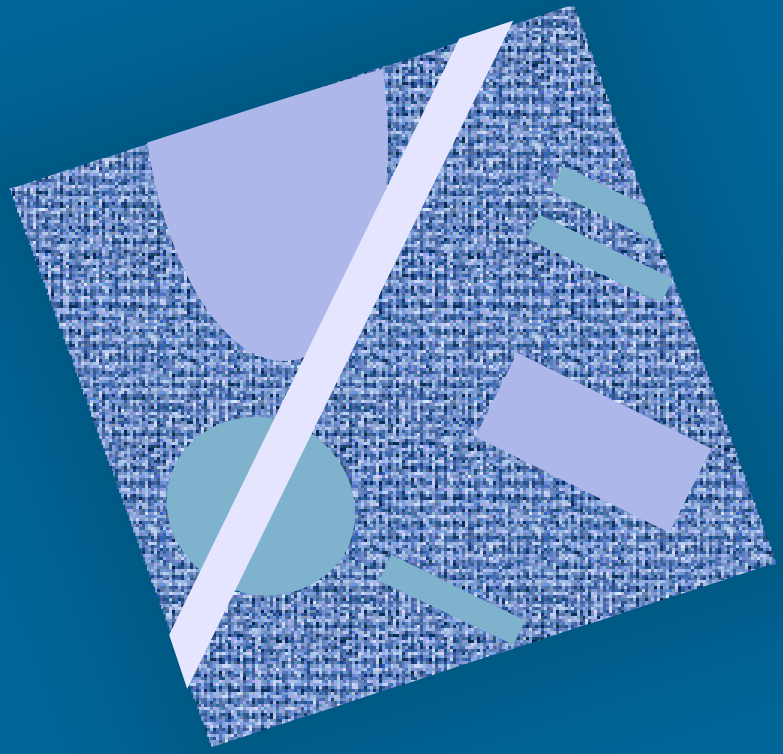
Luento 6
Tiedon esitysmuodot



Lukujärjestelmät

Luvut, merkit, merkkijonot, totuusarvot, oliot

Kuvat, äänet, hajut(?)

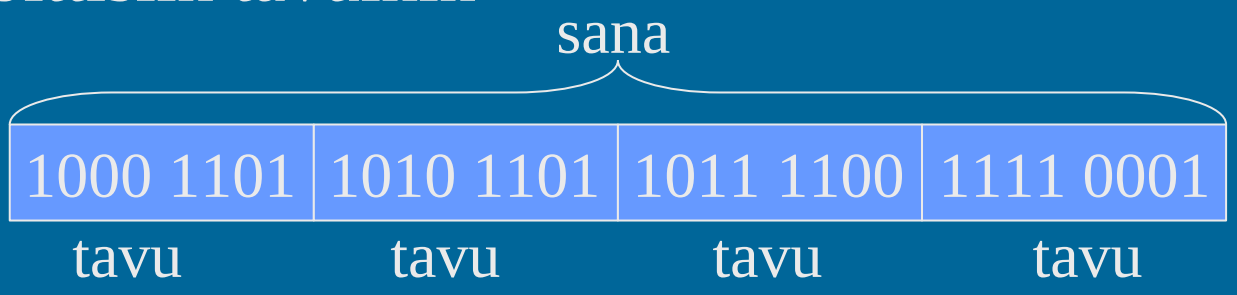
Ohjelmat

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

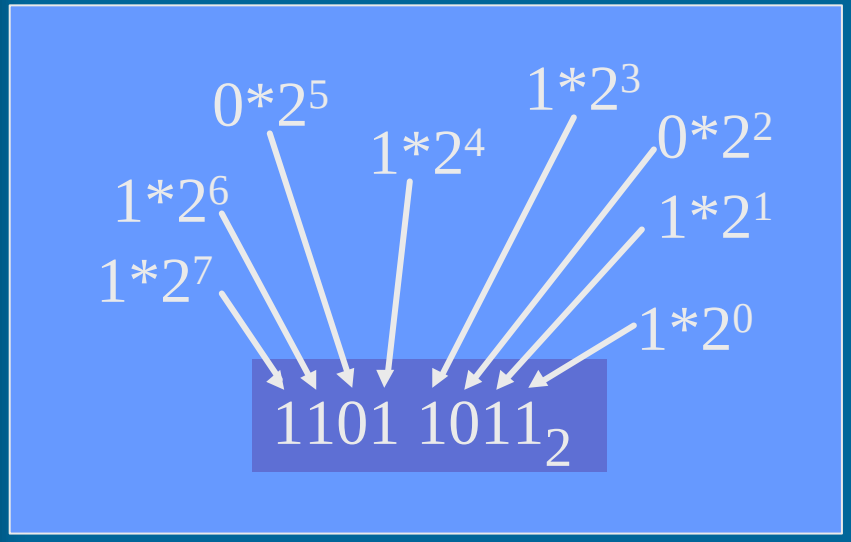
Suorittimen ymmärtämä tieto (9)

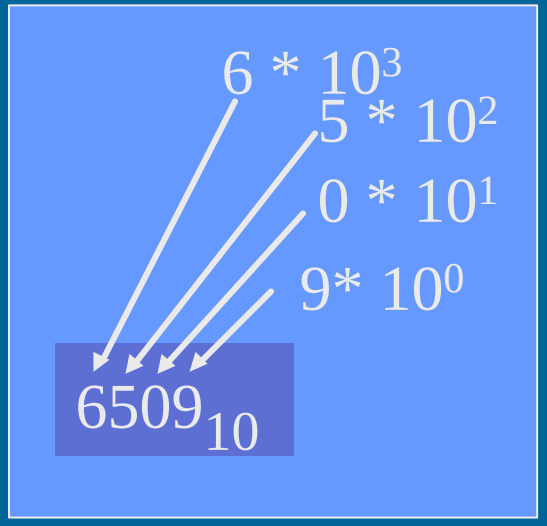
- Kaikki tieto koneessa on koodattuna biteiksi
- Muistissa voidaan esittää kaikki tieto sovitulla esitystavalla (koodauksella)
- Suoritin osaa tehdä operaatioita joillakin esitystavoilla koodatuille tiedoille
  - kokonaisluvut ja liukuluvut (aina)
  - totuusarvot, merkit ja merkkijonot (joskus)
  - kuvat,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

TTK-91: kokonaisluvut

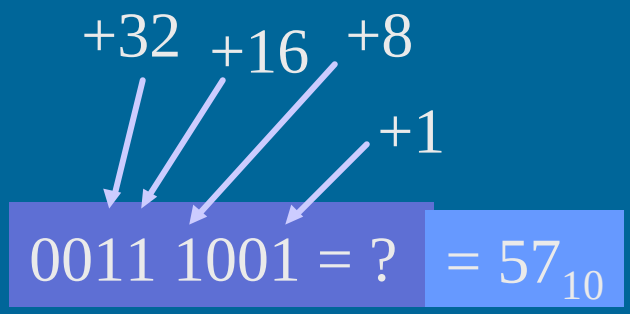
Binäärijärjestelmä (2)

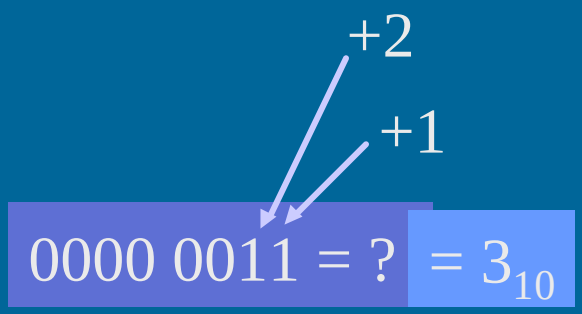
- Kantaluku 2, numerot 0 ja 1
  - numeroiden painoarvot oikealta vasemmalle:
    $1=2^0$, $2=2^1$, $4=2^2$, $8=2^3$, $16=2^4$, $32=2^5$, …
  - kymmenjärjestelmässä painoarvot ovat
    $1=10^0$, $10=10^1$, $100=10^2$, $1000=10^3$, …

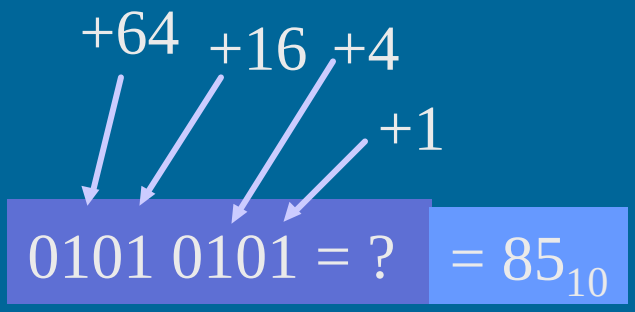




Binäärilukuesimerkkejä

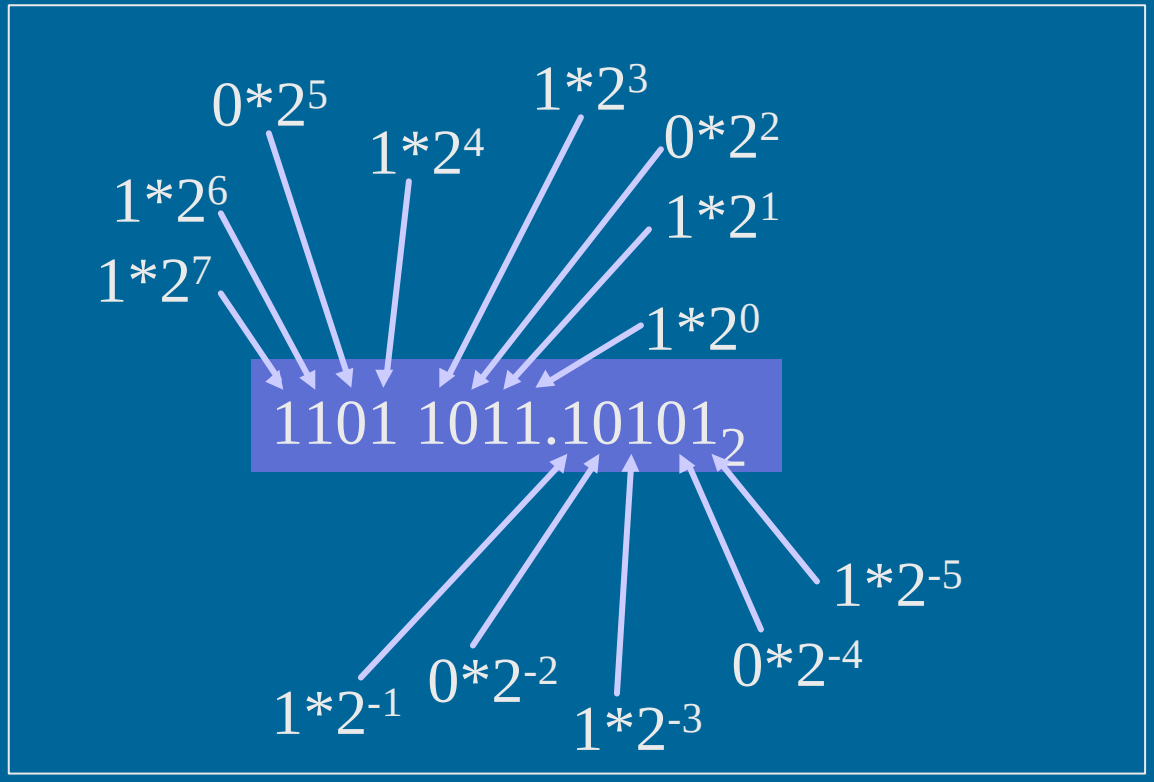




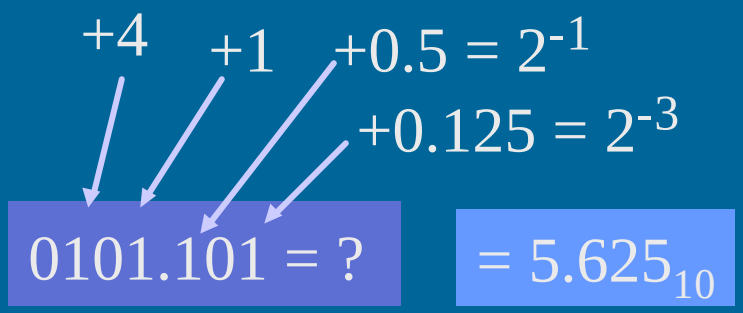


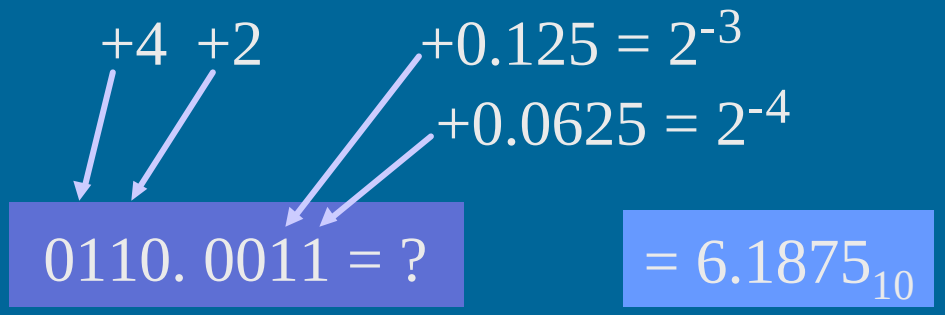
Binääripiste (2)

- Binääriluvuilla voi olla myös binääriosa (vrt. desimaaliosa)



Binääripiste-esimerkkejä

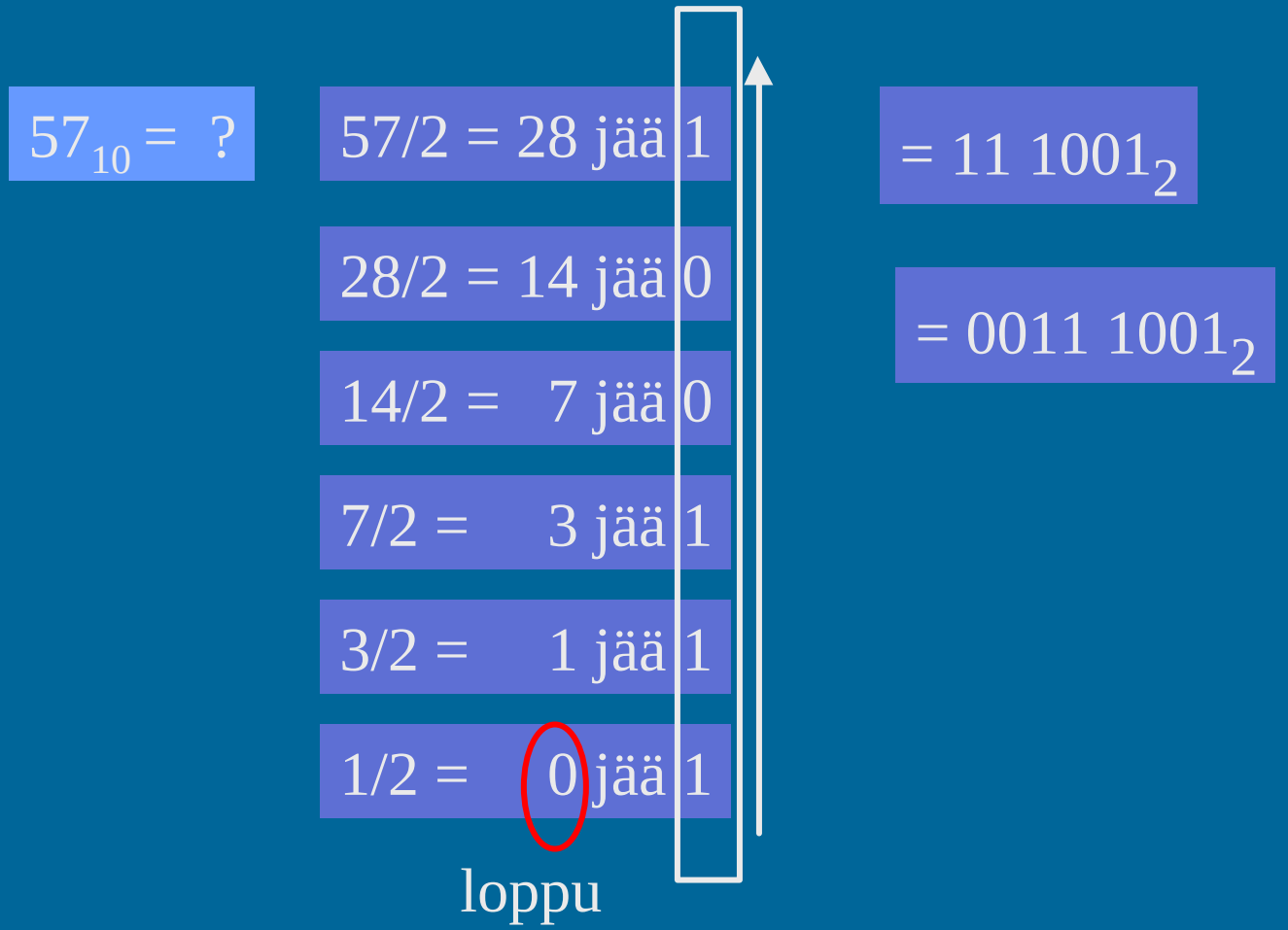




Muunnokset lukujärjestelmien välillä (5)

- 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 ⇒ 2-järj kokonaislukuesimerkki (11)



10-järj ⇒ 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 ⇒ 2-järj binääriosaesimerkki

$0.1875_{10} = ?$

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

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

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

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

loppu

$= 0.0011_2$

$= 0.00110000000000000000_2$

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

$= 479AF_{16}$

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

16-järj: $120ADF_{16}$ 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 $= 1074657_8$

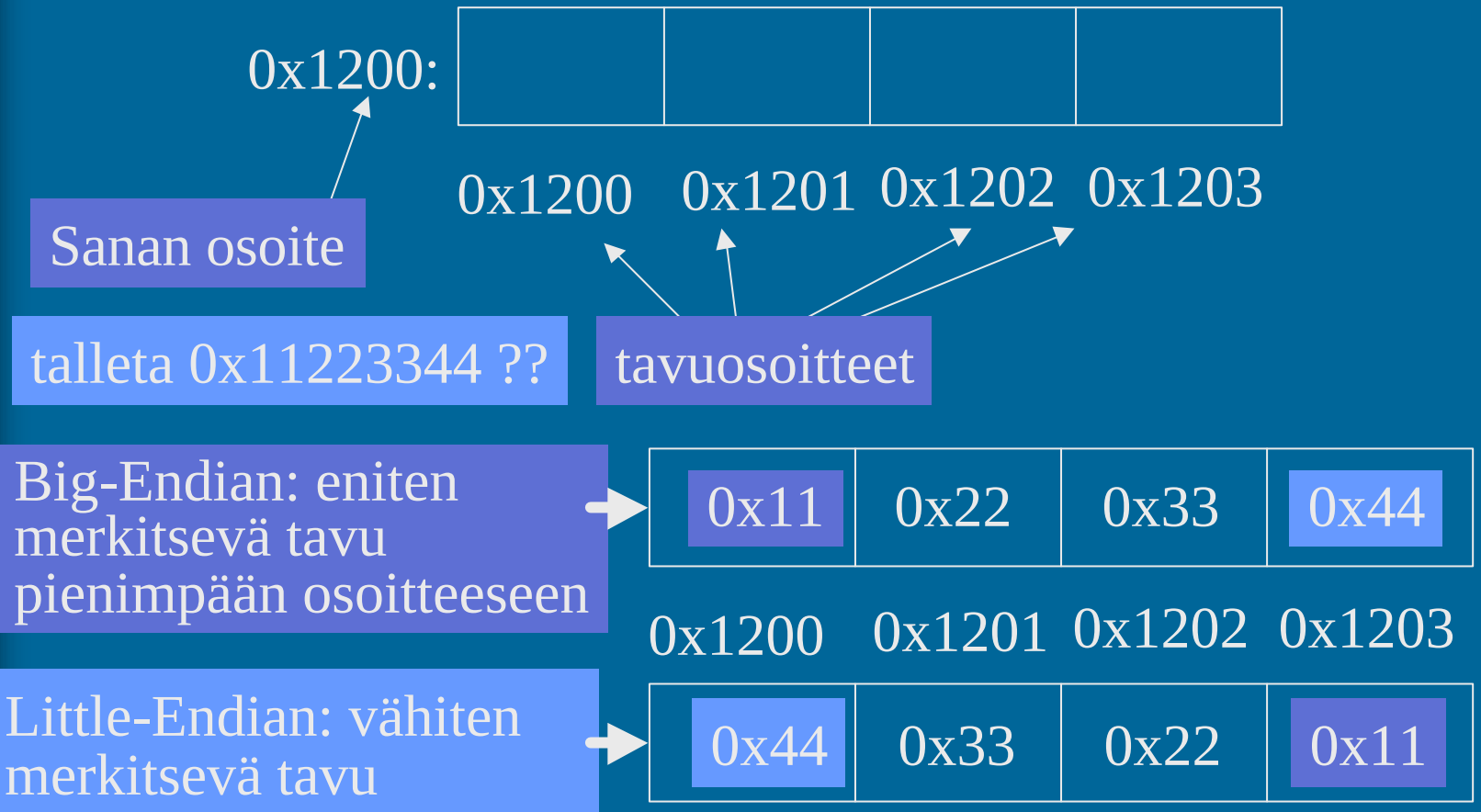
$= 0001074657_8 = \underline{0}1074657$

8-järj: 120371_8 1 2 0 3 7 1

binääri: 001 010 000 011 111 001

Big vs. Little Endian (3)

- Miten monitavuiset arvot talletetaan?



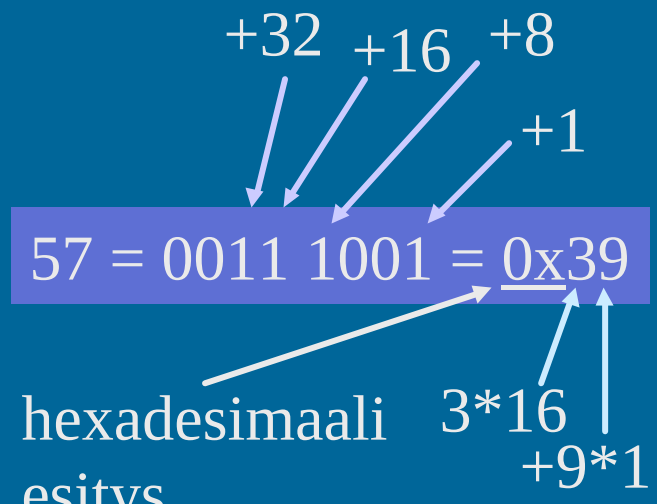
Big vs. Little Endian (5)

- 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

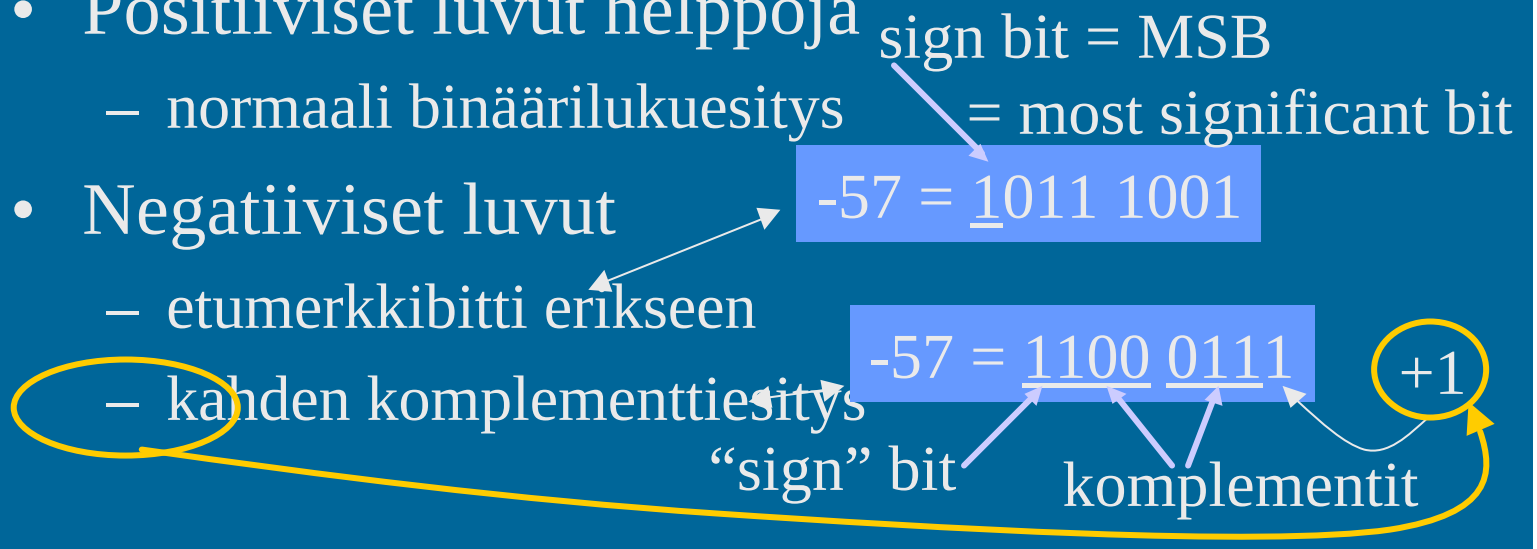
TTK-91: big-endian

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



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:

ones complement: -0 = 1111 1111
-1 = 1111 1110

Liukuluvut

- Tietokoneessa ei ole realilukuja tai rationaalilukuja (matemaattiset käsitteet)
- Aina on rajallinen esityksen tarkkuus
  - lukuja π, **p**2, tai 1/3 ei voi esittää tarkasti
  - 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

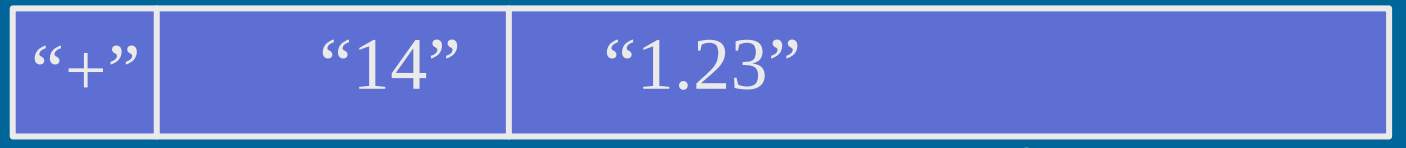
Liukulukujen esitys (4)

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

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

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

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



sign exponent mantissa or significand

(exponentti) (mantissa)

IEEE 32-bit Floating Point Standard (3)

IEEE Standard 754

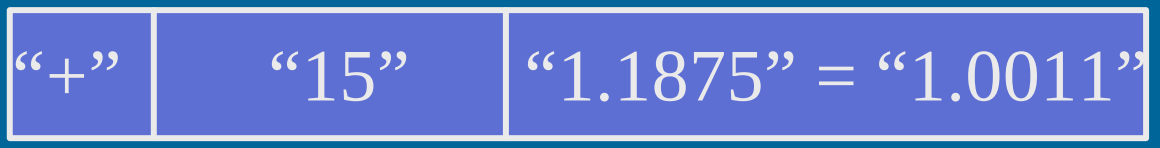
“+”	“14”	“1.1875” = “1.0011”

sign exponent mantissa or significand

(2:n potenssi)

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

IEEE 32-bit FP Standard

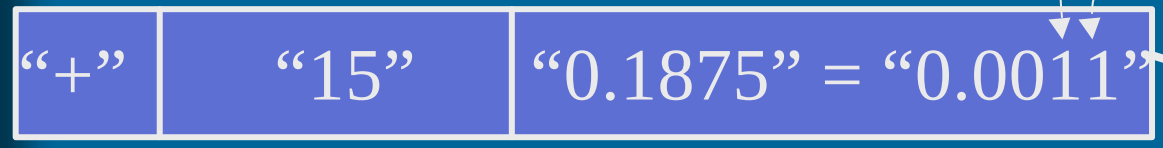


- 8 bittiä eksponentille, lisättynä 127:llä (biased form)

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

  - eksponentit 0 and 255 erikoistapauksia
    - talletettu arvoalue: **1 - 254** ⇒ todellinen arvoalue: **-126 - 127**

IEEE 32-bit FP Standard (7)



1/8 = 0.1250
1/16= 0.0625
0.1875

sign exponent mantissa or significand

- 23 bittiä mantissalle, siten että ...

1) Binääripiste (.) on heti ensimmäisen bitin jälkeen

2) Mantissa on normalisoitu: vasemmanpuolimmainen bitti on1

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)

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

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

IEEE 32-bit FP Values (6)

sign
1 bit

exponent
8 bits

mantissa or significand
23 bits

X = ?

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

$= 1.1111_2 * 2$

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

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

$= 1.9375 * 2$

$= 3.875$

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:

muistiosoite tai (pienehkö) vakio

Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri

M = muistinoutojen määrä toiseen operandiin (addressing mode)
(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)

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

Kone-kielen tiedon osoitus-moodit

jos Ri = R0,
niin pelkkä ADDR

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 tarkoituksiin (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

  - 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 vektorioperaatioita 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 videokuvaa 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, veneen tyyppi, tunteen palo, ….

ks. HS artikkeli 5.5.2000

- Sovelluskohtaisesti, ei vielä yleisiä standardeja
  - kokonaisluvut (diskreetti data)
  - liukuluvut (jatkuva data)
- Ei omia konekäskyjä, manipulointi omilla aliohjelmilla

-- Jakson 6 loppu --