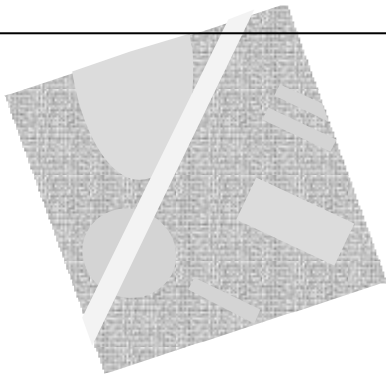


## Luento 6

# Tiedon esitysmuodot



- Lukujärjestelmät
- Kokonaisluvut
- Liukuluvut
- Merkit, merkkijonot
- Totuusarvot
- Kuvat, äänet, hajut(?)

22.5.2002Copyright Teemu Kerola, K20021

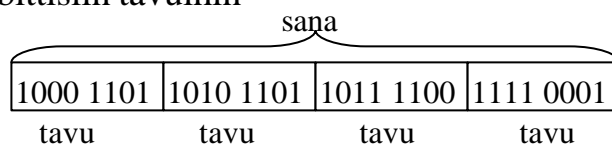
## Tiedon tyypit

- **Kommunikointi ihmisen kanssa**
  - kuva, ääni, merkit, ...
- **Laitteiston sisäinen talletus**
  - kuvaformatit, ääniformatit, pakkausstandardit, ...
  - kokonaisluvut, liukuluvut, merkit, merkistöt
  - ohjelmat
- **Suorittimen omana lajinaan ymmärtämät tyypit**
  - on olemassa konekäskyjä tälle tietotyyppille
  - kokonaisluvut
  - liukuluvut (useimmat suorittimet nykyään)
  - totuusarvot (jotkut suorittimet)
  - merkit (jotkut suorittimet)
  - konekäskyt

22.5.2002Copyright Teemu Kerola, K20022

## Tiedon esitys laitteistossa <sup>(4)</sup>

- Kaikki tieto koneessa on binääribitteinä (0 tai 1)
  - binäärijärjestelmän numerot: 0, 1
  - helppo toteuttaa piireillä
  - helppo suunnitella logiikkaa Boolean algebran avulla
- Muisti jaettu tasapituisiin sanoihin (word)
  - sana = word = 32 bittiä (16 bittiä, 64 bittiä, ...)
- Usein sana on jaettu tasapituisiin (byte) 8-bittisiin tavuihin



22.5.2002

Copyright Teemu Kerola, K2002

3

## Tiedon esitys laitteistossa <sup>(4)</sup>

- Tietoa siirretään muistiväylää pitkin sanoina
  - joskus useampi kuin yksi sana kerrallaan (lohko)
- Prosessorin rekisterit ovat yleensä yhden tai kahden sanan mittaisia
  - 1 sana: kokonaisluku, pieni liukuluku
  - 1 sana: 1 merkki tai 4 merkkiä
  - 2 sanaa: pitkä kokonaisluku, iso liukuluku

22.5.2002

Copyright Teemu Kerola, K2002

4

## Tiedon esitys <sup>(7)</sup>

- Kysymys: miten esittää eri tyyppisiä tietoja?
- Vastaus: koodataan ne biteiksi
  - kaikki tieto on koneessa bitteinä
- Kaikelle käsitellylle tiedolle on omat koodausmenetelmänsä
  - kaikkia koodausmenetelmiä ei ole standardoitu
  - samalla tietotyypille voi olla useita koodausmenetelmiä
    - kokonaisluvut, liukuluvut, merkit, merkkijonot, kuvat, ...
  - ongelma: ymmärtävätkö koneet toisiaan?
    - tiedon esitysmuotoa voidaan joutua muuttamaan kun tietoa siirretään koneelta toiselle

22.5.2002

Copyright Teemu Kerola, K2002

5

## Suorittimen ymmärtämä tieto <sup>(9)</sup>

- Kaikki tieto koneessa on koodattuna biteiksi
- Muistissa voidaan esittää kaikki tieto sovitulla esitystavalla (koodauksella)
- Suoritin osaa tehdä operaatioita joillakin esitystavoilla koodatuille tiedoille
  - kokonaisluvut ja liukuluvut (aina)
  - totuusarvot, merkit ja merkkijonot (joskus)
  - kuvat ja äänet (ei yleensä ellei erikoistunut suoritin)
  - hajut (ei vielä)
- Muiden tietojen käsittely tapahtuu ohjelmallisesti
  - esim. merkkejä voidaan käsitellä kokonaislukuoperaatioilla ja aliohjelmilla

TTK-91: kokonaisluvut
--------------------------

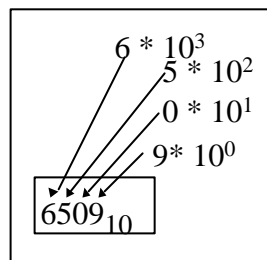
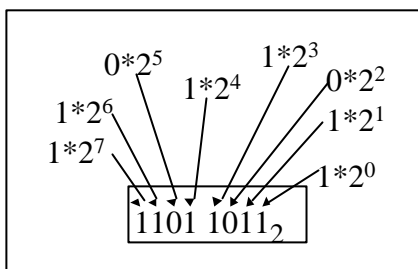
22.5.2002

Copyright Teemu Kerola, K2002

6

## Binäärijärjestelmä (2)

- Kantaluku 2, numerot 0 ja 1
  - numeroiden painoarvot oikealta vasemmalle:  
 $1=2^0, 2=2^1, 4=2^2, 8=2^3, 16=2^4, 32=2^5, \dots$
  - kymmenjärjestelmässä painoarvot ovat  
 $1=10^0, 10=10^1, 100=10^2, 1000=10^3, \dots$

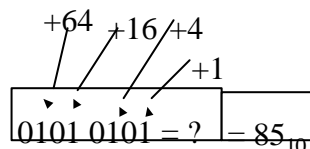
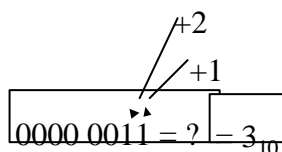
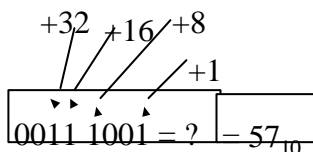


22.5.2002

Copyright Teemu Kerola, K2002

7

## Binäärilukuesimerkkejä



22.5.2002

Copyright Teemu Kerola, K2002

8

## Bonäärilukujen laskutoimitukset

+	0	1
0	0	1
1	1	10

	1	1	
1	0	1	1
0	1	1	0
=====			
1	1	1	0
0	1		

45  
+ 12  
----  
57

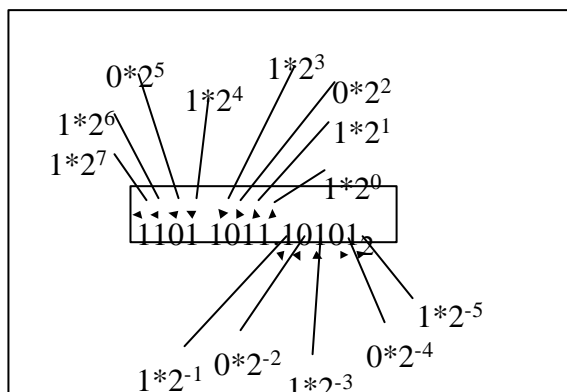
*	0	1
0	0	0
1	0	1

		1	0	1
*		1	0	1
=====				
		1	0	1
+	1	0	1	
=====				
	1	1	0	0
	1	0	0	1

5  
\* 5  
  
  
  
= 25

## Binääripiste (2)

- Binääriluvuilla voi olla myös binääriosa (vrt. desimaaliosa)



## Binääripiste-esimerkkejä

$$\begin{array}{r}
 +4 \quad +1 \quad +0.5 = 2^{-1} \\
 \quad \quad \quad +0.125 = 2^{-3} \\
 \hline
 0101.101 = ? \quad = 5.625_{10}
 \end{array}$$

$$\begin{array}{r}
 +4 \quad +2 \quad +0.125 = 2^{-3} \\
 \quad \quad \quad +0.0625 = 2^{-4} \\
 \hline
 0110.0011 = ? \quad = 6.1875_{10}
 \end{array}$$

22.5.2002

Copyright Teemu Kerola, K2002

11

## Muunnokset lukujärjestelmien välillä <sup>(5)</sup>

- 2-järjestelmä  $\Rightarrow$  10-järjestelmä
  - esitettiin jo edellä
- 10-järjestelmä  $\Rightarrow$  2-järjestelmä
  - kokonaisosa ja desimaaliosa erikseen
  - kokonaisosa:
    - jaa toistuvasti 2:lla, kunnes 0 jäljellä
    - ota jakojännökset käännetyssä järjestyksessä

22.5.2002

Copyright Teemu Kerola, K2002

12

## 10-järj $\Rightarrow$ 2-järj kokonaislukuesimerkki <sup>(11)</sup>

$57_{10} = ?$

$57/2 = 28$  jää 1

$28/2 = 14$  jää 0

$14/2 = 7$  jää 0

$7/2 = 3$  jää 1

$3/2 = 1$  jää 1

$1/2 = 0$  jää 1

loppu

$= 11\ 1001_2$

$= 0011\ 1001_2$

22.5.2002
Copyright Teemu Kerola, K2002
13

## 10-järj $\Rightarrow$ 2-järj desimaaliosa $\Rightarrow$ binääriosia

- Kerrotaan toistuvasti desimaaliluvun desimaaliosa 2:lla, kunnes
  - desimaaliosa = 0 (tarkka binääriesitys)
  - tarpeeksi numeroita haluttuun tarkkuuteen
- Tulos saadaan ottamalla saatujen desimaalilukujen kokonaisosat (0 tai 1) lasketussa järjestyksessä

22.5.2002
Copyright Teemu Kerola, K2002
14

## 10-järj $\Rightarrow$ 2-järj binääriosaesimerkki

$0.1875_{10} = ?$

$2 * 0.1875 = 0.375 = 0 + 0.375$

$2 * 0.375 = 0.75 = 0 + 0.75$

$2 * 0.75 = 1.5 = 1 + 0.5$

$2 * 0.5 = 1.0 = 1 + 0.0$

$2 * 0.0 = 0.0 = 0 + 0.0$

$= 0.0011_2$

$\downarrow$  loppu

$= 0.00110000000000000000_2$

22.5.2002
Copyright Teemu Kerola, K2002
15

## Heksadesimaaliesitys <sup>(6)</sup>

- Binäärilukuja käyttö on tarpeellista, mutta niitä on ikävä kirjoittaa
  - liikaa numeroita
- Kirjoitetaan ne 16-järjestelmässä eli heksadesimaalijärjestelmässä
- 4 bittiä vastaa aina yhtä 16-järjestelmän numeroa
- Yksi 16-järjestelmän numero vastaa aina 4 bittiä
- 16-järjestelmän numerot ovat:  
 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E ja F
 

A	B	C	D	E	ja	F
10	11	12	13	14		15

22.5.2002
Copyright Teemu Kerola, K2002
16



## Heksadesimaaliesimerkkejä <sup>(11)</sup>

binääri: 0100 0111 1001 1010 1111

16-järj: 4 7 9 A F = 479AF<sub>16</sub>

= 0004 79AF<sub>16</sub> = 0x 479AF

16-järj: 120ADF<sub>16</sub> 1 2 0 A D F

binääri: 0001 0010 0000 1010 1110 1111

22.5.2002 Copyright Teemu Kerola, K2002 17

## Oktaaliesimerkkejä <sup>(13)</sup>

Numerot: 0, 1, 2, 3, 4, 5, 6, 7

binääri: 01 000 111 100 110 101 111

8-järj: 1 0 7 4 6 5 7 = 1074657<sub>8</sub>

= 0001074657<sub>8</sub> = 01074657

8-järj: 120371<sub>8</sub> 1 2 0 3 7 1

binääri: 001 010 000 011 111 001

22.5.2002 Copyright Teemu Kerola, K2002 18

## Big vs. Little Endian <sup>(3)</sup>

- Miten monitavuiset arvot talletetaan?

0x1200: 

--	--	--	--

Sanan osoite

talleta 0x11223344 ??

tavuosoitteet

0x1200 0x1201 0x1202 0x1203

Big-Endian: eniten merkitsevä tavu pienimpään osoitteeseen

0x11	0x22	0x33	0x44
------	------	------	------

0x1200 0x1201 0x1202 0x1203

Little-Endian: vähiten merkitsevä tavu pienimpään osoitteeseen

0x44	0x33	0x22	0x11
------	------	------	------

22.5.2002 Copyright Teemu Kerola, K2002 19

## Big vs. Little Endian <sup>(5)</sup>

- Monitavuisen tiedon (sana-) osoite on sama molemmissa tapauksissa
- Tavujen sisäinen järjestys vaihtelee
- Suorittimen suunnittelija päättää
  - Matematiikkapiirien tulee tietää miten luvut esitetty
  - Täytyy ottaa huomioon siirrettäessä tietoa verkon yli
- Power-PC: bi-endian - molemmat moodit käytössä
  - voidaan valita ohjelmakohtaisesti
  - etuoikeutetussa tilassa voidaan vielä valita erikseen
  - suoritin osaa laskea kummallakin tavalla talletetuilla luvuilla

TTK-91: big-endian

22.5.2002 Copyright Teemu Kerola, K2002 20

## Kokonaislukujen esitysmuoto (8)

Kaikki esitetty biteillä 0 ja 1  
ei etumerkkejä  
ei desimaalipistettä

+32 +16 +8 +1

57 = 0011 1001

- Etumerkittömät kokonaisluvut helppoja
- Positiiviset luvut helppoja
  - normaali binäärilukuesitys
- **Negatiiviset luvut**
  - etumerkkibitti erikseen
  - kahden komplementtiesitys

sign bit = MSB  
= most significant bit

-57 = 1011 1001

-57 = 1100 0111 (+1)

“sign” bit      komplementit

22.5.2002      Copyright Teemu Kerola, K2002      21

## Negatiiviset luvut (8)

- Etumerkkibitti erikseen
  - sign bit = MSB  
= most significant bit
- Yhden komplementtiesitys
- Kahden komplementtiesitys
- Vakiolisäys
  - lisää 127 (= 2\*\*8-1)
  - tai jokin muu luku

57 = 0011 1001

-57 = 1011 1001      talletusmuoto

-57 = 1100 0110

-57 = 1100 0111 (+1)

“sign” bit      komplementit

-57 = 0100 0110

-57 + 127 = 70

22.5.2002      Copyright Teemu Kerola, K2002      22

## Yhden ja kahden komplementti

- Yhden komplementti:

- ykköset nolliksi ja nollat ykkösiksi

ones complement: -0 = 1111 1111
-1 = 0000 0000
-57 = 1100 0110

- Kahden komplementti

- ykköset nolliksi ja nollat ykkösiksi
- ja lisätään vielä ykkönen

57 = 0011 1001
-57 = 1100 0110
+ 1
=====
1100 0111

+2 = 0000 0010
+1 = 0000 0001
+0 = 0000 0000
-0 = 0000 0000
-1 = 1111 1111
-2 = 1111 1110

22.5.2002

Copyright Teemu Kerola, K2002

23

## Kahden komplementti

- Käytössä useimmiten

- Etu: vain yksi nolla

- Yhden komplementtissa 2 nollaa

- +0 = 0000 0000 -0 = 1111 1111

- Helpot muunnokset: arvo  $\leftrightarrow$  esitysmuoto

- Miten arvo -75 esitetään?

- 75 = 0100 1011  $\Rightarrow$
- -75 = 1011 0100 + 0000 0001 = 1011 0101

- Mitä arvoa esitysmuoto 1100 1100 tarkoittaa?

- 1100 1100 - 1 = 1100 1011  $\Rightarrow$  0011 0100 = 52
- eli 52 = 0011 0100, kahden komplementtimuodossa = 1100 1011 + 1 = 1100 1100

22.5.2002

Copyright Teemu Kerola, K2002

24

- Kahden komplementtimuodossa vähennyslasku korvautuu yhteenlaskulla

$$- 57 - 52 = 57 + (-52); \quad -57 + 52$$

57: 0011 1001
-52: <u>1100 1100</u> +
1 0000 0101 = 5

-57: 1100 0111
52: <u>0011 0100</u> +
1111 1011 = -5

22.5.2002

Copyright Teemu Kerola, K2002

25

## Liukuluvut

- Tietokoneessa ei ole realilukuja tai rationaalilukuja (matemaattiset käsitteet)
- Aina rajallinen esityksen tarkkuus
  - lukuja  $\pi$ , SQRT(2) tai  $1/3$  ei voi esittää tarkasti
  - luvut 1.000000000 ja luvut 1.000000001 ovat yhtäsuuria (joissakin esityksissä)
- Yleinen realilukuja vastaava esitysmuoto on liukulukuesitysmuoto float, double, real
  - 32 bittiä, noin 7-8 desimaalinumeron tarkkuus
  - 64 bittiä, noin 16-17 desimaalinumeron tarkkuus

22.5.2002

Copyright Teemu Kerola, K2002

26

## Liukulukujen esitys <sup>(4)</sup>

$+1.23 = +1.23 * 10^0$
------------------------

$+123.0 = +1.23 * 10^2$
-------------------------

$+0.123 = +1.23 * 10^{-1}$
----------------------------

$-0.000\ 000\ 000\ 123 = -1.23 * 10^{-10}$
--

$+123\ 000\ 000\ 000\ 000 = +1.23 * 10^{14}$
--

“+”	“14”	“1.23”
-----	------	--------

sign	exponent	mantissa or significand
	(exponentti)	(mantissa)

22.5.2002
Copyright Teemu Kerola, K2002
27

## IEEE 32-bit Floating Point Standard <sup>(3)</sup>

IEEE  
Standard 754

“+”	“15”	“1.1875” = “1.0011”
-----	------	---------------------

sign	exponent	mantissa or significand
	(2:n potenssi)	

- Etumerkki
  - 1 bitti etumerkille, 1 ⇒ “-”, 0 ⇒ “+”
  - etumerkki bitti *S* ⇒ etumerkin arvo =  $(-1)^S$

22.5.2002
Copyright Teemu Kerola, K2002
28

## IEEE 32-bit FP Standard

“+”	“15”	“1.1875” = “1.0011”
sign	exponent	mantissa or significand

8 bittiä eksponentille, lisättynä 127:llä (biased form)

exponent = 5	store	5+127 = 132 = 1000 0100
exponent = -1	store	-1+127 = 126 = 0111 1110
exponent = 0	store	0+127 = 127 = 0111 1111

esitysmuodot 0 and 255 erikoistapauksia: hyvin pieni, ääretön, ei luku

- talletettu arvoalue: 1 - 254 ⇒ todellinen arvoalue: -126 - 127

Entä kun eksponentti on **15**?

127 + 15 = 142 = 1000 1110
-------------------------------

22.5.2002 Copyright Teemu Kerola, K2002 29

## IEEE 32-bit FP Standard <sup>(7)</sup>

“+”	“15”	“0.1875” = “0.0011”
sign	exponent	mantissa or significand

• 23 bittiä mantissalle, siten että ...

- 1) Binääripiste (.) on heti ensimmäisen bitin jälkeen
- 2) Mantissa on normalisoitu: vasemmanpuolimmainen bitti on 1
- 3) Vasemmanpuolimmaista (eniten merkitsevä) bittiä (1) ei talleteta (implied bit, piilobitti)

mantissa eksponentti

0.0011	“15”
--------	------

1.1000	“12”
--------	------

1000	“12”
------	------

24 bitin mantissa!

1/8 = 0.1250  
1/16 = 0.0625  
0.1875

22.5.2002 Copyright Teemu Kerola, K2002 30

### IEEE 32-bit FP Values <sup>(8)</sup>

$23.0 = +10111.0 * 2^0 = +1.0111 * 2^4 = ?$

$4+127=131$

0	1000 0011	011 1000 0000 0000 0000 0000
---	-----------	------------------------------

sign      exponent      mantissa or significand  
 1 bit      8 bits                      23 bits

$1.0 = +1.0000 * 2^0 = ?$

$0+127 = 127$

0	0111 1111	000 0000 0000 0000 0000 0000
---	-----------	------------------------------

sign      exponent      mantissa or significand  
 1 bit      8 bits                      23 bits

22.5.2002
Copyright Teemu Kerola, K2002
31

### IEEE 32-bit FP Values <sup>(6)</sup>

0	1000 0000	111 1000 0000 0000 0000 0000
---	-----------	------------------------------

sign      exponent      mantissa or significand  
 1 bit      8 bits                      23 bits

$X = ?$	$X = (-1)^0 * 1.1111 * 2^{(128-127)}$
	$= 1.1111_2 * 2$
	$= (1 + 1/2 + 1/4 + 1/8 + 1/16) * 2$
	$= (1 + 0.5 + 0.25 + 0.125 + 0.0625) * 2$
$= 1.9375 * 2$	$= 3.875$

22.5.2002
Copyright Teemu Kerola, K2002
32



## Merkit <sup>(4)</sup>

- Yleensä 1 tavu per merkki
- ASCII, 7 bittiä/merkki (+ tark. bitti?)  

'A' = 0x41, 'a' = 0x61, LF = 0x0A
- EBCDIC, 8 bittiä/merkki
- ISO/IEC 8859-15 ('Latin-9'),
  - 8-bittiä/merkki, 256 eri merkkiä käytössä
  - mukana myös ä, ö, š, €

22.5.2002

Copyright Teemu Kerola, K2002

33

## UCS ja Unicode <sup>(5)</sup>

- UCS - Universal Character Set
- Samat merkistöt, eri standardit
- 2 tavua eli 16 bittiä per merkki
  - 65536 merkkiä koko maailmassa käytössä oleville n. 200000 symbolille
- Kontrollimerkit
  - 0x0000-001F and 0x0080-009F
  - 0x007F = DELETE, 0x0020 = SPACE
- UCS:ssä myös 8-bittiset koodi ”rivit”
  - eri alueille tai tarkoituksiin (zone) omat 8-bittiset koodinsa

22.5.2002

Copyright Teemu Kerola, K2002

34

## UCS ja Unicode

- Merkit välillä 0x0000-00FF samassa järjestyksessä kuin Latin-9 merkistössä
  - 16-bittisen UCS:n ”rivi 00” = 8-bittinen Latin-9
- Myös muut aakkoset:
  - I-zone = Kanji (0x4E00-9FFF, 20992 merkkiä)
- Ei omia konekäskyjä, manipulointi aliohjelmilla

22.5.2002

Copyright Teemu Kerola, K2002

35

## Merkkijonot

- Yleensä peräkkäin talletettu joukko tavuja
- Lisäksi tarvitaan jollain tavalla koodata merkkijonon pituus
  - laitetaan loppuun erikoismerkki
    - C-kieli: `'\0'` = 0x00
  - toteutetaan tietueena 

20	”Ei yleensä nyt enää!”
----	------------------------

**pituus merkkijono**
  - ei omia konekäskyjä, manipulointi aliohjelmilla
    - kokonaisluku- ja bittimanipulointikäskyt
    - joissakin koneissa ”strcpy”- ja strcmp”-käskyt

22.5.2002

Copyright Teemu Kerola, K2002

36

## Totuusarvot <sup>(4)</sup>

- Boolean TRUE ja FALSE
- Yleensä koodattu TRUE=1, FALSE=0
  - muttei aina!
  - totuusarvolauseke **A and B** = kokonaislukulauseke **A\*B**
- Usein Boolean arvo per sana
  - loput 31 bittiä nollia
  - ohjelmointikielten Boolean-muuttujat
- Joskus pakatussa muodossa 32/arvoa per sana
- Ei omia konekäskyjä, manipulointi aliohjelmilla
  - kokonaisluku ja bittimanipulointikäskyt
  - haluttu käsky ”**JTRUE ...**” voidaan toteuttaa käskynä ”**JPOS ...**” (jos TRUE =1)

22.5.2002

Copyright Teemu Kerola, K2002

37

## Kuvat

- Monta kuvastandardia
  - yleisyys, siirrettävyys, pakkaustiheys
  - näyttöä varten tarvittavan laskennan määrä
- Kuvatiedoston alussa otsake kertoo talletusformaatin
- Viiva- ja vektorikuvat
  - kuva koodattuna objekteina
    - ympyrä, monikulmio, käyrä, alueen väri
- Rasterikuvat
  - kuva koodattuna pisteinä
    - kunkin pisteen väri koodattu esim. 24 bitillä

22.5.2002

Copyright Teemu Kerola, K2002

38

## Kuvat

- Kuvat ovat yleensä pakattu mahdollisimman vähän tilaa vievää muotoon
  - optimoitu tilan, ei laskennan mukaan
  - purkaminen voi vaatia paljon laskentaa
- GIF, JPEG, TIFF, BMP, ....
- Ei omia konekäskyjä, manipulointi aliohjelmilla

22.5.2002

Copyright Teemu Kerola, K2002

39

## Videokuva

- Vie hyvin paljon muistitilaa
- Talletus kuva kerrallaan, esim. 25 kuvaa/sek
  - 1 sekunti hyvälaatuista videokuva pakkaamattomassa muodossa 20 MB
- Talletus ”incrementaalisesti”
  - kun seuraava kuva poikkeaa edellisestä vain vähän
  - talleta vain muutokset edelliseen

22.5.2002

Copyright Teemu Kerola, K2002

40

## Videostandardit

- MPEG (Moving Pictures Expert Group)
- AVI (Audio Visual Interleave)
- MOV, INDEO, FLI, GL, DVD, ...
- ei omia konekäskyjä, manipulointi aliohjelmilla
- tai erikoisproessoreilla (GPU), joiden käskykanta suunniteltu (jonkin standardin mukaisten kuvien) kuvankäsittelyyn
  - grafiikkakorteilla

22.5.2002

Copyright Teemu Kerola, K2002

41

## Grafiikkakortit

- Esim. 4-64 MB (dual-port) muistia
  - 2 lukua/kirjoitusta samanaikaisesti
- tai 'tavallista', mutta hyvin nopeaa RAMia
- Nopea väylä (ennen PCI, nyt AGP) suorittimelle
- Näytönohjaus monitoristandardien (VGA, XGA, RGB, ..) mukaisesti
- Oma suoritin (GPU)
  - lukee videodataa ja generoi näytettävän kuvan näyttöpuskuriin, josta monitori sen näyttää
- Voi olla integroitu emolevyn kanssa

22.5.2002

Copyright Teemu Kerola, K2002

42

## Äänet

- Täydellinen äänidata
  - 44100 näytettä/sek, 16 b/näyte, 88KB/sek
- Syntetisoitu ääni
  - MIDI-käskyjä
    - Music Instrument Digital Interface
    - ”Soita nuotti N voimakkuudella V”
- ei omia konekäskyjä, manipulointi aliohjelmilla
- tai erikoisprosessoreilla, joiden käskykanta suunniteltu äänen käsittelyyn
  - äänikortit

22.5.2002

Copyright Teemu Kerola, K2002

43

## Äänikortit

- Esim. 4-64 MB VRAM- tai RAM-muistia
- nopea väylä (esim. PCI) suorittimelle
- oma suoritin, joka lukee äänidataa ja generoi äänet kaiuttimille tai vahvistimeen
  - kaiuttimet tai vahvistin kiinni äänikortilla
- Voi olla integroitu emolevyn tai grafiikkakortin kanssa

22.5.2002

Copyright Teemu Kerola, K2002

44

## Maku, haju, tunto ja muu data <sup>(3)</sup>

- Tähtien kirkkaus, hajut, ks. HS artikkeli 5.5.2000  
veneen tyyppi, tunteen palo, ....
- Sovelluskohtaisesti, ei vielä yleisiä standardeja
  - kokonaisluvut (diskreetti data)
  - liukuluvut (jatkuva data)
- Ei omia konekäskyjä, manipulointi omilla aliohjelmilla

22.5.2002

Copyright Teemu Kerola, K2002

45

## Konekäskyjen esitysmuoto muistissa <sup>(4)</sup>

- 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

22.5.2002

Copyright Teemu Kerola, K2002

46

## 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

M = muistinoutojen määrä toiseen operandiin (ennen mahdollista muistiin talletusta)

muistiosoite tai (pienekkö) vakio

(addressing mode)

00 eli 0 kpl, rekisteri tai välitön osoitus  
 01 eli 1 kpl, suora osoitus  
 10 eli 2 kpl, epäsuora osoitus  
 ( 11 eli 3 kpl, epäkelpo arvo → poikkeustilanne )

22.5.2002 Copyright Teemu Kerola, K2002 47

## Konekäskyn operandit ja tulos

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

jos  $R_i = R_0$ ,  
 niin pelkkä ADDR

Konekielen tiedon osoitusmoodit

- arvo:  $R_i + ADDR$
- muistipaikan  $M[R_i + ADDR]$  sisältö
- muistipaikan  $M[ M[R_i + ADDR] ]$  sisältö

22.5.2002 Copyright Teemu Kerola, K2002 48



## Taulukkojen esitysmuoto

- Peräkkäisrakenteena, kuten esimerkit aikaisemmin
- riveittäin tai sarakkeittain
- ei omia konekäskyjä, manipulointi aliohjelmilla tai loopeilla
  - paitsi ns. vektorikoneet, joilla on omia konekäskyjä vektoriopeeraatioita varten
- Indeksoitu tiedonosoitusmoodi tulee yksiulotteisten taulukoiden käyttöä

22.5.2002

Copyright Teemu Kerola, K2002

49

## Tietueiden esitysmuoto

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

22.5.2002

Copyright Teemu Kerola, K2002

50

## 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

22.5.2002

Copyright Teemu Kerola, K2002

51

## -- Jakson 6 loppu --



22.5.2002

Copyright Teemu Kerola, K2002

52