Luento 6
Tiedon esitysmuodot

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

 Copyright Teemu Kerola 2002

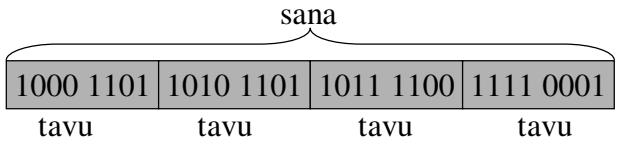
Tiedon tyypit (3)

- 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

 Copyright Teemu Kerola 2002

Tiedon esitys laitteistossa (3)

- 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)



14.8.2002 Copyright Teemu Kerola 2002 3

Tiedon esitys laitteistossa (2)

- 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

14.8.2002 Copyright Teemu Kerola 2002 4

Tiedon esitys (7)

- 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

 Copyright Teemu Kerola 2002

Suorittimen ymmärtämä tieto (10)

- Kaikki tieto koneessa on koodattuna biteiksi
- Muistissa voidaan esittää kaikki tieto millä tahansa sovitulla esitystavalla (koodauksella)
- Suoritin osaa tehdä operaatioita joillakin esitystavoilla koodatuille tiedoille

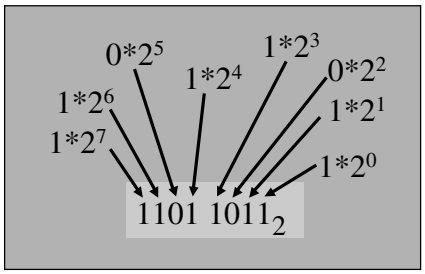
TTK-91: kokonaisluvut

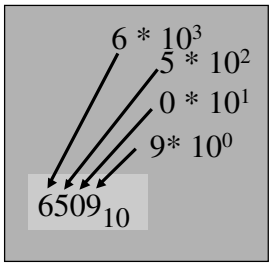
  - kokonaisluvut ja liukuluvut (aina)
  - totuusarvot, merkit ja merkkijonot (joskus)
  - kuvat ja äänet (ei yleensä ellei erikoistunut suoritin)
  - hajut (ei vielä)
- Muiden tietojen käsittely tapahtuu ohjelmallisesti
  - esim. merkkejä voidaan käsitellään kokonaislukuoperaatioilla ja aliohjelmilla

 Copyright Teemu Kerola 2002

Binäärijärjestelmä (2)

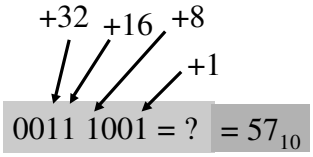
- Kantaluku 2, numerot 0 ja 1
  - numeroiden painoarvot oikealta vasemmalle: $1=2^0$, $2=2^1$, $4=2^2$, $8=2^3$, $16=2^4$, $32=2^5$, …
  - kymmenjärjestelmässä painoarvot ovat $1=10^0$, $10=10^1$, $100=10^2$, $1000=10^3$, …

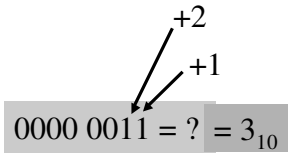


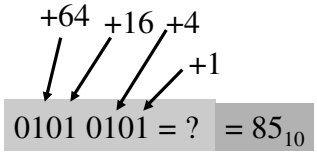


14.8.2002 Copyright Teemu Kerola 2002 7

Binäärilukuesimerkkejä







14.8.2002 Copyright Teemu Kerola 2002 8

Binäärilukujen laskutoimitukset (3)

+	0	1
0	0	1
1	1	10

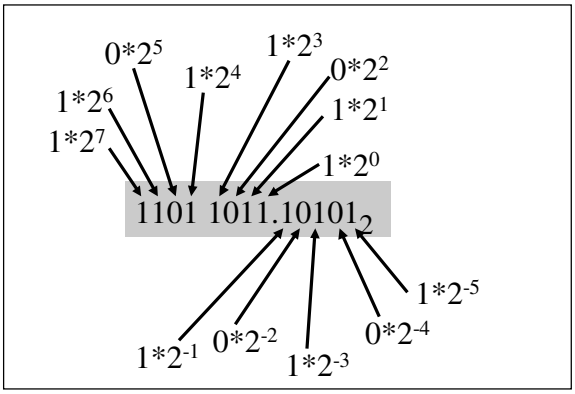
```
  1 1
 101101      45
  +1100     +12
 ======     ===
 111001     =57
```

*	0	1
0	0	0
1	0	1

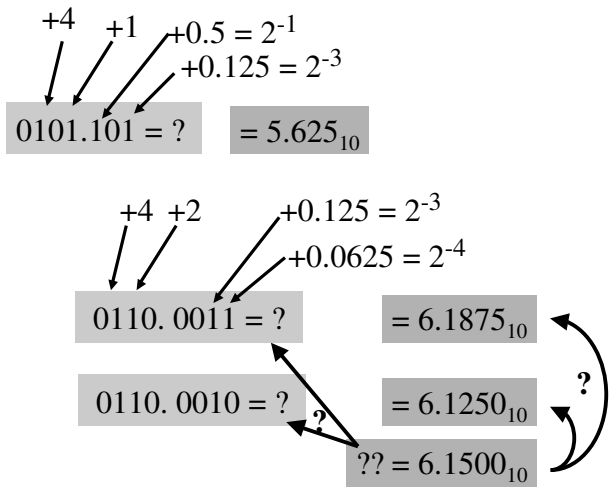
```
    101      5
   *101     *5
 ======
    101
  +101
 ======
  11001     =25
```


Binääripiste (2)

- Binääriluvuilla voi olla myös binääriosa (vrt. desimaaliosa)



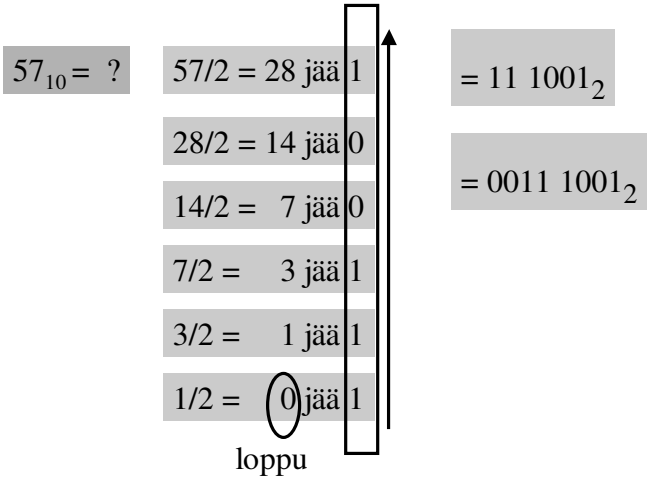
Binääripiste-esimerkkejä (10)



Muunnokset lukujärjestelmien välillä (5)

- 2-järjestelmä ⇒ 10-järjestelmä
  - esitettiin jo edellä
- 10-järjestelmä ⇒ 2-järjestelmä
  - kokonaisosa ja desimaaliosa erikseen
  - kokonaisosa:
    - jaa toistuvasti 2:lla, kunnes 0 jäljellä
    - ota jakojäännökset käännetyssä järjestyksessä

10-järj ⇒ 2-järj kokonaislukuesimerkki (11)



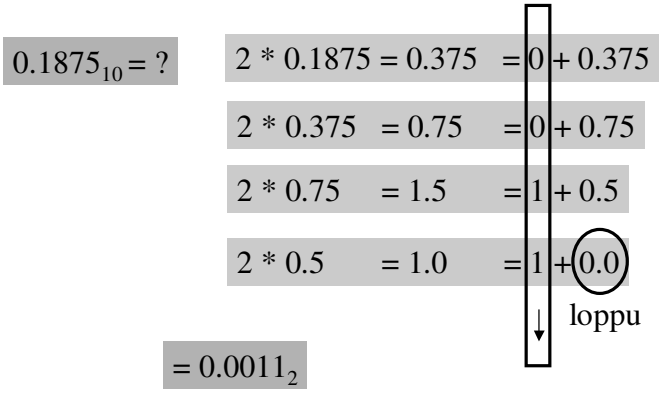
 Copyright Teemu Kerola 2002

10-järj ⇒ 2-järj desimaaliosa ⇒ binääriosa (2)

- 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

 Copyright Teemu Kerola 2002

10-järj ⇒ 2-järj
desimaaliosa ⇒ binääriosa (9)



$= 0.0011_2$

$= 0.00110000000000000000_2$

Heksadesimaaliesitys (6)

- 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

Heksadesimaaliesimerkkejä (11)

binääri: 0100 0111 1001 1010 1111

16-järj: 4 7 9 A F = $479AF_{16}$

= $0004\ 79AF_{16}$ = 0x 479AF

16-järj: $120ADF_{16}$ 1 2 0 A D F

binääri: 0001 0010 0000 1010 1110 1111

14.8.2002 Copyright Teemu Kerola 2002 17

Oktaaliesimerkkejä (7)

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_8

= 0001074657_8 = 01074657

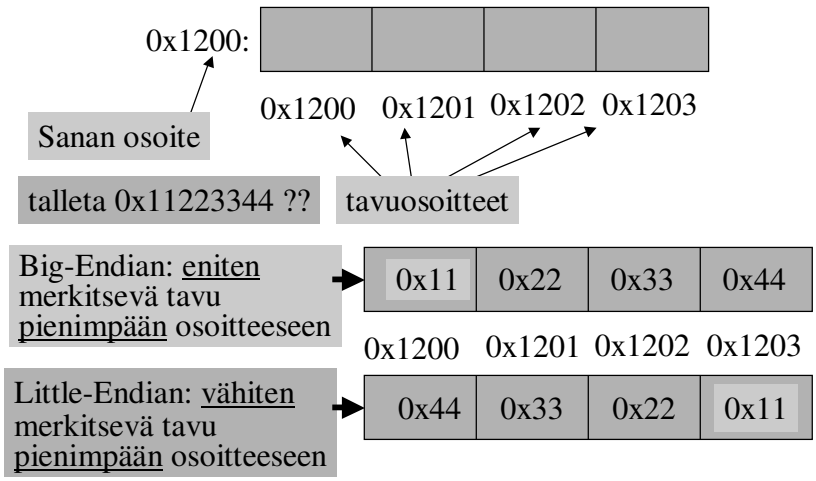
8-järj: 120371_8 1 2 0 3 7 1

binääri: 001 010 000 011 111 001

14.8.2002 Copyright Teemu Kerola 2002 18

Big vs. Little Endian (3)

- Miten monitavuiset arvot talletetaan?



14.8.2002 Copyright Teemu Kerola 2002 19

Big vs. Little Endian (5)

- 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

14.8.2002 Copyright Teemu Kerola 2002 20

Negatiiviset luvut (4)

+57 = 0011 1001

- Etumerkkibitti erikseen

sign bit = MSB
= most significant bit

luku -57 = 1011 1001 talletus-muoto

- Yhden komplementtiesitys

-57 = 1100 0110

"sign" bit

+1

- Kahden komplementtiesitys

-57 = 1100 0111

"sign" bit

- Vakiolisäys
  - Lisää 127 (=2^8 –1)
  - tai joku muu luku …

-57 = 0100 0110

-57 + 127 =70

 Copyright Teemu Kerola 2002

Kahden komplementti (6)

+57 = 0011 1001

- Useimmiten käytössä
- Etu: vain yksi nolla
  - yhden komplementissa kaksi nollaa
    - +0 = 0000 0000 -0 = 1111 1111
- Helpot muunnokset: arvo ↔ esitysmuoto
  - miten arvo -57 esitetään?
    - 1100 0110 **+1** = 1100 0111
  - mitä arvoa esitysmuoto 1100 0111 tarkoittaa?
    - - (0011 1000 **+1**) = -0011 1001 = -57

 Copyright Teemu Kerola 2002

Liukuluvut (3)

- Tietokoneessa ei ole realilukuja tai rationaalilukuja (matemaattiset käsitteet)
- Aina rajallinen esityksen tarkkuus
  - lukuja π, 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

 Copyright Teemu Kerola 2002

Liukulukujen esitys (4)

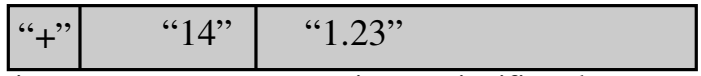
$+1.23 = +1.23 * 10^{0}$

$+123.0 = +1.23 * 10^{2}$

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

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

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



sign exponent (exponentti) mantissa or significand (mantissa)

 Copyright Teemu Kerola 2002

IEEE 32-bit Floating Point Standard (3)

IEEE Standard 754

"+"	"14"	"1.1875" = "1.0011"
sign	exponent	mantissa or significand
	(2:n potenssi)	

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

 Copyright Teemu Kerola 2002

IEEE 32-bit FP Standard (3)

"+"	"15"	"1.1875" = "1.0011"
sign	**exponent**	mantissa or significand

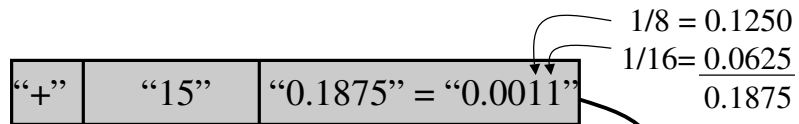
- 8 bittiä eksponentille, lisättynä 127:llä (biased form)

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 ja 255 erikoistapauksia
    - laajennettu arvoalue: hyvin pienet luvut, NaN, ±∞
  - talletettu arvoalue: **1 - 254** ⇒ tod. arvoalue: **-126 - 127**
    (esitysmuoto) (arvoalue)

 Copyright Teemu Kerola 2002

IEEE 32-bit FP Standard (7)



- 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"

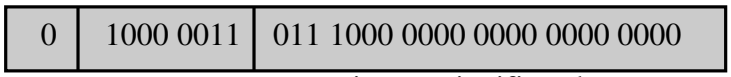
24 bitin mantissa!

14.8.2002 Copyright Teemu Kerola 2002 27

IEEE 32-bit FP Values (9)

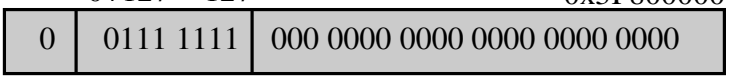
$23.0 = +10111.0 * 2^0 = +1.0111 * 2^4 = ?$

4+127=131



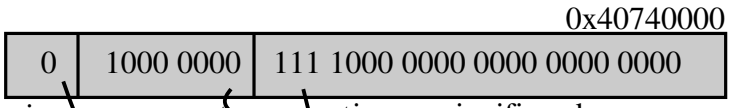
$1.0 = +1.0000 * 2^0 = ?$

0+127 = 127 0x3F800000



14.8.2002 Copyright Teemu Kerola 2002 28

IEEE 32-bit FP Values (6)



X = ?

$X = (-1)^0 * \mathbf{1}.1111 * 2^{(128-127)}$

$= 1.1111_2 * 2$

$= (1+ 1/2 + 1/4 + 1/8 + 1/16) * 2$

$= (1 + 0.5 + 0.25 + 0.125+0.0625) * 2$

$= 1.9375 * 2$ $= 3.875$

Merkit (5)

- 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

UCS ja Unicode (5)

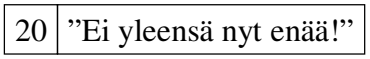
- UCS - Universal Character Set
- Samat merkistöt, eri standardit
- 2 tavua eli 16 bittiä per merkki
  - 65536 merkkiä koko maailmassa käytössä oleville n. 200000 symbolille
- Kontrollimerkit
  - 0x0000-001F and 0x0080-009F
  - 0x007F = DELETE, 0x0020 = SPACE
- UCS:ssä myös 8-bittiset koodi ”rivit”
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

UCS ja Unicode (3)

- 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

Merkkijonot (5)

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



pituus merkkijono

  - ei omia konekäskyjä, manipulointi aliohjelmilla
    - kokonaisluku- ja bittimanipulointikäskyt
    - joissakin koneissa ”strcopy” ja ”strcmp” käskyt

 Copyright Teemu Kerola 2002

Totuusarvot (4)

- 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
  - haluttu käsky ”***JTRUE ...”*** voidaan toteuttaa käskynä ”***JPOS …”*** (jos TRUE = 1)

 Copyright Teemu Kerola 2002

Kuvat (4)

- 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

 Copyright Teemu Kerola 2002

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

 Copyright Teemu Kerola 2002

Videokuva

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

 Copyright Teemu Kerola 2002

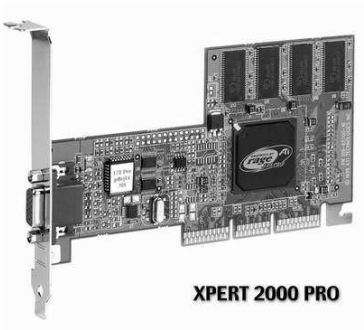
Videostandardit

- 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

 Copyright Teemu Kerola 2002

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 näyttö-puskuriin, josta monitori sen näyttää
- Voi olla integroitu emolevyn kanssa



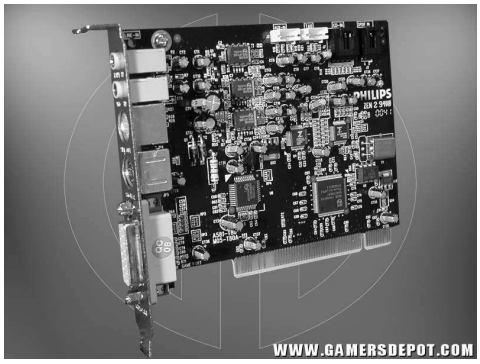
 Copyright Teemu Kerola 2002

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

 Copyright Teemu Kerola 2002

Äänikortit



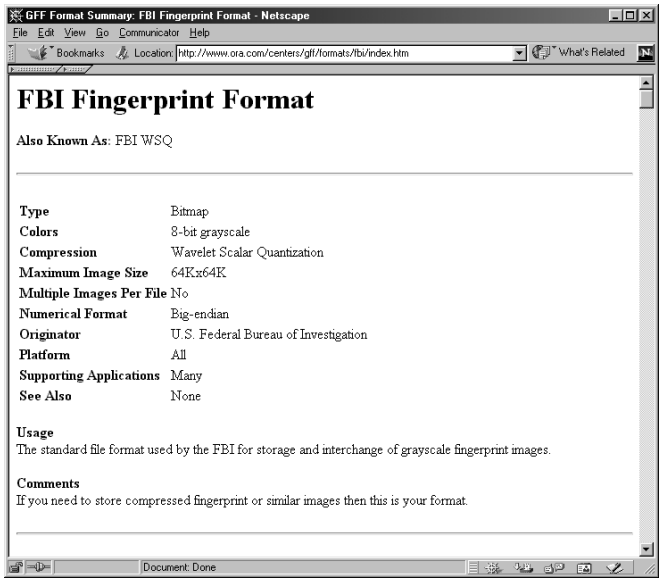
- Esim. 4-64 MB VRAM tai RAM muistia
- Nopea väylä (esim. PCI) suorittimelle
- Oma suoritin, joka lukee äänidataa ja generoi äänet kaiuttimille tai vahvistimeen
  - kaiuttimet tai vahvistin kiinni äänikortilla
- Voi olla integroitu emolevyn (tai grafiikkakortin) kanssa

Maku, haju, tunto ja muu data (3)

ks. HS artikkeli 5.5.2000

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

-- Luennon 6 loppu --



14.8.2002 Copyright Teemu Kerola 2002 43