

Luento 7

Tiedon esitysmuodot (jatk)

Tiedon muuttumattomuuden tarkistus

Järjestelmän sisäinen muisti

Pariteetti

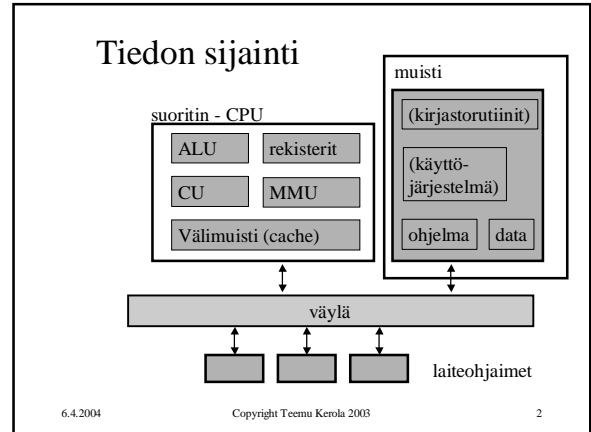
Hamming-koodi

Välimuisti

Tavallinen muisti

Muistien historiaa

6.4.2004
Copyright Teemu Kerola 2003
1



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

6.4.2004
Copyright Teemu Kerola 2003
3

TTK-91 konekäskyn rakenne

• Käskyn esitys bittitasolla on aina:

OPER käskykoodi 8 bit field	Rj 3 bit	M 2 bit	Ri 3 bit	ADDR osoiteosa 16 bit field
31	24	21	19	16 15
				0

Rj = käskyn ensimmäinen operandi
 Ri = indeksirekisteri (R0 ≡ 0)
 M = muistinoutojen määrä toiseen operandiin (ennen mahdollista muistiin talletusta)

muistiosoite tai (pienehkö) vakio (addressing mode)

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)

6.4.2004
Copyright Teemu Kerola 2003
4

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 muistiosoiteena

jos Ri = R0, niin pelkkä ADDR

Konekielen tiedon osoitusmoodit

}

- arvo: Ri + ADDR
- muistipaikan M[Ri+ADDR] sisältö
- muistipaikan M[M[Ri+ADDR]] sisältö

6.4.2004
Copyright Teemu Kerola 2003
5

Taulukkojen esitysmuoto

- Peräkkäisrakenteena, kuten esimerkit aikaisemmin
- Riveittäin tai sarakeittain
- Ei omia konekäskyjä, manipulointi aliohjelmilla tai loopeilla
 - Poikkeus: vektorisuorittimet, joilla
 - vektorirekisterit (esim. 64 liukulukua) tavallisten rekistereiden lisäksi
 - omia konekäskyjä vektoriopeaatioita varten
- Indeksoitu tiedonosoitusmoodi tukee 1-ulotteisten taulukoiden käyttöä

6.4.2004
Copyright Teemu Kerola 2003
6

Tietueiden esitysmuoto

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

6.4.2004

Copyright Teemu Kerola 2003

7

Olioiden esitysmuoto

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

6.4.2004

Copyright Teemu Kerola 2003

8

Tiedon muuttumattomuus

- Muuttumattomuus vaikuttaa itsestäänselvyydeltä, mutta ...
- Laitteisto voi voittaa ja aiheuttaa muutoksia tallennettuihin tietoihin
 - Bitti vaihtaa tilaa (0 ->1 tai 1 ->0)
- Ohjelmassa voi olla virhe, joka aiheuttaa tiedon tallennuksen väärään paikkaan
 - Koko muistipaikan sisältö vaihtuu

6.4.2004

Copyright Teemu Kerola 2003

10

Entä jos bitti kääntyy?

- Bitti vaihtaa tilaa valvomattomasti
- Bitti voi muuttua muistissa tai tiedon siirrossa sekä väylällä että koneiden välillä
- Syitä
 - sähköinen komponentti tai johdin vaurioituu
 - lämpöherkkä komponentti (lämmentyään lakkaa toimimasta)
 - jännitepiikki
 - elektromagneettinen säteily
 - irtonainen johdin (huono liitos)

6.4.2004

Copyright Teemu Kerola 2003

11

Tiedon tarkistus ⁽³⁾

- Tiedon oikeellisuutta ei voi tarkistaa yleisessä tapauksessa
- Laitteistovirheitä voidaan havaita ja joskus automaattisesti korjata
 - muistipiirissä voi olla vika (staattinen vika)
 - sopiva alkeishiukkanen voi muuttaa bitin tiedonsiirron aikana (transienti virhe)
 - korjaamattomasta virheestä voi aiheutua häiriö
- Tietokannan eheys on eri asia!

Lisää tietoa?



Tietokanta kurssit

6.4.2004

Copyright Teemu Kerola 2003

12

Tiedon muuttumattomuuden tarkistus ⁽²⁾

- Perusidea: otetaan mukaan ylimääräisiä bittejä, joiden avulla virheitä voidaan havaita ja ehkä myös korjata
- Järjestelmä suorittaa tarkistukset automaattisesti joko laitteistotasolla tai ohjelmiston avulla
- Ei tepsii ohjelmavirheiden aiheuttamiin ei-toivottuihin muutoksiin, koska yleensä tarkistusbitit ovat kunnossa

6.4.2004 Copyright Teemu Kerola 2003 13

Esimerkki ohjelmistotason tarkistusmerkistä ⁽²⁾

- Henkilötunnus: 120464-121C

$120464121 \% 31 = 12$

0123456789 ABCDEFHJKLMNPRSTUVWXY
 10 11 12 ↑ ↑ ↑↑ 30

- Tarkistusmerkin avulla voidaan tarkistaa, että mikään yksittäinen merkki ei ole väärin
 - havaitsee yhden merkin virheen 120464-123C
 - virhettä ei voi automaattisesti korjata!! Miksi?
 - välimerkkiä (-, +, A) ei tarkasteta lainkaan!

6.4.2004 Copyright Teemu Kerola 2003 14

Bittitason tarkistukset ⁽⁵⁾

- Muisti- ja 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?
 - lisäpiuhojen tarve väylällä? Hetu: +10%
- Tarkistukset/korjaukset laitteisto- vai SW-tasolla? Hetu: ohjelmistotasolla

6.4.2004 Copyright Teemu Kerola 2003 15

Ylimääräisten bittien käyttö

- Yksi ylimääräinen bitti
 - Voidaan havaita yhden bitin virhe
 - Pariteettibitiksi
- Useampia ylimääräisiä bittejä
 - Voidaan havaita useamman bitin virhe tai korjata yhden tai muutaman bitin virhe
 - Hamming-koodaus pienehkölle bittiryhmälle (esim. tavu)
 - CRC (Cyclic Redundancy Code) isommalle määrälle

6.4.2004 Copyright Teemu Kerola 2003 16

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

6.4.2004 Copyright Teemu Kerola 2003 17

Hamming etäisyys ⁽³⁾

- 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 kodiin.

$'A' = 0x41 = 0100\ 0001$

$'B' = 0x42 = 0100\ 0010$

$'C' = 0x43 = 0100\ 0011$

- ISO Latin-1:n Hamming etäisyys: 1
- Pariteettibitin kanssa Hamming etäisyys: 2
 - mikä todennäköisyys 2 bitin (vs. 1 bitin) virheeseen?
 - riittävän pieni? $(\text{Prob}\{\text{"yhdessä bitin virhe"}\})^2$

6.4.2004 Copyright Teemu Kerola 2003 18

Hamming etäisyys

- Kertoo mahdollisuudet bittivirheiden havaitsemiseen tai korjaamiseen
- Merk. d joukon pienin Hamming etäisyys
- m -bittinen virhe havaitaan, kun $d \geq m+1$
- m -bittinen virhe voidaan korjata, kun $d \geq 2m+1$

6.4.2004 Copyright Teemu Kerola 2003 19

Virheen korjaava Hamming koodi⁽⁵⁾

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?

6.4.2004 Copyright Teemu Kerola 2003

Hamming koodi⁽⁹⁾

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

Data + parit. **100 1100** 4 bittiä dataa, 3 pariteettibittiä

Bitti nro: 765 4321

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

Kutakin databittiä n tarkistavat ne pariteettibitit joiden summana n voidaan esittää. Parillinen pariteetti.

$6 = 4 + 2 \Rightarrow$ databittiä 6 tarkistavat par. bitit 4 ja 2

6.4.2004 Copyright Teemu Kerola 2003 21

Virheen korjaava Hamming koodi⁽⁸⁾ (ECC)

Data: **100 1100** (parillinen pariteetti)

Bitti nro: 765 4321 765 4321 421

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

6.4.2004 Copyright Teemu Kerola 2003 22

Virheen korjaava Hamming koodi (ilman animaatioita)

Data: **100 1100** **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

6.4.2004 Copyright Teemu Kerola 2003 23

Hamming koodi

Data Bits	Single-Error Correction		Single-Error Corr & Double-Error Detection	
	Check Bits	%increase	Check Bits	%increase
8	4	50	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91

Table 5.2 [Sta03]

6.4.2004 Copyright Teemu Kerola 2003 24

Hamming: 8 databittiä

Positio	12	11	10	9	8	7	6	5	4	3	2	1
Bin. lukuna	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bitti	D8	D7	D6	D5		D4	D3	D2		D1		
Tark. bitti					C8				C4		C2	C1
Tallettu	0	0	1	1	0	1	0	0	1	1	1	1
					D3 muuttuu 0->1							
Noudettu	0	0	1	1	0	1	1	0	1	1	1	1
Tark. bitti					0				0		0	1

Fig 5.10 [Sta03]
25

6.4.2004

Copyright Teemu Kerola 2003

CRC - Cyclic Redundancy Code ⁽⁷⁾

- Tiedonsiirrossa käytetty tarkistusmenetelmä
- Tarkistussumma (16 bittiä) isolle tietojoukalle
 - laske CRC = f (viesti) % 2¹⁶ (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 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

6.4.2004

Copyright Teemu Kerola 2003

26

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

6.4.2004

Copyright Teemu Kerola 2003

27

Laitteiden monistaminen ⁽⁴⁾

- Monta muistipiiriä tai levyä, samat tiedot monistettu
- Monta suoritinta, samat käskyjen suoritukset monistettu
- Monta laitteistoa, samat ohjelmat monistettu
 - äänestysmenetely: enemmistö voittaa
 - monimutkainen, hidas?
 - virheelliseksi havaittu laitteisto suljetaan pois häiriköimästä automaattisesti?
- Eri- tai samantyyppiset 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>

6.4.2004

Copyright Teemu Kerola 2003

28

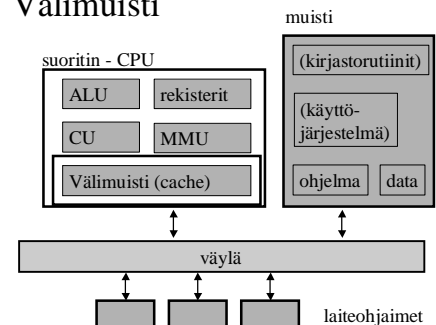
- Pariteetti
- Hamming-koodi
- Välimuisti
- Tavallinen muisti
- Muistien historiaa

6.4.2004

Copyright Teemu Kerola 2003

29

Välimuisti



6.4.2004

Copyright Teemu Kerola 2003

30

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 (käskyn) välimuistissa
 - (talleta muutettu tieto keskusmuistiin)


6.4.2004 Copyright Teemu Kerola 2003 31

Välimuisti ⁽⁶⁾

Fig. 4.3 [Stal03]
Fig. 4.13 [Stal99]

- Tuntumaton suorittimelle
 - 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?  tietokoneen rakenne -kurssi

Lisää tietoa?  käyttöjärjestelmäkurssit

6.4.2004 Copyright Teemu Kerola 2003 32

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

6.4.2004 Copyright Teemu Kerola 2003 33

Muistin toteutus ⁽⁶⁾

- Eri teknologioita eri tasoihin 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"

6.4.2004 Copyright Teemu Kerola 2003 34

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: staattinen RAM, kalliimpi (~10-20x), 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)

6.4.2004 Copyright Teemu Kerola 2003 35

ROM teknologia ⁽⁸⁾

- ROM - Read-Only Memory
 - tieto säilyy virran katketta (non-volatile)
 - voi käytössä vain lukea, ei voi kirjoittaa
 - esim. järjestelmän alustustiedot (BIOS)
 - kirjoitus lastun valmistusaikana, Mask-ROM
 - ei enää käytössä
 - 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)

6.4.2004 Copyright Teemu Kerola 2003 36

Kirjoitettavia ROM-muisteja ⁽⁶⁾

- 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, kaikki tai lohko kerrallaan
 - nopeampi kuin EEPROM

read-mostly memory


6.4.2004 Copyright Teemu Kerola 2003 37

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


6.4.2004 Copyright Teemu Kerola 2003 38

Muistien historiaa

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



32 numeroa á 50 bittiä


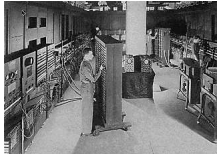


Artzybasheff
Time cover
1951

6.4.2004 Copyright Teemu Kerola 2003 39

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

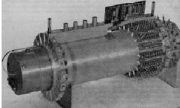
6.4.2004 Copyright Teemu Kerola 2003 40

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

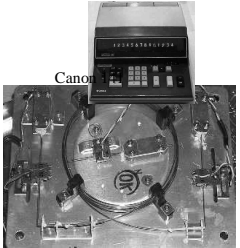


Univac memory

6.4.2004 Copyright Teemu Kerola 2003 41

Muistien historiaa

- Akustinen nikkeliviiveputki
 - magneettikenttä aiheuttaa pituussuuntaisen muodonmuutoksen tankoon ja siten vääntöpulssein johtimeen
 - vähän ajan päästä muutos tuntuu toisessa päässä ja aiheuttaa magneettikentän muutoksen siellä
 - Hazeltine Electronic Corp, 1950?
 - Elliot 401, 1953
 - Canon 141 laskin, 1969
 - Sony Sobax 2500, 1970

Canon

6.4.2004 Copyright Teemu Kerola 2003 42

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)
 - akustisen viiveteknologian kilpailija



Tom Kilburn holding a Cathode Ray Tube

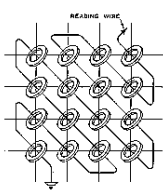
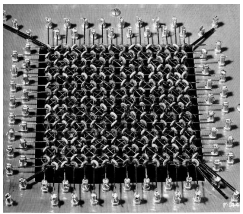


Storing 2048 bits on a CRT in 1947

6.4.2004 Copyright Teemu Kerola 2003 43

Muistien historiaa

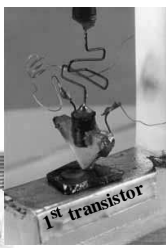
- Ferriittirengas (core) teknologia
 - 1952, Jay Forrester & Bob Everett, MIT (Whirlwind)
 - tieto säilyy ilman virtaa
 - ei häiriinny säteilystä (avaruus, sotilasteknologia)
 - 1955, valtaa muistimarkkinat Williams Tube'ltä


6.4.2004 Copyright Teemu Kerola 2003 44

Muistien historiaa

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




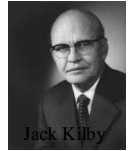

1st transistor



6.4.2004 Copyright Teemu Kerola 2003 45

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






http://www.tomorrowtoday.com/content/newsbriefs/HIRESwafer.htm

6.4.2004 Copyright Teemu Kerola 2003 46

Muistien historiaa

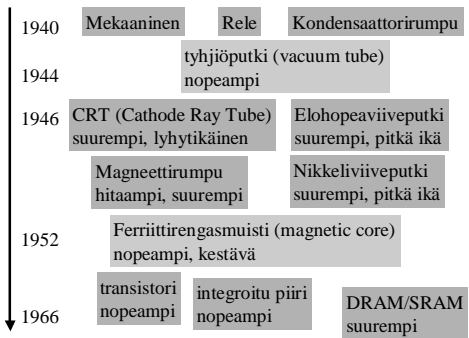
- DRAM
 - Robert Dennard, IBM, 1966
 - (US) National Medal of Technology 1988
 - Intel 1103 (1970)
 - John Reed
 - 1 Kbit
 - valtaa markkinat ferriittirengasmuisteilta 1972
- SRAM
 - 1970, Fairchild Corp



Robert Dennard

6.4.2004 Copyright Teemu Kerola 2003 47

Muistitekologian historiaa (7)



1940 Mekaaninen Rele Kondensaattorirumpu

1944 tyhjiöputki (vacuum tube) nopeampi

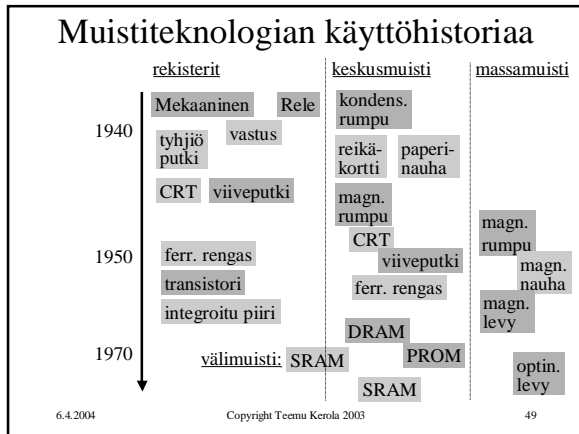
1946 CRT (Cathode Ray Tube) suurempi, lyhytikäinen Elohopeaviiveputki suurempi, pitkä ikä

Magneettirumpu hitaampi, suurempi Nikkeliviiveputki suurempi, pitkä ikä

1952 Ferriittirengasmuisti (magnetic core) nopeampi, kestävä

1966 transistori nopeampi integroitu piiri nopeampi DRAM/SRAM suurempi

6.4.2004 Copyright Teemu Kerola 2003 48



Muistien historiaa

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

address
sector number (14 bits) block no. (11 bits) word no. (10 bits) Sector organisation (IBM S/360/85)

16-associative sector number registers data cache of 16 sectors 64 bytes if valid block word to CPU

sector fan block 0 block 1 block 2 ... block 15
1 cache sector block valid bits

6.4.2004 Copyright Teemu Kerola 2003 50

Muistien historiaa

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

Fujio Masuoka

6.4.2004 Copyright Teemu Kerola 2003 51

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
 - hävisi Flash-muisteille

6.4.2004 Copyright Teemu Kerola 2003 52

-- 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)

6.4.2004 Copyright Teemu Kerola 2003 53