

Jakso 7

**Tiedon muuttumattomuuden tarkistus**  
**Järjestelmän sisäinen muisti**



- Pariteetti
- Hamming-koodi
- Välimuisti
- Tavallinen muisti
- Muistien historiaa

28.5.2002 Copyright Teemu Kerola, K2002 1

### Tiedon tarkistus <sup>(4)</sup>

- Tiedon oikeellisuutta ei voi tarkistaa yleisessä tapauksessa
- Laitteistovirheitä voidaan havaita ja joskus automaattisesti korjata
  - bitti voi muuttua muistissa tai tiedonsiirrossa
    - muistipiirissä voi olla vika (staattinen vika)
    - sopiva alkeishiukkanen voi muuttaa bitin tiedonsiirron aikana (transientti virhe)
- Tietokannan eheys on eri asia!

Lisää tietoa? Tietokanta kurssit

28.5.2002 Copyright Teemu Kerola, K2002 2

### Tiedon muuttumattomuus <sup>(2)</sup>

- Perusidea: otetaan mukaan ylimääräisiä bittejä, joiden avulla virheitä voidaan havaita ja ehkä myös korjata
- Järjestelmä suorittaa tarkistukset automaattisesti joko laitteistotasolla tai ohjelmiston avulla

28.5.2002 Copyright Teemu Kerola, K2002 3

### Esimerkki ohjelmistotason tarkistusmerkistä <sup>(2)</sup>

- Henkilötunnus: 120464-121C
 

$120464121 \cdot 31 = 12$

0123456789 ABCDEFHIJKLMNPRSTUVWXY  
10 11 12

- Tarkistusmerkin avulla voidaan tarkistaa, ettei mikään yksi merkki ole väärin
  - havaitsee yhden merkin virheen 120464-123C
  - virhettä ei voi automaattisesti korjata! 120464-123E

Miksi? 120464-123E

28.5.2002 Copyright Teemu Kerola, K2002 4

### Bittitaso tarkistukset <sup>(7)</sup>

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

28.5.2002 Copyright Teemu Kerola, K2002 5

### Pariteettibitti <sup>(9)</sup>

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

$0010\ 001\ 0 \leftarrow$        $1000\ 1101\ 1111\ 001\ 1 \leftarrow$

28.5.2002 Copyright Teemu Kerola, K2002 6

### Hamming etäisyys <sup>(3)</sup>

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

'A' = 0x41 = 0100 0001

'B' = 0x42 = 0100 0010

'C' = 0x43 = 0100 0011

} 2 bittii

} 1 bittii

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

28.5.2002 Copyright Teemu Kerola, K2002 7

### Hamming koodi <sup>(2)</sup>

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

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

(b) tarvitaan 3 ylimääräistä bittii

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

Täsmälleen 1 databitti identifioituu kerrallaan

28.5.2002 Copyright Teemu Kerola, K2002 8

### Hamming koodi <sup>(9)</sup>

- Käytetään useampia pariteettibittejä
- Havaitsee: kahden bitin muuttuminen
- Korjaa: yhden bitin muuttuminen

Data: 100 1100      4 bittii dataa, 3 pariteettibittii

Bitti nro: 765 4321

Kaikki bitit nro 2<sup>i</sup> ovat pariteettibittejä, muut ovat databittejä (numerot alkavat 1:stä)

Kutakin data-bittii n tarkistavat ne pariteettibitit joiden summana n voidaan esittää. Parillinen pariteetti.

6 = 4 + 2 ⇒ databittii 6 tarkistavat par. bitit 4 ja 2

28.5.2002 Copyright Teemu Kerola, K2002 9

### Virheen korjaava Hamming koodi <sup>(8)</sup>

Data: 100 1100      110 1100

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 (flip)

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

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

2 + 4 = 6 ⇒ korjaa bitti nro 6

28.5.2002 Copyright Teemu Kerola, K2002 10

### Virheen korjaava Hamming koodi (ilman   )

Data: 100 1100      110 1100

Bitti nro: 765 4321      765 4321

Pariteettibitti 1 tarkistaa bittejä 1, 3, 5, 7

Pariteetti

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

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

2 + 4 = 6 ⇒ korjaa bitti nro 6

28.5.2002 Copyright Teemu Kerola, K2002 11

### CRC - Cyclic Redundancy Code <sup>(7)</sup>

- Tiedonsiirrossa käytetty tarkistusmenetelmä
- Tarkistussumma (16 bittii) isolle tietojoukolle
  - laske CRC = f (viesti \* 2<sup>16</sup>) % (X<sup>16</sup> + X<sup>15</sup> + X<sup>2</sup> + 1) (huom! polynomijako)
  - lähetä viesti ja CRC
  - vastaanota viesti ja CRC
  - laske CRC ja tarkista oliko oikein (pitäisi olla 0!)
  - jos p

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

28.5.2002 Copyright Teemu Kerola, K2002 12

### Virheiden tarkistusmenetelmien käyttöalueet

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

28.5.2002 Copyright Teemu Kerola, K2002 13

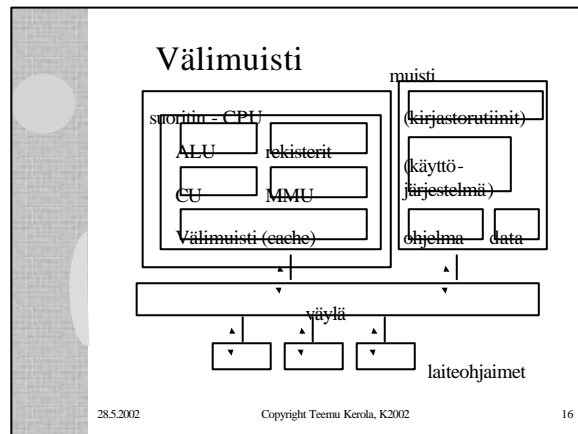
### Laitteiden monistaminen <sup>(6)</sup>

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

Lentokoneet, avaruussukkula, ydinvoimala, ...

28.5.2002 Copyright Teemu Kerola, K2002 14

28.5.2002 Copyright Teemu Kerola, K2002 15



### Välimuisti (cache) <sup>(3)</sup>

- Ongelma: keskusmuisti on aika kaukana suorittimesta
  - rekisterin viittausaika: X
  - muistin viittausaika: 10X
- Ratkaisu: välimuisti lähelle suoritinta
  - pidetään siellä (kopioida) 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
  - tee viittaus dataan (käskyyn) välimuistissa
  - (talleta muutettu tieto keskusmuistiin)

28.5.2002 Copyright Teemu Kerola, K2002 17

### Välimuisti

- Tuntumaton suorittimelle Fig. 4.13 [Stal99]
  - jos viitattu tieto ei saatavilla, niin suoritin vain odottaa muutaman kellopulssin ajan...
- Toteutettu usein nopeammalla teknologialla kuin keskusmuisti (tavallinen muisti)
- Toteutettu usein samalla mikropiirillä kuin suoritin
- Silti iso aikaero: välimuisti 2X, muisti 10X
  - ratkaisu: monitasoiset välimuistit: L1, L2, L3
  - L2 eri piirillä, mutta isompi kuin L1
- TTK-91 koneessa ei ole välimuistia

28.5.2002 Copyright Teemu Kerola, K2002 18

### Miksi välimuisti toimii? <sup>(5)</sup>

- Paikallisuusperiaate: tietynä aikavälillä muistiviitteet kohdistuvat vain koko muistin **pieneen osajoukkoon** (locality principle)
- Ajallinen paikallisuus: on todennäköistä, että **vähän aikaa sitten** viitattuun tietoon viitataan uudelleen kohta (temporal locality)
- Alueellinen paikallisuus: on todennäköistä, että vähän aikaa sitten viitattun tiedon **välille** viitataan lähitulevaisuudessa (spatial locality)

345 23 71 8 305 63 91 2

28.5.2002 Copyright Teemu Kerola, K2002 19

### Miten välimuisti toimii? <sup>(3)</sup>

Fig. 4.13 [Stal99]

- Välimuistissa on useita keskusmuistin lohkoja – esim. 4 sanaa / lohko (cache block / cache line)
- Kun tulee "välimuistihuti" (viitattu data ei välimuistissa), niin koko tuon datan sisältämä lohko ("rivi") kopioidaan muistista välimuistiin
- Uusille lohkoille tehdään tilaa poistamalla vanhoja lohkoja, joihin ei toivottavasti heti viitata – ne lohkot, joihin on kirjoitettu, täytyy kopioida muistiin

28.5.2002 Copyright Teemu Kerola, K2002 20

### Välimuistin ominaisuuksia <sup>(5)</sup>

- Koko – isommasta löytyy enemmän dataa, **mutta se voi olla hitaampi** (Fig. 4.13 [Stal99])
- Kuvausfunktio – miten löytää data välimuistista?
- Poistoalgoritmi – mikä lohko poistetaan seuraavaksi?
- Kirjoituspolitiikka – miten käsitellä WRITE käskyjä?
- Lohkon (rivin) koko – monta pientä: hyvä ajalliselle paikallisuudelle – muutama iso: hyvä alueelliselle paikallisuudelle

28.5.2002 Copyright Teemu Kerola, K2002 21

### Muistin toteutus <sup>(6)</sup>

- Eri teknologioita eri tasoihin muisteihin
- RAM - Random-Access Semiconductor Memory – anna osoite ja lue/kirjoita signaali – mistä vaan voi lukea/kirjoittaa samassa ajassa (volatile memory)

Huom: kaikki nykyiset muistit ovat "random access"

28.5.2002 Copyright Teemu Kerola, K2002 22

### RAM:n kaksi eri teknologiaa <sup>(2)</sup>

- DRAM: dynaaminen RAM, halvempi, hitaampi, tietoja pitää virkistää vähän väliä (esim. joka 2 ms) – tavallinen keskusmuisti (1975-...) useimmissa koneissa – toteutettu kondensaattoreilla, jotka "vuotavat" ...
- SRAM: static RAM, kalliimpi (~20x), nopeampi (~10x), ei vaadi tietojen virkistämistä – välimuisti useimmissa koneissa – muisti superkoneissa (esim. Cray C-90) – toteutettu samanlaisilla logiikkaportteilla (gate) kuin prosessorikin – CMOS valmistusteknologia (Complementary Metal Oxide Semiconductor)

28.5.2002 Copyright Teemu Kerola, K2002 23

### Muistin toteutus <sup>(7)</sup>

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

28.5.2002 Copyright Teemu Kerola, K2002 24

### Kirjoitettavia ROM-muisteja <sup>(6)</sup>

- PROM - Programmable ROM
  - kerran kirjoitettava
  - tiedon päivitys: "polta" tiedot tyhjiin PROM:iin
- EPROM - Erasable PROM
  - tietoja ei voi päivittää sana kerrallaan
  - vanhat tiedot voidaan (kaikki!) poistaa 20 min. UV-säteilyllä, jonka jälkeen päivitetty tiedot voidaan ladata
- EEPROM - Electronically Erasable PROM
  - tietojen pyyhkiminen tavukohtaisesti elektronisesti
- FLASH EEPROM memory
  - tietojen pyyhkiminen nopeasti kerralla elektronisesti
  - normaalijännitteellä
  - nopeampi kuin EEPROM BIOS, CIH-virus


read-mostly memory

28.5.2002 Copyright Teemu Kerola, K2002 25

### Muistien historiaa

- Rumpumuisti
  - 1939, ABC, Atanassoff-Berry Computer, Iowa State College.
    - lähinnä laskin ...
  - 1951, Aiken Mark III
    - erilliset rumpumuistit koodille ja datalle

Artyzbasheff  
Time cover  
1951



30 numeroa á 50 bittia

28.5.2002 Copyright Teemu Kerola, K2002 26

### Muistien historiaa

- Williams Tube
  - 1946, Williams & Kilburn
  - katodisädeputki
  - ensimmäinen suuri "RAM" muisti



Tom Kilburn holding a Cathode Ray Tube

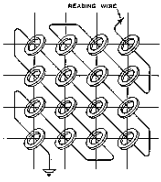
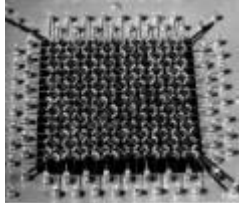


Storing 2048 bits on a CRT in 1947

28.5.2002 Copyright Teemu Kerola, K2002 27

### Muistien historiaa


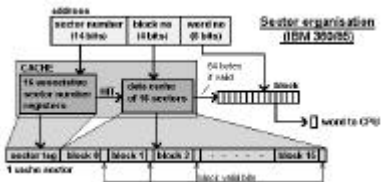
- Ferriittirengas (core) teknologia
  - 1952, Jay Forrester & Bob Everett, MIT (Whirlwind)
  - tieto säilyy ilman virtaa
  - ei häiriinny säteilystä (avaruus, sotilasteknologia)
  - 1955, valtaa markkinat Williams Tube'ita

28.5.2002 Copyright Teemu Kerola, K2002 28

### Muistien historiaa



- Välimuisti (1965, Maurice Wilkes)
  - IBM S/360 Model 85
    - 1968
    - 256 lohkoa á 64 tavua

28.5.2002 Copyright Teemu Kerola, K2002 29

### Muistien historiaa

- DRAM (1966, Robert Dennard, IBM)
  - Intel 1103 (1970)
    - John Reed
    - 1 Kbit
  - valtaa markkinat ferriittirengasmuisteilta 1972
- SRAM (1970, Fairchild Corp)

Robert Dennard

28.5.2002 Copyright Teemu Kerola, K2002 30


### Muistien historiaa

- PROM
  - ???
- EPROM
  - 1971, Dov Frohman, Intel 1701
- EEPROM,
  - 1980, Intel 2816
- Flash EEPROM
  - 1984, Lai et al, Intel

28.5.2002 Copyright Teemu Kerola, K2002 31


### Muistien historiaa

- OROM - optical ROM
  - 1990, James Russell (Russell keksi myös CD-ROM:n)
  - 1998, Wond-OROM-a
    - 128 MB/kortti plus lukulaite
    - ei liikkuvia osia
    - sama nopeus kuin CD-ROM:lla (siis aika hidas!)
    - pieni virrankulutus
    - sopii kannettaviin laitteisiin



28.5.2002 Copyright Teemu Kerola, K2002 32

--  
**Jakson  
7  
loppu**  
--



Whirlwind Project,  
MIT, 1946- ...

28.5.2002 Copyright Teemu Kerola, K2002 33