RAM} = \frac{\text{€}0.10}{\text{MB}}$$

$$C_{VG} = 512 \text{ KB}$$

$$K_{VG} = \frac{\text{€}25}{\text{MB}}$$

$$K = 4096 \times 0.1 + \frac{25}{2} \\ = 409.6 + 12.5 = \text{€}422.1$$

7× sneller < 5% duurder

De ontwerper moet rekening houden met:

1. grootte
2. regellengte
3. organisatie van het voorgeheugen
 - (a) **Direct afgebeeld voorgeheugen:** Elke blok heeft een vaste plaats in het voorgeheugen, elke blok is 1 regel. (som benadeelt bij het gebruik van lussen)
 - (b) **Associatief voorgeheugen:** een register kan om het even waar in het voorgeheugen staan. Elke regel moet daarom wel een uniek label hebben. Er wordt op labels gezocht in het voorgeheugen. (incrementeel en gelijktijdig) bij een misser wordt het label in het voorgeheugen geplaatst. Er moet mogelijk wel een adres worden uit gegooit. AVG zijn het duurst maar niet altijd het snelst. Er is een goede vertaler voor nodig.
 - (c) **Set-associatief voorgeheugen:** elk register kan in het voorgeheugen geplaatst worden maar niet om het even waar. Het adres bepaalt in welke set in het VG het register geplaatst word.
4. gezamenlijk of apart VG(data en instructies)(apart is nodig bij pipelining)
5. aantal voorgeheugens

Wat is pipelining? Welke problemen kunnen er bij opduiken? Welke invloed heeft dit concept op de performance?

Bevelencyclus bestaat uit:

1. ophalen van instructie en ophogen bevelenteller
2. analyseren van instructie
3. berekenen van het adres van de operand
4. eventueel ophalen van operand uit geheugen
5. uitvoeren van instructie

iedere stap wordt in een andere schakeling uitgevoerd. de processor moet niet 1 bevel uitvoeren en dan het volgende maar kan al beginnen zodra een bevel in de volgende stap zit met een nieuw bevel te analyseren.

Problemen:

1. instructie kunnen afhankelijk zijn van andere instructies
2. sprong bevelen (vertraging i.p.v. versnelling)
3. bevelen duren niet allemaal even lang

oplossing voor sprongbevelen:

1. voorspellen
2. een tweede lopende band (werkt niet bij meerdere opeenvolgende sprong bevelen)

Hoe werken multi-core processors? Welke problemen treden er op bij multi-core processors? Wat is snooping? Wat is Hyper-threading?

Multiprocessor: taken verdelen over verschillende cores. De taken moeten wel te verdelen zijn onder de verschillende cores. Meerder programmas uitvoeren kan nu ook verdeeld worden onder de verschillende processors.

Er zijn wel een aantal problemen:

1. **Verbruik en warmteafvoer:** energie besparen door cores uitteschakelen die niet werken en de kloksnelheid lager te leggen dan bij monocoers.
2. **Beheer van de voorgeheugen:** iedere processor heeft zijn eigen VG. Bij een schrijf opdracht moet ieder voorgeheugen worden aangepast. Er wordt een schakeling toegevoegd die alles controleert:
 - (a) Kopieën in het voorgeheugen moeten ongeldig worden gemarkeerd
 - (b) De waardes moeten vervangen worden door de meest recente waarde.
 - (c) Controle op databussen verloopt anders(databus word gebruikt bij een beperkt aantal cores anders wordt een netwerkstructuur gebruikt.:
 - i. Er wordt een update op de bus geplaatst
 - ii. Op deze update moet gecontroleerd worden (snooping)
 - iii. Bij een misser voorziet een ander voorgeheugen de processor
3. **Benutten van de beschikbare rekenkracht:** alle cores moeten evengoed gebruikt worden. Java biedt de klasse thread aan die een abstractie maakt voor het begrip uitvoeringsspoor. De verschillende threads kunnen makkelijk verdeeld worden.

Multi-threading houdt in dat er niet een nieuwe core zich op de chip bevindt maar meerder bevelentellers en accumulatoren. Eén processor regelt dit en kan snelle van taken wisselen door van thread te wisselen.

Geef een overzicht van de verschillende technieken die kunnen gebruikt worden om de snelheid van een computer op te drijven. De beschrijving van iedere techniek afzonderlijk mag beknopt zijn.

1. Voorgeheugen
2. Pipelining
3. Multi-core processors

Hoofdstuk 3

Wat is een assembleerprogramma? Wat is een compiler, wat is een kruiscompiler, hoe zou je te werk gaan om een C-compiler voor een dramamachine te maken? Beschrijf bondig hoe de vertaler voor de DRAMA-machine werkt.

1. Assembleerprogramma, ook wel vertaler genoemd, zorgt er voor dat lagere (of hogere) programmeertaal wordt omgezet naar machinetaal
2. Een compiler zet hogere programmeertaal, zoals C, om naar machine-instructies
3. Kruiscompiler genereert code voor een andere machine dan deze waarop hij zelf uitgevoerd wordt.
4. C-compiler eerst in c schrijven , daarna vertalen met een andere compiler , bijvoorbeeld een c-compiler voor het LINUX-besturingssysteem op een PC. Kan enkel op een Linux pc worden uitgevoerd en niet op een Drama machine zelf
5. Een compiler moet 4 taken uitvoeren
 - (a) Lexicale analyse: herkennen van de woorden v.d. taal (tokens). Vb. '4', '*', ';', 'if'...
 - (b) Syntactische analyse: het herkennen van de individuele opdrachten. Vb. 'b = 3*a'
 - (c) Semantische analyse: het begrijpen van de individuele opdrachten: 3 vermenvuldigen met de inhoud van variabele a en het resultaat wegbergen in de variabele b.
 - (d) Code genereren

Wat is een voorvertaler? Welke functie heeft dit programma? Leg bondig de werking van de voorvertaler uit indien er alleen MACRO en MCREINDE directieven zijn. Geef ook aan hoe de voorvertaler aangepast moest worden indien we macrotaal uitbreiden met een MSPR – directief. Hoe moeten we de voorvertaler uitbreiden indien we het MEVA-directief toevoegen?

1. Voorvertaler ('pre-processor') bestaat erin de macriedefinities te verwijderen uit het programma, en alle macro – oproepen te vervangen door de overeenkomstige code. (= macro- expansie)
2. Als er enkel en allen marco en mcreinde directieven zijn wordt de taak van onze voorvertaler vereenvoudigd. De voorvertaler moet 3 taken uitvoeren.
 - (a) Alles wat niets met macro's te maken heeft , gaat de voorvertaler gewoon kopiëren
 - (b) Macrodefinities verwijderen uit het bronbestand en ze tijdelijk bijhouden (zodat een latere macro-oproep correct geëxpandeerd kan worden)
 - (c) Macro-oproepen expanderen (d.w.z. vervangen door de eventueel lichtjes gewijzigde macrodefinitie)

Vv start in de kopieer modus.

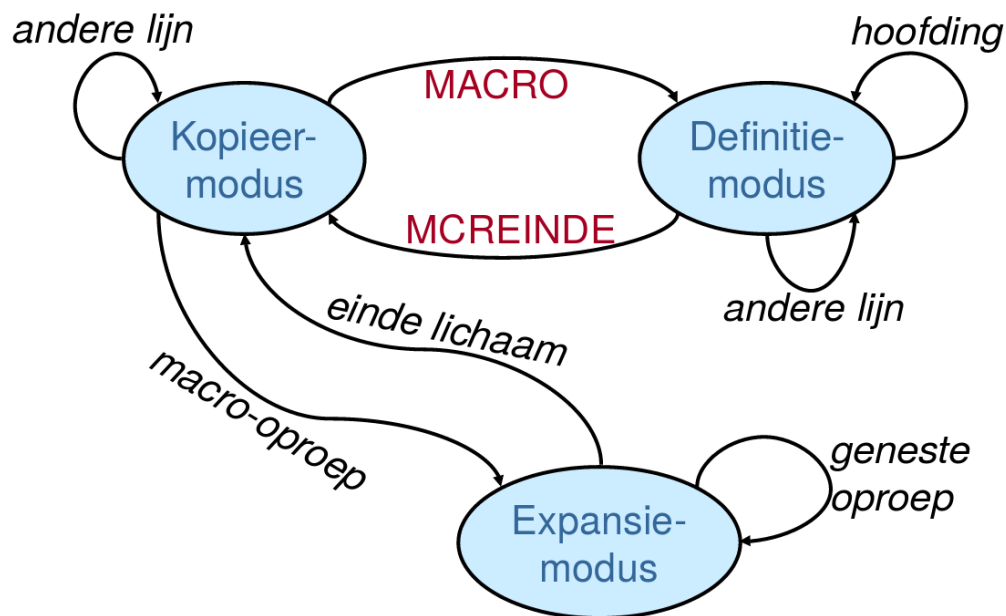
1. In de kopieermodus zal de vv de invoer lijn per lijn inlezen, en in de meeste gevallen de lijn gewoon kopiëren naar het uitvoerbestand, er zijn 3 mogelijk heden
2. De lijn bevat een MACRO-directief. Vv gaat door naar definitiemodus
3. De lijn kan van de vorm xxx arg ... zijn waarbij xxx de naam is van een eerder gedefinieerde marco. Vv zal verband bastleggen tussen de formele parameters van de marco en de actuele parameters. Dit wordt in de argumententabel bijgehouden. De vv gaat nu naar de expansiemodus.
4. In alle andere gevallen wordt de lijn letterlijk gekopieerd naar het uitvoerbestand. De voorvertaler blijft in dit geval in de kopieermodus
5. In de defenitiemodus wordt een nieuwe macro gedefinieerd. De eerste lijn die wordt gelezen in deze toestand = hoofding v.d. marco. Deze bepaalt de nam van de macro, alsook de namen van de formele parameters. De vv creëert een gegevensstructuur. In deze worden de hoofding en het lichaam opgeslagen. Voor de volgende lijnen zijn er 2 mogelijkheden
 - (a) De lijn bevat MCREINDE-directief. De vv gaat terug naar kopieer modus.
 - (b) In alle andere gevallen wordt de lijn ongewijzigd toegevoegd aan de gegevensstructuur waarin de macrodefinitie opgeslagen wordt. De vv blijft dan in defenitiemodus.
6. In de expansiemodus wordt een macro-oproep vervangen door het lichaam van die macro. De vv zal nu het lichaam opgeslagen in de gegevensstructuur lijn per lijn lezen ipv het invoerbestand te lezen. Er moeten 2 soorten substituties doorgevoerd worden.
 - (a) Elk symbool van de vorm <NAAM> wordt vervangen door de overeenkomstige actuele parameter uit de argumenten tabel. Indien de baan van de parameter niet voorkomt in de argumententabel, zal de vv een foutmelding afdrukken.
 - (b) Een symbool van de vorm \$ETIKET is een lokaal symbolisch adres en moet vervangen worden door een uniek globaal symbolisch adres. Het \$-teken in \$etiket wordt vervangen door .teller, waarbij teller de waarde van de interne macro-oproepteller is.

Na het uitvoeren van alle substituties, zal de vv nagaan wat er met de resulterende lijn moet gebeuren. Er zijn (weer) twee mogelijkheden

1. De lijn is een macro-oproep. In dit geval zal de vv tijdelijk de expansie van de huidige macro uitstellen en eerst de nieuwe macro-oproep expanderen. Dit houdt in dat de vv bijhoudt waar hij gekomen is bij de expansie van de huidige macro-oproep, daarna een nieuwe argumententabel opstelt en tenslotte begint met het lezen van het lichaam van de zojuist opgeroepen macro. Uiteraard blijft de vv in de expansiemodus. Na de expansie aan deze laatste macro-oproep zal hij verdergaan met de huidige expansie
2. In andere gevallen wordt de gewijzigde lijn naar het uitvoerbestand gekopieerd.

Indien het volledige lichaam gelezen is, wordt de argumententabel vernietigd en indien deze expansie geen genestelde expansie was, gaat de vv terug naar de kopieermodus.

1. In de kopieermodus worden deze extra macrodirectief verwerkt. De vv zoekt in de GET(globale etikettentabel) het lijnnummer van het etiket; het kopiëren gaat verder bij de lijn met dit lijnnummer. Indien het nog niet opgenomen is in de tabel, blijft de voorvertaler lijnen overslaan tot het etiket gevonden is (merk op dat de vv geen 2 stappen nodig heeft zoals de vertaler! Wel is het mogelijk dat door die sprongen de invoer meer dan een x gelezen zal worden.)
2. Als men het meva directief gaan toevoegen, wordt deze nagekeken in de expansie-modus. de expressie wordt geëvalueerd. Indien de variabele reeds voorkomt in de ALVT(argumenten-en-lokale-variabelentabel) , dan wordt de waarde daar aangepast; zo niet, indien de variabele voorkomt in de GVT, dan wordt de waarde daar aangepast; zo niet wordt de variabele en de waarde toegevoegd aan de ALVT. De macro-conditiecode variabele (MCC) krijgt een nieuwe waarde



Wat is een absolute lader, beschrijf de werking. Hoe wordt de lader in het geheugen geladen?

1. De absolute lader plaatst het vertaalde programma ongewijzigd in het geheugen.
2. De absolute lader zal de volgende acties ondernemen.
 - (a) De lengte inlezen
 - (b) De opeenvolgende getallen van het programma inlezen en wegbergen in opeenvolgende geheugenplaatsen (te beginnen bij adres 0000)
 - (c) De uitvoering van het ingelezen programma starten (naar adres 0000 springen)
3. Niet zeker over dit antwoord . De drama bronprogramma wordt opgezet naar een uitvoerbaar programma, dit uitvoerbaar programma bestaat uit 3 delen
 - (a) Hoofding. Hierin wordt de lengte en het laadadres vermeld. Deze info heeft de lader nodig om het programma op de juiste plaats en op een correcte wijze in het geheugen te brengen.

- (b) Programmacode (de machinecode en data). Deze wordt door de lader in het geheugen geladen.
- (c) Symbooltabel, die bedoeld is voor andere programma's bv het speurprogramma of de binder

Wat is/wat doet een relocerende lader? Hoe gaat een relocerende lader te werk? Welke informatie heeft de lader hiervoor nodig en van waar krijgt hij die?

1. Laders die de code kunnen aanpassen = relocerende lader. Dit betekent dat het uitvoerbaar programma extra informatie zal bevatten.
2. Als de lader een programma op een ander adres wilt laden dan moeten er sommige operanden worden aangepast. Dit doet die via een relocatietabel die zich achteraan in het uitvoerbaar programma bevindt. Met de tabel kan de lader de code op een correcte wijze aanpassen. De relocerende lader zal in stappen werken
 - (a) Het uitvoerbaar programma wordt elttelijk in het geheugen geladen, vanaf het laadadres
 - (b) De relocatiestap: de code zal aangepast worden m.b.v. de relocatietabel

Voordeel: de programmeur wordt ontlast van het opgeven van een laadadres.

Wat is dynamische relocatie?

De relocatie wordt nu tijdens de uitvoering van het programma doorgevoerd. Dit kan met een redelijk eenvoudige wijze gerealiseerd worden m.b.v. een basisregister. Dit register behoort tot het centraal verwerkingsorgaan (cvo = cpu (= central processing unit(=processor))) en wordt geïnitieerd door het besturingsprogramma met het beginadres van het geladen programma. Telkens het cvo de inhoud van een geheugenregister met een bepaald adres wil ophalen of wijzigen, zal de hardware automatisch de inhoud van het basisregister bijtellen bij dit adres. Het adres wordt dus dynamisch geresloceerd. Vb p 59 deel 3

Wat is de binder? Wat is zijn taak? Welke stappen voert hij daarbij uit? Welke informatie nodig en van wie krijgt hij die?

1. De binder is een koppelprogramma
2. De taak van een binder is om afzonderlijke vertaalde modules samen te voegen tot een groot uitvoerbaar programma, dat dan door de lader in het geheugen kan geladen worden.
3. De binder voert de volgende stappen uit
 - (a) Plaats voorzien voor de verschillende modules
 - (b) De machinecode van al de modules samenvoegen
 - (c) Het startadres voor de uitvoering bepalen op basis van de (enige geldige) start-aanduiding en de plaats waar die module in het uitvoerbaar bestand komt te staan
 - (d) De code reloceren . Alle overige modules worden immers opgeschoven

- (e) De externe referenties in rekening brengen . (=binden)
 - (f) In dien het resultaat zelf nog reloceerbaar moet zijn, moet de binder ook een relocatietabel opstellen
4. Het reloceren en het bijnden zijn de moeilijkste taken. Deze vereisen dat de binder respectievelijk beschikt over een allocatietabel en een globaal symbooltabel.in de allocatietabel wordt bijgehouden welke plaats voorzien is voor elke modulede globale symbooltabel geeft weer wat de waarde is van elk globaal symbolisch adres. Deze tabellen kunnen slechts opgesteld worden nadat alle modules onderzocht werden
 5. De binder werkt in twee stappen
 - (a) De allocatietabel en de globale symbooltabel wordt opgesteld .

Allocatie: de binder leest de modules actereenvolgens in en bepaalt het beginadres voor elke module. Deze beginadressen worden in de allocatietabel bijgehouden symbolen: de machinecode wordt in deze stap overgeslagen. De #symbool-sectie wordt gebruikt om de globale symbooltabel op te stellen: - lokale en externe symbolische adressen worden voorlopig genegeerd - globale etiketten worden samen met hun waarde, die zelf verhoogd wordt met het nieuwe beginadres van de module, in de globale symbooltabel toegevoegd
vb p 74-75

De code wordt samengevoegd, gerelocerd, gebonden en het start adres word bepaald en er wordt eventueel een relocatie tabel opgesteld.

Wat is een vertolker? Hoe werkt het? Welke voor- en nadelen zijn er? Wat zijn de toepassingen van een vertolker?

1. Een vertolker is een programma dat een ander programma inleest en het vertolkt
2. De eerste vertolkers werkten als volgt: de vertolker leest telkens een broninstructie in, vertaalt deze naar de overeenkomstige sequentie van machine-instructies en voert deze onmiddellijk uit. (niet af)

Wat is een speurprogramma (debugger)? Wat zijn breekpunten? Hoe kunnen deze mogelijk geïmplementeerd worden?

Syntaxisfouten kunnen opgespoord worden door de vertaler. Logische fouten niet. Hier-voor heb je een debugger nodig.

De programmeur kan zelf controleren hoe de computer de opdrachten uit voert met behulp van een aantal bevelen. Deze bevelen kunnen uitgebreid zijn. Maar er zijn een aantal bevelen die elke debugger heeft:

1. Breek punten plaatsen, verwijderen en een overzicht geven van de geplaatste breekpunten.
2. De inhoud van geheugenregister weergeven en overschrijven.
3. De inhoud van een accumulator afdrukken en wijzigen.
4. Instructies in stappen uitvoeren.
5. Het programma starten en herstarten.

Breek punten kunnen worden toegepast door een subroutine op te roepen die breekpunten behandelt. Het programma zal een tabel bijhouden waarin voor ieder breekpunt de oorspronkelijke instructies en het adres in staat.

Hoofdstuk4

Leg uit: PO.

Programma-onderbrekingen zijn erg belangrijk zowel om de efficiënte werking van de computer te kunnen realiseren als om de correcte werking van programma's en computers te garanderen. Enkele voorbeelden zijn overloop en oneindige lussen. Als er geen onderbreking is bij oneindige lussen zal de computer zelfs opnieuw moeten worden opgestart. De PO kunnen in vijf verschillende klassen worden opgedeeld:

1. **externe oorzaken:** een reset knop kan een voorbeeld zijn
2. **randapparatuur:** de besturingseenheden van de randapparaten kan de aandacht vragen van de processor
3. **machine fouten:** De ergste machine fout is het wegvallen van de spanning. Indien er een batterij is zullen alle gegevens worden opgeslaan.
4. **programmafouten:** deling door 0, oneindige lussen, ongeldige instructies, ...
5. **geprogrammeerde PO:** soms wil het programma onderbroken worden en kan dit niet met een SBR routine

Als er een PO wordt aangevraagd zal de CVO eerst de instructie afmaken en hierna kijken of hij op de PO ingaat. Dat doet die als volgt:

1. Staat er een globaal masker op ga dan door met de volgende instructie(**Globale maskers:** geen enkele PO wordt toegestaan)
2. Kijk of de prioriteit van de PO-vlaggen groter is dan het onderbrekingsniveau
3. Negeer oproepen die gemaskerd zijn
4. Als er geen vlaggen meer zijn ga dan door met de volgende bevelcyclus, anders
 - (a) Bewaar de huidige toestand van de processor
 - (b) Zet de PO-vlag af
 - (c) Pas de toestand van de processor aan door de onderbrekingsniveaus, de bevelenteller, CC, OVI en SOI aan te passen.

Wat is directe geheugentoegang:? Waarom wordt dit toegepast? Welke voordelen en nadelen (o.a. cycle stealing)?

Directe geheugentoegang wordt toegepast bij het besturen van randapparaten. Bij DGT zal de bestuurder van de randapparaten de gegevens rechtstreeks in het geheugen plaatsen. De processor met slechts een opdracht geven aan de bestuurder, deze zal dan zelf de opdracht afmaken zonder hulp van de processor. Op het einde van de opdracht zal de bestuurder een PO aanvragen. Dit is veel minder in het geval dat men in en uitvoer zou regelen met PO bevelen. Dankzij de DGT zal de processor zich met iets anders kunnen bezig houden. De tijd winst hangt af de rotatie wachttijd en het debiet. Maar niet enkel de schijfbestuurder maar ook de processor heeft toegang tot het geheugen nodig.(voor bevelen op te halen en gegevens op te slaan) aangezien er maar 1 bus is en die maar door 1 apparaat tegelijkertijd gebruikt zal kunnen worden zullen de processor en schijfbestuurder soms extra lang moeten wachten. We zeggen dat de schijfbestuurder geheugencycli steelt van de processor.

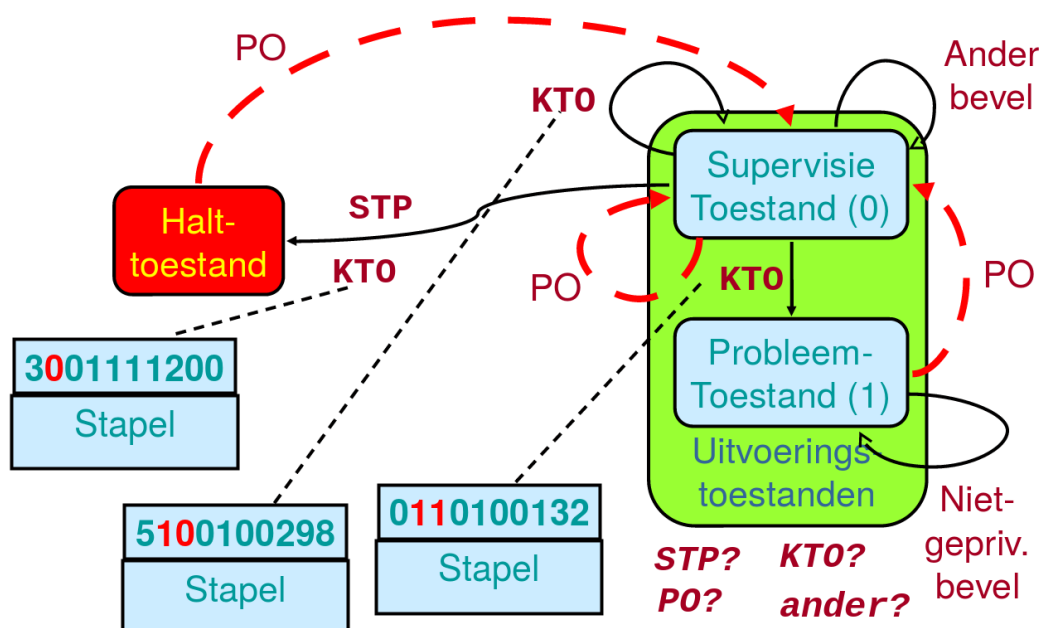
Dit kan men beperken tot een minimum door een apart gegevenspad tussen de bestuurder en het geheugen plaatsen.

Welke processortoestanden zijn er aanwezig op de DRAMA-machine? Hoe gaan we van de ene toestand in de andere over en wat zijn de redenen daarvoor (maak een diagramma).

1. Uitvoeringstoestand: uitvoeren van de bevelencyclus
 - (a) Supervisie toestand: Uitvoering van de het besturingsprogramma
 - (b) Probleemtoestand: Uitvoering van een gewoon programma
2. Halttoestand: De processor voert geen bevelen meer uit. Hij reageert wel nog op PO-vlaggen

Er vallen drie dingen op aan het schema:

1. Iedere PO brengt de processor naar supervisie toestand
2. Bij terug keer uit een PO gaat de processor terug naar zijn vorige toestand
3. Het STP bevel brengt de processor in de halt toestand



Wat zijn geprivilegieerde bevelen? Waarom zijn bepaalde bevelen geprivilegieerd? Hoe moet de processor aangepast worden om zulke bevelen te kunnen implementeren? Waarom is het zinloos om geprivilegieerde bevelen te implementeren als er geen geheugenbescherming voorzien is?

Geprivilegieerde bevelen kunnen enkel worden uitgevoerd als de processor zich in de supervisietoestand bevindt. Dus enkel bevelen van het BP. In drama zijn dit de:

1. in- en uitvoerbevelen: INV en UTV

2. Programma-onderbrekingsbevelen: MKL, MKH, TSM en TSO
3. Bevelen die de processortoestand wijzigen: KTO en STP

Deze bevelen zijn geprivilegieerd omdat ze de correcte werking van de computer kunnen ondermijnen. De processor zal geprivilegieerde bevelen uitvoeren in supervisie toestand en een PO9 valg plaatsen in probleemtoestand.

Wat is een Supervisie-oproep? Is hij essentieel voor de PC? Geef een schema met wat er gebeurt en duidt aan wat er gedaan wordt door de software en wat door de hardware. Kan je enkele voorbeelden geven?

Omdat een programma dus geen beroep kan doen op de randapparatuur door de geprivilegieerde bevelen is er een oplossing: Supervisie-oproep. Het programma gaat beroep doen op het BP om deze taken uit te voeren. OND X is de DRAMA equivalent van een supervisie oproep. X geeft aan van welke dienst van de BP dat het programma gebruik wil maken. Voor het schema zie pagina 174 met uitleg op 172,173 en 175.

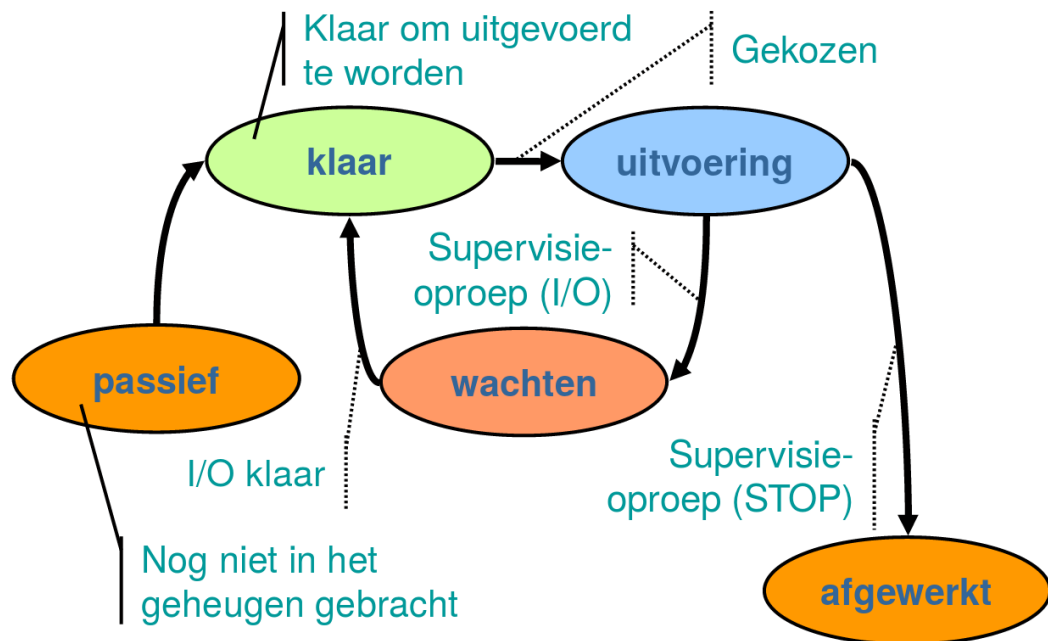
Wat is multiprogramatie? Geef een duidelijk voorbeeld van multiprogramatie. Wat is het verband met een time-sharing systeem?

Omdat een processor vaak staat te wachten op de gegevens van randapparaten kan de processor m.b.v. PO een ander opdracht uitvoeren. Een voorbeeld staat op pagina 179-180. Bij ieder programma zal de processor moeten bepalen hoelang het aan een programma werkt. Ook bij een time-sharing systeem zal de processor dit moeten doen. Ieder gebruiken wil namelijk evenveel tijd krijgen van een processor. Het zou ongeoorloofd zijn als gebruiker 1 de tijd van de processor krijgt wanneer hij wil en gebruiker 2 moet wachten tot gebruiker 1 de processor niet nodig heeft. De doorvoer of throughput is het aantal afgewerkte programma's per tijdseenheid. De computer kan sneller werken wat hij kan meer programma's per tijdseenheid afwerken maar toch zal de gebruiker merken dat de computer zwaar belast is en een programma trager gaat werken. Dit komt omdat de processor met verschillende programma's bezig is en dus minder bezig is met een programma.

Welke taken biedt een besturingssysteem aan? Hoe kan de gebruiker hier een beroep op doen?

De OS moet een aantal taken voltooien:

1. Geheugenbeheer: Het geheugen van de computer is beperkt, als er dan ook nog eens verschillende programma's tegelijkertijd in het geheugen zitten moet dit goed geregeld worden. Het BP zal een tabel bijhouden waar welk programma in het geheugen zit
2. Processorbeheer: De processor zal verschillende bestanden moeten uitvoeren. Het zal prioriteiten moeten stellen. Programma's kunnen zich in bepaalde toestanden vinden. Dit wordt bijgehouden in een tabel.



De processor kan dus enkel kiezen tussen programma's die klaar zijn. Bij timesharing zal de processor de tijd moeten verdelen onder verschillende gebruikers. Programma's die af zijn moeten een STP-bevel aanvragen via supervisie om het programma te beëindigen.

1. beheer van in- en uitvoer: De computer zal een tabel opstellen van alle randapparaten, hun status en al hun opdrachten.
2. Bestandenbeheer: De computer zal zelf onthouden waar bestanden staan a.d.h.v. een tabel.
3. informatie beheer: Het BP bevat zelf ook heel wat informatie zoals de klok. Deze informatie kan worden opgevraagd door programma's. een klok kan gerealiseerd worden met een kristal oscillator of op de frequentie van het net.

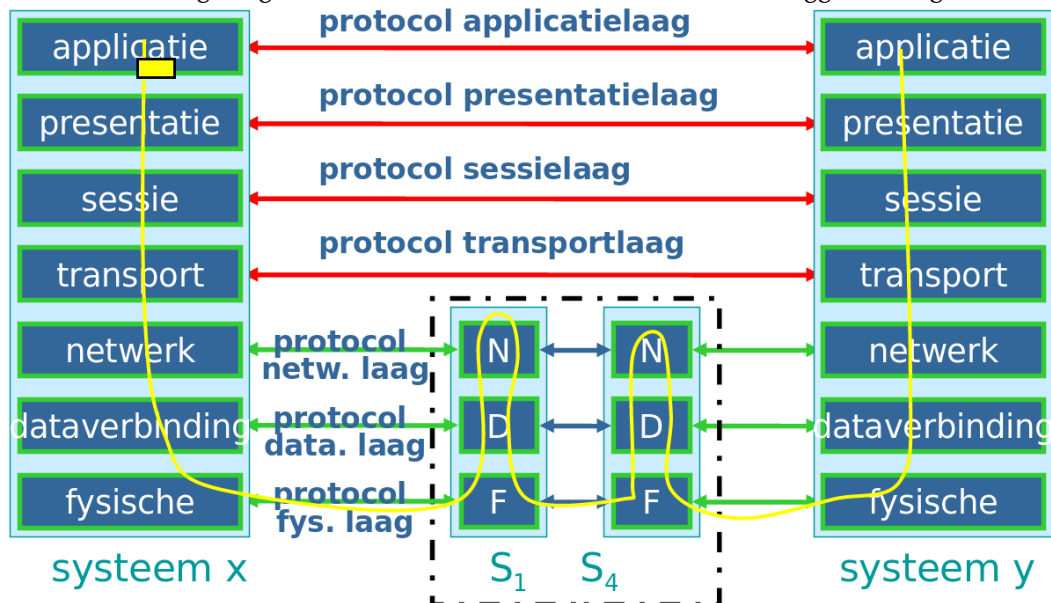
Hoofdstuk5

Wat is het verschil tussen lijnschakelen, boodschapschakelen en pakketschakelen?

Lijnschakelen: bij lijnschakelen wordt er een verbinding opgestart over één lijn tussen de verzender en de ontvanger. Er kan informatie worden uitgewisseld over de lijn tot de verbinding verbroken wordt. **Boodschapschakelen:** alle informatie die moet worden verzonden wordt in een boodschap gestoken. Deze wordt verzonden over het net van schakelaars. Er treedt een grote vertraging op want ieder schakelaar gaat de informatie inlezen, opslaan en dan pas doorsturen. De vertraging kan nog groter worden als er bestanden in de wachtrij van een schakelaar staan. **Pakketschakelen:** in plaats van een hele boodschap te verzenden gaan we de boodschap opdelen in kleinere pakketten en deze te verzenden. De pakketten zullen minder vertraging ondervinden. De pakketten kunnen wel in verschillende volgorde aankomen, de ontvanger zal ze daarom moeten sorteren.

Wat is een protocol? Wat wordt bedoelt met "netwerken hebben een gelaagde structuur". Schets de 7 lagen van het OSI-model en bespreek kort de functie van iedere laag. **Welke weg legt een boodschap af van computer A naar computer B over 2 schakelaars? hoeveel lagen heeft het Internet en waarom zijn er zoveel lagen nodig?**

Een protocol is een stel van regels die twee entiteiten volgen bij een onderlinge samenwerking. Bij netwerken zullen deze entiteiten samenwerken met een gelaagde structuur van protocollen. Een laag zal gebruik maken van de diensten van de onderliggende laag.



Fysische laag: versturen van de bits over de kabel.

1. Aard van de gebruikte kabel
2. Kenmerken van de connectoren
3. Voorstelling van bits op de kabel

Dataverbinding laag: creatie van een foutvrij communicatie kanaal.

1. Herkennen van pakketten

2. Detectie van fouten
3. Controle van de pakketenstroom (flow control)

Netwerk laag: verkeer van pakketten doorheen één of meerdere netwerken. 2 soorten diensten:

1. Virtuele verbinding, verzekerd dat alle pakketten aankomen in de juist volgorde
2. Datagramdiensten, zal zijn best doen om ieder pakket af te leveren, de volgorde maakt geen belang.

Aandachtspunten:

1. Routebepaling
2. Kostenberekening
3. Onderlinge verbinding van netwerken

Transport laag: betrouwbare communicatiekanalen

1. Efficiënt gebruik van het netwerk
2. Eventueel opdelen in pakketten en terug samenvoegen
3. Controle van de boodschappenstroom

Sessie laag: organisatie van de samenwerking

1. Organisatie van de dialoog (zender/ontvanger)
2. Plaatsen van synchronisatie punten.

Presentatie laag: behoud van de betekenis van de uitgewisselde informatie

1. Keuze van de voorstelling van gegevens
2. Encryptie
3. Compressie van data

Applicatie laag: hoogste laag met protocollen voor applicaties van de gebruiker, standaard applicaties, deelproblemen, Het internet heeft maar 5 lagen, dit komt omdat er nooit een formeel model gedefinieerd of beschreven is, het is de loop de jaren gegroeid. De overeenkomsten zijn redelijk groot tussen het OSI-model en het internet model. Het Internet model heeft geen presentatie laag of sessie laag. Er worden bestaande standaarden voor gebruikt.

Wat is de functie van de dataverbinding laag? Hoe worden de taken gerealiseerd? (Gegeven: code voor zender en ontvanger. Worden er met dit protocol fouten opgevangen? Welke wel en welke niet? Toon aan met een tekening. Geef in woorden de aanpassingen ter verbetering.) zie hiervoor pagina 27-34 in het boek.

De dataverbinding laag moet een betrouwbaar kanaal realiseren. Het heeft vier taken:

1. **Afbakenen van pakketten:** de ontvanger moet het begin en einde van ieder pakket herkennen. We kunnen ieder pakket laten starten en eindigen met eenzelfde bitpatroon. Maar dat patroon mag niet in de data voorkomen. We gebruiken 01111110. Als er 5 keer een 1 achter elkaar staat in de data gaat men na de vijfde 1 een 0 plaatsen. De ontvanger gaat deze weglaten.
2. **Detectie van fouten:** er moet een controle zijn op fouten. Een voorbeeld is de pariteit bit. Dit is enkele een controle op een fout van 1 bit. In de werkelijkheid worden er complexere systemen toegepast.
3. **Stroombeheersing:** Een snellere zender mag een tragere ontvanger niet overspoelen (flow control) de ontvanger geeft een expliciete of impliciete toelating.
4. **Pakket lay-out: vast patroon:**
 - (a) Beginpatroon(01111110)
 - (b) Hoofding
 - (c) Data
 - (d) Controle bits
 - (e) Eindpatroon (01111110)

Wat is het 1-bit vensterprotocol (beschrijven in woorden)? Welke aanpassingen zijn nodig voor satellietverbindingen?

Het 1-bit data protocol kan enkel worden toegepast als er dataverkeer in 2 richtingen is toegestaan. Er wordt een pakket verzonden naar de ontvanger, deze stuurt een boodschap terug zodat het volgende pakket verzonden kan worden. In de boodschap staat een volgnummer die aangeeft welk het volgende pakket is. De pakketten worden met 1-bit genummerd (0 of 1) afwisselend moet er een pakket met 0 en 1 aankomen. De zender stuurt enkel een volgend pakket als er op tijd een bevestiging aankomt. Anders wordt het vorige pakket opnieuw verzonden. Bij een duur communicatiekanaal is het onaanvaardbaar dat het maar een kleine tijd nuttig gebruikt zal worden. Daarom zal er toelating zijn om meerdere pakketten te verzenden. Door met meerdere bits te werken voor pakketnummers kan er gecontroleerd worden welk pakket aankomt en welk opnieuw verzonden moet worden.

Wat is ethernet? Hoe ziet het er fysisch uit? Beschrijf de werking ervan (De regels die gebruikt worden om ethernet te realiseren). Bij ethernet moeten alle pakketten een minimale grootte hebben, waarom is dit zo? Welke structuur heeft ethernet. Beschrijf het protocol. Geef enkele voor en nadelen.

Ethernet is een merknaam, het is een busnetwerk met als fysische laag een aantal opties, meestal worden er gevlochten koperparen gebruikt. Ieder systeem dat wordt aangesloten heeft een interface bord. De kabel verbindt de verschillende computers en de uiteinden van de kabel worden gepast afgesloten. Als een computer een pakket uitstuurt wordt dit in beide richtingen uitgestuurd. Als 2 computers tegelijkertijd sturen zullen de signalen interfereren en doet er zich een botsing voor. De botsing kan pas worden opgemerkt als de computer zelf een verstoord signaal ontvangt. Dit gaat over micro seconden want de snelheid over de kabel is 200000km/s en lokale netwerken zijn niet langer dan 3KM. Zonder regels 18% van het netwerk wordt nuttig gebruikt

1. Regel 1: een computer mag slechts starten met het uitzenden van een pakket, indien het computersysteem geen signaal ontvangt. (36% wordt nuttig gebruikt)

2. Regel 2: tijdens het uitzenden van een pakket blijft een zendend computersysteem luisteren naar het signaal op de kabel. Het uitzenden stopt zodra er een botsing wordt vastgesteld. Dit systeem zendt gedurende korte tijd een stoor signaal uit.
3. Regel 3: na een botsing wachten de betrokken systemen een willekeurige tijd vóórdat het pakket opnieuw uitzenden. Een speciaal algoritme zorgt ervoor dat bij herhaalde botsingen de willekeurige tijd steeds groter wordt. Na 16 mislukte pogingen wordt er een foutmelding verstuurd.

Dit principe wordt beschreven door **CSMA/CD**:

1. **MA (multiple access)**: iedere computer heeft gelijktijdige toegang tot het netwerk.
2. **CS (Carrier sense)**: computers kunnen detecteren of de kabel in gebruik is.
3. **CD (Collision detect)**: computers kunnen botsingen detecteren.

Voordelen:

1. Bij weinig gebruik van het netwerk is het uitzenden van pakketten foutloos.
2. Bij veel gebruik wordt er nog steeds een 90% van de beschikbare transmissiecapaciteit gebruikt.

Nadelen:

1. Omdat alle tijden willekeurig worden gekozen kan het zijn dat 1 computer steeds benadeeld wordt. In realtime systemen is dit onaanvaardbaar
2. Er is een minimale lengte van pakketten nodig. Anders kan een computer niet detecteren dat er storing is. (64 Kb)

Wat is de functie van de netwerk laag? Welke diensten biedt ze aan en welke taken neemt ze op zich? Welke informatie heeft ze nodig om deze taken te realiseren?

De netwerk laag moet het mogelijk maken om pakketten te kunnen verzenden. Dit gebeurt tussen een computer en een schakelaar of onderling tussen schakelaars en dus niet langer meer tussen twee computers.

Twee types van diensten

1. Virtuele verbindingen
2. Datagramdiensten
 - (a) Geen garantie van aflevering.
 - (b) Pakketten moeten niet in volgorde aankomen.

Bij datagram diensten volstaat het dat een pakket kan verstuurd worden, het zal een hoofding aan een pakket toevoegen waarin het netwerkadres van de bestemming staat, het netwerkadres van de zender en het protocol van de transport laag. Er is ook routing nodig zodat de pakketten over schakelaars verzonden kunnen worden. Ieder schakelaar zal een routingstabel hebben.

Besprek de transport laag. Wat is TCP? Leg het verloop van een verbinding via TCP uit. Wat is de functie van de transport laag?

IP-adres en poortnummer vormen een paar. Dit paar identificeert dus een computersysteem en het proces dat op het computer systeem moet worden uitgevoerd. poort nummer 80 is gereserveerd voor de WWW server.

Virtuele verbinding tussen twee computersystemen:

1. Opzetten van verbinding
2. Gebruik van verbinding
 - (a) Pakketten moeten in de juiste volgorde worden gezet.
 - (b) 1-bit vensterprotocol zou snelheid te veel beperken.
3. Verbreken van de verbinding

Wat zijn domeinnamen? Hoe worden ze beheerd, en leg dit uit a.d.h.v. het voorbeeld www.bedrijf.com. Kan je een scenario bedenken waarbij dit proces fout afloopt? Wat is hun structuur? Hoe zet je domeinnamen om naar een IP? Pas toe voor www.dexia.fr Hoe werken domeinnamen.

En IP-adres is niet geschikt als een webadres. Daarom maken we gebruik van symbolische namen die iedereen kan onthouden.

1. **2-letter land codes**(eu, nl ,be)
2. **Generieke namen**
 - (a) **.com**: commerciële organisaties
 - (b) **.org**: niet commerciële organisaties
 - (c) **.net**: bedrijven die netwerkdiensten aanbieden
 - (d) **.int**: internationale organisaties
 - (e) Recent uitgebreid en in de toekomst zullen er nog meer komen
3. **Organisaties binnen de Verenigde staten(deze zouden beter onder .vs staan maar we kunnen geen drastische wijzigingen meer doorvoeren)**
 - (a) **.edu**: universiteiten
 - (b) **.gov**: de overheid
 - (c) **.mil**: het leger

Voor de omzettingen van domeinnamen naar IP-adressen worden databanken opgericht. Eén databank voor alle computersystem is onmogelijk. Als deze wegvalt is het internet onbruikbaar. Er worden verschillende naambeheerders(name server) gebruikt. voor ieder domein is er een aparte naambeheerder die alle IP-adressen in het domein kent. De root-naambeheerder bevat alle adressen van naambeheerders.

Een voorbeeld: ww.cs.vu.nl

1. De naam wordt verzonden naar de Root-naambeheerder die .nl terugstuurt
2. De naam wordt naar de .nl naambeheerder verstuurt. Die stuurt .vu.nl terug
3. De naam wordt door gegeven aan de naambeheerder van .vu.nl die .cs.vu.nl terugstuurt

4. De naam wordt tenslotte doorgestuurd naar cs.vu.nl naambeheerder die het adres van www.cs.vu.nl terugstuurt.(130.37.20.20)

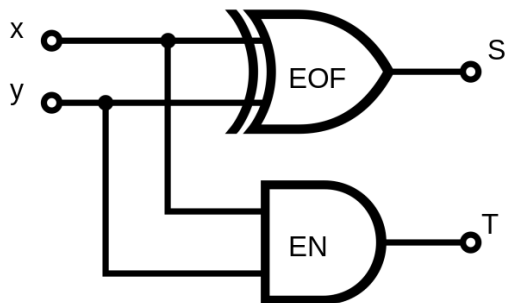
Dit principe wordt versnelt door lokale mapping. Gedurende een tijd mag een lokale naambeheerder de informatie opslaan waardoor er sneller doorverwezen kan worden. Dit systeem is kwetsbaar, als de root-naambeheerder uitvalt liggen alle onderliggende domeinen uit. Daarom zijn er verschillende naambeheerders met dezelfde informatie.

Hoofdstuk6

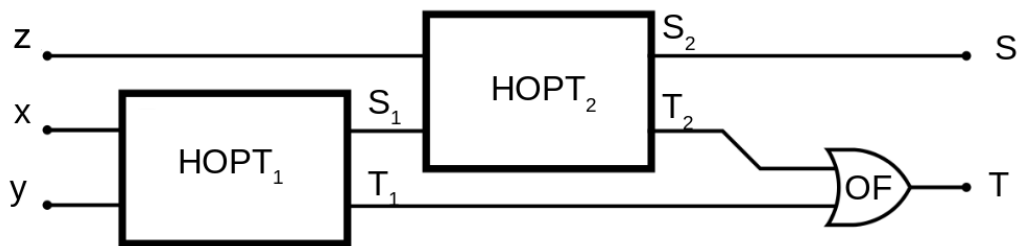
Vragen over de 2-complement voorstelling en de bewegende komma voorstelling worden bijna ieder jaar gevraagd.

Teken een halve en hele opteller. Toon hoe met deze basiscomponenten gehele binaire getallen in 2-complement opgeteld kunnen worden.

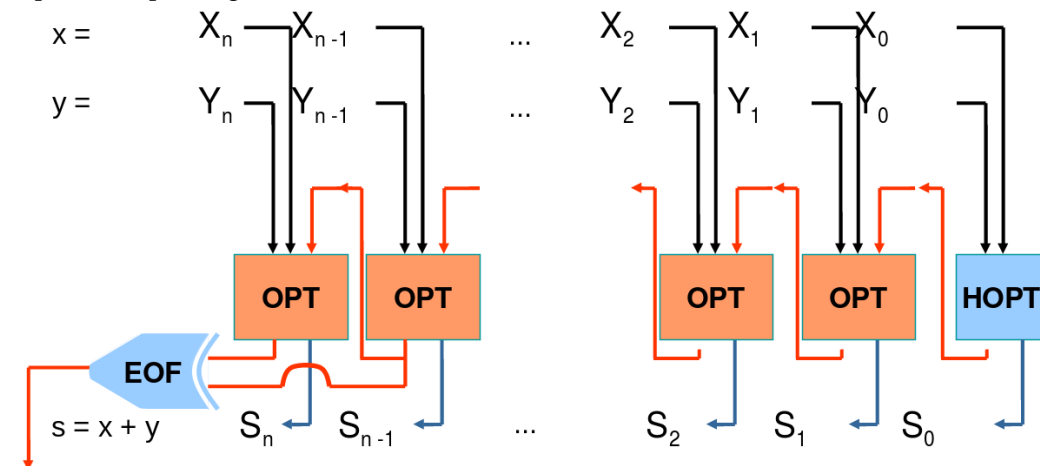
Halve opteller:



Opteller:



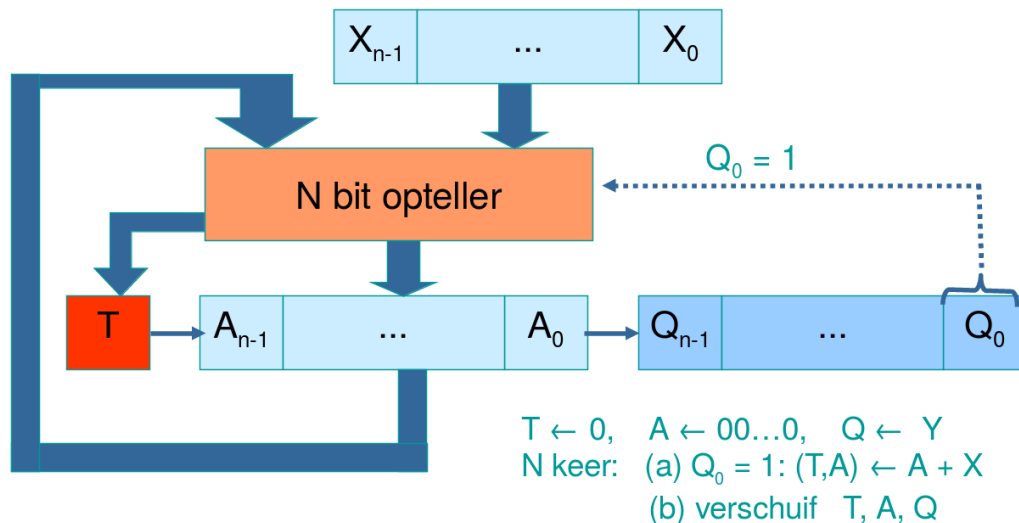
De parallel-optelling:



Overloop (indien 1)

Leg uit: Schakelschema van de vermenigvuldiging. Hoeveel stappen maakt deze? Waar bevindt de uitkomst zich? Wat gebeurt er tijdens elke stap?

$$X \times Y$$



→ = verschuif naar rechts

I	Bewerking	T	A	Q
0	initialiseer	0	00000	<u>01101</u>
1	$A \leftarrow A + X$	0	11011	<u>11011</u>
	verschuif	0	01101	<u>10110</u>
2	verschuif	0	00110	<u>11011</u>
	$A \leftarrow A + X$	1	00001	<u>11011</u>
3	verschuif	0	10000	<u>11101</u>
	$A \leftarrow A + X$	1	01011	<u>11101</u>
4	verschuif	0	10101	<u>11110</u>
	verschuif	0	01010	<u>11111</u>

Bespreek de voorstelling van de gehele getallen in een computer. Geef voor elke methode de voorstelling van -143 in een getal van 10 bits. Bespreek de methoden en vergelijk ze onderling. Zijn bewerkingen met de verschillende voorstellingswijzen even moeilijk? Wat is een toepassing voor de verhoogde komma-voorstelling? Met welke voorstellingsmethode kan je het makkelijkste rekenen?

Voorstelling met voorteken: De meest linkse bit is de bit van het teken. Voor een 0 is dit een + en voor een 1 is dit -.

1. Het getallenbereik is symmetrisch, er zijn evenveel positieve als negatieve getallen.
2. 0 kan men als +0 en -0 voorstellen.

3. Veranderen van teken is simpel. Voer een niet-bewerking op de eerste bit uit.
4. Je kan met een of-bewerking op de eerste bit simpel nagaan of 2 getallen hetzelfde teken hebben.

2-complement voorstelling:

Een positief getal wordt voorgesteld door hun binaire voorstelling en negatieve getallen door het complement van hun absolute waarde t.o.v. 2^n . dus $2^{(n-1)}$ is kleiner of gelijk aan x is kleiner dan 2^n . Dit werkt ook voor ieder r-talig stelsel. De DRAMA machine maakt gebruik van een 10-complementvoorstelling. Het is echter geen strikt r-talig talstelsel meer. Maar omdat de plaatsen van de cijfers toch nog overeenkomen met de machten is het een semi-polyadisch of half r-talig talstelsel.

1. Het bereik is asymmetrisch
2. 0 heeft maar één voorstelling
3. Het teken kan bepaald worden met de eerste bit
4. Het veranderen van teken is niet complex al lijkt het dit wel.

Verhoogde voorstelling: Alle getallen worden verhoogd met een bepaald getal N. meestal neemt men voor N $2^{(n-1)}$.

$$-2^{(n-1)} = 000\dots 000\ 0 = 100\dots 000\ 2^{(n-1)} - 1 = 111\dots 111$$

1. Asymmetrisch
2. Eén voorstelling voor 0. Bestaat niet enkel uit 0 bits.
3. Teken is simpel te bepalen a.d.h.v. de eerste bit.
4. Het veranderen van teken is simpel.

1-complement voorstelling: Een negatief getal is het complement t.o.v. $2^{(n-1)}$

1. Symmetrisch
2. Twee voorstellingen voor 0. (+0 en -0)
3. Het teken veranderen is simpeler dan bij 2-complement

-143: met 10 bits: $143 = 10010011$

Voorstelling met voorteken: 1010010011 ($1000000000 = -0$, $0000000000 = +0$)

2-complement voorstelling: 1101101101 ($1111111111 = -1$)

Verhoogde voorstelling: $0 = 1000000000$ en $147 = 10100010011$, $0-147 = -147 = 0010010011$

1-complement voorstelling: 1101101100 ($1111111111 = -0$)

2-complementsvorm, 10-bit: Zet -77 en 47 om, maak de bewerking -77 - 47, wat zijn de toegepaste rekenregels? Wat is het grootste positieve en het kleinste negatieve getal dat je dan kan voorstellen?

Grootst mogelijk getal: $2^{(n-1)} - 1 \Rightarrow 511$

Kleinst mogelijk getal: $-2^{(n-1)} \Rightarrow -512$

$ \begin{array}{rcl} 47 & = & 0000101111 \\ 77 & = & 0001001111 \\ \\ -47 & & 1111010001 \\ -77 & = & 1110110001 \\ \text{(omdraaien en 1 optellen)} & & \\ \\ -77-47 = & & \\ & & 1110110001 \\ + & & 1111010001 \\ \hline & & \cancel{X}1110000010 \\ = & & 1110000010 \end{array} $	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px solid black; padding-right: 5px;"> $\begin{array}{r} 47 \\ 23 \\ 11 \\ 5 \\ 2 \\ 1 \end{array}$ </td> <td style="width: 50%; padding-left: 5px;"> $\begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{array}$ </td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"> $\begin{array}{r} 77 \\ 38 \\ 13 \\ 9 \\ 4 \\ 2 \\ 1 \end{array}$ </td> <td style="padding-left: 5px;"> $\begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{array}$ </td> </tr> </table>	$ \begin{array}{r} 47 \\ 23 \\ 11 \\ 5 \\ 2 \\ 1 \end{array} $	$ \begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{array} $	$ \begin{array}{r} 77 \\ 38 \\ 13 \\ 9 \\ 4 \\ 2 \\ 1 \end{array} $	$ \begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{array} $
$ \begin{array}{r} 47 \\ 23 \\ 11 \\ 5 \\ 2 \\ 1 \end{array} $	$ \begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{array} $				
$ \begin{array}{r} 77 \\ 38 \\ 13 \\ 9 \\ 4 \\ 2 \\ 1 \end{array} $	$ \begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{array} $				

Gehele negatieve getallen, welke voorstelling hebben wij hiervoor gezien, bespreek ze en zet -177 om naar die voorstellingen (10 bits voorstelling). Leg ook telkens je omzetting algoritme uit.

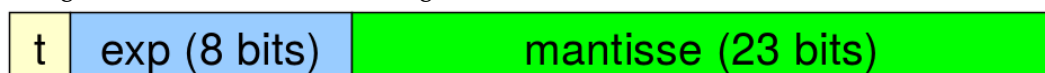
$$177 = 0010110001$$

- 2-complement: 0 en 1 omdraaien en optellen met 1
:1101001111
- 1-complement: 0 en 1 omdraaien
:1101001110
- voorteken: eerste 0 vervangen door een 1
:1010110001
- verhoogde voorstelling: inverteren en 1 optellen
:0010110001 $\rightarrow$ 1101001110 (inverteren)
:1101001111 (1 optellen)

177	1
88	0
44	0
22	0
11	1
5	1
2	1
1	1
	0
	0

Bewegende kommavoorstelling in DRAMA, basis 1000, 10 decimalen per getal: 1 tekenbit (0=Pos, 1=Neg), 2 decimalen voor de exponent (in +50 verhoogde voorstelling), 7 decimalen voor de mantisse (komma staat voor het eerste cijfer). Wat is het bereik van deze voorstelling, duidt aan op een getallen as. kunnen alle getallen in dit bereik voorgesteld worden? Bereken de som van volgende getallen volgend bovenstaande voorstelling. Voor de nodige stappen uit en verklaar wat je doet. Wat is de getalwaarde van het resultaat. $0520012000 + 1519999999$. Wat verandert er wanneer de basis groter of kleiner wordt?

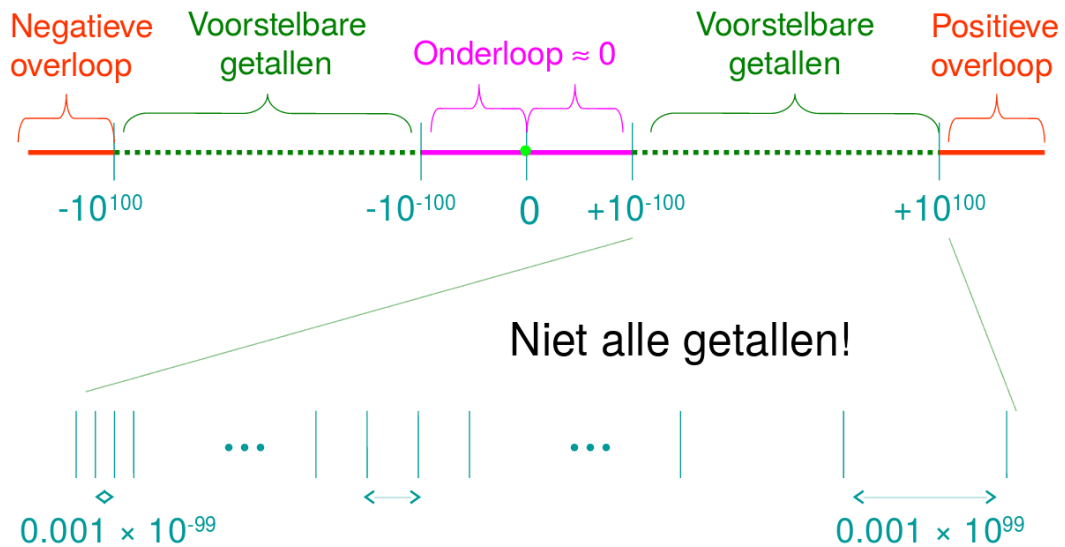
Niet alle getallen kunnen worden voorgesteld. Het grootste getal is 0999999999 en het kleinste positieve getal is 0500000001. Hiertussen kunnen alle waarden worden afgerond zodat ze kunnen worden voorgesteld. De nauwkeurigheid is niet altijd even correct. Als de basis groter wordt zal de nauwkeurigheid dalen maar het bereik toenemen.



- t: tekenbit van de mantisse
- exp: exponent van de basis
- mantisse: $\frac{1}{100} \leq |mantisse| < 1$

$$\begin{aligned}
 0520012000 &= 0.0012000 \times 1000^2 \\
 1519999999 &= -0.9999999 \times 1000^1 \\
 &\approx -0.0009999 \times 1000^2 \\
 &= 1520009999
 \end{aligned}$$

$$\begin{aligned}
 0520012000 + 1520009999 &= \\
 0012000 - 0009999 &= 0002001 \\
 = 0520002001 &= 0512001000 \\
 = 2001000 \times 1000^1
 \end{aligned}$$



Relatieve afstand $\approx$ constant

Op welke manier worden niet numerieke gegevens voorgesteld in de computer. Welke drie voorstellingswijzen hebben wij gezien. Geef enkele kenmerken van deze voorstellingswijzen.

EBCDIC (Extended Binary Coded Decimal Interchange Code)

1. IBM $\Rightarrow$ uitbereiding op BCD-code
2. 8 Bit voorstelling
3. Tabel eerste 4 cijfers een kolom en laatste 4 cijfers een rij
4. 16^2 voorstellingen
5. Romeins alfabet, cijfers en symbolen

ASCII (American Standards Code for Information Interchange)

1. 7 bits (eerste drie bits voor de rij en laatste 4 voor de kolom)
2. 128 verschillende tekens
3. Controle tekens (0x00 t.e.m 0x1F)

Unicode: ASCII is niet genoeg, geen accenten of chinese tekens.

1. Alle symbolen
2. JAVA ondersteunt UNICODE waardoor populariteit toeneemt
3. 16 bits
4. 65.536 verschillende tekens
5. Eerste 256 waarde komen overeen met ASCII zodat omzetting gemakkelijk is
6. Uitgebreid tot 1.114.112 symbolen waarvan 100.000 gebruikt zijn