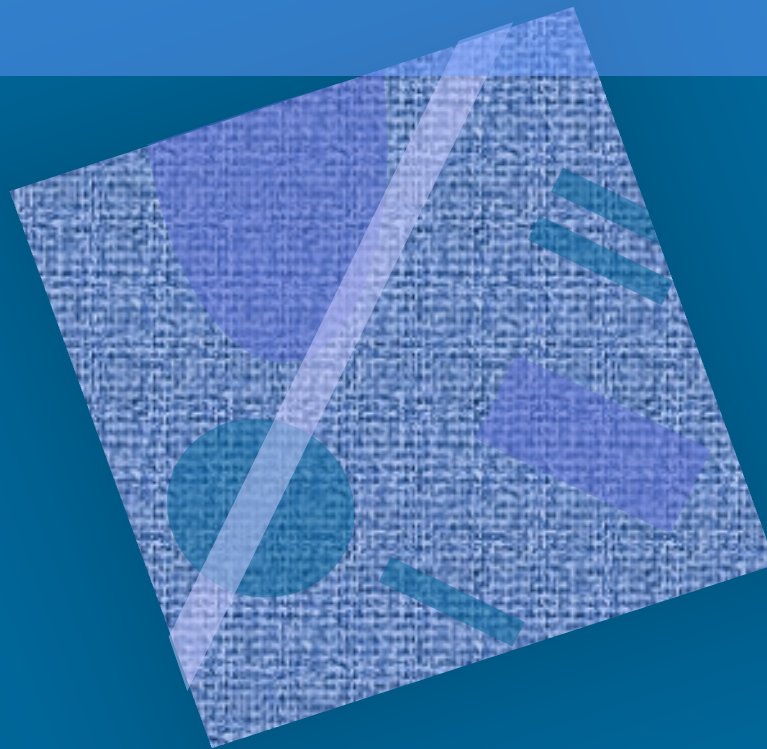


Luento 6 (verkkoluento 7)

# Tiedon muuttumattomuuden tarkistus Järjestelmän sisäinen muisti

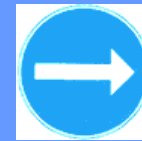


Pariteetti, Hamming-koodi  
Välimuisti, muisti

# Tiedon tarkistus

- Tiedon oikeellisuutta ei voi tarkistaa yleisessä tapauksessa
- Laitteistovirheitä voidaan havaita ja joskus automaattisesti korjata
  - Bitti voi muuttua muistissa tai tiedon siirrossa
    - muistipiirissä voi olla vika (staattinen vika)
    - sopiva alkeishiukkanen voi muuttaa bitin tiedonsiirron aikana (transientti vika)
  - Vian takia tieto muuttuu virheelliseksi (virhe)
  - Korjaamattomasta virheestä voi aiheutua häiriö
- Tietokannan eheys on eri asia!

Lisää  
tietoa?



Tieto-  
kanta  
kurssit

# Tiedon muuttumattomuus

- Virheitä tapahtuu
- Otetaan mukaan ylimääräisiä bittejä, joiden avulla virheitä voidaan havaita ja ehkä myös korjata
- Järjestelmä suorittaa tarkistukset automaattisesti joko laitteistotasolla tai ohjelmiston avulla

# Bittitason tarkistukset

- Muistipiirit, levyt, väylät, tiedonsiirrot Hetu: 120464-121C  
(merkkejä, ei bittejä)
- Monenko bitin muuttuminen havaitaan? Hetu: 1
- Monenko bitin muuttuminen voidaan automaattisesti korjata? Hetu: 0
- Havaitsemiseen ja/tai korjaamiseen tarvitaan enemmän (ylimääräisiä) bittejä
  - lisämuistitilan tai levytilan tarve? Hetu: +10%
  - lisäpiuhojen tarve väylällä?
- Tarkistukset/korjaukset Hetu: ohjelmistotasolla  
laitteisto- vai ohjelmistotasolla?

# Pariteettibitti

- Yksi ylimääräinen bitti per tietoalkio
  - sana, tavu, tietoliikennepaketti
- Parillinen (pariton) pariteetti: 1-bittien kokonaislukumäärä on aina parillinen (pariton)
- Havaitsee: 1 bitti
- Korjaa: 0 bittiä
- Esimerkki (parillinen pariteetti)

0010 001 0

1000 1101 1111 001 1

par.tilantarve: 12.5%

6.25%

# Suomalainen IBAN

Havaitsee: Kaikki yhden merkin virheet  
Useimmat kahden merkin virheet  
Kaikki kahden merkin vaihtumiset  
Useimmat muut virheet

Pankkitilin numero

500015-123

50001500000123 FI00

50001500000123 15 18 00

50001500000123 15 18 00 mod 97 = 61

98-61 = 37

FI37 5000 1500 0001 23

# Hamming etäisyys (R. Hamming 1950)

- Montako bittiä jossain koodijärjestelmässä (esim ISO Latin-1) esitetyllä koodilla (esim. 'A' = 0x41 = 0100 0001) täytyy muuttua, että se muuttuu johonkin toiseen (mihin tahansa) lailliseen koodiin.

'A' = 0x41 = 0100 0001

2 bittiä

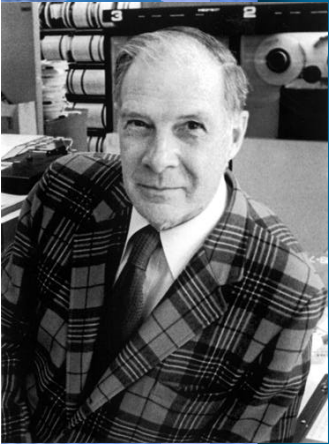
'B' = 0x42 = 0100 0010

1 bittiä

'C' = 0x43 = 0100 0011

- ISO Latin-1:n Hamming etäisyys: 1
- Pariteettibitin kanssa Hamming etäisyys: 2
  - mikä todennäköisyys 2 bitin (vs. 1 bitin) virheeseen?
  - riittävän pieni?

$$\text{Prob}\{ \text{"2 bitin virhe"} \} = (\text{Prob}\{ \text{"1 bitin virhe"} \})^2$$



# Virheen korjaava Hamming koodi (R. Hamming 1950)

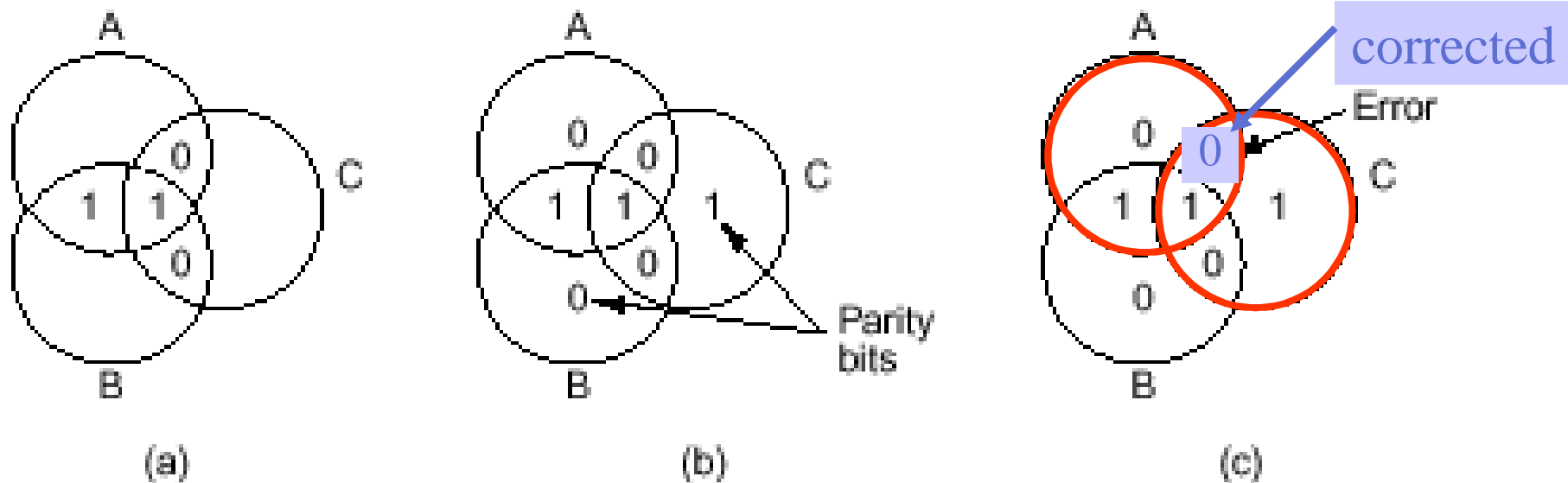


Figure 2-14. (a) Encoding of 1100. (b) Even parity added. (c) Error in AC.

[Tane99]

(a) Kukin databitti (4 kpl) kuuluu erilaisiin pariteettijoukkoihin (3 kpl)

(b) Tarvitaan 3 ”ylimääräistä” bittiä!

(c) Joukot A ja C havaitsevat virheen ja siten paikallistavat virheellisen bitin

Täsmälleen 1 databitti identifioituu kerrallaan!

entä jos virhe pariteettibitissä?



# Virheen korjaava Hamming koodi

Data: oikein 100 1100 virheellinen 110 1100 (parillinen pariteetti)  
Bitti nro: 765 4321 765 4321

Pariteettibitti 1 tarkistaa bittejä 1, 3, 5, 7

Pariteettibitti 2 tarkistaa bittejä 2, 3, 6, 7

Pariteettibitti 4 tarkistaa bittejä 4, 5, 6, 7

Tapahtuu virhe: bitti 6 muuttuu (flips)

Pariteettibitti 2 tarkistaa bittejä 2, 3, 6, 7: VIRHE

Pariteettibitti 4 tarkistaa bittejä 4, 5, 6, 7: VIRHE

$2+4 = 6 \Rightarrow$  korjaa bitti nro 6

1 = 00**1**  
2 = 0**1**0  
3 = 0**1****1**  
4 = **1**00  
5 = **1**0**1**  
6 = **1****1**0  
7 = **1****1****1**

# CRC - Cyclic Redundancy Code

- Tiedonsiirrossa käytetty tarkistusmenetelmä
- Tarkistussumma (16 bittiä) isolle tietojoukolle
  - laske  $CRC = f(\text{viesti}) \% 2^{16}$  (ota 16 viimeistä bittiä)
  - lähetä viesti ja CRC
  - vastaanota viesti ja CRC
  - laske CRC ja tarkista, oliko se sama kuin viestissä
  - jos pielessä, niin pyydä uudelleenlähetytä

## CRC-CCITT CRCs detect:

All single- and double-bit errors

All errors of an odd number of bits

All error bursts of 16 bits or less

In summary, 99.998% of all errors

# Virheiden tarkistusmenetelmien käyttöalueet

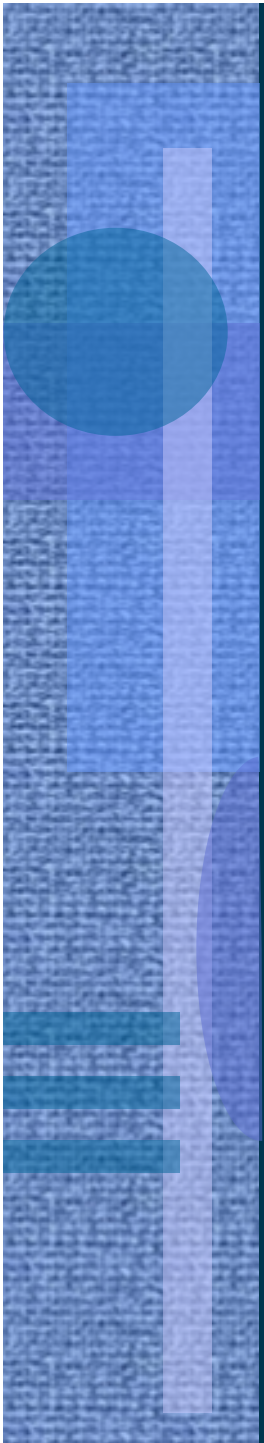
- Mitä lähempänä suoritinta, sitä tärkeämpää tiedon oikeellisuus on
- Sisäinen väylä, muistiväylä
  - virheet lennossa korjaava Hamming koodi
- Paikallisverkko
  - uudelleenlähetyksen vaativa CRC
  - kun tulee virheitä, niin niitä tulee yleensä paljon
    - Hamming koodi ei riitä kuitenkaan
    - pariteettibitti päästää läpi (esim.) 2 virheen paketit

# Laitteiden monistaminen

- Monta muistipiiriä tai levyä, samat tiedot monistettu
- Monta suoritinta, samat käskyjen suoritukset monistettu
- Monta laitteistoa, samat ohjelmat monistettu
  - äänestysmenettely: enemmistö voittaa
  - monimutkainen, hidas?
  - virheelliseksi havaittu laitteisto suljetaan pois häiriköimästä automaattisesti?
- Eri tai saman tyyppiset laitteistot, samankaltaiset ohjelmat
  - samat speksit, samat syötteet, eri ohjelmoijat

“Four of the five computers (IBM AP-101) on the Columbia ran identical software and compared results with each other before giving the go-ahead to take a specific action. The fifth computer (also IBM AP-101) ran a different version of the software and was used only if the others failed.”

<http://www.hq.nasa.gov/office/pao/History/computers/contents.html>



6.11.2012

Copyright 2012 Teemu Kerola

13

# Välimuisti (cache)

- Ongelma: keskusmuisti on aika kaukana suorittimesta

rekisterin viittausaika: X  
muistin viittausaika: 10X

- Ratkaisu: välimuisti lähelle suoritinta

- pidetään siellä kopioita viime aikoina viitatuista keskusmuistin alueista

välimuistin viittausaika: 2X

- Jokainen muistiviite on nyt seuraavanlainen

- jos data ei ole välimuistissa, niin hae se sinne
  - suoritin odottaa tällä aikaa, laitteistototeutus!
- tee viittaus dataan (käskyyn) välimuistissa
- (talleta muutettu tieto keskusmuistiin)

# Muistin toteutus

- Eri teknologioita eri tasoisiin muisteihin
- RAM - Random-Access Semiconductor Memory
  - anna osoite ja lue/kirjoita signaali
  - mistä vaan voi lukea/kirjoittaa samassa ajassa
  - virta pois  $\Rightarrow$  tiedot häviävät (volatile memory)

Kaikki nykyiset keskusmuistit ovat ”random access”

# RAM:n kaksi eri teknologiaa

- DRAM: dynaaminen RAM, halvempi, hitaampi, tietoja pitää virkistää vähän väliä (esim. joka 2 ms)
  - tavallinen keskusmuisti (1975-..) useimmissa koneissa
  - toteutettu kondensaattoreilla, jotka ”vuotavat” ...
- SRAM: staattinen RAM, kalliimpi (~10-20x), nopeampi (~10-50x), vie tilaa enemmän, ei vaadi tietojen virkistämistä
  - välimuisti useimmissa koneissa
  - muisti superkoneissa (esim. Cray C-90)
  - toteutettu samantlaisilla logiikkaportteilla (gate) kuin prosessorikin



# Nykyaikaisen keskusmuistin termejä

- SDRAM (synchronous dynamic random access memory)
  - Kello tahdittaa siirrot väylälle
  - Sisäinen puskuri, useita muistioperaatioita jonossa
- DDR (double data rate) SDRAM
  - Yhden kellopulssin aikana kaksi datasiirtoa, sekä nousevalla että laskevalla pulssin osalla

Kello, väylä -  
Lisää tietoa?



Tietokoneen  
rakenne

# ROM teknologia

- ROM - Read-Only Memory
  - tieto säilyy virran katkettua (non-volatile)
  - Voi käytön aikana vain lukea, ei voi kirjoittaa
    - esim. järjestelmän alustustiedot (BIOS)
  - kirjoitus lastun valmistusaikana, Mask-ROM
    - ei enää käytössä
  - huono puoli: kerran väärin, aina väärin (ehkä...)
  - päivitys: laita valmistajalta saatu uusi lastu paikalleen
  - tietoa voi lukea mistä vain samassa ajassa (random access)
  - yleensä hitaampi kuin RAM (~10x)

# Kirjoitettavia ROM-muisteja

- PROM - Programmable ROM
  - kerran kirjoitettava
  - tiedon päivitys: ”polta” tiedot tyhjään PROM:iin
- EPROM - Erasable PROM
  - tietoja ei voi päivittää sana kerrallaan
  - vanhat tiedot voidaan (kaikki!) poistaa 20 min. UV-säteilyllä, jonka jälkeen päivitettyt tiedot voidaan ladata
- EEPROM - Electronically Erasable PROM
  - tietojen pyyhkiminen tavukohtaisesti elektronisesti
- FLASH EEPROM memory
  - tietojen pyyhkiminen nopeasti kerralla elektronisesti
  - normaalijännitteellä, kaikki tai lohko kerrallaan
  - nopeampi kuin EEPROM
- Flash Drive, SSD – Solid State Disk
  - Flash-muistilla toteutettu iso muisti
  - Käyttöjärjestelmä näkee kovalevynä

read-  
mostly  
memory

HISTORIAA

NYKYAIKA

# SSD

- Flash-muisti paketoituna erilaisiin koteloihin
- Rajallinen määrä (100 000?) kirjoituskertoja samaan muistipaikkaan
  - Kirjoitettavia fyysisiä muistialueita kierrätetään
  - Vara-alueita valmiiksi allokoituna
- Kaksi teknologiaa
  - nor: luku/kirj sana kerrallaan, hitaampi
  - nand: luku/kirj lohko kerrallaan, nopeampi

# SSD vs. kovalevy (2012)

- I/O per sek: SSD 100x nopeampi
- Kaistanleveys: SSD vähän nopeampi
- Saantiaika: SSD 10-100x nopeampi
- Kapasiteetti: kovalevy 10x isompi
  
- SSD kestää tärinää, ei kulu, vaatii vähemmän virtaa, vie vähemmän tilaa, on kalliimpi (per MB)

# Muisti- hier- arkia

- Flash?
- SSD?
- Levypal-  
velin?
- Pilvi?

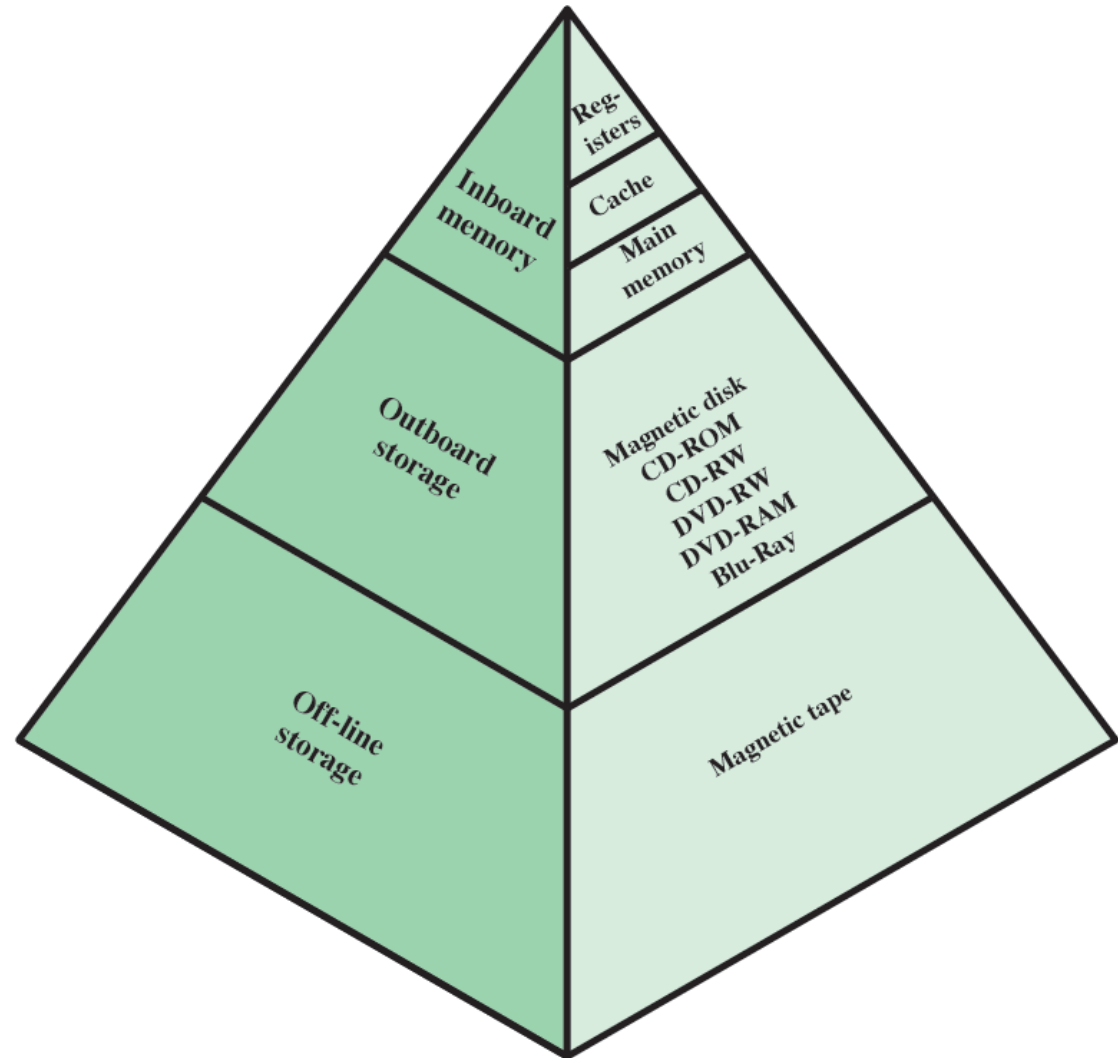


Figure 4.1 The Memory Hierarchy

# -- Luennon 3 loppu --

- Akustinen elohopeaviiveputki
  - kvartsikide muutti sähkövirran akustiseksi signaaliksi (ja päin vastoin) pietsosähköisen ilmiön avulla
  - 1000 bittiä per 1.45m putki
  - Luku: odota, että oikea luku ehtii putken päähän, lue se ja kirjoita se takaisin putken toiseen päähän
  - W. Shockley & J.P. Eckert, 1946
  - M. Wilkes, EDSAC – Electronic Delay Storage Automatic Calculator, 1949

Eckert



Wilkes