

## Luento 6 Tiedon esitysmuodot

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

07/11/2001

Copyright Teemu Kerola 2001

1

## Tiedon tyypit <sup>(3)</sup>

- 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 tietotyypille
  - kokonaisluvut
  - liukuluvut (useimmat suorittimet nykyään)
  - totuusarvot (jotkut suorittimet)
  - merkit (jotkut suorittimet)
  - konekäskyt

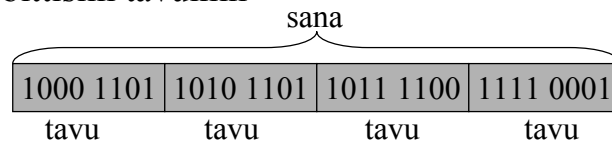
07/11/2001

Copyright Teemu Kerola 2001

2

## Tiedon esitys laitteistossa <sup>(3)</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 8-bittisiin tavuihin (byte)



07/11/2001

Copyright Teemu Kerola 2001

3

## Tiedon esitys laitteistossa <sup>(2)</sup>

- Tietoa siirretään muistiväylää pitkin sanoina
  - joskus useampi kuin yksi sana kerrallaan (lohko)
- Suorittimen 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

07/11/2001

Copyright Teemu Kerola 2001

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

07/11/2001

Copyright Teemu Kerola 2001

5

## Suorittimen ymmärtämä tieto <sup>(10)</sup>

- Kaikki tieto koneessa on koodattuna biteiksi
- Muistissa voidaan esittää kaikki tieto millä tahansa 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

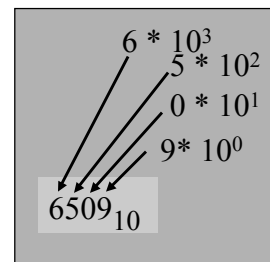
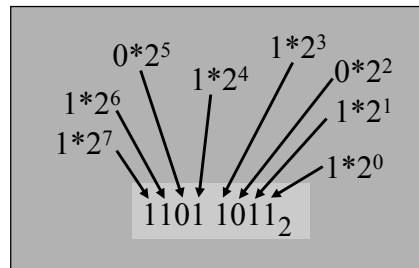
07/11/2001

Copyright Teemu Kerola 2001

6

## Binäärijärjestelmä (2)

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

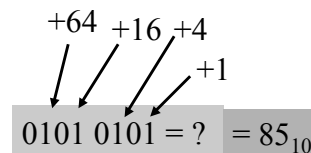
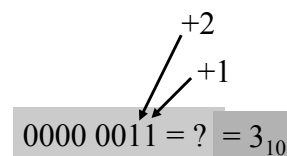
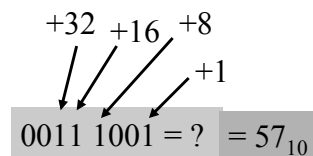


07/11/2001

Copyright Teemu Kerola 2001

7

## Binäärilukuesimerkkejä



07/11/2001

Copyright Teemu Kerola 2001

8

## Binäärilukujen laskutoimitukset <sup>(3)</sup>

+	0	1
0	0	1
1	1	10

$$\begin{array}{r}
 11 \\
 101101 \\
 +1100 \\
 \hline
 111001
 \end{array}
 \begin{array}{l}
 45 \\
 +12 \\
 =57
 \end{array}$$

*	0	1
0	0	0
1	0	1

$$\begin{array}{r}
 101 \\
 *101 \\
 \hline
 101 \\
 +101 \\
 \hline
 11001
 \end{array}
 \begin{array}{l}
 5 \\
 *5 \\
 \\
 =25
 \end{array}$$

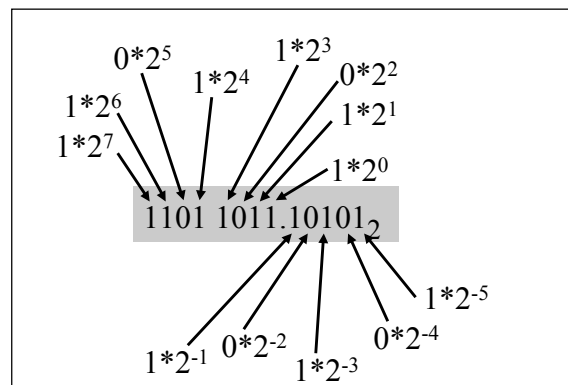
07/11/2001

Copyright Teemu Kerola 2001

9

## Binääripiste <sup>(2)</sup>

- Binääriluvuilla voi olla myös binääriosa (vrt. desimaaliosa)

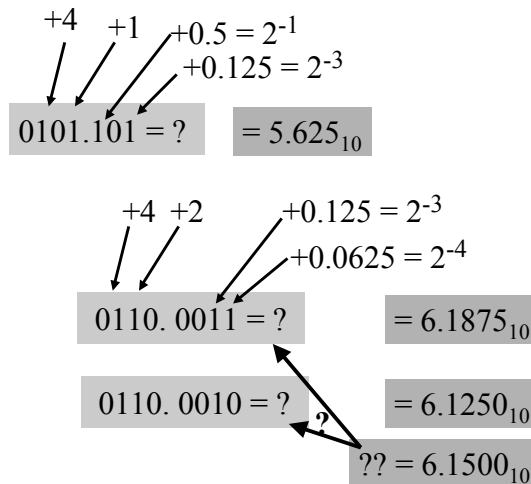


07/11/2001

Copyright Teemu Kerola 2001

10

## Binääripiste-esimerkkejä <sup>(10)</sup>



07/11/2001

Copyright Teemu Kerola 2001

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 jakojäännökset käännettyssä järjestyksessä

07/11/2001

Copyright Teemu Kerola 2001

12

**10-järj  $\Rightarrow$  2-järj**  
**kokonaislukuesimerkki <sup>(1)</sup>**

$57_{10} = ?$	$57/2 = 28$ jää 1	$= 11\ 1001_2$
	$28/2 = 14$ jää 0	$= 0011\ 1001_2$
	$14/2 = 7$ jää 0	
	$7/2 = 3$ jää 1	
	$3/2 = 1$ jää 1	
	$1/2 = 0$ jää 1	
	loppu	

07/11/2001 Copyright Teemu Kerola 2001 13

**10-järj  $\Rightarrow$  2-järj**  
desimaaliosa  $\Rightarrow$  binääriosia <sup>(2)</sup>

- 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

07/11/2001 Copyright Teemu Kerola 2001 14

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

$0.1875_{10} = ?$	$2 * 0.1875 = 0.375 = 0 + 0.375$	↓ loppu
	$2 * 0.375 = 0.75 = 0 + 0.75$	
	$2 * 0.75 = 1.5 = 1 + 0.5$	
	$2 * 0.5 = 1.0 = 1 + (0.0)$	

$= 0.0011_2$

$= 0.0011000000000000000000_2$

07/11/2001 Copyright Teemu Kerola 2001 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  

10
11
12
13
14
15

07/11/2001 Copyright Teemu Kerola 2001 16



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

binääri: 0100 0111 1001 1010 1111

16-järj:  $\underbrace{0100}_4 \underbrace{0111}_7 \underbrace{1001}_9 \underbrace{1010}_A \underbrace{1111}_F = 479AF_{16}$   
 $= 0004\ 79AF_{16} = \underline{0x}\ 479AF$

16-järj:  $120ADF_{16}$     1 2 0 A D F

binääri:  $\underbrace{0001}_1 \underbrace{0010}_2 \underbrace{0000}_0 \underbrace{1010}_A \underbrace{1110}_D \underbrace{1111}_F$

07/11/2001

Copyright Teemu Kerola 2001

17

## Oktaaliesimerkkejä <sup>(7)</sup>

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

binääri: 01 000 111 100 110 101 111

8-järj:  $\underbrace{01}_1 \underbrace{000}_0 \underbrace{111}_7 \underbrace{100}_4 \underbrace{110}_6 \underbrace{101}_5 \underbrace{111}_7 = 1074657_8$   
 $= 0001074657_8 = \underline{0}1074657$

8-järj:  $120371_8$     1 2 0 3 7 1

binääri:  $\underbrace{001}_1 \underbrace{010}_2 \underbrace{000}_0 \underbrace{011}_3 \underbrace{111}_7 \underbrace{001}_1$

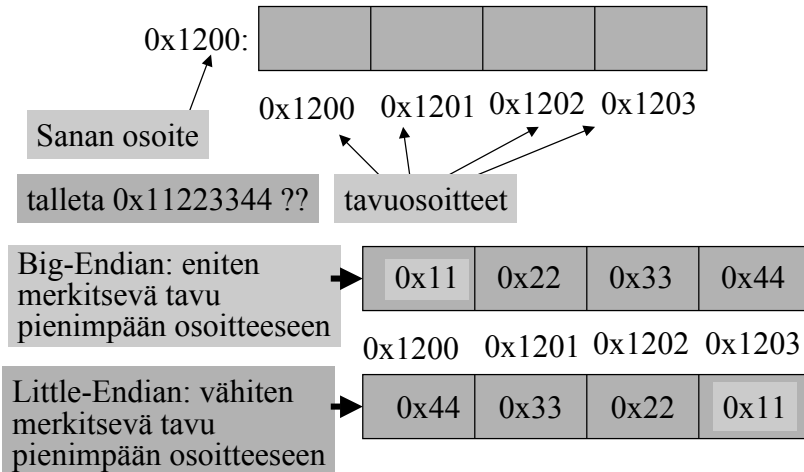
07/11/2001

Copyright Teemu Kerola 2001

18

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

- Miten monitavuiset arvot talletetaan?



07/11/2001

Copyright Teemu Kerola 2001

19

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

- Monitavuisen tiedon (sana-) osoite on sama molemmissa tapauksissa
- Tavujen järjestys on erilainen
- 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

07/11/2001

Copyright Teemu Kerola 2001

20

## Negatiiviset luvut (4)

- Etumerkkibitti erikseen
  - $+57 = 0011\ 1001$
  - $-57 = \underline{1}011\ 1001$
  - sign bit = MSB  
= most significant bit
- Yhden komplementtiesitys
  - $-57 = 1100\ 0110$
  - “sign” bit
- **Kahden** komplementtiesitys
  - $-57 = 1100\ 0111$
  - “sign” bit
  - $+1$
- Vakiolisäys
  - Lisää  $127 (=2^8 - 1)$
  - tai joku muu luku ...
  - $-57 = 0100\ 0110$
  - $-57 + 127 = 70$

07/11/2001 Copyright Teemu Kerola 2001 21

## Kahden komplementti (7)

- Useimmiten käytössä
  - $+57 = 0011\ 1001$
- Etu: vain yksi nolla
  - yhden komplementtissa kaksi nollaa
    - $+0 = 0000\ 0000$     $-0 = 1111\ 1111$
- Helppo esitysmuodon muunnos
  - miten luku 57 esitetään?
    - $1100\ 0110 \underline{+1} = 1100\ 0111$
  - mitä lukua esitysmuoto 1100 0111 tarkoittaa?
    - $-(0011\ 1000 \underline{+1}) = -0011\ 1001 = -57$

07/11/2001 Copyright Teemu Kerola 2001 22

## Liukuluvut <sup>(3)</sup>

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

07/11/2001

Copyright Teemu Kerola 2001

23

## 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)

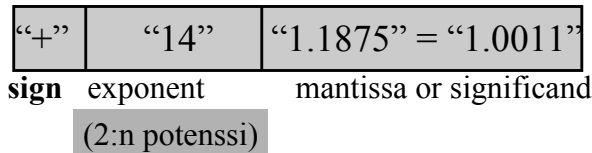
07/11/2001

Copyright Teemu Kerola 2001

24

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

IEEE  
Standard 754



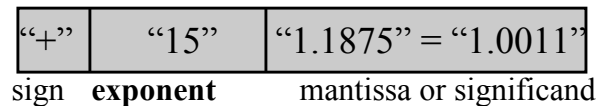
- Etumerkki
  - 1 bitti, 1 ⇒ “-”, 0 ⇒ “+”
  - etumerkkibitti  $S$  ⇒ etumerkin arvo =  $(-1)^S$

07/11/2001

Copyright Teemu Kerola 2001

25

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



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

$$\text{exponent} = 5 \xrightarrow{\text{store}} 5 + 127 = 132 = 1000\ 0100$$

$$\text{exponent} = -1 \xrightarrow{\text{store}} -1 + 127 = 126 = 0111\ 1110$$

$$\text{exponent} = 0 \xrightarrow{\text{store}} 0 + 127 = 127 = 0111\ 1111$$

- eksponentit 0 ja 255 erikoistapauksia:
  - hyvin pienet luvut, NaN,  $\pm\infty$
- talletettu arvoalue: **1 - 254** ⇒ tod. arvoalue: **-126 - 127**

07/11/2001

Copyright Teemu Kerola 2001

26

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

“+”	“15”	“0.1875” = “0.0011”
sign	exponent	<b>mantissa</b> or significand

$1/8 = 0.1250$

$1/16 = \frac{0.0625}{0.1875}$

• 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”
--------	------

<b>1000</b>	“12”
-------------	------

24 bitin mantissa!

07/11/2001
Copyright Teemu Kerola 2001
27

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

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

$4+127=131$

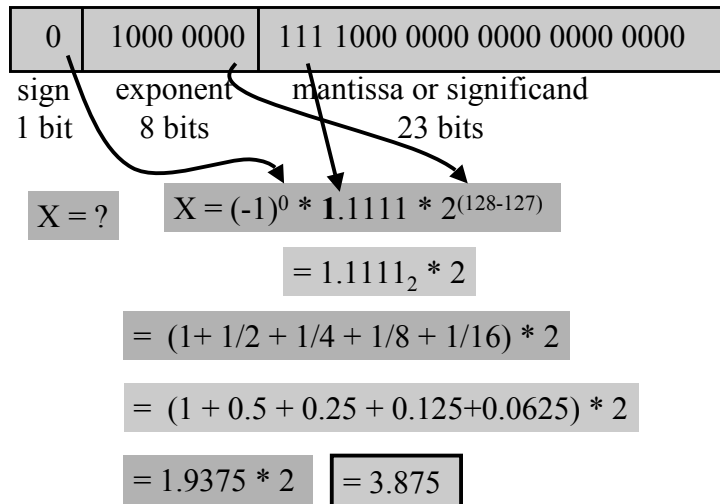
0	1000 0011	011 1000 0000 0000 0000 0000
sign	exponent	mantissa or significand
1 bit	8 bits	23 bits

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

$0+127 = 127$

0	0111 1111	000 0000 0000 0000 0000 0000
sign	exponent	mantissa or significand
1 bit	8 bits	23 bits

07/11/2001
Copyright Teemu Kerola 2001
28

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

07/11/2001

Copyright Teemu Kerola 2001

29

Merkit <sup>(5)</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, €

Lisää tietoa: ks.

<http://www.tieke.fi/edisty/edis699/stand699.htm>

07/11/2001

Copyright Teemu Kerola 2001

30

## 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

07/11/2001

Copyright Teemu Kerola 2001

31

## UCS ja Unicode <sup>(3)</sup>

- Merkit välillä 0x0000-00FF (16 bittiä) samassa järjestyksessä kuin Latin-9 merkistössä (8 bittiä)
  - 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

07/11/2001

Copyright Teemu Kerola 2001

32



## Merkkijonot <sup>(5)</sup>

- Yleensä peräkkäin talletettu joukko tavuja
- Lisäksi tarvitsee 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

07/11/2001

Copyright Teemu Kerola 2001

33

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

- Boolean TRUE ja FALSE
- Yleensä koodattu TRUE=1, FALSE=0
  - muttei aina!
  - Totuusarvolauseke *A and B* = kokon.lukulauseke *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
  - käskyn ”*JNZER ...*” asemesta käsky ”*JTRUE ...*”

07/11/2001

Copyright Teemu Kerola 2001

34

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

- 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

07/11/2001

Copyright Teemu Kerola 2001

35

## 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

07/11/2001

Copyright Teemu Kerola 2001

36

## 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

07/11/2001

Copyright Teemu Kerola 2001

37

## Videostandardit

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

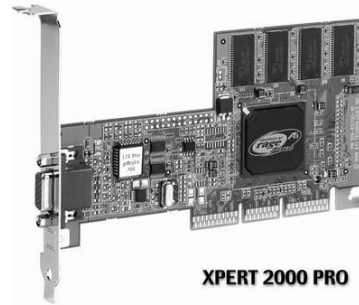
07/11/2001

Copyright Teemu Kerola 2001

38

## Grafiikkakortit

- Esim. 4-64 MB VRAM (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



07/11/2001

Copyright Teemu Kerola 2001

39

## 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 ...
- Erikoisproessoreilla, joiden käskykanta suunniteltu äänen käsittelyyn
  - äänikortit

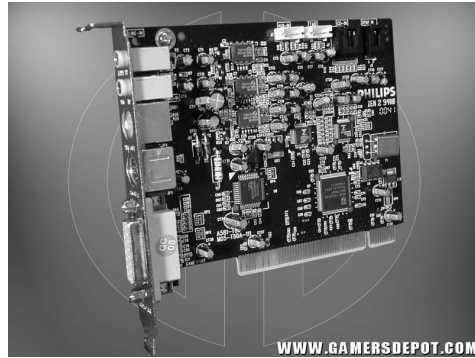
07/11/2001

Copyright Teemu Kerola 2001

40

## 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



07/11/2001

Copyright Teemu Kerola 2001

41

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

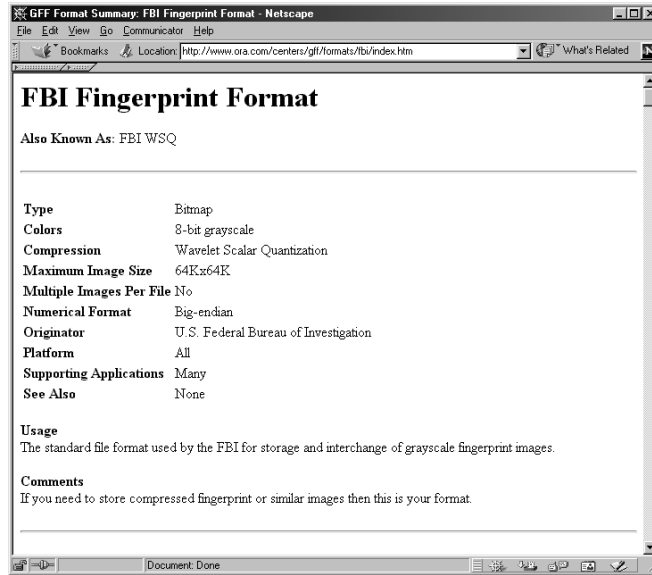
- Tähtien kirkkaus, hajut, ks. HS artikkeli 5.5.2000  
veneiden tyyppi, tunteen palo, ....
- Toteutus sovelluskohtaisesti, ei vielä yleisiä standardeja
  - kokonaisluvut (diskreetti data)
  - liukuluvut (jatkuva data)
- Ei omia konekäskyjä, manipulointi omilla aliohjelmilla

07/11/2001

Copyright Teemu Kerola 2001

42

## -- Luennon 6 loppu --



07/11/2001

Copyright Teemu Kerola 2001

43