Luento 7
Tiedon esitysmuodot (jatk)
Tiedon muuttumattomuuden tarkistus
Järjestelmän sisäinen muisti

Ohjelman esitysmuoto
Rakenteellinen tieto
Pariteetti
Hamming-koodi
Välimuisti
Tavallinen muisti
Muistien historiaa

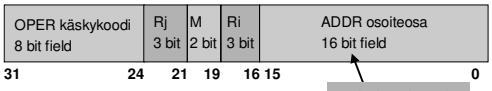
Konekäskyjen esitysmuoto muistissa (4)

- Konekohtainen, jokaisella omansa
- Käskyt ovat 1 tai useamman tavun mittaisia
  - SPARC, kaikki käskyt: 1 sana eli 4 tavua
  - PowerPC, kaikki käskyt: 1 sana eli 4 tavua
  - Pentium II: 1-16 tavua, paljon variaatioita
- Käskyillä on yksi tai useampi muoto, kussakin tietty määrä erilaisia kenttiä
  - opcode, Ri, Rj, Rk, osoitusmoodi
  - pitkä tai lyhyt vakio

TTK-91, kaikki käskyt: 1 sana, 1 muoto

TTK-91 konekäskyn rakenne

- Käskyn esitys bittitasolla on aina:



muistiosoite tai (pienehkö) vakio

Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri (R0 ≡ 0)

M = muistinoutojen määrä toiseen operandiin (addressing mode)
(ennen mahdollista muistiin talletusta)
00 eli 0 kpl, välitön osoitus (STORE: suora osoitus)
01 eli 1 kpl, suora osoitus (STORE: epäsuora osoit.)
10 eli 2 kpl, epäsuora osoitus (STORE: epäkelpo arvo)
(11 eli 3 kpl, epäkelpo arvo → poikkeustilanne)

Konekäskyn operandit ja tulos

- Tulos: rekisteri Rj
  - paitsi WRITE- tai PUSH-käskyissä muistipaikan sisältö
- Ensimmäinen operandi: rekisteri Rj
- Toinen operandi
  - laske ensin arvo Ri+ADDR ja käytä sitä sellaisenaan tai käytä sitä muistisoitteena
  - arvo: Ri + ADDR
  - muistipaikan M[Ri+ADDR] sisältö
  - muistipaikan M[M[Ri+ADDR]] sisältö

jos Ri = R0, niin pelkkä ADDR

Konekielen tiedon osoitusmoodit

Taulukkojen esitysmuoto

- Peräkkäisrakenteena, kuten esimerkit aikaisemmin
- Riveittäin tai sarakeittain
- Ei omia konekäskyjä, manipulointi aliohjelmilla tai loopeilla (paitsi ns. vektorikoneet, joilla on omia konekäskyjä vektorioperaatioita varten)
- Indeksoitu tiedonosoitusmoodi tukee 1-ulotteisten taulukoiden käyttöä

Tietueiden esitysmuoto

- Peräkkäisrakenteena
- Osoite on jonkin osoitemuuttujan arvo
- Ei omia konekäskyjä, manipulointi aliohjelmilla tai kääntäjän generoimien vakiolisäysten avulla
- Indeksoitu tiedonosoitusmoodi tukee tietueiden käyttöä

Olioiden esitysmuoto

- Kuten tietueet, yleensä varattu keosta (heap)
- Useat olion kentistä sisältävät vuorostaan osoitteen keosta suoritusaikana varattuun toiseen olioon
- Metodit ovat aliohjelmien osoitteita
- Ei omia konekäskyjä, manipulointi aliohjelmilla

Tiedon muuttumattomuus (2)

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

Esimerkki ohjelmistotason tarkistusmerkistä (2)

- Henkilötunnus: 120464-121C

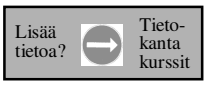
120464121 % 31 = 12

0123456789 ABCDEFHJKLMNPRSTUVWXY

- Tarkistusmerkin avulla voidaan tarkistaa, että mikään yksi merkki ei ole väärin
  - havaitsee yhden merkin virheen
  - virhettä ei voi automaattisesti korjata!! Miksi?

Tiedon tarkistus (4)

- 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 virhe)
- Tietokannan eheys on eri asia!



Bittitason tarkistukset (5)

- Muistipiirit, levyt, väylät, tiedonsiirrot — 120464-121C
- 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 laitteisto- vai SW-tasolla? — Hetu: ohjelmistotasolla

Pariteettibitti (9)

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

0010 001 0 1000 1101 1111 001 1

Hamming etäisyys (3)

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

´A' = 0x41 = 0100 0001
´B' = 0x42 = 0100 0010 — 2 bittiä
´C' = 0x43 = 0100 0011 — 1 bittiä

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

Hamming koodi (4)

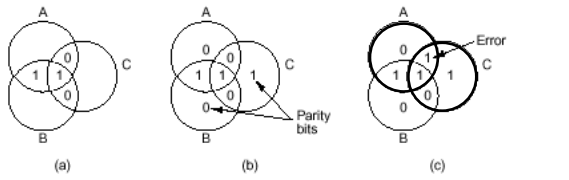


Figure 2-14. (a) Encoding of 1100. (b) Even parity added. (c) Error in *AC*.

[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!

Hamming koodi (9)

- Käytetään useampia pariteettibittejä
- Havaitsee: 2:n bitin muuttuminen
- Korjaa: 1 bitin muuttuminen

Data + parit. 100 1100
Bitti nro: 765 4321

4 bittiä dataa, 3 pariteettibittiä

Kaikki bitit nro 2^i ovat pariteettibittejä, muut ovat databittejä (numerot alkavat 1:stä)

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

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

Virheen korjaava Hamming koodi (8)

(ECC)

Data: 100 1100 (100) 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

1 = 001
2 = 010
3 = 011
4 = 100
5 = 101
6 = 110
7 = 111

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 ⇒ korjaa bitti nro 6

Virheen korjaava Hamming koodi (ilman animaatioita)

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

1 = 001
2 = 010
3 = 011
4 = 100
5 = 101
6 = 110
7 = 111

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 ⇒ korjaa bitti nro 6

CRC - Cyclic Redundancy Code (7)

- Tiedonsiirrossa käytetty tarkistusmenetelmä
- Tarkistussumma (16 bittiä) isolle tietojoukolle
  - laske CRC = f (viesti) % 2^{16}
  - lähetä viesti ja CRC
  - vastaanota viesti ja CRC
  - laske CRC ja tarkista oliko oikein
  - jos pielessä, niin pyydä uudelleenlähetystä

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

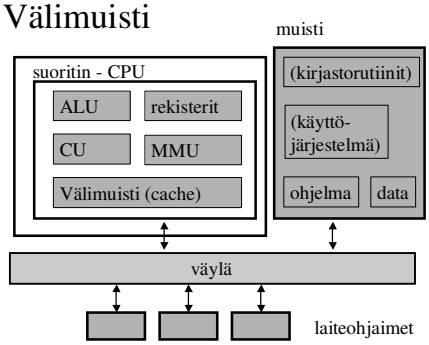
- Mitä lähempänä suoritinta, sitä tärkeämpää tiedon oikeellisuus on
- Sisäinen väylä, muistiväylä
  - virheet korjaava Hamming koodi
- Paikallisverkko
  - uudelleenlähetyksen vaativa CRC
  - kun tulee virheitä, niin niitä tulee yleensä paljon
    - Hamming koodi ei riitä kuitenkaan
    - pariteettibitti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

Laitteiden monistaminen (6)

- 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)
- Äänestysmenettely: enemmistö voittaa
  - monimutkainen, hidas?
  - virheelliseksi havaittu laitteisto suljetaan pois häiriköimästä automaattisesti?

Lentokoneet, avaruussukkula, ydinvoimala, …

Välimuisti



Välimuisti (cache) (3)

- 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)

Välimuisti (6)

- 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 nykyään usein samalla mikropiirillä kuin suoritin
- Silti iso aikaero: välimuisti 2X, muisti 10X
- TTK-91 koneessa ei ole välimuistia

Lisää tietoa? ⇒ käyttö-järjestelmä kurssit

07/11/2001 Copyright Teemu Kerola 2001 25

Muistin toteutus (6)

- 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 ⇒ tiedot häviävät — (volatile memory)

Huom: kaikki nykyiset muistit ovat ”random access”

07/11/2001 Copyright Teemu Kerola 2001 27

RAM:n kaksi eri teknologiaa (2)

- 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 (~10-20x), nopeampi (~10x), ei vaadi tietojen virkistämistä
  - välimuisti useimmissa koneissa
  - muisti superkoneissa (esim. Cray C-90)
  - toteutettu samanlaisilla logiikkaporteilla (gate) kuin prosessorikin
  - CMOS valmistusteknologia (Complementary Metal Oxide Semiconductor)

07/11/2001 Copyright Teemu Kerola 2001 28

ROM teknologia (8)

- ROM - Read-Only Memory
  - tieto säilyy virran katkettua — (non-volatile)
  - 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)

07/11/2001 Copyright Teemu Kerola 2001 29

Kirjoitettavia ROM-muisteja (6)

- 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 päivitetyt 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

07/11/2001 Copyright Teemu Kerola 2001 30

07/11/2001 Copyright Teemu Kerola 2001 31

Muistien historiaa

- Kondensaattorirumpu
  - 1939, ABC, Atanasoff-Berry Computer, Iowa State College.
    - lähinnä laskin, ei toiminut
  - kondensattorit pyörivän rummun pinnalla

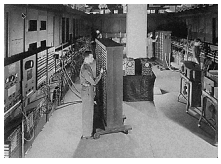
32 numeroa á 50 bittiä

Artzybasheff *Time* cover 1951

07/11/2001 Copyright Teemu Kerola 2001 32

Muistien historiaa

- Elektroniputki
  - logiikka, muisti
- ENIAC, 1945
  - Electronic Numerical Integrator and Computer
  - J.W. Mauchly, J.P. Eckert, J. von Neumann
  - 18,000 elektr. putkea
  - 70,000 vastusta
  - 5 milj. juotettua liitosta
  - tykinammusten ja pommien radanlaskenta



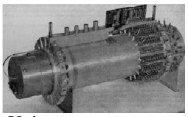
07/11/2001 Copyright Teemu Kerola 2001 33

07/11/2001 Copyright Teemu Kerola 2001 34

Muistien historiaa

- Akustinen elohopeaviiveputki
  - kvartsikide muutti sähkövirran akustiseksi signaaliksi (ja päin vastoin) pietsosähköisen ilmiön avulla
  - 1000 bittiä per 1.45m putki
  - W. Shockley & J.P. Eckert, 1946
  - M. Wilkes, EDSAC – Electronic Delay Storage Automatic Calculator, 1949
  - Mauchly & Eckert, UNIVAC, 1951 (ens. kaupallinen tietokone USA:ssa)

Wilkes

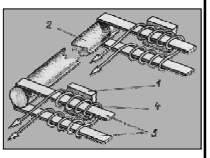


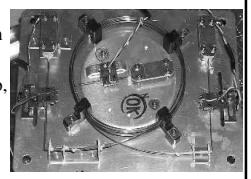
Univac memory

07/11/2001 Copyright Teemu Kerola 2001 35

Muistien historiaa

- Akustinen nikkeliviiveputki
  - magneettikenttä aiheuttaa pituussuuntaisen muodonmuutoksen tankoon ja siten vääntöpulssin johtimeen
  - vähän ajan päästä muutos tuntuu toisessa päässä ja aiheuttaa magneettikentän muutoksen siellä
  - Hazeltine Electronic Corp, 1950?
  - Elliot 401, 1953
  - Canon 141 laskin, 1969



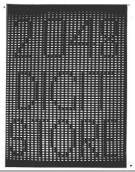


07/11/2001 Copyright Teemu Kerola 2001 36

Muistien historiaa

- Williams Tube
  - 1946, Williams & Kilburn
  - katodisädeputki
  - ensimmäinen suuri ”RAM” muisti
  - kallis: $1000 / 1 kk / putki
  - Small Scale Experimental Machine (”Baby”), 1947
  - Ferranti Mark I, ensimmäinen yleiskäyttöinen kaupallinen tietokone, 1951 (10000 bitin muisti)

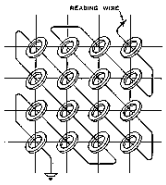
Tom Kilburn holding a Cathode Ray Tube

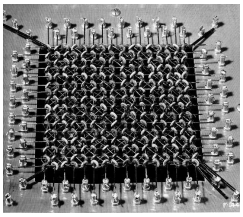


Storing 2048 bits on a CRT in 1947

Muistien historiaa

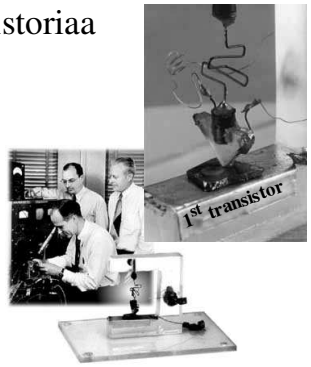
- Ferriittirengas (core) teknologia
  - 1952, Jay Forrester & Bob Everett, MIT (Whirlwind)
  - tieto säilyy ilman virtaa
  - ei häiriinny säteilystä (avaruus, sotilasteknologia)
  - 1955, valtaa muistimarkkinat Williams Tube'lta





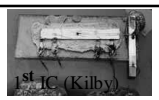
Muistien historiaa

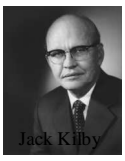
- Transistori
  - J. Bardeen, W.B. Shockley ja W. Brattain, ATT Bell Labs, 1948
    - Nobel 1956
  - MIT TX-0, 1957, ensimmäinen transistoroitu tietokone

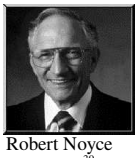


Muistien historiaa

- Integroitu piiri (ei enää johtoja)
  - Jack Kilby, Texas Instruments, 1958
    - Nobel 2000
    - ensimmäinen käsikäyttöinen laskin
  - Robert Noyce, Fairchild Semiconductor, 1959
    - "planar process" valmistusmenet.
    - perusti Intelin G. Mooren kanssa
  - IBM S/360, 1964

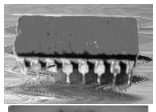



Jack Kilby


Robert Noyce

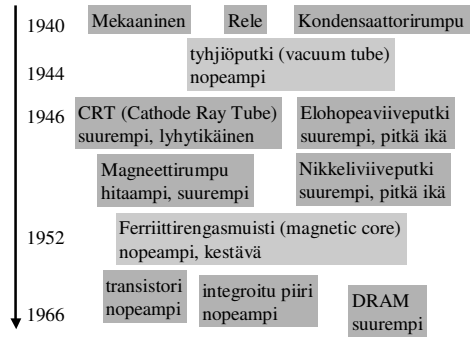
Muistien historiaa

- DRAM
  - Robert Dennard, IBM, 1966
    - (US) National Medal of Technology 1988
  - Intel 1103 (1970)
    - John Reed
    - 1 Kbit
  - valtaa markkinat ferriitti-rengasmuisteilta 1972
- SRAM
  - 1970, Fairchild Corp

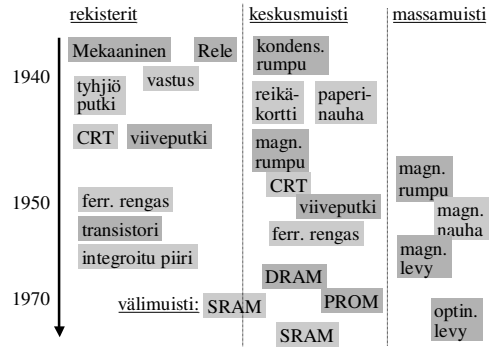


Robert Dennard

Muistiteknologian historiaa (7)

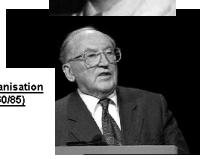


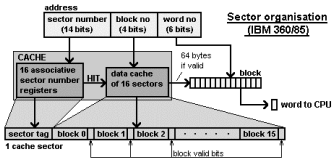
Muistiteknologian käyttöhistoriaa



Muistien historiaa

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





Muistien historiaa

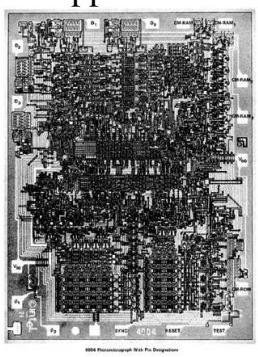
- PROM
  - ???
- EPROM
  - 1971, Dov Frohman, Intel 1701
- EEPROM,
  - 1980, Intel 2816
- Flash EEPROM
  - 1984, Fujio Masuoka, Toshiba

Fujio Masuoka

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

-- Luennon 7 loppu --

Intel 4004, 1971

- Faggin, Hoff, Mazor
- Ens. suoritin lastulla 3x4 mm, $200
- 2300 transistoria
- 4 bitin sana
- Laskinta varten
- Sama laskentateho kuin Eniacilla (18000 tyhjiöputkea)