

Kertausluento 2 (lu04, lu05, lu06)

Aliohjelmien toteutus

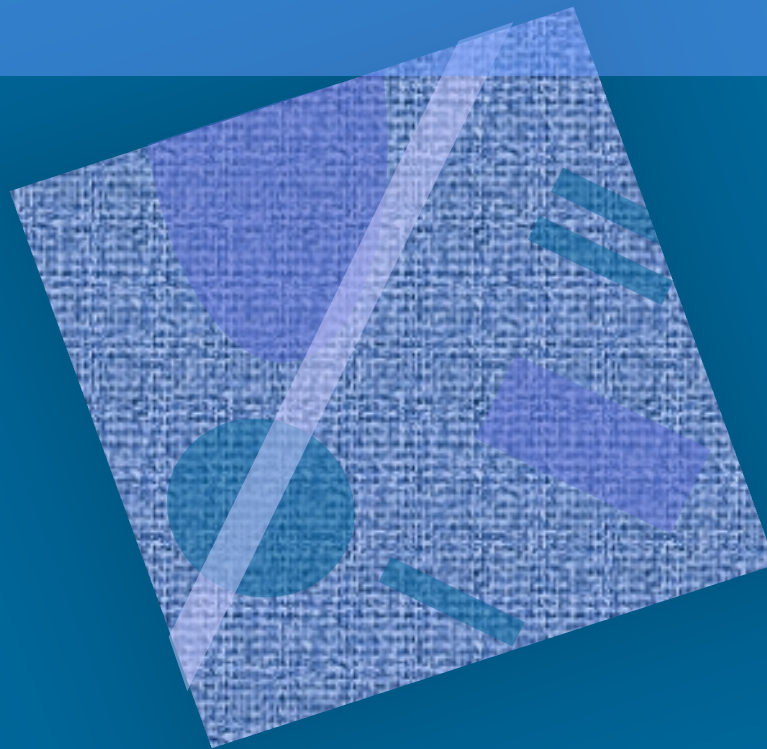
Suoritin, väylä, tiedon esitys



Tyypit, Parametrit
Aktivointitietue (AT)
AT-pino
Käskyjen suoritusyksi
Suorittimen tilat
Poikkeukset ja keskeytykset
Kokonaisluvut
Liukuluvut
Muut tiedot

Luento 4

Aliohjelmien toteutus



Tyypit
Parametrit
Aktivointitietue (AT)
AT-pino
Rekursio

Aliohjelmatyypit

- Korkean tason ohjelmointikielen käsitteet
 - aliohjelma, proseduuri
 - parametrit
 - funktio
 - parametrit, paluuarvo
 - metodi
 - parametrit, ehkä paluuarvo
- Konekielen tason vastaavat käsitteet
 - aliohjelma
 - parametrit ja paluuarvo(t)

Parametrit ja paluuarvo

- Muodolliset parametrit

- määritelty aliohjelmassa ohjelmointihetkellä
- tietty järjestys ja tyyppi
- paluuarvot

```
Tulosta (int x, y)  
Laske(int x): int
```

- käsittely hyvin samalla tavalla kuin parametreillekin

- Todelliset parametrit ja paluuarvo

- todelliset parametrit sijoitetaan muodollisten parametrien paikalle kutsuhetkellä suoritusaikana
- paluuarvo saadaan paluuhetkellä ja sitä käytetään kuten mitä tahansa arvoa

```
Tulosta (5, apu);  
x = Laske( y+234);
```

Parametryypit

- Arvoparametri
 - välitetään parametrin arvo (eli sen kopio) kutsuhetkellä
 - arvo voidaan lukea
 - alkuperäistä arvoa ei voi muuttaa, mutta arvon kopiota voi muuttaa
- Viiteparametri
 - välitetään parametrin osoite
 - arvo ja osoite voidaan lukea, arvoa voi muuttaa
- Nimiparametri
 - välitetään parametrin nimi
 - nimi (merkkijono) kuvataan arvoksi kutsuhetkellä
 - semantiikka määräytyy vasta kutsuhetkellä

```
Swap(i, k);      Swap(i, T[i]);
```

```
tmp = i;  
i = T[i];  
T[i] = tmp; # ?
```

Aliohjelmien toteutuksen osat

- Paluuosoite
 - kutsukohtaa seuraava käskyn osoite
- Parametrien välitys
- Paluuarvon välitys
- Paikalliset muuttujat
- Rekistereiden allokointi (varaus)
 - kutsuvalla ohjelman osalla voi olla käytössä rekistereitä, joiden arvon halutaan säilyä!
 - pääohjelma, toinen aliohjelma, sama aliohjelma, metodi, ...
 - käytettyjen rekistereiden arvot pitää aluksi tallettaa muistiin ja lopuksi palauttaa ennalleen

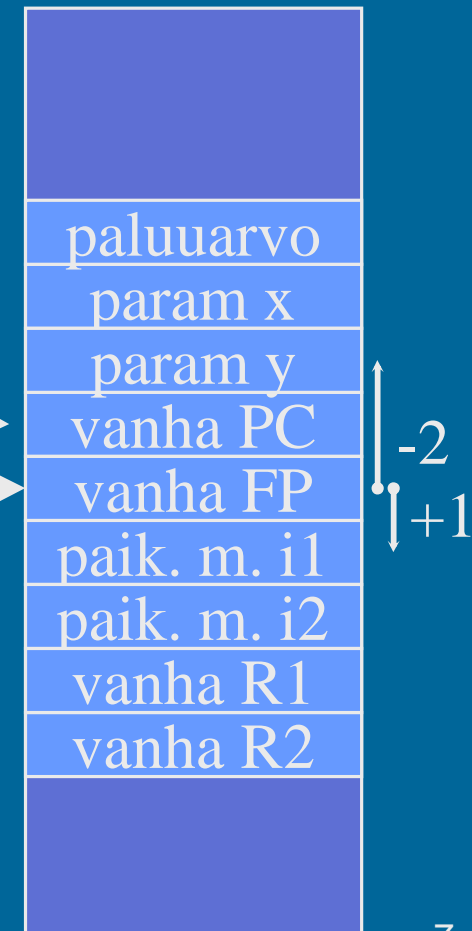
Aktivointitietue

(activation record,
activation frame)

```
int funcA (int x,y);
```

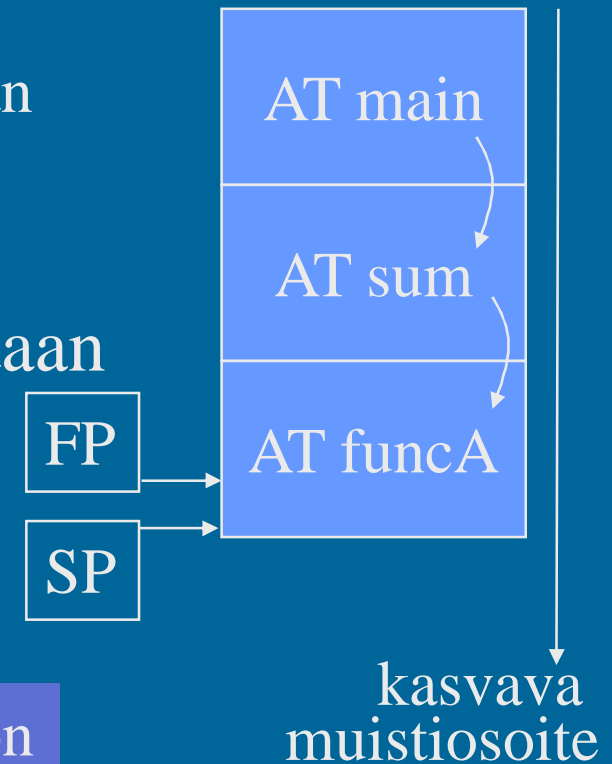
- Aliohjelman toteutusmuoto (ttk-91)

- funktion paluuarvo
(tai kaikki paluuarvot)
- kaikkien (sisäänmeno- ja ulostulo-)
parametrien arvot
- paluuosoite
- kutsukohdan aktivointitietue **FP**
- kaikki paikalliset muuttujat ja
tietorakenteet
- aliohjelman ajaksi talletettujen
rekistereiden alkuperäiset arvot



Aktivointitietueiden hallinta

- Aktivointitietueet (AT) varataan ja vapautetaan dynaamisesti (suoritusaikana) pinosta (muistista)
 - SP (=R6) osoittaa pinon pinnalle
- Aktivointitietuepino
 - FP (R7) osoittaa voimassa olevan AT:n sovittuun kohtaan (ttk-91: vanhan FP:n osoite)
- Pinossa olevaa AT:tä rakennetaan ja puretaan käskyillä:
 - PUSH, POP, PUSHR, POPR
 - CALL, EXIT (SVC, IRET)



Aliohjelman käytön toteutus

- Toteutus jaettu eri yksiköille

Kutsuva
rutiini

– varaa tilaa paluuarvolle pinosta

– laita parametrit (arvot tai osoitteet) pinoon

– talleta vanha PC ja FP, aseta uudet PC ja FP

CALL
käsky

– varaa tilaa paikallisille muuttujille

– talleta käytettävien rekistereiden
vanhat arvot pinoon

prolog

Kutsuttu
rutiini

– (itse aliohjelman toteutus – varsinainen työ)

– palauta rekistereiden arvot

– vapauta paikallisten muuttujien tila

epilog

EXIT
käsky

– palauta PC ja FP

– vapauta parametrien tila

Kutsuva
rutiini

– ota paluuarvo pinosta

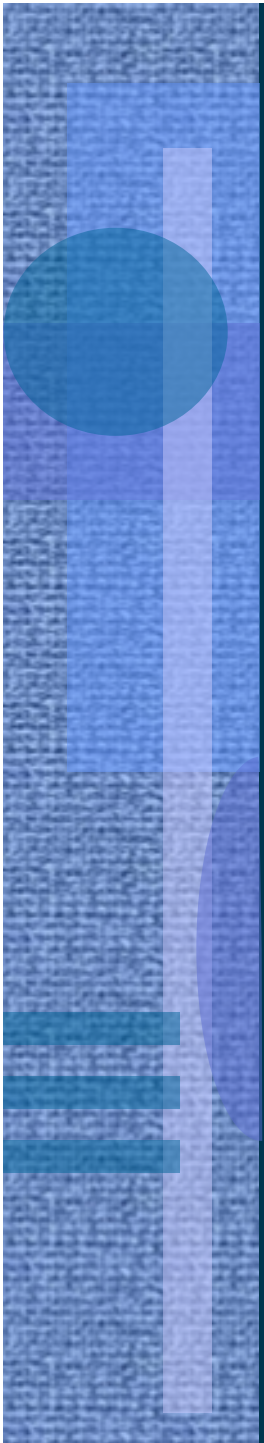
KJ-palvelun kutsu (proseduraalisesti)

(kesk.)

- Samalla tavalla kuin aliohjelman kutsu
 - CALL käskyn asemesta SVC
- Tila paluuarvolle?
- Parametrit pinoon vai rekistereissä?
- SVC kutsu
 - Kutsuttavan rutiinin numero operandina
- IRET paluu
- Paluuarvo (OK, virhe) pois pinosta tarkistusta varten

fOK = ReadBlock (fp, 64)

```
...  
PUSH SP, =0 ;paluuarvo  
PUSH SP, =FileBuffer  
PUSH SP, CharCnt  
PUSH SP, FilePtr  
  
SVC SP, =ReadFile  
  
POP SP, R1  
JNZER R1, FileTrouble  
...
```



26.3.2010

Copyright Teemu Kerola 2010

11

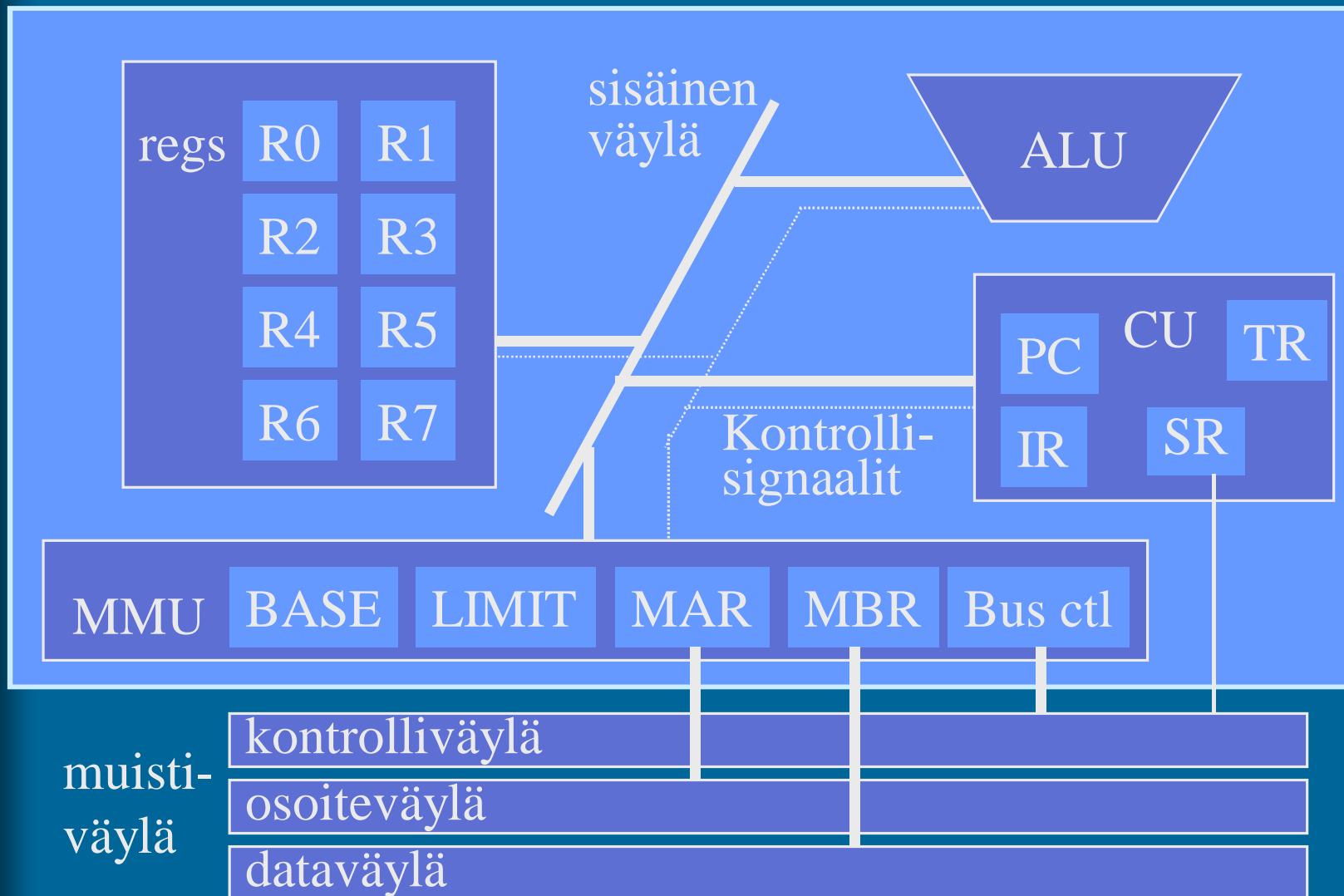
Luento 5

Suoritin ja väylä



Suorittimen rakenne
Väylän rakenne
Käskyjen suoritussykli
Suorittimen tilat
Poikkeukset ja keskeytykset
TTK-91:n ja simulaattorin rakenne

TTK-91 suorittimen rakenne



TTK-91 konekäskyn rakenne

- Käskyn esitys bittitasolla on aina:



Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri (R0 ≡ 0)

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

00 eli 0 kpl, välitön osoitus (STORE: suora osoitus)

01 eli 1 kpl, suora osoitus (STORE: epäsuora osoit.)

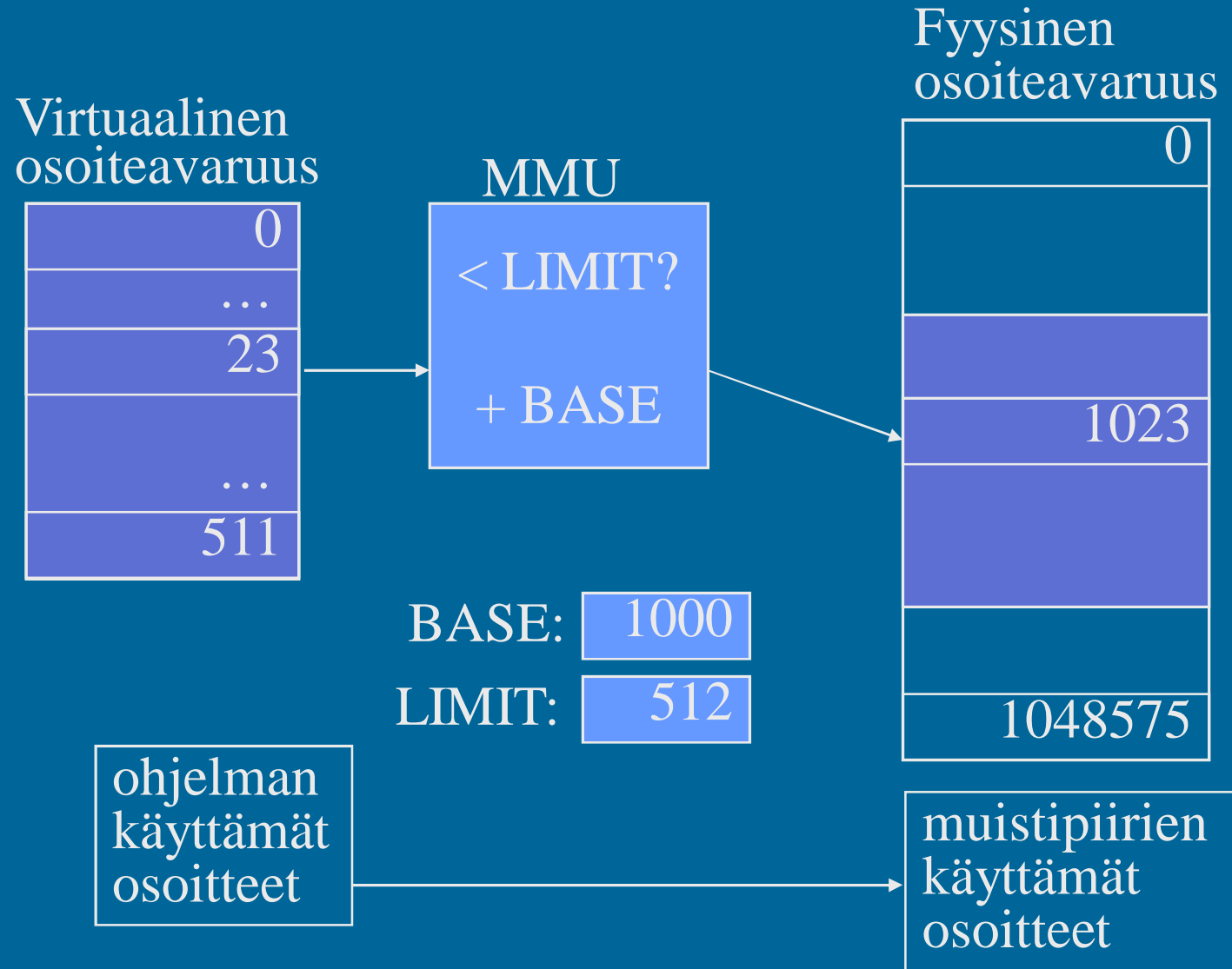
10 eli 2 kpl, epäsuora osoitus (STORE: epäkelpo arvo)

(11 eli 3 kpl, epäkelpo arvo → poikkeustilanne)

muistiosoite tai
(pienempi) vakio

(addressing
mode)

TTK-91 virtuaalimuisti



Virtuaalimuistin osoitteenmuunnos menetelmiä

- Kanta- ja rajarekisteriin perustuva
 - base ja limit rekisterit (esim. ttk-91, 8086, ...)
- Sivuttava
 - sivutaulut
 - virtuaaliavaruus jaettu saman kokoisiin sivuihin
- Segmentoiva
 - virtuaaliavaruus jaettu ohjelman mukaan erillisiin eri kokoisiin segmentteihin
 - koodi segmentti, data segmentti, ...

Lisää
tietoa?



käyttö-
järjestelmä
kurssit

Lisää
tietoa?



käyttö-
järjestelmä
kurssit

Keskeytysten käsittely

- Jokainen mahdollinen keskeytystyyppi on ennalta tunnettu, eli mitään todella yllättävää ei tapahdu
- Jokaiselle keskeytystyypille on oma käyttöjärjestelmän tuntema keskeytyskäsitteilyrutiini interrupt handler
- Jokaisen käskyn suorituksen jälkeen tarkistetaan keskeytysten olemassaolo SR:stä ja haaraudutaan keskeytyskäsitteilyrutiiniin tarvittaessa
 - joskus keskeytykset on estetty (ttk-91:ssä SR:n bitti D)
 - paluu käsitteilyrutiinista ”return-from-interrupt-handler” käskyllä (esim. IRET, tms)
- ”Yllättävä aliohjelmakutsu”

Keskeytyskäsitteijä

(kesk.)

- Osa käyttöjärjestelmää
- Ennen keskeytyskäsitteijään hyppäämistä asetetaan suoritin ja MMU etuoikeutettuun käyttöjärjestelmätilaan (supervisor state)
 - SR:n bitti P on päällä => etuoikeutettu tila eli (P = Privileged) käyttöjärjestelmä tila
 - käyttöjärjestelmätilassa saa viitata mihin tahansa kohtaan muistia (MMU: BASE=0, LIMIT="hyvin iso")
 - käyttöjärjestelmätilassa saa käyttää kaikkia konekäskyjä (esim. IRET tai ClearCache)
- Käsitteijästä paluun yhteydessä MMU:n tila ja suorittimen tila asetetaan ennalleen

Suorittimen tilan muuttaminen



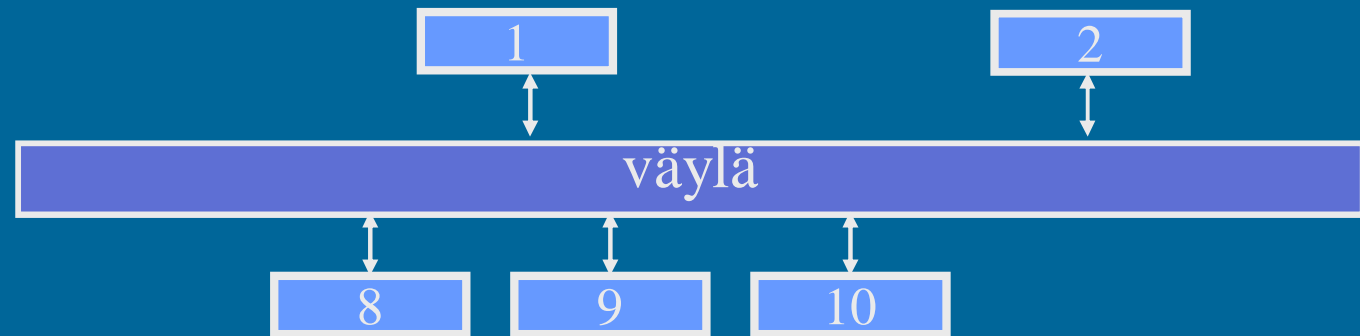
- Käyttäjätila → etuoikeutettu tila
 - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
 - keskeytyskäsittelijä tarkistaa onko (oliko) oikeutta tilan vaihtoon (interrupt handler)
- Etuoikeutettu tila → käyttäjätila
 - etuoikeutettu konekäsky “return from interrupt handler” esim. IRET (Pentium II)
 - palauttaa kontrollin keskeytyneeseen kohtaan ja suorittimen tilan keskeytystä edeltäneeseen tilaan

TTK-91 Nouto- ja suoritussykli vielä vähän tarkemmin

(kesk.)



Väylät



- Kullakin laitteella oma osoite
- Yksi lähettää, kaikki kuulevat, vain ”oikea” laite vastaanottaa
- Paljon erilaisia
- Lähellä suoritinta olevat ovat nopeampia

Lisää tietoa?



Tietokoneen rakenne-kurssi

TTK-91 koneen simulaattori

- Tavallinen Javalla kirjoitettu ohjelma
- TTK-91 koneen osat tietorakenteina
 - rekisterit, MMU, CU, muisti
- Simuloi käskyjen suoritussykliä käsky kerrallaan
- Toteuttaa myös TTK-91 koneen käyttöjärjestelmän osat osana tavallista ohjelmaa
 - assembler käääntäjä, lataaja, debugger, kesk. käsittelijät
- Graafinen käyttöliittymä

Ks. Processor.java Titokoneen koodissa

TTK-91 käskyn suoritusyksi

hae käsky simuloidusta muistista

$IR = \text{mem}[PC]$

pura käsky osiin (OPER, Rj, M, Ri, ADDR) ja
laske osoiteosan arvo TR (ADDR tai $\text{regs}[Ri] + \text{ADDR}$)

$ADDR = IR \bmod 32768$ $TR = \text{regs}[Ri] + \text{ADDR}$

tee tarvittava määrä (M) operandin
hakuja muistista rekisteriin TR

$TR = \text{mem}[TR]$

valitse aliohjelma operaatiokoodin (OPER) perusteella

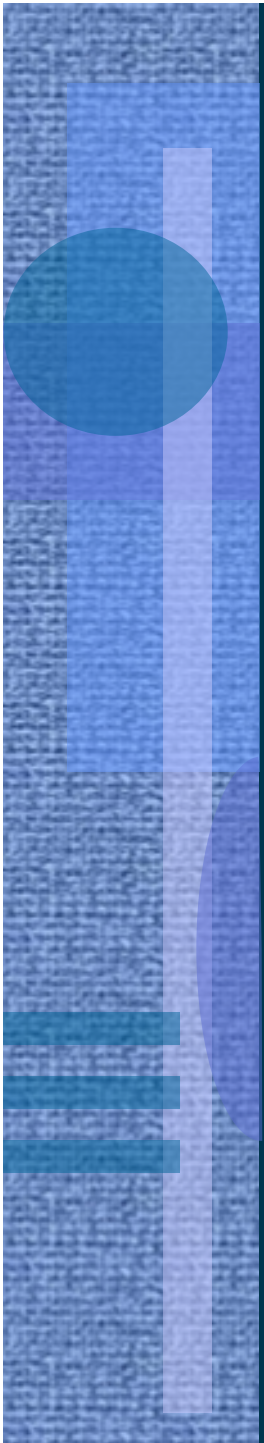
$\text{if } (\text{opcodeOK}[\text{OPER}] = \text{FALSE}) \text{ then } \text{SR.U} = 1;$

simuloi konekäskyn suorituksen muutokset
rekistereihin (R0...R7, SR, PC, MAR, MBR)

$\text{ADD } Rj, M \text{ ADDR}(Ri) \Rightarrow \text{regs}[Rj] += TR;$

lopetta suoritus jos SVC tai keskeytys

$\text{SR.O} = \dots$



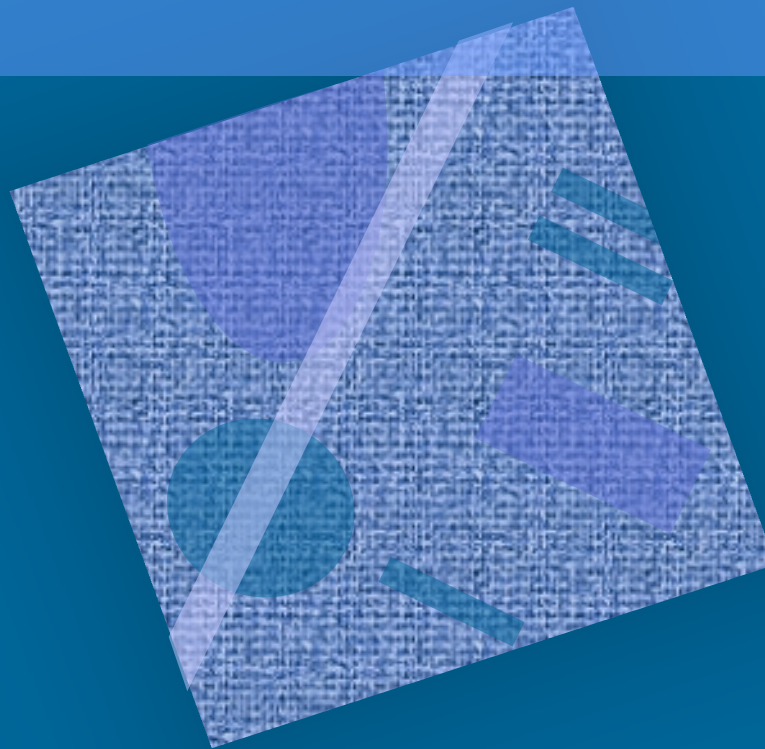
26.3.2010

Copyright Teemu Kerola 2010

24

Luento 6

Tiedon esitysmuodot



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

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
 - 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

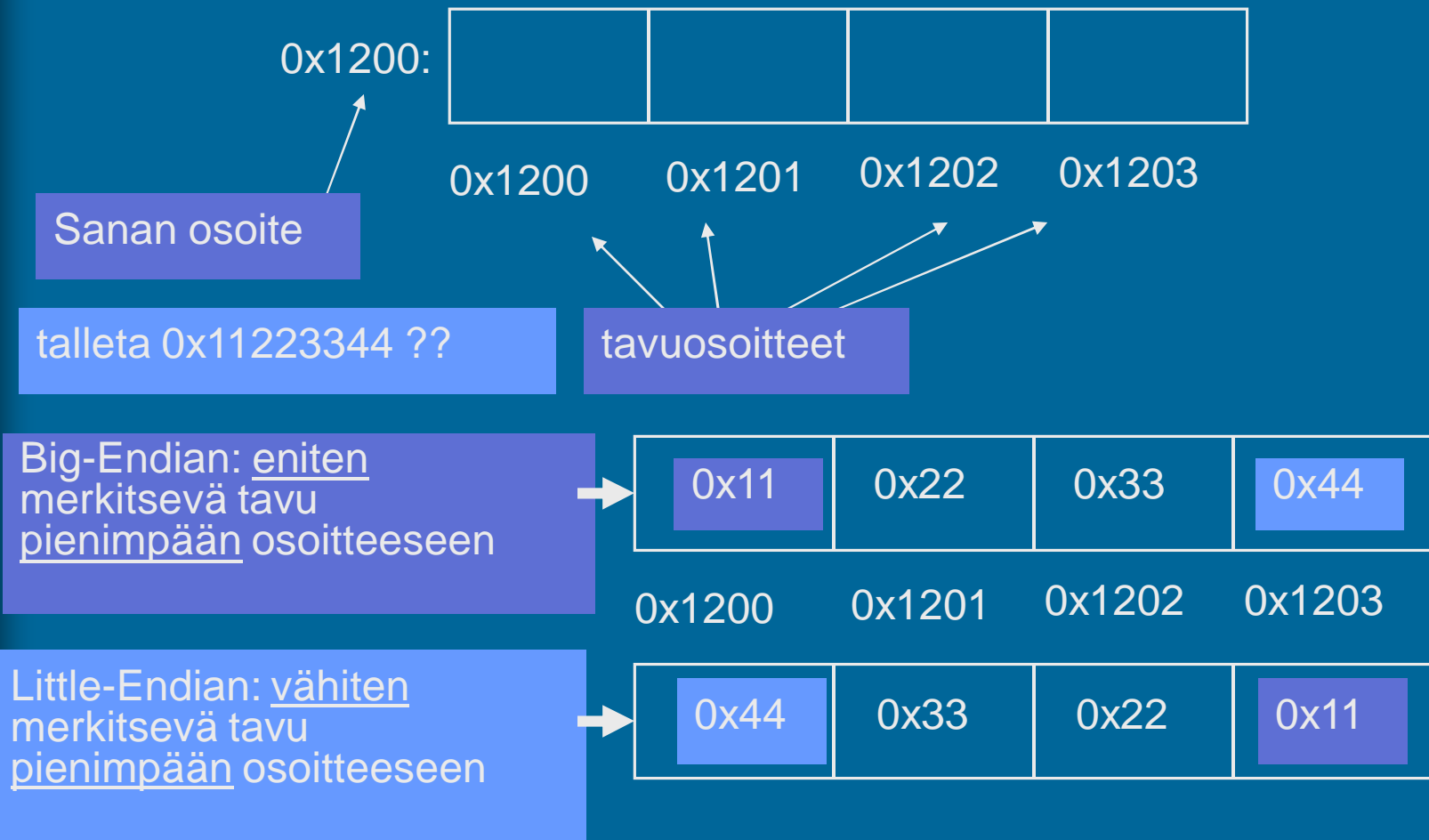
Suorittimen ymmärtämä tieto

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

TTK-91:
kokonaisluvut

Big vs. Little Endian

- Miten monitavuiset arvot talletetaan?



Negatiiviset luvut

- Etumerkkibitti erikseen
- Yhden komplementtiesitys
- **Kahden komplementtiesitys**
- Vakiolisäys
 - Esim lisää 127 ($=2^8 - 1$)
 - yleensä: $2^{\text{bittilkm}} - 1$
 - Talleta etumerkittömänä

arvo talletus
+57 = 0011 1001

sign bit = MSB
= most significant bit

luku -57 = 1011 1001 talletusmuoto

-57 = 1100 0110

"sign" bit

+1

-57 = 1100 0111

"sign" bit

-57 = 0100 0110

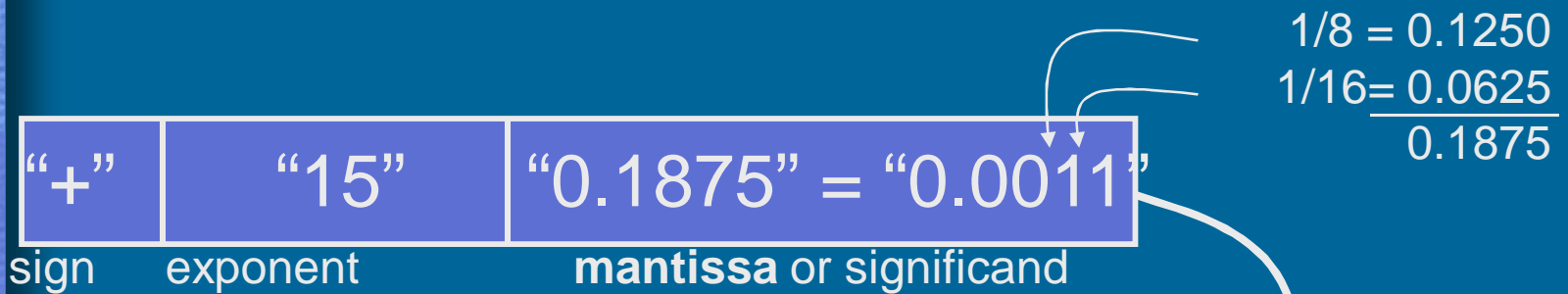
-57 + 127 = 70

+57 = 1011 1000

+57 + 127 = 184

IEEE 32-bit FP Standard

(kesk.)



- 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!

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
- 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

Merkkijonot

- Yleensä peräkkäin talletettu joukko tavuja
- Lisäksi tarvitsee jollain tavalla koodata merkkijonon pituus

– laitetaan loppuun erikoismerkki, tai

- 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"` konekäskyt

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

-- Luennon 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