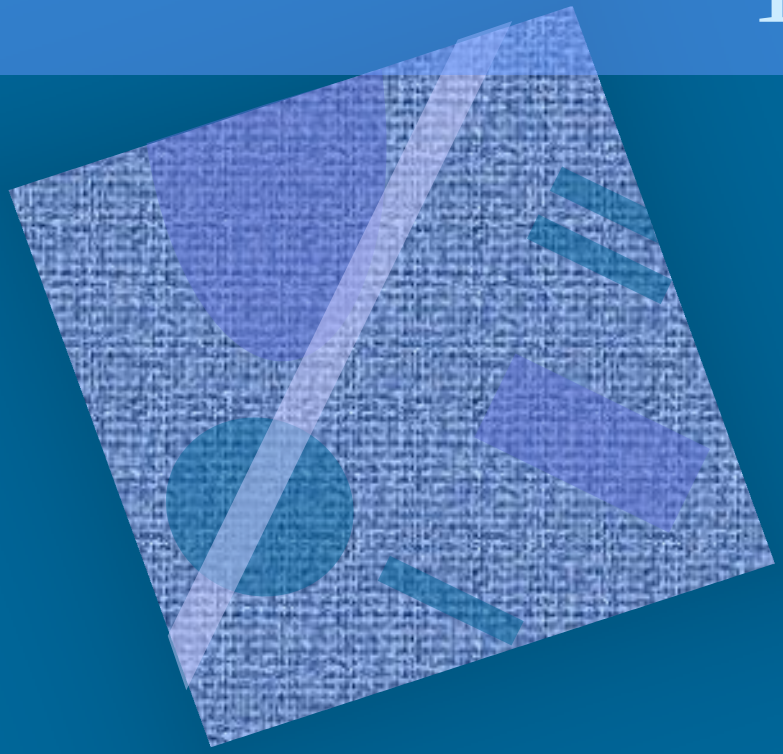
Luento 2 (verkkoluento 2)
Ttk-91 järjestelmä



Ttk-91 laitteisto

Tiedon sijainti

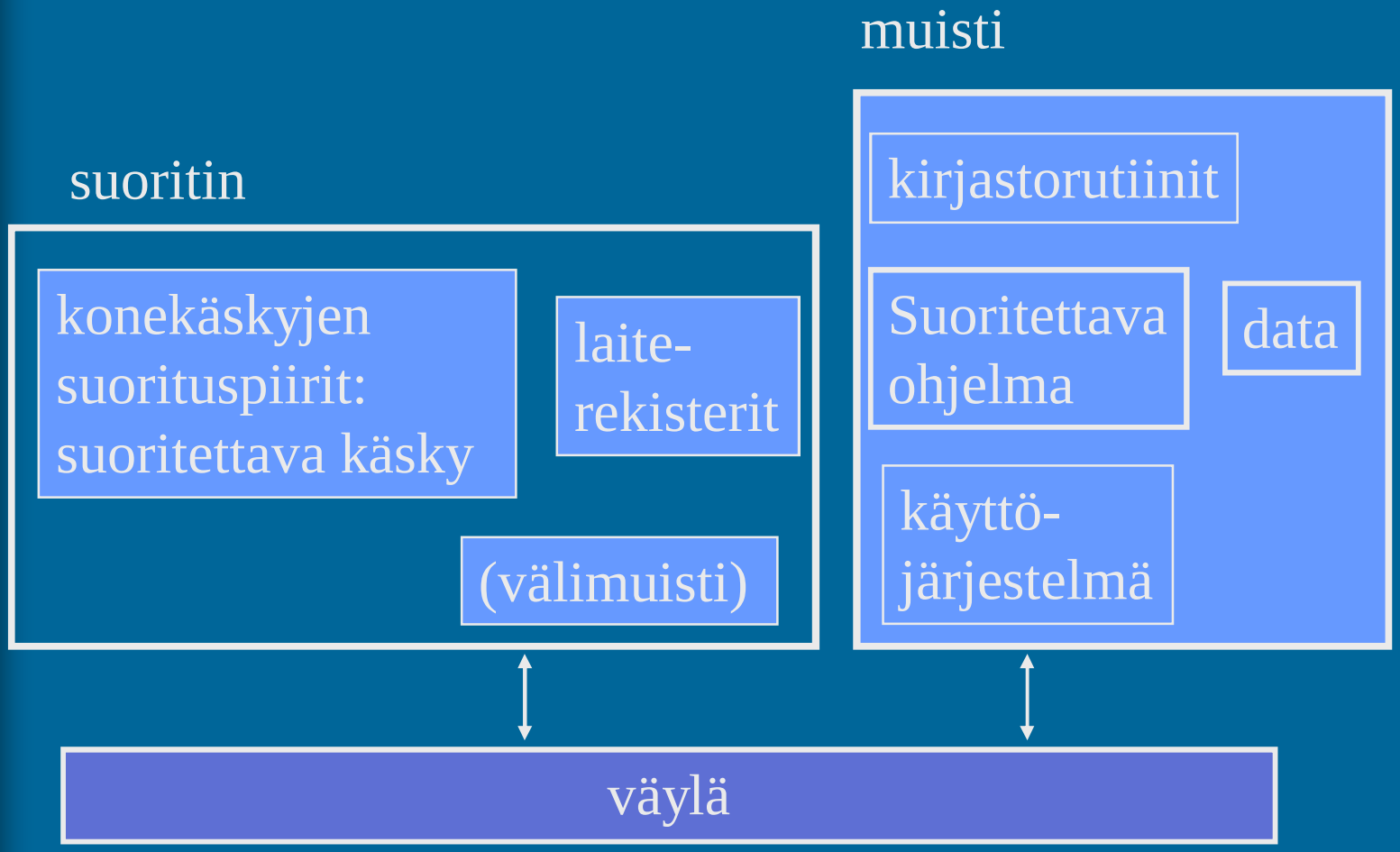
Muistitilan käyttö

Ttk-91 konekieli

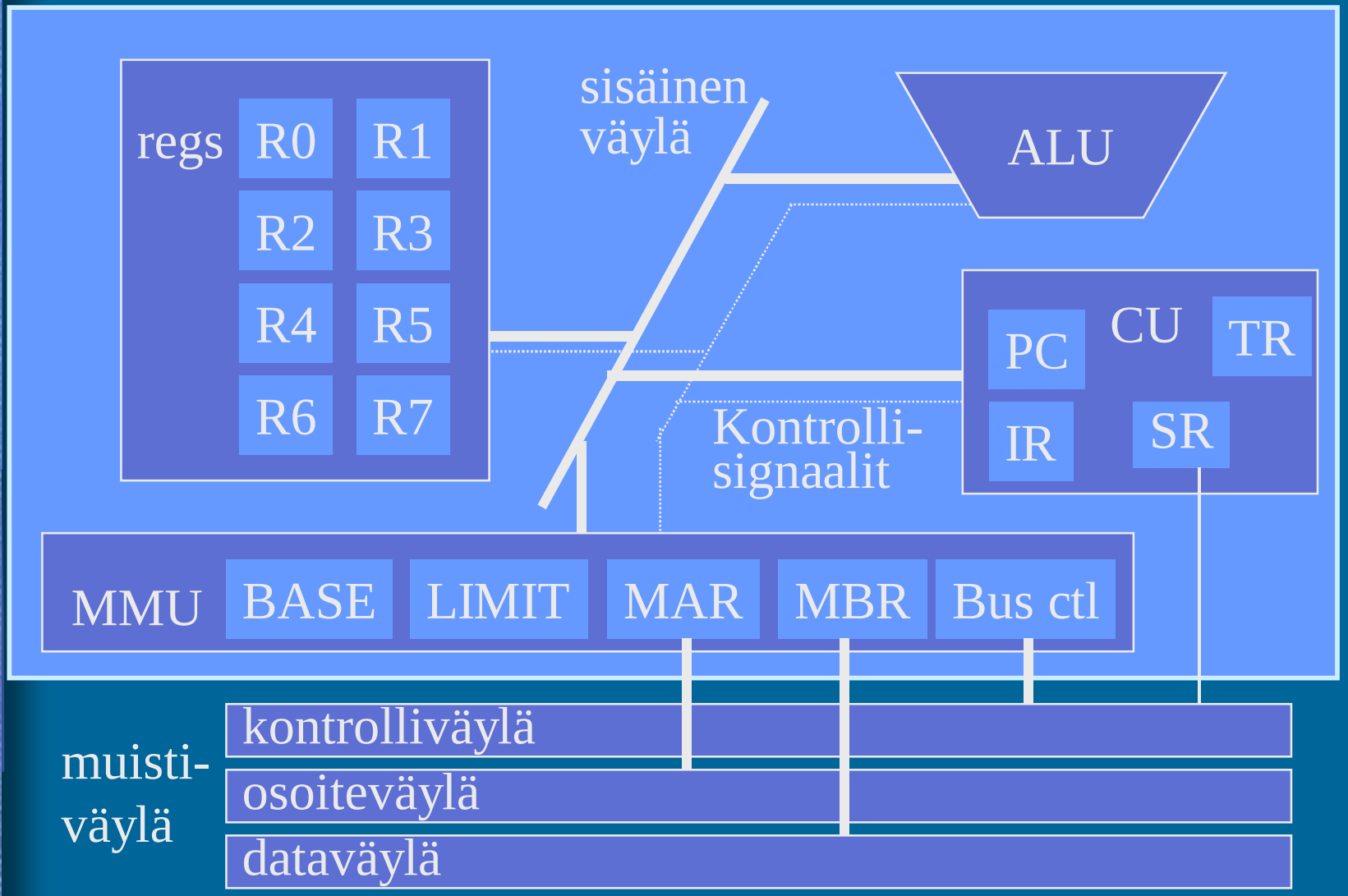
Tiedon osoitus ttk-91:ssä

Indeksointi, taulukot, tietueet

Suorituksenaikainen suorittimen ja muistin sisältö



TTK-91 suorittimen rakenne



TTK-91 konekäskyn rakenne

- Ttk-91 käskyn esitys bittitasolla on **aina**:

(pienehkö) muisti-osoite tai vakio

Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri (0 o ei käytetä)

M = muistinoutojen määrä toiseen operandiin (ennen operaation suoritusta)

(addressing mode)

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

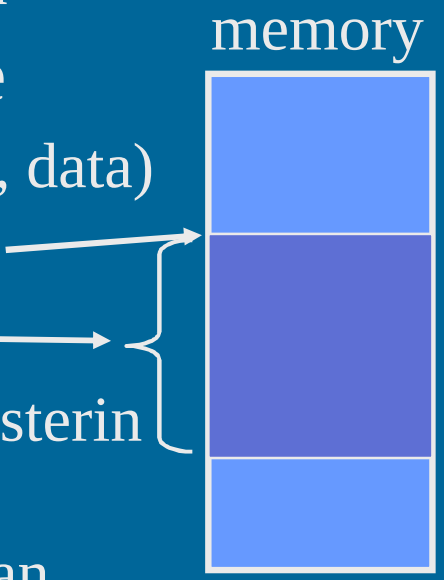
01 eli 1 kpl, suora osoitus (STORE: epäsuora osoit.)

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

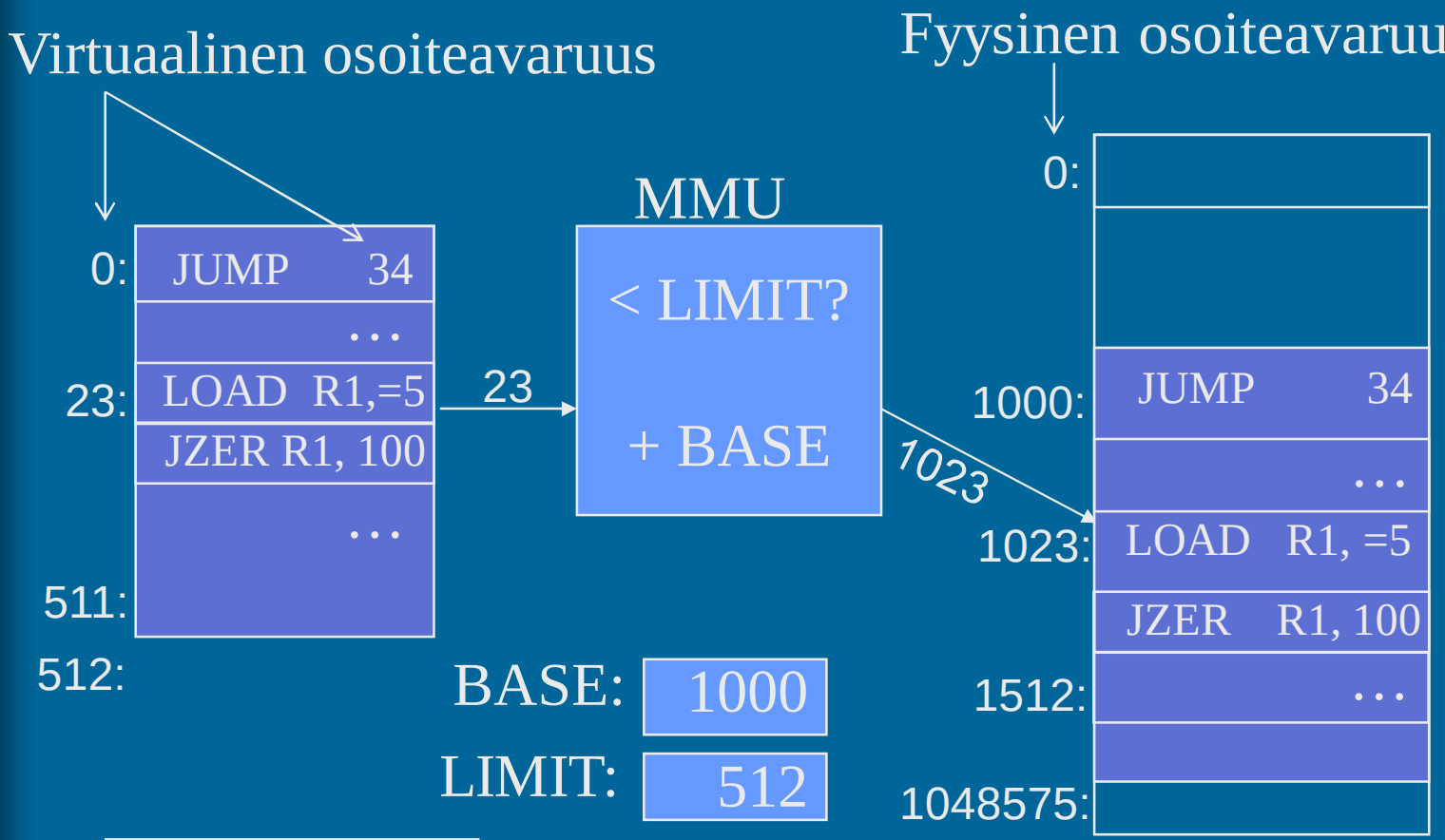
(11 eli 3 kpl, epäkelpo arvo ® virhe, poikkeustilanne)

TTK-91 Muistinhallintayksikkö (MMU)

- Muistiinviittausrekisterit
  - MAR - Memory Address Register, muistiosoite
  - MBR - Memory Buffer Register, luettava/kirjoitettava arvo suorittimella
- Ohjelman käytössä oleva muistialue
  - vain tähän alueeseen voi viitata (koodi, data)
  - BASE - muistisegmentin alkuosoite
  - LIMIT - muistisegmentin koko
  - kaikki osoitteet suhteellisia BASE rekisterin arvoon
    - esim. jos BASE=8000, niin ohjelman osoite 34 viittaa muistiosoitteeseen 8034
  - käyttöjärjestelmä asettaa ja valvoo



TTK-91 Kanta ja rajarekisterit



ohjelman (suorittimen) käyttämät osoitteet

muistipiirien käyttämät osoitteet

Tiedon sijainti suoritusaikana

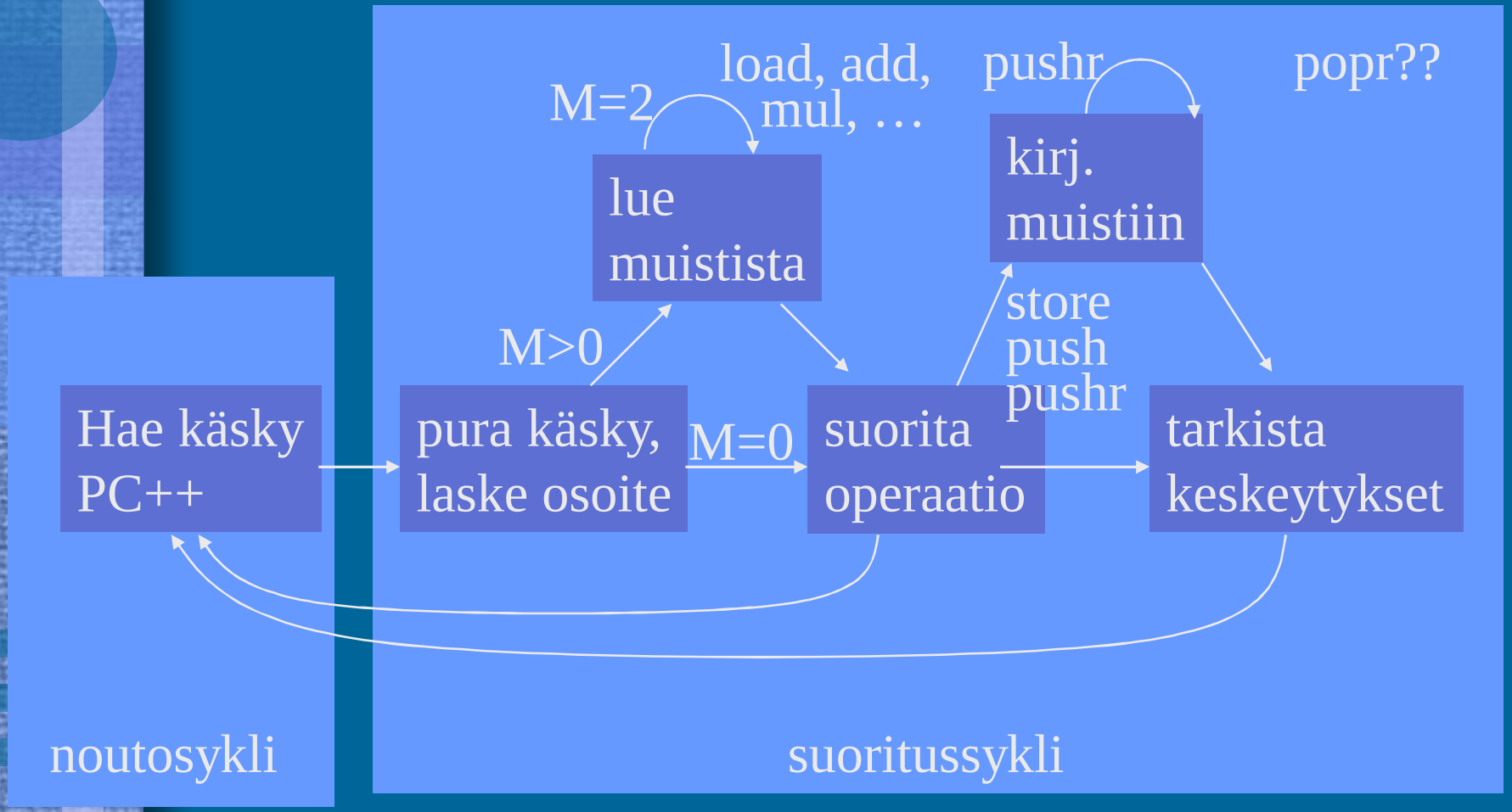
- Muistissa (=keskusmuistissa)
  - iso — Esim. 256 MB, tai 64 milj. 32 bitin sanaa
  - hidas — Esim. 50-150 ns
  - data-alueella vai konekäskyssä vakiona?
- Rekisterissä
  - pieni — Esim. 256 B, tai 64 kpl 32 bitin sanaa
  - nopea — Esim. 1 ns

TTK-91: 8 kpl + PC + ...

Milloin muuttujan X arvo pidetään muistissa ja milloin rekisterissä?
Missä päin muistia? Miten siihen viitataan?

Keskustele

TTK-91 Nouto- ja suoritussykli vielä vähän tarkemmin



Keskustele

TTK-91 operaatiot

- Muistiinviittaukset
  - tavalliset: load & store, aritmetiikan yhteydessä
  - pino-operaatiot (aliohjelmien toteuttamista varten)
- I/O käskyt
- Kokonaislukuoperaatiot
- Loogiset operaatiot totuusarvoille
- Bittien siirtokäskyt (shift instructions)
- Kontrollin siirtokäskyt
  - mistä löytyy seuraavaksi suoritettava käsky? (ellei se ole seuraavassa muistipaikassa)
- Muut käskyt

TTK-91 muistiinviittausoperaatiot

- LOAD
  - käskyä käytetään myös rekistereiden arvojen kopiointiin (Move oper.)

```
LOAD R1, X
LOAD R5, @ptrX
LOAD R0, R5
```

- STORE
  - tallettaa aina muistiin

```
STORE  R2, X
STORE  R3, Tbl(R4)
```

- PUSH, POP, PUSHR, POPR
  - aliohjelmien toteut-tamista varten
  - käsitellään myöhemmin

```
PUSH  SP, R1   ; store to stack
POP   SP, R1   ; take from stack
```

TTK-91 I/O operaatiot

- IN

```
IN      R3, =KBD
```

  - lue arvo (kokonaisluku) rekisteriin annetulta laitteelta (vain KBD määritelty)
- OUT

```
OUT     R2, =CRT
```

  - tulosta arvo (kokonaisluku) rekisteristä annetulle laitteelle (vain CRT määritelty)
- Laitteet?
  - KBD - näppäimistö, stdin
  - CRT - näyttö, stdout
  - ei muita! (ei levyä, ei verkkoa, …)

TTK-91 kokonaislukuoperaatiot

- LOAD (”move”)
- ADD, SUB
- MUL
- DIV, MOD

```
LOAD  R3, R1  ; R3 ¬ R1
```

```
ADD  R3,  X   ; R3 ¬ R3+ Mem(X)
```

```
SUB  R3, =1   ; R3 ¬ R3-1
```

```
MUL  R3, Tbl(R1)   ; R3 ¬ R3 * Mem(Tbl+R1)
```

```
LOAD  R1,=14
DIV     R1,=3          ; R1 ¬ 4
LOAD  R1,=14
MOD    R1,=3           ; R1 ¬ 2
```


Keskustele

TTK-91
loogiset (bitti)operaatiot (4)

- NOT, AND, OR, XOR
  - kaikille 32 bitille
  - yksi bitti kerrallaan

```
LOAD  R1, =12        ; R1 = 000…000 1100
LOAD  R2, =5         ; R2 = 000…000 0101
```

```
AND   R1, R2         ; R1 = 000…000 0100
OR    R1, R2         ; R1 = 000…000 1101
XOR   R1, R2         ; R1 = 000…000 1001
NOT   R1             ; R1 = 111…111 0011
```

TTK-91 bittien siirtokäskyt

- SHL, SHR, SHRA
  - siirrä bittejä vasemmalle tai oikealle
  - täytä nollilla (tai etumerkkibitillä)

```
LOAD R1,=5    ; R1 = 000…000 00101 = 5
SHL  R1,=1    ; R1 = 000…000 01010 = 10
```

  - yhden bitin siirto vasemmalle on sama kuin 2:lla kertominen! (jos vas. puolinen etumerkkibitti ei vaihdu)
  - positiivisilla luvuilla yhden bitin siirto oikealle on sama kuin 2:lla jakaminen!

```
LOAD R1,=5    ; R1 = 000…000 00101 = 5
SHR  R1,=1    ; R1 = 000…000 00010 = 2
```

```
LOAD R1,=-5   ; R1 = 111…111 11011 = -5
SHRA R1,=1    ; R1 = 111…111 11101 = -3
```

TTK-91 kontrollin siirtokäskyt

- JUMP `JUMP Loop`
- COMP `COMP R3, =27` `COMP R2, X`
  - asettaa tilarekisteriin SR vertailun tuloksen: L, E tai G
- JLES, JEQU, JGRE, JNLE, JNEQU, JNGRE
  - perustuu tilarekisterin tietoon eli viimeksi suoritettuun COMP-käskyyn `JGRE Loop`
- JNEG, JZER, JPOS, JNNEG, JNZER, JNPOS
  - perustuu annetun rekisterin arvoon `JPOS R1, Loop`
- CALL, EXIT (käsitellään myöhemmin)
- SVC `SVC SP, =HALT ; ohjelman suoritus päättyy`

TTK-91 muut käskyt

- NOP

```
NOP
```

  - No OPeration, tyhjä käsky, älä tee mitään
  - varaa kuitenkin muistia yhden sanan (32 bittiä)
  - suoritetaan samoin kuin muutkin käskyt
    - kuluttaa aikaa
    - käskyllä voi olla osoite, johon voi hypätä

```
        JZER  R4, OHI
        ....
OHI     NOP
        ....
```

Tiedon osoitusmuodot TTK-91

- Vain jälkimmäiselle operandille
  - Ensimmäinen operandi on aina rekisteri
- Välitön operandi (ei muistiosoitusta)
  - OPER Rj, =ADDR(Ri) M = 0 = 0b00
  - Jälkimmäinen operandi: ADDR+Ri
  - Kumpi vain voi puuttua (ADDR=0 tai Ri=R0)
- Suora (indeksoitu) muistiosoitus
  - OPER Rj, ADDR (Ri) M = 1 = 0b01
  - Jälkimmäinen operandi: Mem(ADDR+Ri)
- Epäsuora (indeksoitu) muistiosoitus
  - OPER Rj, @ADDR(Ri) M = 2 = 0b10
  - Jälkimmäinen operandi: Mem(Mem(ADDR+Ri))

Indeksointi

```
LOAD R4,=Tbl(R3)
LOAD R4,Tbl(R3)
LOAD R4,@Tbl(R3)
```

- Laske aina ensin ns. tehollinen muistiosoite (effective address, EA):

EA= Tbl + (R3) = 201

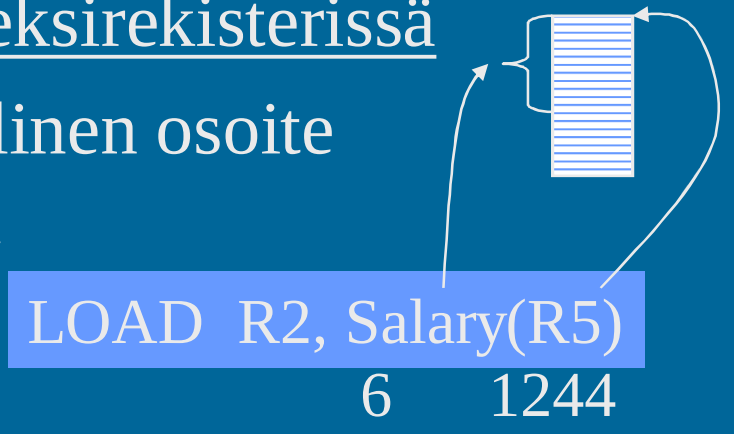
- Sitten katso moodia ja tee niin monta muistinoutoa kuin tarvitaan (tai ei yhtään)
  - ”=”: 0 kpl — R4 ¬ 201 (välitön)
  - tyhjä: 1 kpl — R4 ¬ Mem[201] = 11 (indeksoitu)
  - ”@”: 2 kpl — R4 ¬ Mem[Mem[201]] = Mem[11] = 300 (epäsuora)

pelkkä rekisterin nro @-merkin jälkeen Þ vain 1 muistinouto

STORE käsky Þ 1 kpl vähemmän noutoja ja yksi tallennus

Indeksoinnin käyttö taulukkojen ja tietueiden yhteydessä

- Taulukot
  - taulukon alkuosoite vakiona
  - taulukon indeksi indeksirekisterissä
- Tietueet
  - tietueen alkuosoite indeksirekisterissä
  - tietueen kentän suhteellinen osoite tietueen sisällä vakiona



olio

record object

TTK-91 assembler kääntäjän ohjauskäskyt

- Eivät generoi lainkaan konekäskyjä
  - suoritetaan käännösaikana
- EQU - Equal
  - antaa arvon symbolille symbolitauluun
- DC - data constant
  - varaa yhden sanan tilaa muistista, antaa sille alkuarvon ja antaa osoitteen symbolin arvoksi (symbolitauluun!)
  - esim. muuttujan tai ison vakion määrittely
- DS - data segment
  - varaa monta sanaa tilaa muistista, antaa arvon symbolille
  - alkuarvot ovat epämääräisiä!
  - esim. taulukon tai tietueen tilan varaus

```
Sata    EQU 100

LOAD R1, =Sata

X       DC  50

LOAD R1, X

Tbl     DS  200

LOAD R3,Tbl(R1)
```

TTK-91 symbolinen konekieliohjelma

hello.k91

```
X       DC      13
Y       DC      15

MAIN    LOAD    R1, X
        ADD     R1, Y
        OUT     R1, =CRT
        SVC     SP, =HALT
```

Mitkä ovat seuraavien symbolien arvot?

X? MAIN? CRT? ADD?

https://www.cs.helsinki.fi/group/titokone/v1.100/kayttoohje/manual_fi.html

-- Loppu --

Some typical 80x86 intructions and their function

Instruction	Function
JE name	If equal (CC) EIP = name}; EIP − 128 ≤ name < EIP + 128
JMP name	{EIP = NAME};
CALL name	SP = SP − 4; M[SP] = EIP + 5; EIP = name;
MOVW EBX,[EDI + 45]	EBX = M [EDI + 45]
PUSH ESI	SP = SP − 4; M[SP] = ESI
POP EDI	EDI = M[SP]; SP = SP + 4
ADD EAX,#6765	EAX = EAX + 6765
TEST EDX,#42	Set condition codea (flags) with EDX & 42
MOVSL	M[EDI] = M[ESI]; EDI = EDI + 4; ESI = ESI + 4

Fig. 3.32 [PaHe98]