

Samenvatting

Structuur en organisatie van computersystemen (Universiteit Leuven)

SOCS : samengevat door mij (DUH). Tis nie 100% volledig, ma 'tzou genoeg moete zijn voor't examé... enjoy, GL, HF, don't die... yet...

Thank you list :

- Jeroen Goossens
- "Tom"
- Daan Daele (een aantal schrijf- en inhoudsfouten verbeterd + verduidelijkingen)

Hoofdstuk 2: Computergeheugens en randapparatuur

Deel 1: Electronica en schakelalgebra

Basis (logische) schakelingen:

- ⤴ EN
- ⤴ OF
- ⤴ NIET

2.2.1 Enkele begrippen uit de electriciteit

elektrisch potentiaalverschil = elektrische spanning (E) eenheid V

Stroomsterkte (I): hoeveel elektrische ladingen per tijdseenheid zich door een doorsnede van het medium verplaatsten. Eenheid Ampère

Stroom = verplaatsing elektronen. Elektrische stroom is tegengesteld aan de elektronen stroom.

Bij elektrische spanning gaan de elektronen vloeien.

Stroomsterkte is evenredig met de spanning en omgekeerd evenredig met de weerstand.

$$I = E / R \text{ (wet van ohm)}$$

- R weerstand (eenheid ohm).

Spanningsbron: stroom vloeit van + naar - (dus de elektronen van - naar +)

2.2.2 Transistoren

Drie klassen materialen:

- ⤴ Geleiders
- ⤴ Niet geleiders (isolatoren == diëlektrica)
- ⤴ Half-geleiders: bv silicium, germanium
 - ⤴ onder normale omstandigheden gedragen ze zich als niet-geleiders.
 - ⤴ Door vreemde atomen in de kristalstructuur in te brengen geleid het wel
 - ⤴ “doperen van de halfgeleider”

p-type : plaats voor extra elektronen

n-type : elektronen op overschot

combinaties:

p+n = Diode

P+n+p of n+p+n = transistor

p-type naar het n-type, niet omgekeerd (bij p is plaats, elektronen vloeien naar p, dus stroom van p naar n). Diodes worden niet meer gebruikt.

Transistor (samenvoeging gedopeerde halfgeleiders):

- 3 externe connectoren

- ⤴ De collector (C)
- ⤴ De emitter (E)
- ⤴ De basis(B).
- ⤴ zeer snelle binaire schakelaar.

NEN poort : twee transistoren in serie

NOF poort : twee transistoren in parallel

Bipolaire transistoren en MOS-transistoren

Bipolaire transistor : beide polariteiten spelen een rol bij de werking

Twee voornaamste bipolaire technologieën:

TTL (transistor-transistor logic)

ECL (emitter coupled logic) -> sneller, maar meer energie en plaats nodig op chip

MOS (Metal Oxide Semiconductor)

3 varianten:

- ⤴ PMOS: transistor sluit als de basis geaard wordt
- ⤴ NMOS: transistor sluit als er aan basis een positieve spanning gelegd wordt
- ⤴ CMOS: Complementaire MOS: combi van PMOS en NMOS die in stabiele toestand geen energie verbruikt.

VERSCHIL TUSSEN BIPOLAIRE EN MOS TRANSISTOREN:

MOS-transistoren

pros:

- minder plaats, minder energie, minder storend op naburige transistoren, eenvoudig en goedkoop te maken

cons:

-schakelen trager dan de bipolaire transistoren

MOS transistoren zijn dus veel compacter (dichter bij elkaar), maakt ze het aantrekkelijkst (ook omdat ze goedkoper zijn). Meeste processoren en geheugens gebruiken CMOS-technologie.

Bipolaire technologie enkel gebruikt wanneer de MOS-technologie te traag is.

2.2.3 Enkele begrippen uit de schakelalgebra

NEN en NOF poorten zijn speciaal: uit deze kan elke logische schakeling gemaakt worden.

(Bij het bouwen van grotere schakelingen kan men soms NEN of NOF poorten weglaten. Bij het bouwen van een XOF schakeling met enkel NEN poorten creëert men bvb 2x een dubbele negatie (door einde EN + begin OF) waardoor vier poorten weggelaten kunnen worden.)

basis bewerkingen:

- EN bewerking
- OF bewerking
- NIET bewerking

2.2.5 Wet van Moore

Wet van Moore : elke 18 maanden 2X zoveel transistoren op een chip

gevolgen:

- ⤴ kostprijs van een chip hetzelfde gebleven, kost transistor exponentieel gedaald
- ⤴ hogere snelheid - dichter bij elkaar
- ⤴ miniaturisatie
- ⤴ voortdurende capaciteits verdubbeling

Deel 2: RAM- en ROM geheugens

2.3 Geheugens met onmiddellijke toegang

In het engels: RAM "Random access memories". Er kan zowel gelezen worden als geschreven. Vandaag 2 soorten: SRAM (statische ram) en DRAM (dynamische ram)

2.3.1 Het verleden: ferrietkernegeheugens

WAT?

Ferrite Cores --> naam Core Memory

Ferrietkernen geordend in magnetiseerbare ringetjes.

Niet meer gebruikt

voordelen:

- ⤴ Niet vluchtig. Als spanning weg was bleef de inhoud bewaard.

- ⚡ Niet gevoelig voor straling.

nadelen: niet compact, duur (moeilijk te maken), vrij traag.

2.3.2 Statische geheugens

SRAM

Een geheugencel is een schakeling die een binaire waarde kan onthouden = schakeling met twee stabiele toestanden → (uitvoer NEN- (of NOF-) poorten zijn elkaars tegengestelde).

SRAM (statische RAM) = aanleggen van een spanning is voldoende om informatie te bewaren

Er zijn veel varianten, vaak worden de bipolaire transistoren vervangen door MOS-transistoren.

CMOS-geheugen

Zeer laag energieverbruik

NOVRAM (non volatile ram)

2.3.3 Dynamische geheugens --- DRAM

Dynamisch geheugen

- ⚡ condensator = geladen/ontladen
- ⚡ transistor = schakelaar

Schrijf operatie = 1. positieve spanning aanleggen aan de bitlijn of de bitlijn aarden
2. positieve spanning aanleggen aan woordlijn voor het sluiten van de transistor

Lezen frist de condensator opnieuw op

- OF opgedreven tot de maximum spanning
- OF volledig ontladen

logisch circuit verfrist deeltjes van het geheugen

Voordelen DRAM tov SRAM geheugen:

- ⚡ minder transistoren dus minder plaats nodig op de chip
- ⚡ Lagere energieconsumptie

- ⤴ Compacter geheugen
- ⤴ DRAM is goedkoper

Nadelen tov SRAM zijn:

- ⤴ Extra logische schakelingen nodig voor het opfrissen van het geheugen
- ⤴ Er wordt destructief gelezen, bij elke leesoperatie moet de bit herschreven worden
- ⤴ Het is trager door de opfrissing en herschrijven
- ⤴ DRAM is veel omgevingsgevoeliger. Licht kan de lading van de condensator beïnvloeden

2.3.4 Geheueenchips en geheugenmodules

RAM

- ⤴ 2D arrays (matrix)
- ⤴ veel varianten

Geheugenmodules

- afzonderlijke printplaat
- SIMM (Single Inline Memory Module) = 32-bit =hoek van 30 graden in geheugenexpansieslots
- DIMM (Dual Inline Memory Module) = 64-bit =hoek van 90 graden in geheugenexpansieslots

2.3.5 Eigenschappen

capaciteit

- maximaal aantal gegevensbits die opgeslagen kunnen worden

snelheid

- technologie
- soort RAM geheugen

- geheugenbestuurder

tijd

- toegangstijd = tijd die geheugen nodig heeft om een opdracht uit te voeren

- geheugencyclustijd = tijd die minimaal moet verstrijken tussen twee opdrachten

2.3.6 Verbeterde DRAMs

FPM DRAM: Fast page mode DRAM: oudste vorm nog in gebruik. Geheugencellen in opeenvolgende kolommen kunnen sneller uitgelezen worden.

EDO DRAM: Extended Data Out DRAM: een nieuw adres kan worden aangeboden nog voor de vorige operatie compleet is. De geheugentoegangen overlappen elkaar.

SDRAM: Synchronous DRAM: de data en adreslijnen worden door één klok gestuurd, geheugen kan sneller gelezen worden. (de vorige zijn asynchroon).

DDR SDRAM: double data rate SDRAM: 2X snelheid van SDRAM

RDRAM: Rambus DRAM: Nog snellere DDR SDRAM

VRAM: Video RAM: gebruikt voor grafische systemen. Gelijkaardig aan FPM DRAM.

2.3.7 Nieuwe ontwikkelingen

MRAM: magneto-resistieve RAM met twee magnetiseerbare lagen

voordelen MRAM: combinatie goede eigenschappen van andere geheugentypes

- compact (zoals DRAM)

- snel (zoals SRAM)

- niet-vluchtig (zoals Flasgeheugen)

nadelen MRAM:

- PRIJS!!!!

2.4 Niet vluchtige geheugens – leesgeheugens

ROM: Read only memory

niet vluchtig

meerdere varianten

2.4.1 ROM – Read only memory

1 of 0 = transistors aanwezig of afwezig

compacter dan DRAM (1 bit = 1 bouwsteen(transistor))

toegangstijd ongeveer gelijk aan DRAM

cyclustijd korter dan bij DRAM

nadelen:

- kunnen niet meer gewijzigd worden
- soms te duur (niet kosteneffectief om speciale maskers te maken voor kleine oplagen)

2.4.2 PROM – Programmable ROM

- 1X geschreven worden
- een zekering zorgt dat het niet kan overschreven (na eenmalige beschrijving) worden
- het opblazen van de zekering schijft nullen overal waar geen 1 staat

2.4.3 EPROM – Erasable PROM

- wissen en herschrijven mogelijk
- blootstellen aan een UV licht wist alles (met speciaal toestel)
- veel wissen en herschrijven verslechtert de eigenschappen van de chip

2.4.4 EEPROM – Electrically EPROM

- Electrically Erasable Programmable Read Only Memory
- ter plaatse herschrijven en wissen mogelijk (meest flexibel)
- nadelen:
 - half zo snel als EPROM, 10x trager als SRAM of DRAM
 - duur

- BIOS wordt er in opgeslagen

2 varianten:

- full featured (duurste, wissen gebeurt op byte-niveau)
- flash memory

2.4.5 Flashgeheugen

- opgedeeld in blokken van enkele kilobytes
- wissen gebeurt op blok-niveau

- capaciteit en herschrijfbaarheid(10.000 tot 100.000 keer) limiteren competitie met harde schijven

Deel 3: Hulpgeheugens

2.5 Magneetschijven

2.5.1 Fysisch principe

magnetische laag op de schijf zorgt voor bijhouden gegevens

schrijfmechanisme :

- elektrische stroom door geleider creëert een magnetisch veld, dat dan de schijf magnetiseert.

leesmechanisme :

- bewegende schijf die beweegt langs een geleider wekt een stroom op die dan verwerkt word.

2.5.2 Harde schijf

Platen (platters)

- aluminium, glas, composietmateriaal
- ultraglad
- verticaal gestapeld
- holle as
- constante snelheid
- niet magnetiseerbaar

Magnetiseerbare laag

- vroeger : ijzeroxide
- nu : dunne film

Lees- en schrijfkoppen

- gemonteerd op arm (allemaal verbonden)

Logische opdeling van de schijf

sporen (tracks)

- concentrische ringen
- van buiten naar binnen genummerd beginnend bij 0
- 800-2000 tracks/cm
- verdeeld in sectoren = kleinste hoeveelheid die in een keer gelezen/geschreven wordt

lineaire bitdichtheid niet constant = PROBLEEM

oplossing : zones (meer zones op de buitenringen, grotere capaciteit)

cilinder = sporen die boveneen liggen op verschillende platen

Een sector

- ⤴ kleinste eenheid dat gelezen of geschreven wordt

Bestaat uit:

- ⤴ HD (hoofding)
- ⤴ Data
- ⤴ ECC (Error Correcting Code) = stelt de bestuurder in staat kleine fouten te verbeteren

formatteren

- moet alvorens informatie op te kunnen bewaren
- laag-niveau : fabriek: schijf onderverdeeld in sporen en sectoren
- hoog-niveau : tabel weggeschreven door besturingssysteem

De lees- en schrijfkoppen

- positieve of negatieve stroom door spoel in schrijfelement
 - stroom toegevoegd: schrijf operatie
 - geen stroom toegevoegd: lees operatie
- verticale en horizontale magnetisering (vertikaal = veel plaats & precisie vereist)
- lees- en schrijfkoppen zitten in een glijder (slider) = te klein voor rechtstreekse montage
- 1. Ferriet-koppen -> 2. dunne-filmkoppen -> 3. magneto-resistievelkoppen (MR-koppen)

Landingszone (binnenkant van de schijf) = traagste lineaire snelheid (schijf niet gebruikt)

Behuizing

Koppen zweven 0.2 nanometer boven de schijf

headcrash = kop botst met plaat → kan gebeuren door kleinste vuildeeltje (bv vingerafdruk)

Vermijden door:

- luchtdicht afsluiten (NIET luchtledig) door gebruik van Winchester-techniek

Snelheid van het schijfstation

De snelheid hangt af van:

- ✧ Zoektijd (seektime): de koppen moeten boven het juiste spoor gebracht worden (5-15ms)
- ✧ Rotatiewachttijd (latency): wachten tot de juiste sector onder de kop komt. Snelheid is recht evenredig met rotatiesnelheid (ms)
- ✧ Transporttijd (transfer time): sector moet gelezen of geschreven worden. Hangt af van: rotatiesnelheid, bitdichtheid, grootte van een sector. Product van de eerste twee zijn samen de overdrachtsnelheid (of debiet).
- ✧ Opstarttijd (spinup time): soms zal de schijf stil worden gelegd om energie te sparen. Het kan enkele seconden duren vooraleer de schijf weer op volle snelheid is.
- ✧ Kalibreren: door het snelle toerental worden de schijven zeer warm, ze zetten uit. De kop moet soms weer kalibreren om de juiste hoogte boven de plaat te hebben. Kan rampzalige gevolgen hebben voor programma's die snel info verwachten.

Schijfbestuurder (controller)

bestuurder

- ontvangt en verstuurt signalen van en naar de processor en van en naar de schijf
- heeft buffer

Types:

- 1. IDE -> 2. EIDE -> 3. ATA -> 4. ATAPI -> 5. SATA
- SCSI (Small Computer System Interconnection)

2.5.3 Soepele schijf

Diskette (Floppy Disk)

Verschillen met harde magneetschijven:

- buigzaam plastic/karton
- metalen sluiters
- koppen glijden over het oppervlak (warmte en sleet treden op)
- trager

2.5.4 RAID

RAID (Redundant Array of Inexpensive/individual Disks)

Methode om schijven sneller te maken

+

techniek om schijven betrouwbaarder te maken

Er zijn 6 elementaire niveaus te beginnen van 0(niveau 0→5) + 1 uitbreiding van niveau 5 die niveau 6 genoemd wordt(0→6). P54.

4 concepten

- spiegeling (mirroring): data wordt naar twee schijven weggeschreven
- ont dubbeling (duplexing): uitbreiding spiegeling met verdubbeling bestuurders
- gebruiken van stroken (data striping): grote hoeveelheid wordt opgedeeld en naar verschillende schijven geschreven (sneller)
- pariteits strook: informatie reconstrueren van uitgevallen schijf

2.5.5 Solid State Disks (SSD)

Voordelen:

- snel
- betrouwbaar

Nadelen:

- verslijten
- verliezen ook data als spanning wegvalt
- DUUR!!!!
- niet zo groot als harde schijven (qua GBs)

2.5.6 Varianten

Veel varianten

niet compatibel met elkaar

ex : ZIP, JAZ (ieder heeft een eigen soort drive nodig)

2.6 Optische schijven

CD (Compact Disk)

- Rode boekje (technische gegevens voor standaardisering)
- Digitaal signaal opgeslagen
- pits and lands
- gemaakt met plastic, aluminium, en een sterke rode laser
- afspelen gebeurt met een laser en een foto-elektrische cel
- Lichtgolven reflecteren of botsen om een 1 (overgang tussen pits en lands) of een 0 (geen overgang) te weergeven.
- putjes worden in een spiraal gebrand
- CLV (Constant Linear Velocity) [CDs]
- CAV (Constant Angular Velocity) [Harde Schijven]

2.6.1 CD-ROM

Positioneren van de laserstraal moet voortdurend bijgesteld worden

Gemiddelde toegangstijd

- tijd nodig om snelheid aan te passen
- zoektijd
- rotatie wachttijd

gemiddelde toegangstijd ongeveer 100 ms

totale capaciteit : 650 Mbyte

2.6.2 CD-R

- Een keer schrijven, meerdere keren lezen
- in een doorlopende operatie geschreven
- failsafes => vermindert de kans op onderbreking van de gegevensstroom
 - grote buffers (in RAM)
 - image-bestand
- oranje boekje(technische gegevens standaardisering) (1989)
- CD-ROM/XA = incrementeel schrijven = multi-sessie-CD
- spoor (track) = aantal sectoren dat in een keer geschreven wordt

2.6.3 CD-RW

Compact disk rewritable

twee vaste toestanden:

- kristallijn (reflecteerd goed)
- amorf (reflecteerd slecht)

laser met 3 verschillende vermogens

- hoog = putje gemaakt
- gemiddeld = land gemaakt
- laag = lezen

sandwich structuur:

-legering tussen 2 diëlectrische lagen voor warmte afvoer

CD-RW vs. CD-R

- CD-RW duurder
- soms een voordeel om niet te kunnen herschrijven

2.6.4 DVD

kenmerken :

- kleinere putjes
- dichter gewonden spiraal
- rode laser om schijf te lezen

debiet van 1,4 Mbyte/s

capaciteit :

- enkelzijdig, een laag = 4,7 GB
- enkelzijdig, dubbel laag = 8,5 GB
- dubbelzijdig, een laag = 9,4 GB
- dubbelzijdig, dubbel laag = 17 GB

dubbelzijdig maakt men door 2 enkelzijdige aan mekaar te plakken

2.6.5 DVD-R, DVD-RAM, DVD-RW, DVD+RW

DVD-R is de WORM(Write Once, Read Many) versie van de DVD

DVD-RAM, DVD-RW, DVD+RW zijn herschrijfbaar (varianten zijn niet compatibel met elkaar)

2.6.6 HD-DVD vs. Blu-ray

HD-DVD:

- Toshiba/NEC
- 15 GB/laag

Blu-ray:

- Sony
- 25 GB/laag

Aparte apparatuur nodig voor beide

Blu-ray" gewonnen (19 feb 2008)

2.6.7 Magneto-optische schijven

- zowel magnetische als optische technologieën toegepast
- getemperd glas + magnetiseerbare registratielaag
- stabiel bij kamertemperatuur
- magnetisatie gebeurt vertikaal
- bits worden afgelezen met een laserstraal

schrijven:

- bitpositie opwarmen tot Curietemperatuur (+/- 200°C) of hoger
- elektromagneet creëert magnetisch veld
- bitpositie koelt af en magnetisatie verandert niet meer

lezen:

- laserstraal met laag vermogen
- kerr-effect = weerkaatste straal is gepolariseerd in uurwerkzin of tegengestelde zin afhankelijk van magnetisatie van het oppervlak

capaciteit : 3Gbytes

gemiddelde toegangstijd = 30-80 ms

2.7 Magneetbanden en –cassettes

2.7.1 Magneetbanden

- plastieken lint met magnetiseerbaar laagje
- bits gerangschikt in sporen in langsrichting van band
- 9 sporen (1 byte en 1 pariteitsbit)
- 9 lees-/schrijfkoppen
- blokken van variabele lengte
- hiaat = ongeschreven gedeelte van de band
- blokken hebben geen adres
- sequentiële geheugens = moeten in volgorde gelezen of geschreven worden
- back-up, archivering, grote hoeveelheid data transport

2.7.2 Magneetcassettes

magneetbanden worden al een tijdje niet meer gebruikt

- cassettes worden voor backups gebruikt
- gesloten behuizing, compacter
- meer sporen
- slangvormig schrijven (serpentine recording)
- 1/2", 1/4" (meerdere standaarden die niet compatibel zijn), 8mm, 4mm (beide 8 en 4 hebben sporen die op 5 graden geschreven/gelezen worden)

2.8 Massageheugens

mediawisselaar

- zeer grote capaciteiten
- trage toegankelijkheid (tientallen seconden)

- bestaan uit groot aantal verwijderbare media
- gebruikt voor automatische back-ups

Deel 4: Communicatie verbindingen

2.9 Gegevenstransport

Serieel

- bits 1 per 1 verstuurd
- punt-tot-punt verbinding

Parallel

- meerdere bits tegelijkertijd
- busverbinding
- mogelijk tijdelijk uitstel

2.10 een busverbinding

bus = gemeenschappelijk pad tussen meerdere componenten

- adresbus = geeft adres door
- gegevensbus = geeft de data door
- controlebus = geeft controle signalen door

gemultiplexte bus = adres en gegevenslijnen zijn gemeenschappelijk (eerst adres, dan data)

2.10.1 Busbreedte = aantal verbindingen

- hoe breder, hoe duurder
- meer draden, meer plaats, grotere connectoren
- breedte van adresbus bepaalt het adres bereik
- databus 8, 16, of 32 lijnen
- breedte van controlebus hangt af van de hoeveelheid verschillende opdrachten

2.10.2 Busprotocol

busprotocol = gebruiksregels van de bus

- toegang tot de bus
- welke signalen via welke lijn
- timing van elk signaal (wanneer op bus en hoelang)
- vrijgave van de bus

2.10.3 Synchrone of asynchrone bussen

synchrone bus = alles op kadans van een klok

asynchrone bus = elke busactie neemt alle tijd nodig in beslag, maar nooit meer.

2.10.4 Busarbitrage

gebruik van de bus onderhevig aan strikte regels (welk component eerst)

- component moet eerst Master over de bus worden
- Master & slave = alleen de Master mag een busoperatie starten

gecentraliseerd = apart component beslist

gedecentraliseerd = apparaten beslissen onderling

2.10.5 Busoperaties

verschillende soorten operaties

voorbeelden:

- lees-cyclus
- schrijf cyclus
- bloktransfer cyclus
- lees-wijzig-schrijf cyclus (test and set)

- programmaonderbreking (interrupt)

- ...

2.11 Voorgeheugens (cache)

ontworpen om de kloof tussen processor wachttijd en geheugen toegankelijkheid te verkleinen

2.11.1 Het principe

door twee soorten geheugen (snel, duur, klein en traag, goedkoop, groot) te combineren blijft er snel, redelijk, en groot over.

werking:

- processor zoekt in de cache
- treffer : info gevonden
- misser : verder zoeken in het hoofdgeheugen

2.11.2, 2.11.3 organisatievormen

boekhoudkundige informatie:

- geldig (bevat de cache een kopie)
- welke regel

manieren om regels uit het hoofdgeheugen toe te wijzen

- direct afgebeeld voorgeheugen
 - elk geheugenblok heeft een vaste plaats in het voorgeheugen
 - meeste voorgeheugens zijn van dit type
- associatief voorgeheugen
 - uniek labelveld
 - eerste 3 cijfers van het adres zijn uniek
 - gelijktijdig of incrementeel vergelijken om hits te zoeken in de cache
 - missers vervangen een ander stuk data in het voorgeheugen

- duurste, maar niet noodzakelijk het beste
- set-associatief voorgeheugen
 - compromis tussen associatief en direct afgebeeld
 - elke regel kan nu op enkele verschillende plaatsen staan (adres bepaalt)

2.11.4 Vervangingsstrategieën

- soms ligt de keuze vast
- minst gebruikte word verwijderd (least recently used)
- de regel die al het langste in voorgeheugen zit wordt vervangen (first in, first out)
- willekeurig

2.11.5 Ontwerpparameters

- grootte
- regellengte (aantal bytes die tegelijk uit het hoofdgeheugen worden opgehaald)
- organisatie
- gezamenlijk of gesplitst
- aantal

bij pipelining word het geheel duurder en complexer, maar heel veel sneller

2.11.6 Prestatieverbetering

trefkans:

- grootte
- organisatie
- lokaliteitseigenschappen van het programma zelf

2.12 Pipelining

techniek die de snelheid van de processor opdrijft zonder de kloksnelheid te laten toenemen

Bevelencysclus:

1. ophalen instructie plus ophogen bevelenteller (BT)
2. analyseren instructie
3. berekenen van het adres van de operand
4. ophalen (indien nodig) van de operand uit het geheugen
5. uitvoeren van de instructie

2.12.1 Het principe

alles wordt aan een lopende band gedaan

veronderstel dat elke stap even lang duurt

2.12.2 Prestatie verbetering

opdrachten overlappen = sneller

2.12.3 De complexere werkelijkheid

probleem = adressen

- instructie kan afhankelijk zijn van een vorige instructie
- na elk sprongbevel staan er op de band nog enkele instructies die niet meer uitgevoerd moeten worden

oplossing:

- hardware detecteert afhankelijkheden
- afhankelijkheden worden niet toegestaan = NOP (No-Operation) instructies

probleem = sprong bevelen

oplossing:

- sprong bevel naar voor schuiven, NOP inlassen
- geen bevelen ophalen tot het adres berekend is

probleem = voorwaardelijke sprong

oplossing:

- tweede lopende band

- voorspellen of al dan niet gesprongen moet worden

2.13 Multiprocessoren

- hoger verbruik
- meer warmte
- meerdere processoren = meerdere programma's tegelijk uitvoeren
- monoprocessor-systeem wisselt af tussen verschillende programma's (illusie van gelijktijdigheid)
- meerdere processoren in dezelfde behuizing

problemen:

- elektrisch verbruik + warmte afvoer → multiprocessoren verplicht lagere klokfrequentie als snelste monoprocessoren
- beheer van de voorgeheugens (coherentie behouden)
- het benutten van de beschikbare rekenkracht

2.14 Hardware hyper-threading

- deel van de processor ontdubbelen (alles wat met de toestand van een thread te maken heeft)
- toestand opslaan van de threads niet nodig
- threads waar men moet wachten worden overgeslaan
- besturingsysteem moet aangepast worden om logische en echte processoren te onderscheiden

2.15 Computerprestaties

2.15.1 MIPS, MFLOPS, SPEC

MIPS (Million instructions per second)

- niet alle instructies even langdurig
- niet alle instructies even frequent
- niet alle processoren gebruiken dezelfde instructieset

- niet alle instructies werken op dezelfde gegevenstypes

MFLOPS (Million Floating Point Operations per second)

- andere bewerkingen worden ook gebruikt
- verschillende processoren verschillen in bevelenset
- randapparatuur telt ook mee

SPEC (Standard Performance Evaluation Corporation)

- organisatie zonder winstoogmerk
- definiëren de standaarden (Benchmarks)

SPEC int = getallen en lijsten

SPEC fl = zwaar rekenwerk

SPEC maat is het gemiddelde van de uitslag van deze twee programma's

Benchmarks:

Whetstone = wetenschappelijk en grafische toepassingen benchmark

Dhrystone = geen bewegende komma benchmark

2.15.2 Gebruikersstandpunt

Responsetijd (responsetime)

- tijd tussen het geven van een opdracht en het zien dat er iets gebeurt

Doorlooptijd (turnaroundtime)

- tijd om een programma volledig op te starten

Doorvoer (throughput)

- hoeveel programma's afgewerkt zijn

Hoofdstuk 3: Basisprogrammatuur (adviseer om dit hoofdstuk zeker goed in het boek te leren)

Vertaalprogramma == assembleerprogramma = lijn per lijn vertaling

voorvertaler = expandeert macro oproepen

lader = zet het programma in het geheugen

binder = verschillende modules samenvoegen tot 1 groot programma

vertolker = interpreteert bronprogramma

speurprogramma (debugger) = zoekt naar fouten

bootstrap = speciale techniek voor het opstarten van een computer systeem

3.1 De vertaler

machinetaal = computer begrijpt alleen deze taal

vertaler/assembleerprogramma = vertaalt code naar machinetaal

bronprogramma = invoer voor de vertaler

uitvoerbaar programma = uitvoer van de vertaler

3.1.1 DRAMA-machine-instructies

programmaonderbreking = pauze in het programma bij het detecteren van een fout

DRAMA instructies bestaan uit verschillende velden:

- opcode: welke bewerking
- modusveld (1^{ste} en 2^{de} modus-cijfer): hoe de operand berekenen (2^{de} moduscijfer) en interpreteren (1^{ste} modus cijfer)
- accumulatorveld: op welk rekenregister is de instructie van toepassing
- indexveld: welk rekenregister als indexregistrer
- operandveld: adres geheugenregister of getal. Bij INV en UTV poortnummer

3.1.2 De MINI-DRAMA-vertaler

= vereenvoudiging DRAMA-vertaler

Bevat mnemo-technische functiecodes

Bevat decimaal adres

geen symbolisch adressen

geen geheugenreservaties

geen interpretaties

3.1.3 De DRAMA-vertaler

4 verschillende invoerlijnen (idem mini-DRAMA-vertaler):

- ⤴ Commentaar
- ⤴ Constanten (geïnitieerde geheugenplaatsen)
- ⤴ Directieven
- ⤴ Instructies

commentaarlijnen worden overgeslagen

constanten in 10-complement gezet

DRAMA instructies moeilijker te vertalen (t.o.v. mini-DRAMA-instructies):

- 10 registers R0 → R9
- sommige operanden zijn alleen geldig bij bepaalde instructies

- interpretaties hebben een invloed op de inhoud van het modusveld soms niet toegelaten bij bepaalde instructies
- symbolische adressen (oorzaak onmogelijk om elke lijn onmiddellijk te vertalen)

Symbolische Adressen

voorwaartse referentie = adres niet gedefinieerd

achterwaartse referentie = adres gedefinieerd

Tweestapsvertaling

1. werkelijk adres opslaan in symbooltabel = behandeling definities van symb. adressen
2. code genereren met die tabel = eigenlijke vertaling met behandeling referenties

Stap 1: Het opstellen van de symbooltabel

drie gegevensstructuren nodig:

- lijnteller
- programmateller: bewaart adres van het bevel of de constante die vertaald wordt
- symbooltabel: bijhouden symbolische adressen en hun waarde

Initialisatie gegevensstructuren:

1. lijnteller = 1
2. programmateller = 0000 (tenzij programma elders geladen word)
3. symbooltabel = leeg

lijn per lijn word ingelezen

- Commentaarlijnen worden genegeerd
- Word er een symbolisch adres gedefinieerd toevoegen aan symbooltabel + waarde programmateller
 - a) Gewone drama instructie programmateller +1
 - b) Constanten/geheugenreservaties programmateller +aantal constanten/aantal reservaties
- Na behandeling van lijn → lijnteller +1

Stap 2: codegeneratie

- bevelen omzetten naar machine bevelen
- listingbestand opmaken met alle vertalingen

3.1.4 De vertaler voor een hogere programmeertaal

compiler (compiler) = zet hogere programmeertaal om in machinetaal

- meestal tamelijk eenvoudig

taken:

1. lexicale analyse: herkennen van de woorden van een taal (tokens)
2. syntactische analyse: herkennen individuele opdrachten
3. semantische analyse: begrijpen individuele opdrachten
4. code genereren

Voordelen (van eerste code genereren in een lager programmeertaal i.d.p.v. onmiddellijk machine-taal) :

1. eenvoudig testen
2. alleen het assembleerprogramma moet veranderd worden bij het veranderen vd inwendige bevelenopmaak

compilers worden ontworpen in hogere programmeer talen

kruiscompiler (cross-compiler) : compiler die alles kan vertalen (hoog naar machinetaal)

3.2 De MACRO-voorvertaler

Voorvertaler verwijdert macro definities:

- symbool manipulatie is niet hetzelfde als code generatie
- macro's kunnen overal gebruikt worden(niet alleen in drama programma's)

3.2.1 Voorvertaler algoritme (zie eerst volgende blauwe titel)

drie verschillende taken:

1. alles wat niet met macro's te maken heeft, kopiëren
2. macrodefinities verwijderen en tijdelijk bewaren(voor toekomstige macro-expansies)
3. macro-oproepen expanderen (vervangen door eventueel licht gewijzigde macrodefinitie)

3.2.2 Extra macrodirectieven

voorvertaler variabelen en etiketten mogen globaal gebruikt worden

MEVA = bewaart een waarde in een Macro variabele

variabelen kunnen globaal of lokaal zijn

globale variabelen zijn niet altijd zichtbaar

argumenten-en-lokale-variabelentabel = houdt macro expansies bij

Gegevensstructuren voor het bijhouden

Globale vv-variabelen in de globale vv-variabelentabel (GVT) -> met MCC en teller

Lokale vv-variabelen samen met de argumenten tabel: (ALVT)

Globale vv-etiketten in de globale-etikettentabel (GET)

Lokale macro-etiketten in de lokale-macro-etikettentabel (LET) -> in de macro geg struct!

3.3 De Lader

- brengt het programma alleen maar in het geheugen

- zit zelf ook in het geheugen

verschillende lader algoritmes/schema's

- absolute : plaatst het programma ongewijzigd in het geheugen

- relocerende : past het programma aan zodat het in het geheugen past

3.3.1 De absolute lader

ongewijzigde overdracht:

- leest lengte

- leest en bergt in opeenvolgende geheugenplaatsen vanaf 0000

Het meest elementaire geval

DRAMA laadt altijd vanaf 0000

geheugen na transfer :

- hoofding: vermelding laadadres en lengte
- programma code
- symbooltabel

3.3.2 De relocerende lader

de vertaler zegt aan de lader hoe de machinecode moet aangepast worden als het programma niet van geheugenadres 0000 begint

- reloceerbare code genereren
- relatief adres toont aan welke instructie aangepast moet worden
- laad in het geheugen en past daarna de code aan (relocatiestap)
- programma moet reloceerbaar (bevat een relocatietabel) zijn alvorens een relocerende lader te kunnen benutten
- hardware meer taken geven = hogere flexibiliteit

3.3.3 Dynamische relocatie

- relocatie gebeurt in 1 keer voor de uitvoering van het programma

Door gebruik van geheugenbeheereenheid (GBE) (memory management unit : MMU)

- dynamisch reloceren = dynamische adresvertaling → tijdens uitvoeren reloceren
- GBE berekent ALLE adressen
- basisregister = register met het begin adres als eerste lege spot
- inhoud basis register blijft constant
- het lijkt alsof het programma vanaf 0000 geladen is (maar dat is niet zo)
- inladen gebeurt best zonder dynamische relocatie
- lader vult daarna aan bij het basisregister
- geheugenbeheereenheid wordt pas geactiveerd als het inladen begonnen is

vertaler vereenvoudigd door dynamische relocatie want:

- hij mag steeds veronderstellen dat alles vanaf 0000 geladen is
- moet geen relocatietabel opstellen

laderschema komt grotendeels overeen met die van een absolute lader met een wijziging:

- invullen van het basisregister en het inschakelen van de GBE vlak voor het geladen programma gestart wordt

3.4 De binder

modules = kleine stukjes van een programma in afzonderlijke bestanden

binder = koppelt alle modules om tot een groot, werkend programma

3.4.1 Waarom een binder?

Problemen bij grote programma's op broncodeniveau:

- symbolische adressen mogelijk twee keer hetzelfde etiket
- hervertaling te langdurig
- subroutines blijven overal hetzelfde maar men wil de broncode niet uit handen geven

3.4.2 Modules

modules:

- bevatten verwijzingen
- vertaler directief
- externe symbolische adressen

symbolische adressen zijn lokaal ten opzichte van de module (als ze gedefinieerd zijn)

- anders globaal

3.4.3 Bestanden

elke module word in een afzonderlijk bestand bewaard

- geen probleem voor lagere programmeertalen
- beperkingen voor hogere programmeertalen(bvb splitsen van procedures in C)

3.4.4 Hoofdprogramma

modules worden afzonderlijk vertaald

- resultaat is niet uitvoerbaar
- object module

Informatie voor de binder

wat word behouden:

- mnemotechnische naam
- interpretatie word weergeven (expliciet)

- decimaal voorgestelde operanden (seperated by commas)
- getallen worden voorgesteld door de hoeveelheid cijfers ze in beslag nemen
- negatieve getallen worden met hun 10-complement voorgesteld
- indexregister
- adressen van machinebevelen

3.4.5 De taken van de binder

Binder gebruikt informatie over globale etiketten

taken:

1. plaats voorzien voor verschillende modules
2. machinecode samenvoegen
3. startadres bepalen
4. code reloceren
5. externe referenties in rekening brengen
6. relocatietabel opstellen

reloceren en binden zijn de moeilijkste taken

binder werkt in 2 stappen:

1. allocatietabel en globale symbooltabel opstellen

- allocatiesectie (beginadressen)

Bepaalt nieuwe beginadressen voor de modules = beginadres + le,gte vorig module

- symbolensectie

- lokale en externe symbolische adressen worden genegeer

- globale etiketten worden met hun verhoogde waarde in de globale symbolentabel ingevoegd

2. code samengevoegen, eventueel reloceren en binden

invloed van Dynamisch relocatie op binder :

- weinig

Vertaler wordt complexer

- externe symbolische adressen verwerken

- relocatie/bindingstabel toevoegen aan elk object module

Dynamisch binden = binden uitstellen tot object module nodig is

- voordeel : als module niet meer gebruikt word, dan zit het niet meer in het geheugen

3.4.6 Programmabibliotheken

- object modules samen bewaard

3.5 De vertolker

vertaling = elke instructie omzetten in het machine-taal-equivalent

vertolker = voert een programma direct uit

- programma dat een ander programma inleest en vertolkt

- combineert vertaling en uitvoeringsfase

3.5.1 Werking van een vertolker

leest 1 bronsequentie, vertaalt, en voert direct uit

doelcomputer = computer waar programma vertaald word

gastcomputer = computer waar programma uitgevoerd word

als vertolker doelcomputer nabootst, dan is het een simulator

3.5.2 Voordelen t.o.v. vertalers

- vertolker is gemakkelijker te schrijven dan een compiler

- geen geheugenruimte nodig voor het volledig vertaalde programma

- beschikt over broncode, dus kan zinvolle foutmeldingen afdrukken

- hoeft zelf de machine-instructies niet te kennen

3.5.3 Nadelen

⚠ uitvoeren duurt lang

⚠ te veel geanalyseerd(lus)

optimalisaties

⚠ zet ingelezen programma om in een meer geschikte vorm voor computer

⚠ vaak uitgevoerde fragmenten direct in machine code zetten

3.5.4 Toepassingen van vertolking

- nieuwe programmeertaal vertolker is eenvoudiger
- zeer hoog niveau programmeertaal mogelijk permanent vertolken
- simulator
- emulator
- LISP

3.6 Het speurprogramma

speurprogramma (debugger) = spoort fouten op in een programma

breekpunt : plaats waar het programma word onderbroken

symbolische speurprogramma's = debuggers van een hogere programmeertaal

3.7 Het opstarten van de computer

besturingspaneel : controle paneel om computer op te starten (vroeger, niet meer gebruikt)

halt toestand = computer doet niks

uitvoeringstoestand = computer doet iets

bootstrap procedure : starten van een simpel programma en daarmee complexere programma's laden

- test apparatuur
- laad klein programmaatje op een vaste plaats in het secundair geheugen (bootsector)
- start geladen programma dat de rest van het OS inleest

voordelen van ROM

- bestendig geheugen
- beveiligd
- vlugger

Hoofdstuk 4: Besturingssystemen

4.1 Inleiding

besturingssystemen hebben 2 doelstellingen:

1. gebruikersvriendelijker te maken
2. gebruik efficiënter te maken

sommige besturingssystemen zijn voor bepaalde hardware ontwikkeld:

- DOS (Intel)
- OS/2 (Intel)
- VMS (DEC-machines)
- MVS (IBM-Mainframes)

gemeenschappelijke kenmerken

regelt/controleert de werking van een computer

4.2 Het programma-toestandwoord

programmatoestand woord (PTW) - speciaal register dat deel uitmaakt van de processor

4.3 Programma-onderbrekingen

- belangrijk

4.3.1 Inleiding

overloop =====> uitvoering automatisch onderbroken

4.3.2 De bevelencyclus

normale werking van een computer

1. volgende instructie wordt opgehaald
2. instructie wordt geanalyseerd
3. instructie wordt uitgevoerd

sequentiele besturing verbreken door :

- sprongbevelen
- programmaonderbrekingen (1 t.e.m. 9)

4.3.3 Het programma-onderbrekingsmechanisme

bevelen cyclus word nu :

1. Volgende instructie ophalen
2. instructie analyseren
3. instructie uitvoeren
4. check voor programmaonderbrekingen en melden

4.3.4 Soorten onderbrekingen

Programma-onderbrekingen kunnen onderverdeeld worden in 5 klassen:

- ✧ Externe oorzaken, bv reset knop of een wekker.
- ✧ Rand-apparatuur: hulpgeheugens, toetsenbord, etc.
- ✧ Machinefout
- ✧ Programmafout (overloop, deling door 0)
- ✧ Geprogrammeerde PO: soms wil het programma onderbroken worden (soms is SBR niet de beste manier)

4.3.5 Verbieden van programma-onderbrekingen

verbieden:

- opschorten/uitstellen
- negeren

Methodes:

- a) maskers
- b) prioriteiten

Globale Maskers

PO masker (mask)

maskerwoord = neemt 2de helft van PTW in beslag

- enabled
- disabled

als ze aanstaat wordt geen enkele programmaonderbreking toegestaan

globale programmafout (GPF) verhindert overloop en stapeloverloop fouten

Prioriteiten

- soorten worden via hun belangrijkheid gerangschikt

- eerste cijfer in het PTO is het onderbrekingsniveau

4.3.6 PO-routines

PO-(behandelings)routine (interrupt service routine / interrupt handler)

- moet de vorige uitvoering kunnen verder zetten

KTO (Keer Terug na Onderbreking) = eindigt PO-routine

- opgeroepen door apparatuur
- BT krijgt oude waarde terug
- onderbrekingsniveau terug zijn vorige waarde
- conditiecode (cc) en overloop indicatoren krijgen oude waarde terug
- tijdens behandeling van PO-onderbrekingen worden alle vlaggen genegeerd

4.3.7 Geprogrammeerde programma-onderbreking

geprogrammeerde PO's

- geplaatst door speciale instructie : onderbrekingsinstructie : OND

dienstnummer:

- onderscheid tussen verschillende diensten BP

meeste PO's hebben met I/O-apparaten te maken

4.4 Invoer en uitvoer

een van de belangrijkste taken van een besturingssysteem is het beheer van randapparatuur

- geeft commando's
- behandelt PO's
- probeert fouten op te lossen

gebruiker is slechts geïnteresseerd in hoog-niveau opdrachten

- besturingssysteem zet hoog-niveau opdrachten om naar specifieke opdrachten in machinetaal

4.4.1 Randapparaten

elektronische en mechanische gedeeltes

- controller/adapter (bestuurder) bestuurt dikwijls meerdere identieke apparaten

speciale registers = poorten

- opdrachten en gegevens lezen/schrijven

lezen en schrijven kan op 2 manieren gebeuren

1. speciale I/O

2. inpassing van Poorten in het centraal geheugen

- memory mapped I/O

de toestandspoort kan alleen maar gelezen worden

- bestuurder schrijft toestand

fout-toestand = toestand waar het apparaat zich in bevind na enige soort fout

intermezzo:

- getPort (int)

- setPort (int)

uitbreiding van C (C kent interrupt niet) = 4 verschillen

1. alle accumulatoren bewaard op een vaste plaats

2. alle accumulatoren op het einde hersteld

3. routine keert terug via een KTO commando, niet KTG

4. voldoende plaats word gereserveerd (naam afgeleid van PO-routine)

4.4.3 In- en uitvoer organisatievormen

manier waarop bestuurder van een randapparaat opdrachten krijgt van processor

- a) geprogrammeerd
- b) programma onderbreking (PO)
- c) directe geheugentoegang
- d) speciale invoer/uitvoerprocessoren (channels)
- e) satelliet-computers

Met geprogrammeerde in- en uitvoer:

- teken per teken gecontroleerd door de processor
- oudste methode
- alleen nog maar gebruikt op zeer eenvoudige microprocessoren
- zekere synchronisatie nodig tussen processor en bestuurder
- processor moet nagaan of bestuurder zich in de juiste toestand bevind

2 kenmerken

- 1) gegevenstransport gebeurt via de processor
- 2) processor synchroniseert zichzelf met de bestuurder van het randapparaat

polling: vraag-boodschapjes sturen om te zien of apparaat klaar is

toetsenbord:

- lijn = opeenvolgende tekens, afgesloten door het nieuwe lijn symbool (return, enter)
- processor hoeft geen opdracht te geven aan controller

In- en uitvoer m.b.v programma-onderbrekingen

- bestuurder brengt processor op de hoogte (wanneer klaar/bezet is)
- bekijk methodes vanuit het standpunt van de bestuurder
- processor bepaalt wat, bestuurder bepaalt wanneer
- gegevensdebiet = laag

Directe geheugentoegang

direct memory access

- optimalisatie van IO met PO-s
- bestuurder zorgt voor alles buiten de opdracht te geven
- processor zet alles in't geheugen
- tijdwinst hangt af van het soort schijf
- rotatiewachttijd
- debiet

bestuurder steelt geheugen cycli (cycle stealing)

grotere computer installaties => bestuurder verbonden met geheugen via een ander pad

- een bestuurder met directe gegevenstoegang heeft geen gegevenstoestand

Kanaalbestuurders

krijgt een lijst van operaties en onderbreekt de processor pas als hij klaar is

- zeer lang, veel parameters

Satelliet computers

front-end (IO)

back-end (beheer)

voordelen:

- kunnen grote variëteiten van randapparaten besturen
- kan ook voor gewone berekeningen zorgen
- eenvoudiger programmatuur te vervangen
- niet fabrikant afhankelijk

nadelen:

- betrouwbaarheid verslechterd
- oplossingen;
 - een tweede computer
 - verzekering : onderhouds contract (hoe korter, hoe duurder)

randapparatuur-besturings routines (Device Drivers)

- hoe word het bestuurd

- SCSI (Small Computer System Interface)
- buffert boekhouding noodzakelijk
- tijdsafhankelijke problemen kunnen optreden

4.5 Processortoestanden

- halt toestand : geen bevelen opgehaald en doet geen zak
- uitvoeringstoestand : vraagt bevelen op en voert ze ook uit

4.5.2 Probleemtoestand en supervisietoestand

correcte werking garanderen => toegang

supervisie toestand : voert het besturingssysteem uit

probleemtoestand : voert een programma uit (niet OS)

3 overgangen:

1. elke programma onderbreking zet de processor in supervisie toestand
2. terugkeer uit de PO-routine brengt de processor in de vorige staat
3. STP-bevel => halt toestand

4.5.3 Geprivilegieerde bevelen

machinebevelen:

- geprivilegieerde bevelen = alleen uitvoeren als processor in supervisietoestand staat
- niet-geprivilegieerde bevelen

4.5.4 Supervisie-oproep

supervisie oproep (supervisor call) = effect van een supervisie programma

echo-transmissie : ontvangen symbool terug naar het scherm sturen

4.6 Multiprogrammatie

- terwijl randapparaten bezig zijn kan de processor iets anders doen

- "gelijktijdig" verschillende programma's uitvoeren (zelden Idle/werkloos)

- monoprogrammatie = altijd een enkele opdracht

doorvoer = throughput = aantal programma's per tijdseenheid

DOS = monoprogrammatie

grotere systemen gebruiken zelden/nooit monoprogrammatie

NOOIT mogelijk om processor 100% bezig te houden

programma vraagt aan BP om gestopt te worden

- BP verwijdert programma van het geheugen

besturingsprogramma:

- multiprogrammatie essentieel

- waarden van de accumulatoren

- de plaats waar de processor gekomen was (BT)

- de waarde van de conditiecode, de indicatoren, etc.

- als spanning wordt opgezet, wordt code die in ROM-geheugen zit uitgevoerd

- boot-sector bevat lader-programma dat de rest van het OS in leest

4.7 Soorten Besturingssystemen

soorten besturingssystemen:

1. systemen van algemeen nut (general purpose system)

niet interactieve

- stapel verwerking (batch systems)

interactieve

- single user (windows 95/NT/xp de rest is te nieuw voor de prof dus da staat er nog nie)

- multiple users (time-sharing systems)

2. specifiek doel (dedicated systems)

- real-time systems (eg. computer op't vliegtuig, bankautomaten, etc.)

4.8 Taken en diensten van het besturingsprogramma

- ⤴ **Geheugenbeheer:** Het BP moet beslissen welk stuk geheugen aan welk programma gegeven wordt
 - ⤴ leggen de onder en bovengrens van het toegekende geheugen
- ⤴ **Processorbeheer:** Het BP beslist aan welk programma de processor wordt gegeven en hoe lang
 - ⤴ prioriteit-gebaseerde keuzes gemaakt door BP
 - ⤴ time-sharing systems:
 - ⤴ verschillende gebruikers
 - ⤴ responsetime is cruciaal
- ⤴ **Beheer van in- en uitvoer:** Het BP bestuurd alle randapparaten
 - ⤴ BP houdt tabel bij van alle randapparaten
 - ⤴ klaar
 - ⤴ bezig
 - ⤴ defect
- ⤴ **Bestandenbeheer:** Het BP houdt gegevens en programma's van gebruikers bij en stelt ze op aanvraag ter beschikking
 - ⤴ bestand (file) = hoeveelheid informatie die samen hoort via een gebruikersvriendelijke naam
 - ⤴ klok : circuit dat op regelmatige tijdstippen "tikt".. DUH..
 - ⤴ kristaloscillator : zeer nauwkeurig
- ⤴ **Informatiebeheer:** Het BP houdt heel wat informatie bij en stelt deze op aanvraag ter beschikking.
 - ⤴ boekhouding
 - ⤴ facturatie

statistieken

4.9 Kosten/Baten aspecten(bij gebruik BP)

baten:

- correcte werking wordt verbeterd
- gebruikersvriendelijkheid verhoogt
- efficiënter gebruik

kosten:

- apparatuur
- programmatuur
- overhead aan tijd

rendement:

- 80% van de processortijd besteed aan BP
- 20% van de processortijd aan gebruikers programma's

- 80% van de apparatuur dient voor het beheer van die apparatuur
- 20% van de apparatuur dient voor het uitvoeren van bevelen

Tijdsgebruik gebruikerprogramma's

- 75% organisatie
- 25% effectief rekenwerk

Hoofdstuk 5: Computernetwerken (wegens de uitgetekende voorbeelden is het boek hierbij ook vaak vereist)

Hoe?

goedkope, krachtige computers meer en meer beschikbaar

- goedkope componenten

- hoge snelheid verbindingen

waarom?

- gemeenschappelijk gebruik van resources
- verhoogde betrouwbaarheid en beschikbaarheid
- kostenreductie
- snel en eenvoudig communicatiemiddel

via het netwerk kunnen duurdere componenten in de computerinfrastructuur voor alle gebruikers beschikbaar gesteld worden

- printers, scanners, etc.
- reserve componenten beschikbaar in geval van een panne
- servers + pc's = goedkoper dan 1 centrale computer
- computer netwerk = communicatie middel

5.1.2 Componenten in een computer netwerk;

communicatie netwerk

- communicatiediensten

gedistribueerd systeem

- onderling verbonden computersysteem
- een enkel virtueel systeem

lange afstandscommunicatienetwerk

- bestaat uit schakelaars
- bandbreedte : verschil tussen hoogste en laagste frequency

- coaxiale kabels/fiberoptics (glasvezel)

lokaal netwerk = analoog netwerk

5.1.3 schakeltechnieken:

- lijnschakelen
- boodschapschakelen
- pakketschakelen

lijnschakelen:

- opzetten van een verbinding → kanaal met vaste bandbreedte continu beschikbaar
- gebruiken van een verbinding
- verbreken van een verbinding

Probleem: inefficiënt gebruik gereserveerd kanaal

Boodschapschakelen;

- geen reservatie van kanalen
- store en forward mechanisme
- grotere vertraging t.o.v. lijnschakelen

Pakketschakelen:

- boodschappen worden in pakketten opgesplitst
- elk pakket word afzonderlijk verstuurd

Voordeel : communicatielijn niet heel lang bezet

5.1.4 Netwerkarchitectuur

problemen:

- bittransport moet betrouwbaarder zijn
- route voor elk pakket moet bepaald worden
- boodschappen splitsen en samenvoegen

protocol = dienst

- regels tussen 2 entiteiten bij onderlinge samenwerking

gelaagde architectuur

boodschappen, die transporten entiteiten uitwisselen, bestaan uit 2 delen (p14 lezen+bekijken!!)

1. gearceerde deel (proceslaag) : onbelangrijk
2. hoofding : aanvraag om dienst te verlenen

transportlaag geeft de volgende informatie aan de netwerklaag:

- boodschap zelf
- identificatie van de bestemming

OSI (Open System Intercommunication) (referentiemodel) (7 lagen):

1. Fysische laag: verantwoordelijk voor versturen bits over de kabel

- kabels/connectoren

2. Dataverbindingslaag

- creatie van foutenvrij communicatie kanaal
- detectie van fouten
- herkennen van pakketten

-flow control

3. netwerklaag

- verantwoordelijk voor het verkeer van pakketten doorheen een of meerdere netwerken
- routebepaling voor pakketten
- kostendoorrekening naar gebruikers
- onderling verbinden van netwerken

4. transportlaag

- eerste protocols tussen eindstelsel
- realiseert betrouwbare communicatie kanalen
- efficiënt gebruik van netwerken
- boodschappen opsplitsen en daarna samenvoegen
- controle van de boodschapstroom

5. sessielaag

- bied hulpmiddelen aan tussen 2 partijen via een kanaal
- organisatie van de dialoog
- synchronisatie punten creëren (hervatten dialoog mogelijk maken)

6. presentatielaag

- behoud van de betekenis van de uitgewisselde informatie
- keuze van een voorstellingswijze voor de gegevens op de kabel
- encryptie
- compressie

7. applicatielaag

protocols voor ;

- applicaties
- standaard applicaties
- deelproblemen in tal van applicaties

5.1.6 internetmodel;

internet modellen:

1. applicatielaag

2. transportlaag

3. netwerklaag

4. datalinklaag

5. fysische laag

5.2 Fysische laag

analoog : continu veranderbaar

digitaal : discreet , eindig aantal punten zonder tussenliggende punten

transmissiemedia:

- gevlochten koperparen → soms te grote vervorming
- coaxiale kabels (keuze tussen basisband of breedband)

digitale communicatie:

betrouwbaar bittransport niet gewaarborgd:

- signaal wordt verzwakt
- vervormd
- snelheid afhankelijk van frequentie (vertraging)
- ruis

modem : MODuleren- Demoduleren

3 mogelijk te wijzigen kenmerken

- amplitude: gebruik van verschillende spanningsniveaus
- frequency: verschillende tonen
- fase: signaal met andere fase

5.3 Dataverbindingslaag

- grote verwerkings verschillen mogelijk
- geen apparatuur beschikbaar om te synchroniseren

Taken dataverbindingslaag:

- Afbakenen van pakketten

-Detectie van fouten

controlebits : bits waarmee controller kan nagaan of alle bits correct ontvangen zijn

-Stroombeheersing(flow control)

opletten dat de snelle zender de trage ontvanger niet overspoelt

-Pakketlayout:

beginpatroon

Hoofding

data

controlebits

-eindpatroon

Algoritmen en protocols

Protocol 1: geen fouten, geen stroombeheersing

Protocol 2: geen fouten, stroombeheersing: de ontvanger geeft zender toestemming om volgend pakket te versturen (toelatingspakketten)

Protocol 3: stroombeheersing, fouten mogelijk:

2 soorten fouten:

- Pakket niet ontvangen

- Foutief herkend (door controlebits)

⇒ gebruik van waker bij het verloren gaan pakketten

Protocol 4: 1-bit vensterprotocol:

Gebruik van volgnummers in bevestigingen (tegen het kruisen van pakketten)

5.4 Lokale netwerk

5.4.1 Algemene kenmerken

Eenvoud;

- geen afzonderlijke schakelaars
- eenvoudig topologie (geen problemen met routebepaling)

Communicatiepaneel word gedeeld door alle computersytemen:

- kunnen allemaal pakketen versturen.
- Ontvangen alle pakketen

2 lagen

- LLC (Logical Link Control) : gemeenschappelijk voor alle lokale netwerken. Zorgt er voor dat dezelfde dienst word aangeboden op alle lokale netwerken van de netwerklaag

- MAC (Medium Access Control) : typisch voor ieder lokaal netwerk maar onbetrouwbaar transport van pakketten

Ethernet:

Sturen van pakket:

- elektrisch signaal plant zich voort naar beide uiteinden van de kabel

Bij gelijktijdig uitsturen van twee systemen:

- botsing van pakketten (kan gedetecteerd worden)

Wanneer mag een systeem sturen:

regel 1: sturen mag pas wanneer geen signaal opgevangen wordt

regel 2: uitzendende computer blijft luisteren terwijl hij stuur. Stopt met uitzenden bij botsing +stuur stoorsignaal

regel 3: na botsing, willekeurige tijd wachten vooraleer te hervatten

Naam principe: CSMA/CD : Carrier Sense Multiple Access Collision Detect

MA: alle systemen hebben gelijktijdig toegang

CS: computersystemen kunnen detecteren of de kabel in gebruik is

CD: systemen kunnen botsingen detecteren

Voordelen:

- lage belasting : snel starten

- hoge belasting : 90% gebruik van transmissiecapaciteit nog steeds mogelijk

nadelen

- geen waarborg dat pakketten effectief verstuurd kunnen worden door een systeem

- pakketten minimaal 64 bytes

pakket layout : 6 bytes

- hoofding

- controlesom

moderne bekabeling : 100-150m

switch: lijn per lijn buffer ruimte voor pakketten voorzien

Wireless Lan (WiFi)

- accesspoint: celcontrole

- ruggengraatnetwerk (backbone): verbindingen tussen accesspoints

CSMA/CD = ethernet

CSMA/CA = wifi (collision avoidance)

simultaneous send/receive

protocol:

1. check om te versturen (indien nodig, wachten)
2. versturen van pakket en return confirmation als controle bits juist zijn
3. repeat als botsing ontstaat

virtual carrier sense:

- station stuurt kleine aanvraag pakket (request to send)
- ontvanger antwoord (als medium vrij is)
- alle stations die deze berichten krijgen beschouwen de ontvanger als bezet

5.5 Netwerklaag:

twee types diensten:

1. virtuele verbinding

2. datagram dienst:

- geen garantie van aflevering
- volgorde van afleveren niet noodzakelijk hetzelfde als verzendings volgorde

Aan elk computersysteem word een adres toegekend door de beheerders van het netwerk

5.5.2 Protocol:

Pakket moet verstuurd kunnen worden in de hoofding van een pakket bevind zich:

- netwerk adres van de bestemming
- netwerk adres van de zender
- protocol van de transportlaag, waaraan data moet worden gegeven

5.5.3 routing

statische routing : routingstabellen worden op verschillende schakelaars geplaatst

routeringsprotocol : schakelaars wisselen onderling informatie uit over de aangesloten computersystemen

Netwerken verbinden:Internet

niet realistisch om aan te nemen dat alle computers op een netwerk aangesloten zullen worden

- bedrijven: prive en controle
- organisaties : bvb elk departement in KULeuven

de term "inter network" of internet verwijst naar een netwerk dat opgebouwd is door het verbinden van een aantal verschillende netwerken

WAN : lange-afstandsnetwerken

LAN : Lokale netwerken

5.6.1 Adressering: IP adressen= getal van 32 bits

5.6.2 Diensten en protocol

binnen het internet biedt de netwerklaag uitsluitend een datagram service aan

IP : protocol van de netwerk laag (Internet Protocol)

5.6.3 routing

d.m.v. routingstabel :hoe het verder sturen van pakketten

5.7 Transportlaag

TCP: transmission control protocol

UDP: Universal datagram protocol

TCP ondersteunt virtuele verbindingen

UDP ondersteunt een datagram dienst tussen de eindpunten (levert IP adres)

5.8 Naamdiensten

Domeinnamen

TLD's: top level domains.

✎ 2 letter landcodes

- ⤴ Generieke namen (verwijzen naar gelijkaardige organisaties), com,org,net
- ⤴ Namen die verwijzen naar organisaties binnen de US (edu, gov, mil)

Voor ieder top level domain is er een beheerder die de namen toekent en het beleid voor naamkeuze vastlegt.

Omzetting van domeinnamen naar IP-adressen

naam beheerder (name server)

- zet adresnamen om in IP adressen met behulp van een tabel
- zeer stabiel systeem

5.9 Toepassingen

5.9.1email : berichten lezen, versturen, organiseren, en bewaren

- heading (email adres, onderwerp, etc.)
- boodschap

protocols (email)

- SMTP
- POP
- IMAP

URL = Uniform Resource Allocator

HTML = HyperText Markup Language