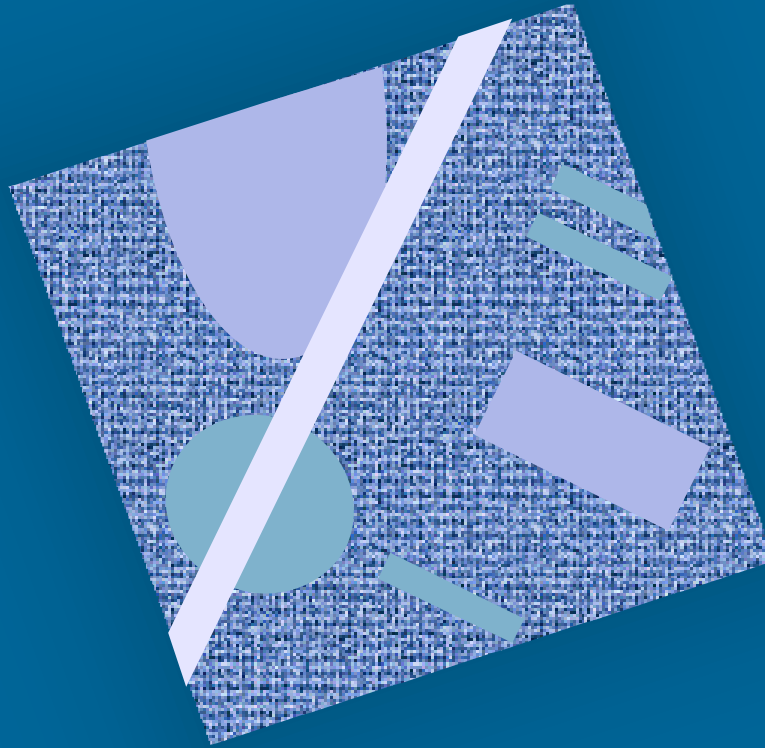


Luento 6

Tiedon esitysmuodot



Lukujärjestelmät

Luvut, merkit,
merkkijonot,
totuusarvot, oliot

Kuvat, äänet, hajut(?)

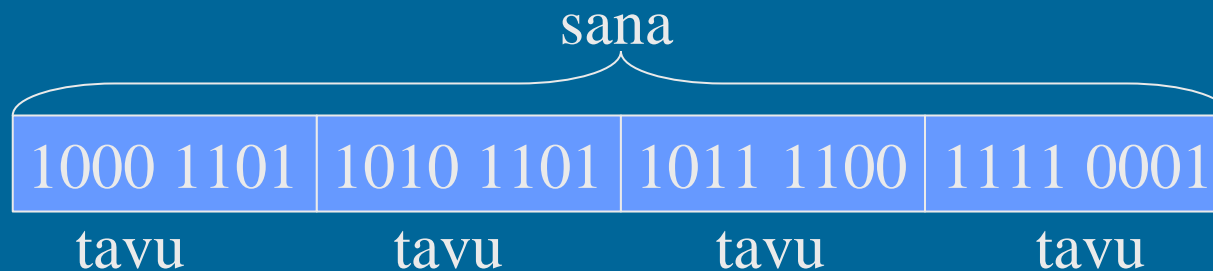
Ohjelmat

Tiedon tyypit

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

Tiedon esitys laitteistossa ⁽⁴⁾

- 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)



Tiedon esitys laitteistossa ⁽⁴⁾

- 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

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

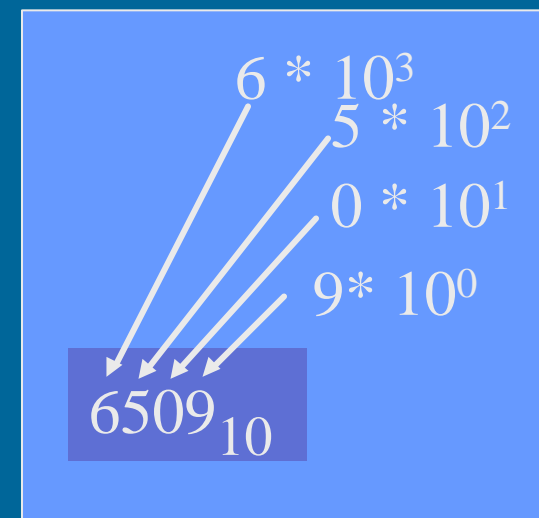
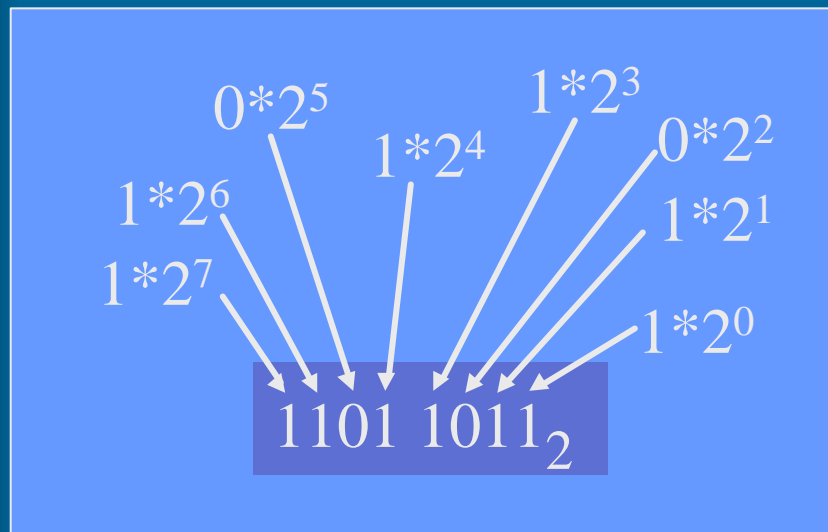
Suorittimen ymmärtämä tieto ⁽⁹⁾

- 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, äänet ja hajut (ei yleensä)
- Muiden tietojen käsittely tapahtuu ohjelmallisesti
 - esim. merkkejä voidaan käsitellä kokonaislukuoperaatioilla ja aliohjelmilla

TTK-91:
kokonaisluvut

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
 $1=10^0, 10=10^1, 100=10^2, 1000=10^3, \dots$



Binäärilukuesimerkkejä

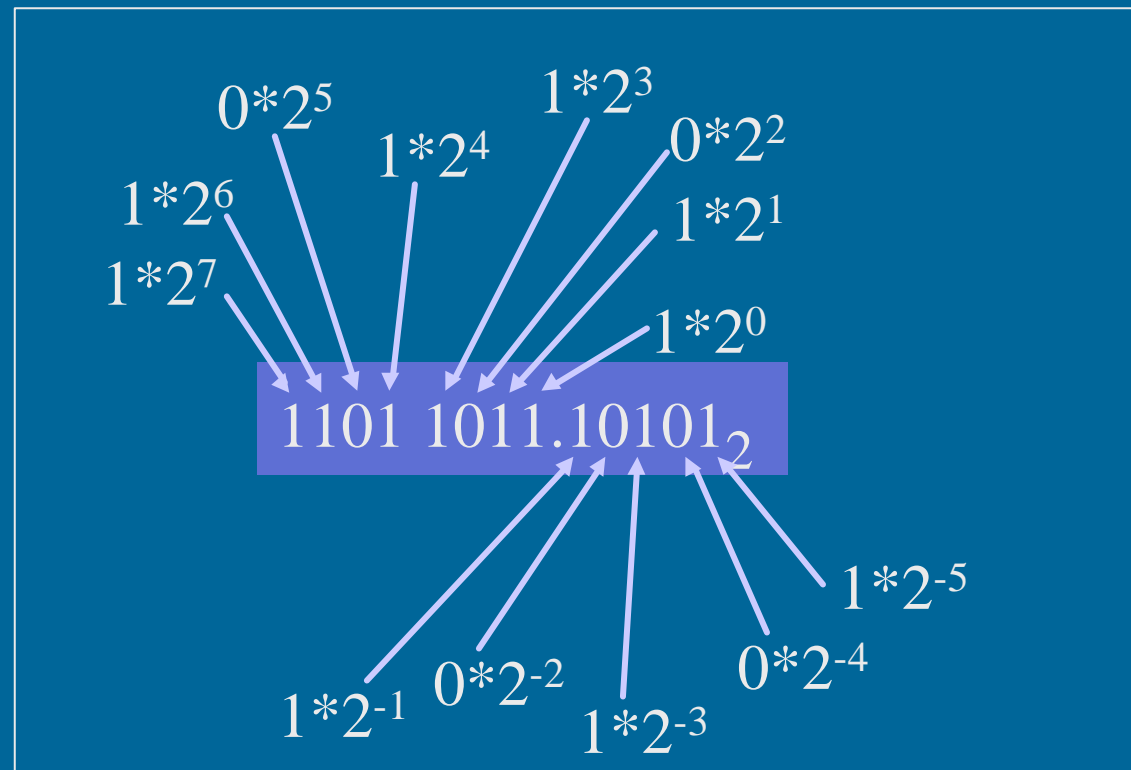
$$\begin{array}{ccccccc} & +32 & +16 & +8 & & & \\ & \swarrow & \swarrow & \swarrow & \swarrow & & \\ 0011 & 1001 & = ? & & = 57_{10} & & \\ & \swarrow & \swarrow & \swarrow & \swarrow & & \\ & +1 & & & & & \end{array}$$

$$\begin{array}{ccccccc} & & & +2 & & & \\ & & & \swarrow & & & \\ 0000 & 0011 & = ? & & = 3_{10} & & \\ & & & \swarrow & & & \\ & & & +1 & & & \end{array}$$

$$\begin{array}{ccccccc} & +64 & +16 & +4 & & & \\ & \swarrow & \swarrow & \swarrow & \swarrow & & \\ 0101 & 0101 & = ? & & = 85_{10} & & \\ & & & \swarrow & & & \\ & & & +1 & & & \end{array}$$

Binääripiste ⁽²⁾

- Binääriluvuilla voi olla myös binääriosaa (vrt. desimaaliosa)



Binääripiste-esimerkkejä

$+4$ $+1$ $+0.5 = 2^{-1}$
 $+0.125 = 2^{-3}$

$0101.101 = ?$ $= 5.625_{10}$

$+4$ $+2$ $+0.125 = 2^{-3}$
 $+0.0625 = 2^{-4}$

$0110.0011 = ?$ $= 6.1875_{10}$

Muunnokset lukujärjestelmien välillä ⁽⁵⁾

- 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 käännetyssä järjestyksessä

10-järj \Rightarrow 2-järj kokonaislukuesimerkki ⁽¹¹⁾

$$57_{10} = ?$$

$$57/2 = 28 \text{ jää } 1$$

$$28/2 = 14 \text{ jää } 0$$

$$14/2 = 7 \text{ jää } 0$$

$$7/2 = 3 \text{ jää } 1$$

$$3/2 = 1 \text{ jää } 1$$

$$1/2 = 0 \text{ jää } 1$$

loppu

$$= 11\ 1001_2$$

$$= 0011\ 1001_2$$

10-järj \Rightarrow 2-järj binääriosalle

- 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

10-järj \Rightarrow 2-järj binääriosaesimerkki

$$0.1875_{10} = ?$$

$$2 * 0.1875 = 0.375 = 0 + 0.375$$

$$2 * 0.375 = 0.75 = 0 + 0.75$$

$$2 * 0.75 = 1.5 = 1 + 0.5$$

$$2 * 0.5 = 1.0 = 1 + 0.0$$



loppu

$$= 0.0011_2$$

$$= 0.001100000000000000000000_2$$

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 = 479AF₁₆

= 0004 79AF₁₆ = 0x 479AF

16-järj: 120ADF₁₆ 1 2 0 A D F

binääri: 0001 0010 0000 1010 1110 1111

Oktaaliesimerkkejä ⁽¹³⁾

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 = 1074657₈

= 0001074657₈ = 01074657

8-järj: 120371₈

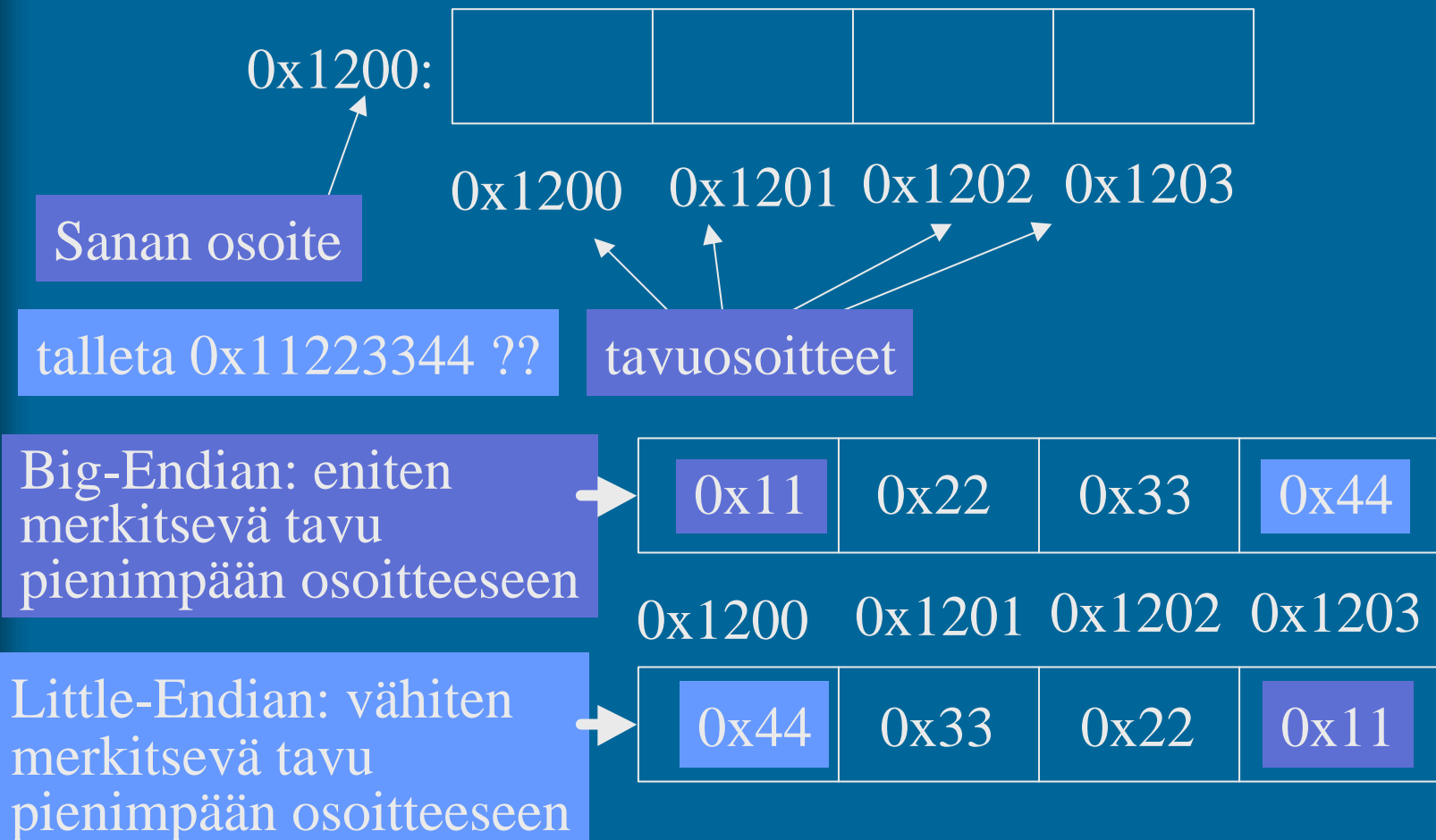
1 2 0 3 7 1

binääri:

001 010 000 011 111 001

Big vs. Little Endian ⁽³⁾

- Miten monitavuiset arvot talletetaan?



Big vs. Little Endian ⁽⁵⁾

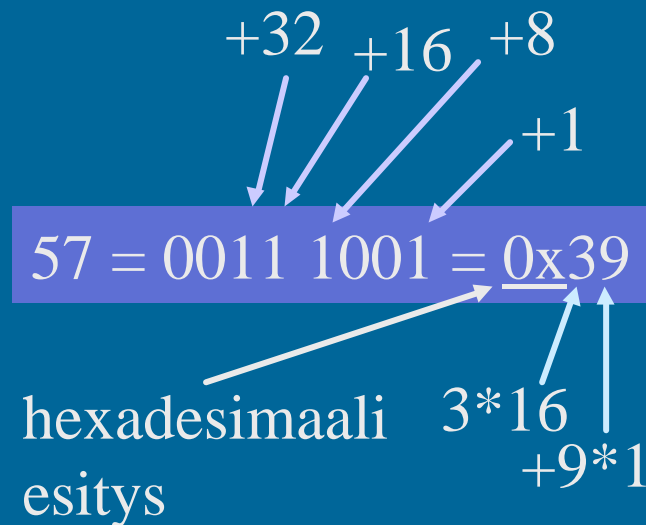
- Monitavuisen tiedon (sana-) osoite on sama molemmissa tapauksissa
- Tavujen sisäinen järjestys vaihtelee
- Suorittimen suunnittelija päättää
 - Matematiikkapiirien tulee tietää miten luvut esitetty
 - Täytyy ottaa huomioon siirrettäessä tietoa verkon yli
- Power-PC: bi-endian - molemmat moodit käytössä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

TTK-91: big-endian

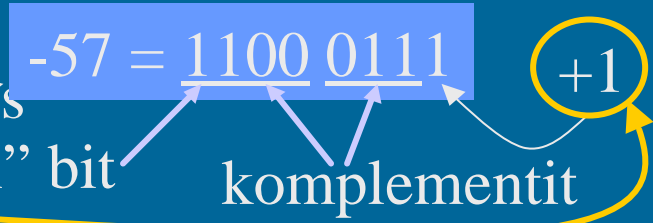
Kokonaislukujen esitysmuoto (8)

Kaikki esitetty biteillä 0 ja 1
ei etumerkkejä
ei desimaalipistettä

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



sign bit = MSB
= most significant bit



Kahden ja yhden komplementti esitykset

- Kahden komplementti käytössä useimmiten

- Tilaa 8 bittiä?

- 7 bittiä datalle
- 1 bitti “etumerkille”

$$+2 = 0000\ 0010$$

$$+1 = 0000\ 0001$$

$$0 = 0000\ 0000$$

$$-1 = 1111\ 1111$$

$$-2 = 1111\ 1110$$

- Yhden komplementti:

$$\text{ones complement: } -0 = 1111\ 11\underline{11}$$

$$-1 = 1111\ 1110$$

Liukuluvut

- Tietokoneessa ei ole realilukuja tai rationaalilukuja (matemaattiset käsitteet)
- Aina on rajallinen esityksen tarkkuus
 - lukuja π , $\sqrt{2}$, tai $1/3$ ei voi esittää tarkasti
 - luvut 1.000000000 ja luvut 1.0000000001 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

Liukulukujen esitys (4)

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

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

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

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

“+”	“14”	“1.23”
-----	------	--------

sign exponent

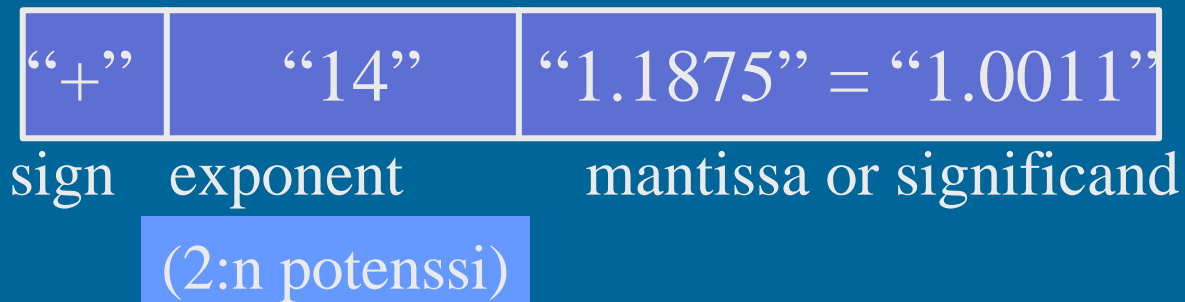
mantissa or significand

(exponentti)

(mantissa)

IEEE 32-bit Floating Point Standard ⁽³⁾

IEEE
Standard 754



- Etumerkki
 - 1 bitti etumerkille, $1 \Rightarrow \text{“-”}$, $0 \Rightarrow \text{“+”}$
 - etumerkki bitti $S \Rightarrow$ etumerkin arvo = $(-1)^S$

IEEE 32-bit FP Standard



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

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

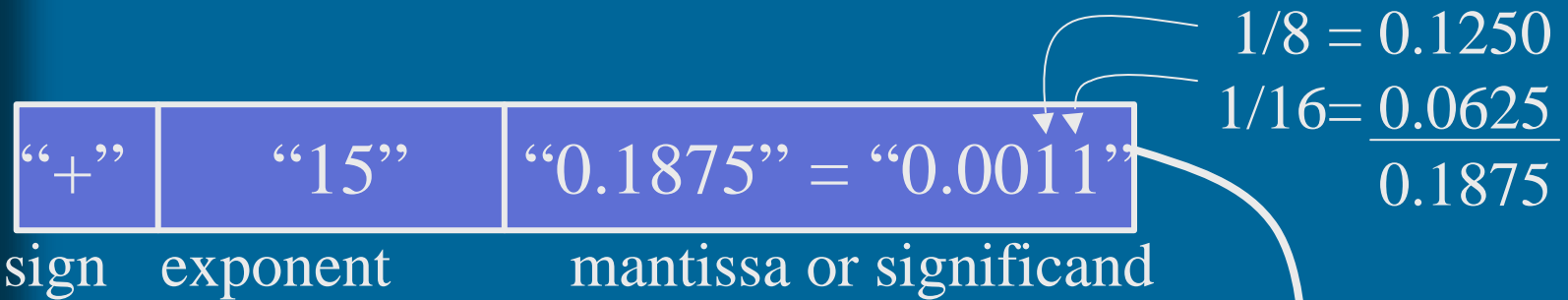
exponent = -1 $\xrightarrow{\text{store}}$ $-1+127 = 126 = 0111\ 1110$

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

– eksponentit 0 and 255 erikoistapauksia

- talletettu arvoalue: **1 - 254** \Rightarrow todellinen arvoalue: **-126 - 127**

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)

mantissa eksponentti

0.0011 “15”

1.1000 “12”

1000 “12”

24 bitin mantissa!

IEEE 32-bit FP Values (8)

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

$4+127=131$

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

IEEE 32-bit FP Values (6)



$X = ?$

$$X = (-1)^0 * 1.1111 * 2^{(128-127)}$$

$$= 1.1111_2 * 2$$

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

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

$$= 1.9375 * 2$$

$$= 3.875$$

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:



Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri

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

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)

muistiosoite tai
(pienehkö) vakio

(addressing
mode)

Konekäskyn operandit ja tulos

- Tulos: rekisteri R_j
 - paitsi WRITE- tai PUSH-käskyissä muistipaikan sisältö
 - Ensimmäinen operandi: rekisteri R_i
 - Toinen operandi
 - laske ensin arvo $R_i + ADDR$ ja käytä sitä sellaisenaan tai käytä sitä muistisoitteena
- jos $R_i = R_0$,
niin pelkkä ADDR
- arvo: $R_i + ADDR$
 - muistipaikan $M[R_i + ADDR]$ sisältö
 - muistipaikan $M[M[R_i + ADDR]]$ sisältö

Kone-
kielen
tiedon
osoitus-
moodit

Merkit ⁽⁴⁾

- 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 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 tarkoitukseen (zone) omat 8-bittiset koodinsa?

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

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

Totuusarvot ⁽⁴⁾

- Boolean TRUE ja FALSE
- Yleensä koodattu TRUE=1, FALSE=0
 - muttei aina!
- 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

Taulukot ja tietueet

- Taulukot peräkkäisrakenteena
 - kuten esimerkit aikaisemmin
 - riveittäin tai sarakkeittain
 - ei omia konekäskyjä, manipulointi aliohjelmilla tai loopeilla (paitsi ns. vektorikoneet, joilla on omia konekäskyjä vektoriopeeraatioita varten)
- Tietueet peräkkäisrakenteena
 - osoite on jonkin osoitemuuttujan arvo
 - ei omia konekäskyjä, manipulointi aliohjelmilla tai kääntäjän generoimien vakiolisäysten avulla

Oliot

- Oliot
 - kuten tietueet, yleensä varattu kasasta (heap)
 - useat olion kentistä sisältävät vuorostaan osoitteen kasasta suoritusaikana varattuun toiseen olioon
 - metodit ovat aliohjelmien osoitteita
 - ei omia konekäskyjä, manipulointi aliohjelmilla

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

Kuvat

- Kuvat ovat yleensä pakattu mahdollisimman vähän tilaa vievää muotoon
 - purkaminen vaatii laskentaa
- GIF, JPEG, TIFF, BMP,
- Ei omia konekäskyjä, manipulointi aliohjelmilla

Video kuva

- 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

Videostandardit

- MPEG (Moving Pictures Expert Group)
- AVI (Audio Visual Interleave)
- MOV, INDEO, FLI, GL, ...
- ei omia konekäskyjä, manipulointi aliohjelmilla (tai erikoisprosessoreilla, joissa erikoiskäskyjä)
 - grafiikkakortit

Äänet

- Syntetisoitu ääni tai täydellinen äänidata
- Syntetisoitu ääni
 - MIDI-käskyjä
 - Music Instrument Digital Interface
 - ”Soita nuotti N voimakkuudella V”
- ei omia konekäskyjä, manipulointi aliohjelmilla (tai erikoisprosessoreilla, joissa erikoiskäskyjä)
 - äänikortit

Ääni

- Täydellinen äänidata
 - Perustuu ääninäytteeseen 44.1 KHz taajuudella
 - 44100 näytettä / sek
 - koodataan esim. 16 bitillä (2 tavua): 88KB /sek
- Äänistandardit
 - MIDI, WAV, AU

Maku, haju, tunto ja muu data ⁽³⁾

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

-- Jakson 6 loppu --

GFF Format Summary: FBI Fingerprint Format - Netscape

File Edit View Go Communicator Help

Bookmarks Location: <http://www.ora.com/centers/gff/formats/fbi/index.htm> What's Related

FBI Fingerprint Format

Also Known As: FBI WSQ

Type	Bitmap
Colors	8-bit grayscale
Compression	Wavelet Scalar Quantization
Maximum Image Size	64Kx64K
Multiple Images Per File	No
Numerical Format	Big-endian
Originator	U.S. Federal Bureau of Investigation
Platform	All
Supporting Applications	Many
See Also	None

Usage
The standard file format used by the FBI for storage and interchange of grayscale fingerprint images.

Comments
If you need to store compressed fingerprint or similar images then this is your format.

Document: Done