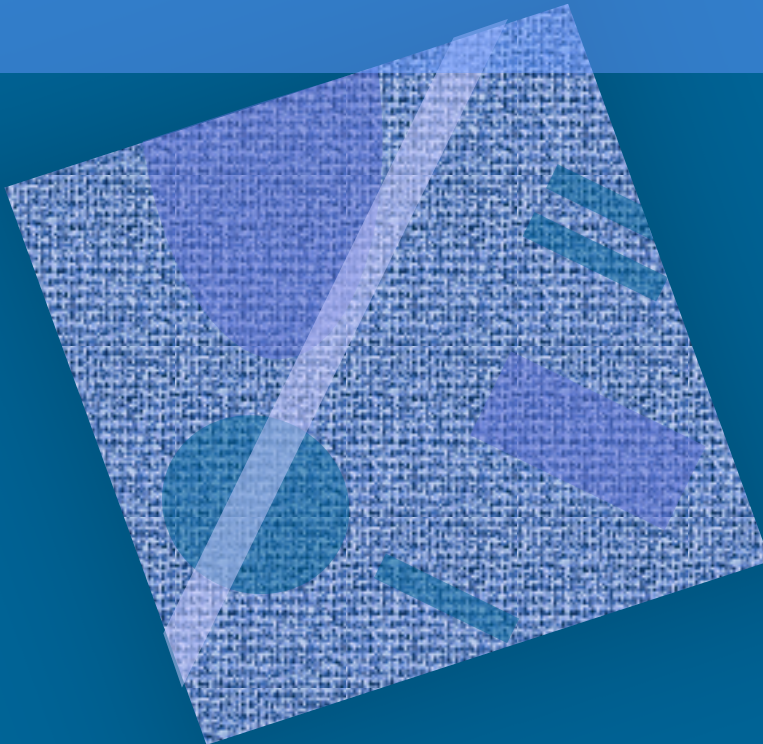


Tiedon esitysmuodot



Lukujärjestelmät

Kokonaisluvut, liukuluvut

Merkit, merkkijonot

Äänet, kuvat, muu tieto

Monitavuinen tieto

Ohjelman esitysmuoto

Rakenteellinen tieto

Tiedon tyypit

- **Kommunikointi ihmisen kanssa**
 - kuva, ääni, merkit, ...
- **Laitteiston sisäinen talletus**
 - kuvaformaatit, ääniformaatit, 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

Tiedon esitys

- 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
 - samalle tietotyypille voi olla useita koodausmenetelmiä
 - kokonaisluvut, liukuluvut, merkit, merkkijonot, kuvat, ...
 - ongelma: ymmärtävätkö koneet toisiaan?
 - tiedon esitysmuotoa voidaan joutua muuttamaan, kun tietoa siirretään koneelta toiselle

Tiedon esitys laitteistossa

- 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 (ennen 16, 32, 48, 64 bittiä, ...)
- Yleensä sana on jaettu tasapituisiin 8-bittisiin tavuihin (byte)



Suorittimen ymmärtämä tieto

- Kaikki tieto koneessa on koodattuna biteiksi
- Muistissa voidaan esittää kaikki tieto millä tahansa sovitulla esitystavalla (koodauksella)
- Joillekin esitystavoille on omat konekäskyt (eli suoritin ymmärtää niitä)
 - Kokonaisluvut ja liukuluvut (lähes aina)
 - Totuusarvot, merkit ja merkkijonot (joskus)
 - Kuvat ja äänet (ei yleensä ellei erikoistunut suoritin)
- Muiden tietojen käsittely tapahtuu ohjelmallisesti (eli usea käsky tai aliohjelma tai metodi)
 - Esim. merkkejä (merkkien esitysmuotoa) voidaan käsitellä kokonaislukuoperaatioilla ja aliohjelmilla
 - Rationaaliluvut, 128-bittiset kokonaisluvut (?), isot taulukot, tietueet, oliot, sormenjäljet, äänet, kuvat, hajut, ...

TTK-91:
kokonaisluvut

Numeromuunnokset

- Binääriluvuista kymmenjärjestelmään

10110011 \rightarrow 179

- Kymmenjärjestelmästä binääriluvuiksi

179 \rightarrow 10110011

- Binääriluvusta heksadesimaaliluvuksi

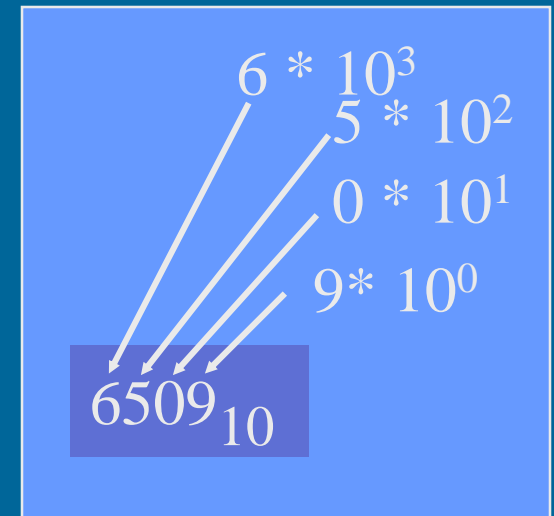
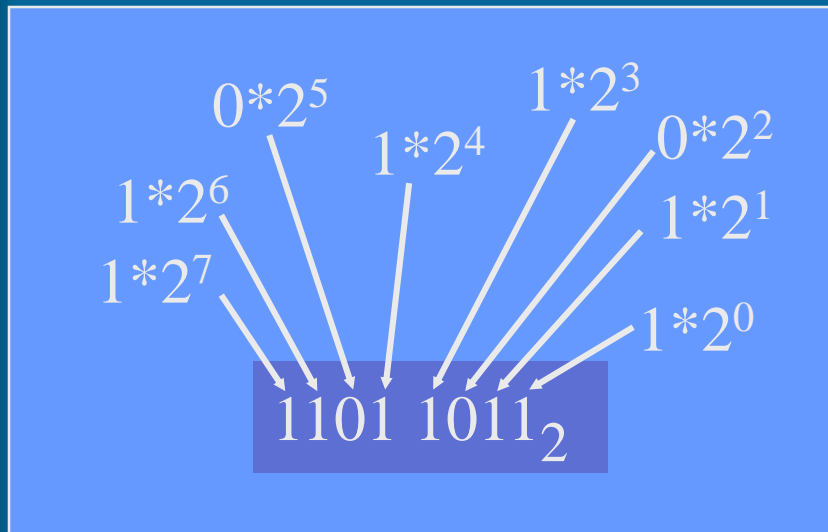
10110011 \rightarrow B3 (tai 0xB3)

- Heksadesimaaliluvusta binääriluvuksi

0xB3 = B3 \rightarrow 10110011

Binäärijärjestelmä

- 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 ⁽⁹⁾

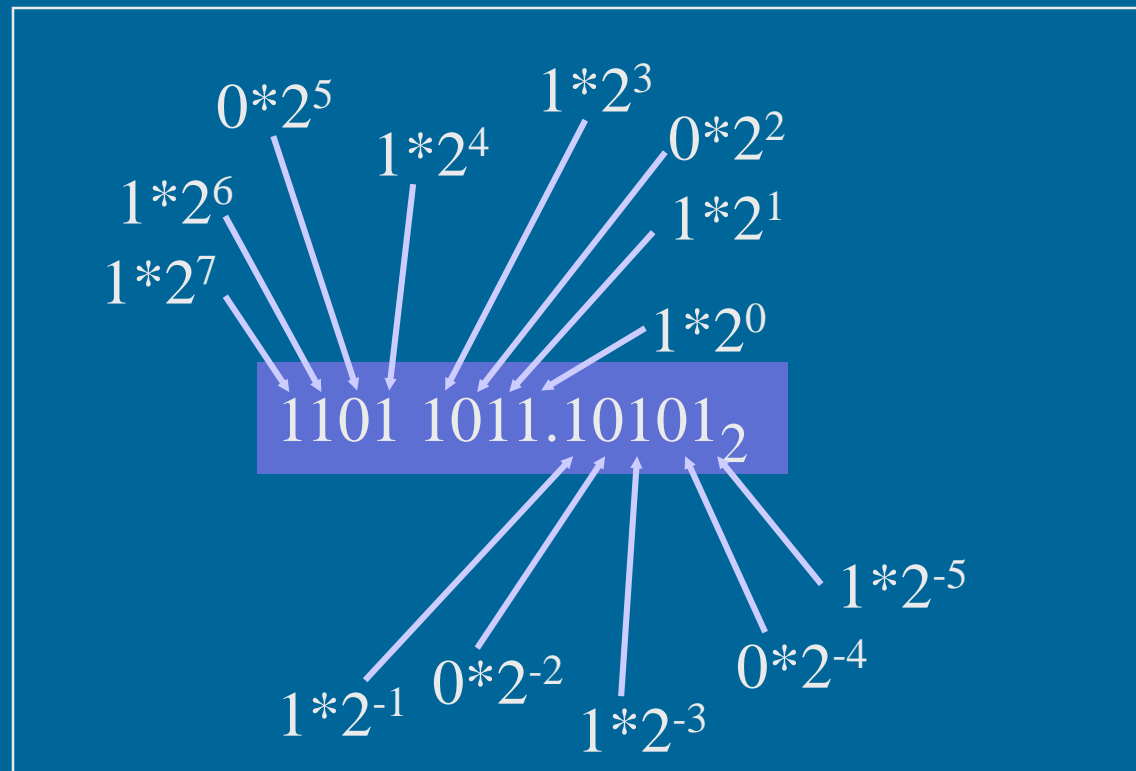
$$\begin{array}{ccccccc} & +32 & +16 & +8 & & & \\ & \swarrow & \swarrow & \swarrow & \swarrow & & \\ 0011 & 1001 & = ? & & = 57_{10} & & \\ & \swarrow & \swarrow & \swarrow & \swarrow & & \\ & +1 & & & & & \end{array}$$

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

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

Binääripiste

- 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

- 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 desimaaliosa \Rightarrow binääriosa

- 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 desimaaliosa \Rightarrow binääriosa

$$0.1875_{10} = ?$$

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

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

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

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

↓
loppu

$$= 0.0011_2$$

$$= 0.001100000000000000000000_2$$

Heksadesimaaliesitys

- 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

binääri:

0100 0111 1001 1010 1111

16-järj:

4 7 9 A F

= 479AF₁₆

= 0004 79AF₁₆ = 0x 479AF

16-järj:

120ADF₁₆

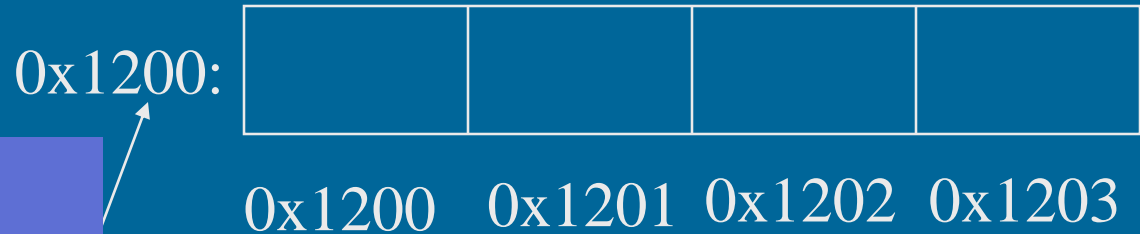
1 2 0 A D F

binääri:

0001 0010 0000 1010 1110 1111

Big vs. Little Endian

- Miten monitavuinen tieto talletetaan?
 - Yleensä per sana tai kaksoissana



sanan osoite
(pienin tavuosoite)

talleta 0x11223344 ??

tavuosoitteet

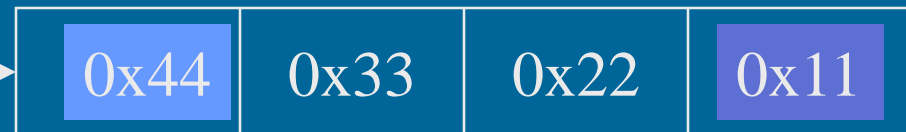
0x11223344

Big-Endian: eniten
merkitsevä tavu
pienimpään osoitteeseen



0x1200 0x1201 0x1202 0x1203

Little-Endian: vähiten
merkitsevä tavu
pienimpään osoitteeseen



0x44332211

“normaali”
tapa

Kokonaisluvut

arvo talletus
 $+57 = 0011\ 1001$

Yleensä positiiviset suoraan binäärinä

- Etumerkkibitti erikseen

sign bit = MSB
= most significant bit

luku $-57 = \underline{1}011\ 1001$ talletusmuoto

- Yhden komplementtiesitys

$-57 = 1100\ 0110$

“sign” bit

- **Kahden** komplementtiesitys

$-57 = 1100\ 0111$

“sign” bit

- Vakiolisäys

– Esim. lisää 127 ($=2^7 - 1$)

• yleensä: $2^{\text{bittilkm}-1} - 1$

– Talleta etumerkittömänä

$-57 = 0100\ 0110$

$-57 + 127 = 70$

$+57 = 1011\ 1000$

$+57 + 127 = 184$

Liukuluku

- Vastaa reaalitylukua tietokoneessa
- Kompromissi
 - Suuruusluokka vs. tarkkuus
- Aina kiinteä maksimitarkkuus
 - Sama määrä merkitseviä numeroita
- Etumerkki, mantissa, suuruusluokka

+5678901.2345678 vs. +5.678901 * 10⁶

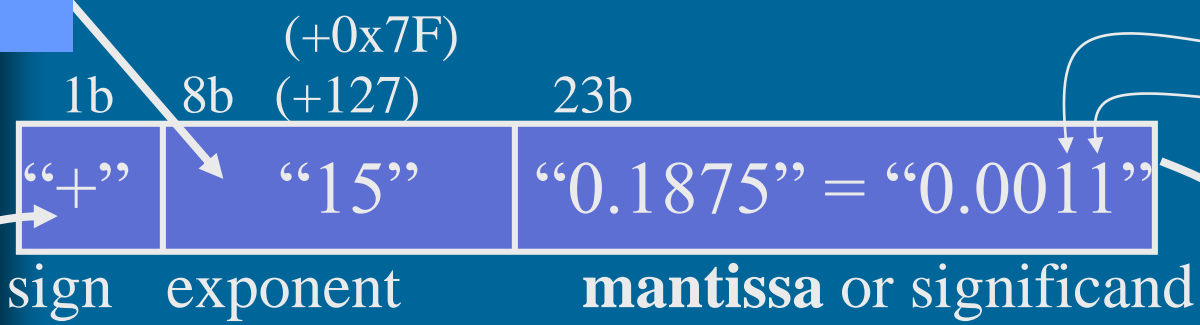
-0.000012345678 vs. -1.234568 * 10⁻⁵

+111.1010101111 vs. +1.11101011 * 2²

IEEE 32-bit FP Standard

vakiolisäys
+127

$$+6144.0_{10} = +1\ 1000\ 0000\ 0000.0_2 = +0.0011 * 2^{15} = \dots$$



$$\begin{aligned} 1/8 &= 0.1250 \\ 1/16 &= \frac{0.0625}{0.1875} \end{aligned}$$

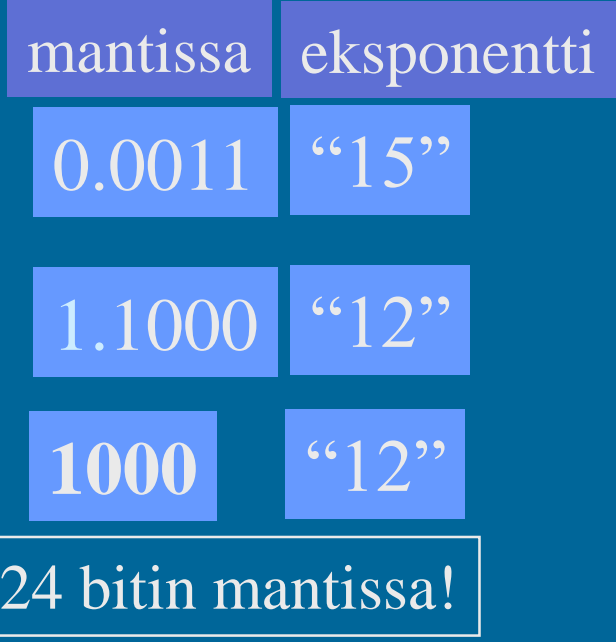
0 = '+',
1 = '-'

- 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)



Miksi käytetään piilobittii?

IEEE 32-bit FP Values

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

$$4+127=131$$

0x41B4 0000



sign exponent mantissa or significand
 1 bit 8 bits 23 bits

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

$$0+127 = 127$$

0x3F80 0000



sign exponent mantissa or significand
 1 bit 8 bits 23 bits

IEEE 32-bit FP Values

0x40780000



sign 1 bit exponent 8 bits mantissa or significand 23 bits

X = ?

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

$$= 1.1111_2 * 2$$

$$= 11.111_2$$

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

$$= 2 + 1 + 1/2 + 1/4 + 1/8$$

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

$$= 3.875$$

$$= 1.9375 * 2$$

$$= 3.875$$

IEEE standardin erikoistapauksia

- ± 0

0/1 0000 0000 0...0

- $\pm \infty$

0/1 1111 1111 0...0

?...?1?...?

(ei kaikki nolliä)

- Quiet NaN

0/1 1111 1111 1?...?1?...?

- Signaling NaN

0/1 1111 1111 0?...?1?...?

- Hyvin pienet, normalisoimattomat luvut

- Exponentti: 2^{-126}

- Piilobitti: 0

0/1 0000 0000 ??...?1?...?

UCS ja Unicode

- UCS - Universal Character Set
- Samat merkistöt, eri standardit
- 2 tavua eli 16 bittiä per merkki
 - 65536 merkkiä koko maailmassa käytössä oleville n. 200000 symbolille
- (32-bit UCS-4 sisältää myös kaikki kiinalaiset merkit)
- 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, esim. UTF-8

Merkkijonot

- 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 (yleensä enää) omia konekäskyjä, manipulointi aliohjelmilla
 - kokonaisluku- ja bittimanipulointikäskyt
 - joissakin koneissa `"strcpy"` ja `"strcmp"` konekäskyt (oletettu jokin tietty merkkijonojen esitysmuoto)

Totuusarvot

- Boolean TRUE ja FALSE
- Yleensä koodattu TRUE=1, FALSE=0
 - muttei aina!
 - Totuusarvolauseke $A \text{ and } B = \text{kokon.lukulauseke } A * B$
- Usein Boolean arvo per sana
 - loput 31 bittiä nollia
 - ohjelmointikielten Boolean muuttujat
- Joskus pakatussa muodossa 32 arvoa per sana
- Ei omia konekäskyjä, manipulointi bittimanipulaatiokäskyillä ja aliohjelmilla
 - Bittimanipulaatiokäskyt ovat yleensä kaikille sanan (tavun) biteille!

Kuvat

- Monta kuvastandardia
 - GIF, JPEG, TIFF, BMP,
 - Yleisyys, siirrettävyys, pakkaustiheys
 - Näyttöä varten tarvittavan laskennan määrä
 - Kuvatiedoston alussa otsake kertoo talletusformaatin
- Pakattu (usein) mahdollisimman vähän tilaa vievää muotoon
 - Optimoitu tilan (tiedonsiirron), ei laskennan mukaan
 - Purkaminen voi vaatia paljon laskentaa
- Ei omia konekäskyjä, manipulointi aliohjelmilla ja/tai grafiikkasuorittimilla (näytönohjaimilla)

Videokuva

- Monta standardia
 - MPEG (Moving Pictures Expert Group)
 - AVI (Audio Visual Interleave)
 - MOV, INDEO, FLI, GL, DVD, ...
- Ei omia konekäskyjä, manipulointi aliohjelmilla ja/tai grafiikkasuorittimilla (näytönohjaimilla)

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 ja/tai erikoissuorittimilla (äänikorteilla)
 - Voi olla integroitu näytönohjaimeen/emolevyyn

Maku, haju, tunto ja muu data

- 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

Konekäskyjen esitysmuoto

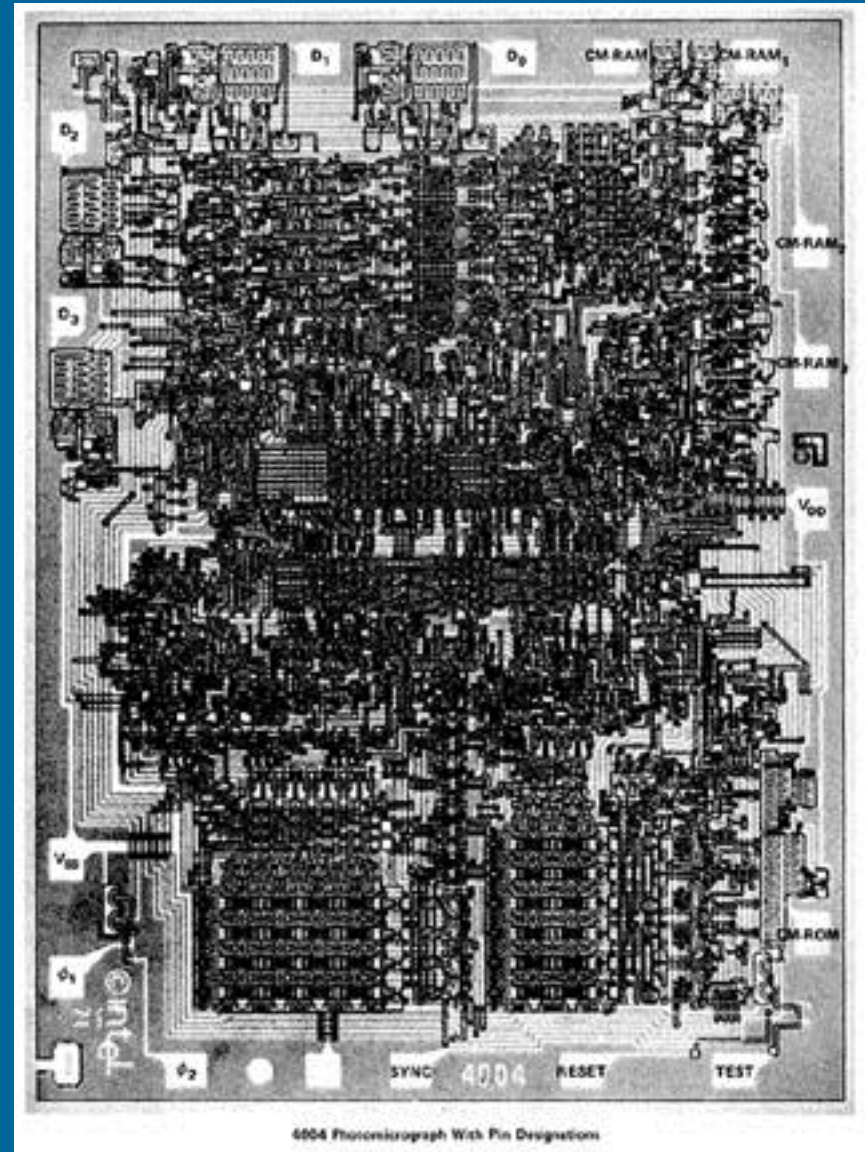
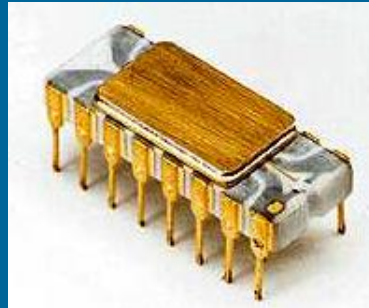
- Konekohtainen, jokaisella omansa
- Käskyt ovat 1 tai useamman tavun mittaisia
 - SPARC, kaikki käskyt: 1 sana eli 4 tavua
 - ARM, kaikki käskyt: 1 sana eli 4 tavua
 - Pentium II: 1-16 tavua, paljon variaatioita
- Käskyillä on yksi tai useampi muoto, kussakin tietty määrä erilaisia kenttiä
 - opcode, Ri, Rj, Rk, osoitusmoodi
 - pitkä tai lyhyt vakio

TTK-91, kaikki käskyt: 1 sana, 1 muoto

-- Loppu --

Intel 4004, 1971

- Faggin, Hoff, Mazor
- Ens. mikroprosessori
- 3x4 mm, \$200
- 2300 transistoria
- 4 bitin sana
- Laskinta varten
- Sama laskentateho kuin Eniacilla (18000 tyhjiöputkea)



4004 Photomicrograph With Pin Designations