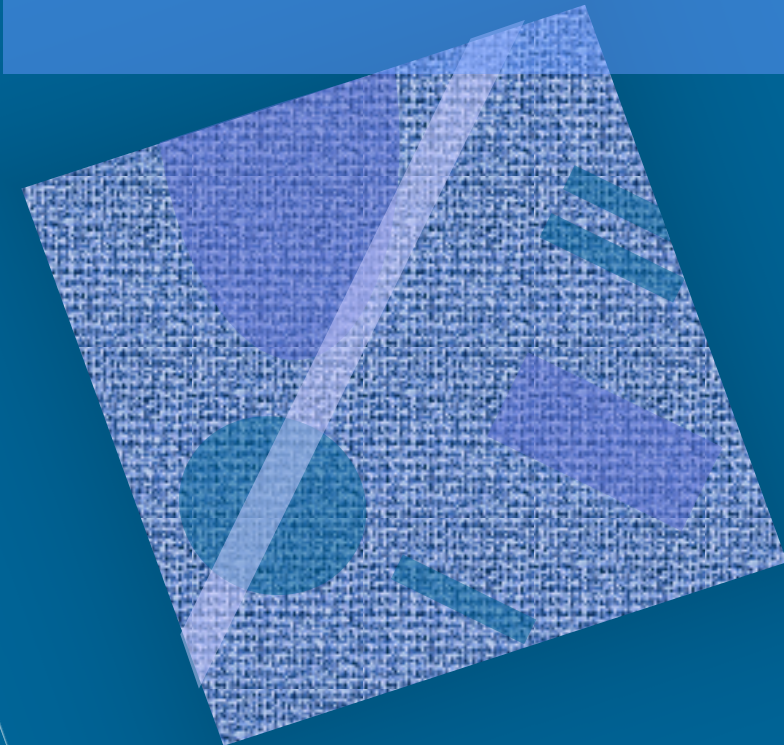


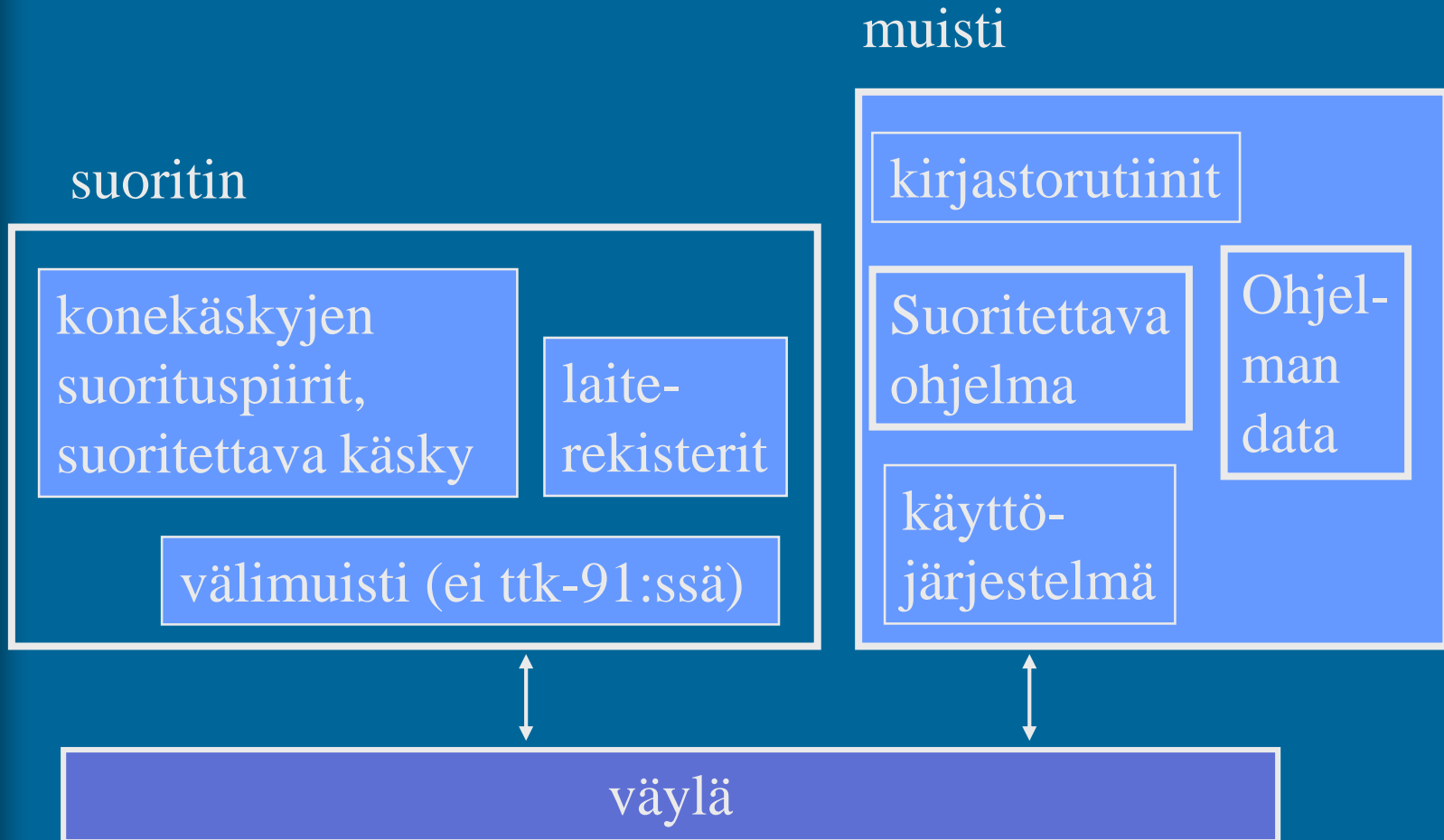
# Luento 5 (verkkoluento 5)

## Suoritin ja väylä



Käskyjen suoritusyksi  
Suorittimen tilat  
Poikkeukset ja keskeytykset

# Suoritusaikainen suorittimen ja muistin sisältö



# Suorittimen toiminta, perusidea

PC=0 alkuarvo  
(ttk-91:ssä)

start

IR = Instruction Register  
(käskyrekisteri)

Käskyn nouto  
IR := mem (PC)

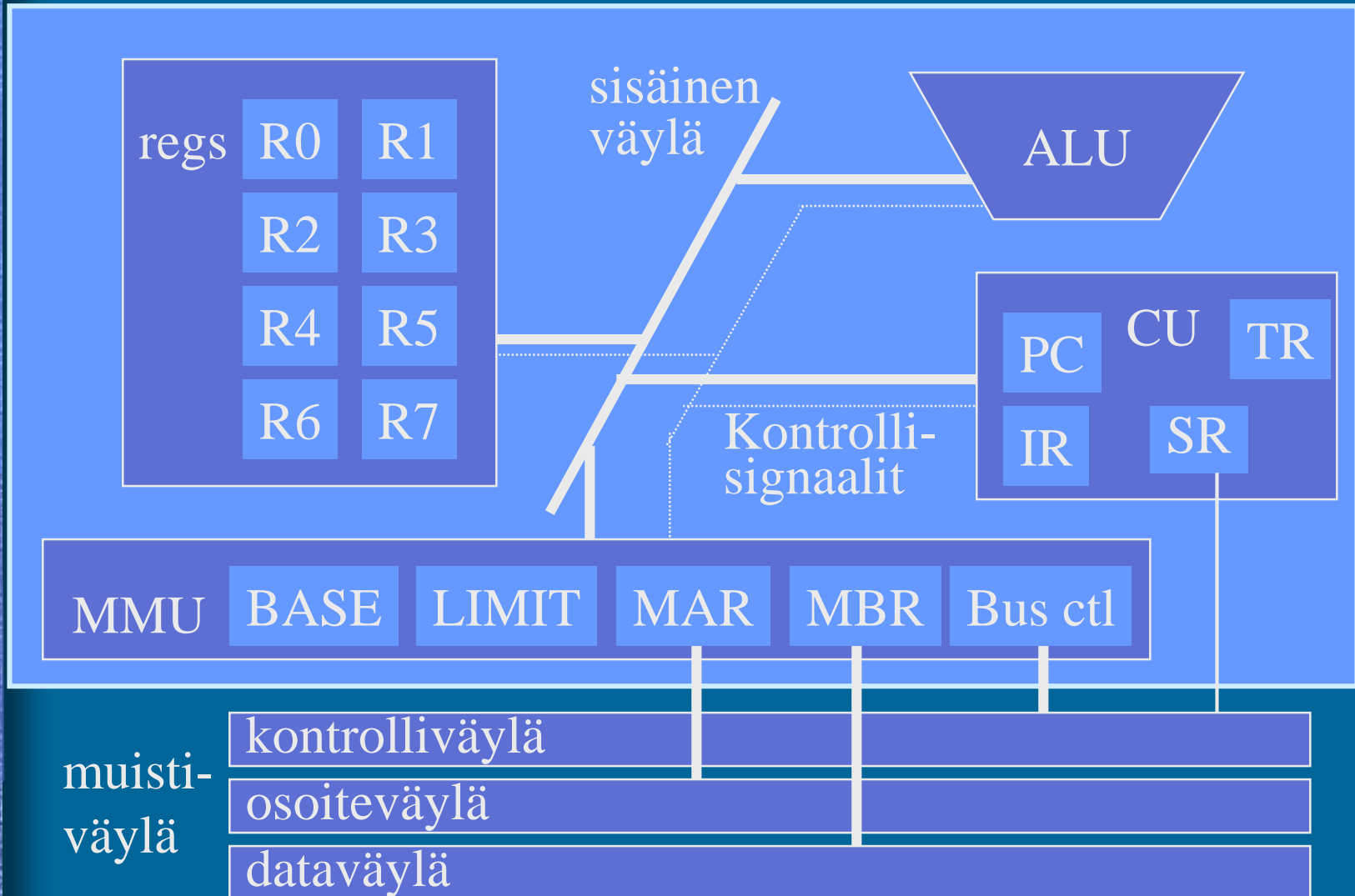
PC = Program Counter  
(paikanlaskuri,  
käskynosoitin, IP)

Käskyjen nouto- ja  
suoritussykli

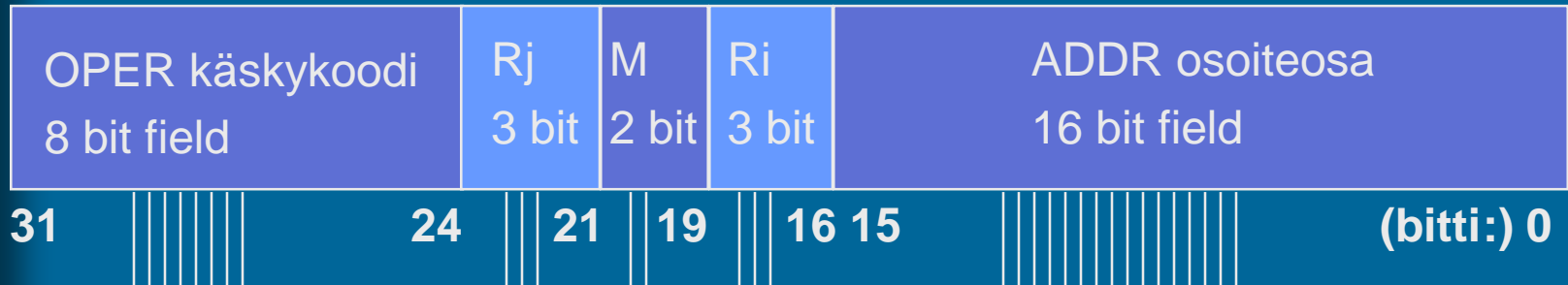
Ohjelmanlaskurin  
kasvatus  
PC := PC+1

IR:n käskyn suoritus  
(voi muuttaa PC:tä)  
(voi sisältää muistiviitteitä)

# TTK-91 suorittimen rakenne



# Käskyn purku ja tehollisen muistiosoitteen (EA) laskemisvaihe



- Purku automaattisesti langoitettuna käskyrekisteristä (IR)
- Muistiosoitteen lasku, tulos tilapäisrekisteriin (TR)
  - Rj: 1. operandi ja tulosrekisteri
  - jos  $R_i=0$ , niin  $TR \leftarrow ADDR$
  - muutoin  $TR \leftarrow (R_i)+ADDR$ 
    - ALU suorittaa laskutoimituksen
    - jos  $ADDR = 0$ , niin  $TR \leftarrow (R_i)$
  - Effective Address (EA) on nyt TR:ssä
- TR:n sisältöä voi käyttää datana sellaisenaan tai muistiosoitteena tai epäsuorana muistiosoitteena
  - Jälkimmäisen operandin määrittely

# Käskyn suorittaminen

- Titokone näyttää vaiheet simuloimalla
- Store R4, @10(R1) ; R1 =20, R4=15
  - Käskyn nouto ja ohjelmanaskurin kasvatus:
    - PC → MAR, “bus read”, odota, mem → MBR
    - MBR → IR
    - PC+1 → PC
      - Esim: PC → ALU1, 1 → ALU2, +, odota, ALU → PC
      - Esim: PC → INCin, odota, INCout → PC (oma piiri)
  - Käskyn suoritus:
    - R1 → ALU1, 10 → ALU2, +, (odota), ALU → TR
    - TMP → MAR, ”bus read”, odota, mem → MBR
    - MBR → MAR, R4 → MBR, ”bus write”, odota, MBR → mem
  - Huom: yhteensä 3 muistiviitettä (tällä käskyllä)

# TTK-91 konekieli

Operaatio 8 bit	Rj 3 b	M 2 b	Ri 3 b	Attribuutti (arvo tai osoite) 16 bittiä
--------------------	-----------	----------	-----------	--

- Kukin käsky 32 bittiä. Oma operaatiokoodi kullakin käskyllä
- Käskyn rekisterien ja attribuuttien tulkinta riippuu käskystä ja osoitusmuodosta (M)
- Tietotyyppi:
  - Vain 32-bittinen kokonaisluku (tai vain bittejä)
  - EI: merkkejä, liukulukuja, totuusarvoja

# TTK-91 rekisterit

- 8 yleisrekisteriä
  - vain näitä rekistereitä voi osoittaa tai lukea/kirjoittaa (suoraan) konekäskyillä
  - kaikki laskenta tapahtuu rekistereiden avulla
    - vain 8 ”muistipaikkaa” varsinaista laskentaa varten
  - R0 työrekisteri
    - indeksirekisterinä == 0 (vakio nolla eli indeksirekisterinä R0:n käyttö tarkoittaa lukua 0 rekisterin R0 sisällön asemesta), eli ”ei indeksointia”
  - R1-R5 työ- ja indeksirekistereitä
    - tyyppi riippuu rekisterin käytöstä konekäskyssä

Aliohjelmien toteutukseen (ei tavalliseen laskentaan)

- pino-osoitin SP (R6)
- ympäristöosoitin FP (R7)

Stack Pointer

Frame Pointer



# TTK-91 Kontrolliyksikkö (CU)

- PC - Program Counter, käskyosoitin
  - seuraavaksi suoritettavan konekäskyn osoite
- IR - Instruction Register, käskyrekisteri
  - suorituksessa oleva konekäsky
- TR - Temporary Register, apurekisteri
  - tilapäinen talletuspaikka käskyn suoritusaikana
- SR - State Register, tilarekisteri
  - suorittimen tila ja rajoitukset tällä hetkellä

# TTK-91 Tilarekisteri SR

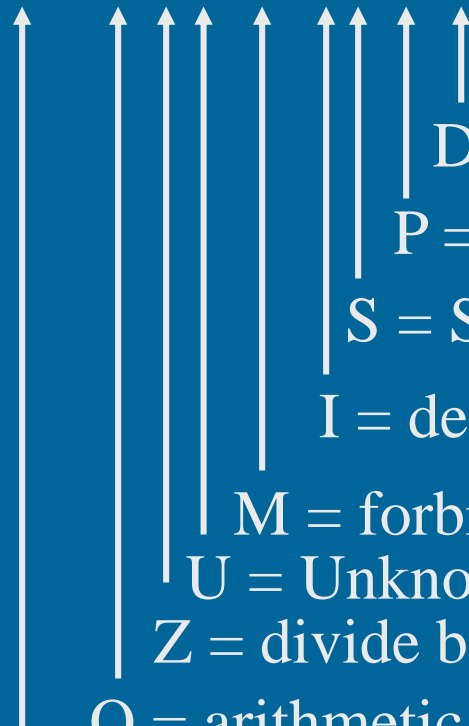
- Tilatietoa siitä, mitä suorittimella tapahtui edellisen käskyn suorituksessa
  - virhetilanteet, poikkeukset, keskeytykset
  - konekäsky olikin käyttöjärjestelmän palvelupyyntö
  - vertailun tulos
- Tilatietoa siitä, mitä systemissä tapahtui viime aikoina
  - käsittelemättömät laitteiden antamat signaalit (laitekeskeytykset, device interrupts)
- Tilatietoa siitä, mitä suoritin saa tehdä jatkossa
  - etuoikeutettu vai tavallinen suoritustila? (kaikki muistialueet ja kaikki käskyt sallittuja, vai ei?)
  - poikkeukset ja keskeytykset sallittuja vai ei?

# Tilarekisteri SR <sup>(9)</sup>

32 bittiä (kunkin arvo 0 tai 1)

SR:

GEL OZUM IS P D ????????



D = Interrupts Disabled (kesk. estetty)

P = Priviledged mode (etuoik. tila)

S = SVC (supervisor call) palvelupyyntö

I = device Interrupt (laitekeskeyty)

M = forbidden Memory address

U = Unknown instruction

Z = divide by Zero

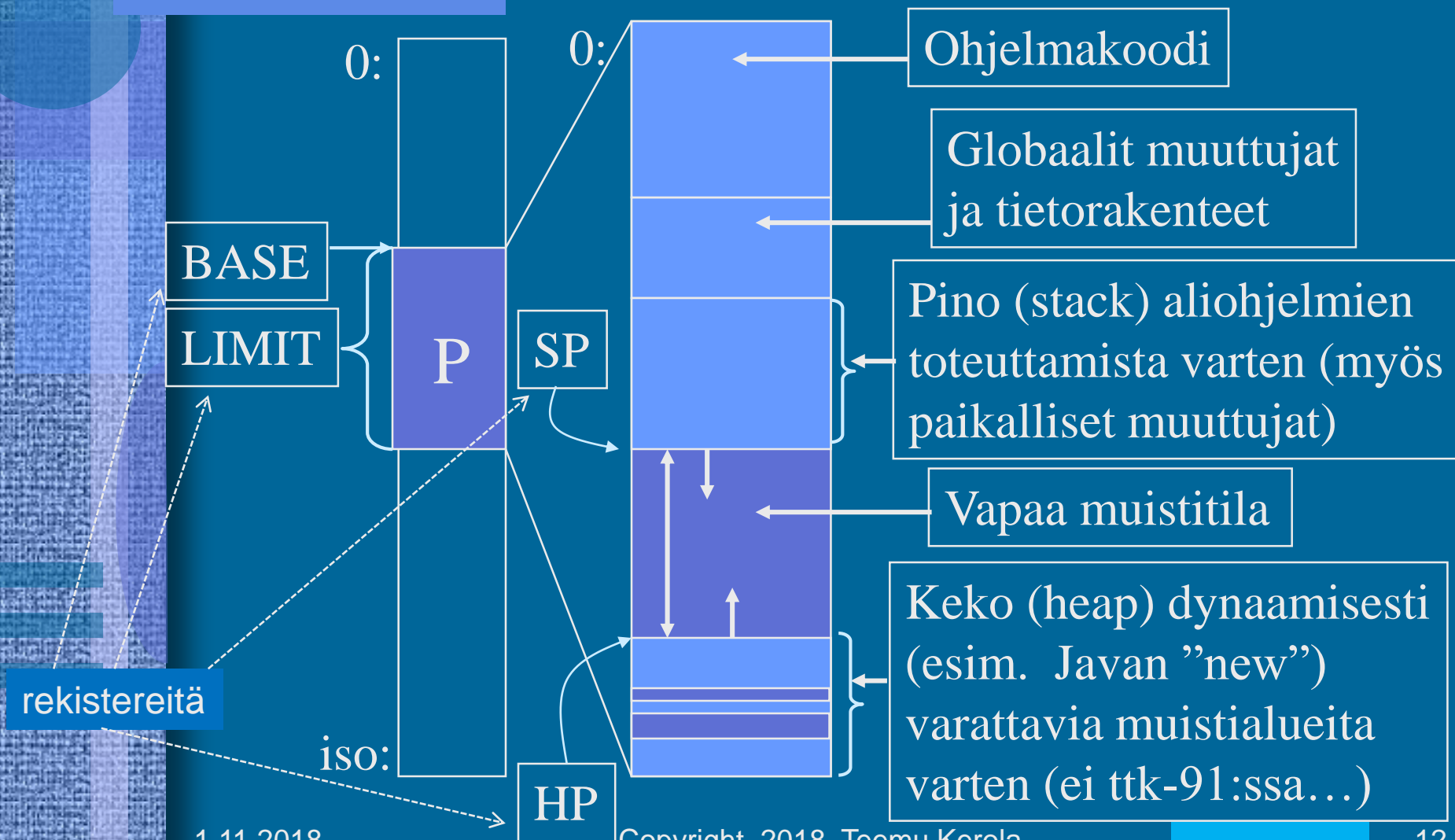
O = arithmetic Overflow

GEL = comparison indicators: Greater, Equal, Less

# Muistitilan käyttö ohjelmalle P

Todellinen fyysinen muisti

P:n näkemä (virtuaalinen) muisti



# Virtuaalimuistin osoitteenmuunnosmenetelmiä

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

Lisää  
tietoa?



käyttö-  
järjestelmä-  
kurssi

Lisää  
tietoa?



käyttö-  
järjestelmä-  
kurssi

# Keskeytysten käsittely

- Keskeyttää suoritussyklin, ”yllättävä aliohjelmakutsu”
- Jokainen mahdollinen keskeytystyyppi on ennalta tunnettu, eli mitään todella yllättävää ei koskaan tapahdu
- Jokaiselle keskeytystyypille on oma käyttöjärjestelmän tuntema **interrupt handler**  
keskeytyskäsittelyrutiini
- Jokaisen käskyn suorituksen lopussa (sen jälkeen) tarkistetaan (laitteisto tarkistaa) keskeytysten olemassaolo SR:stä ja haaraudutaan (PC ja SR talteen, PC:lle ja SR:lle uusi arvo) keskeytyksen tyyppin mukaiseen keskeytyskäsittelijään tarvittaessa
  - Esim. kesk. tyyppi 3:  $PC \leftarrow 3$  tai  $PC \leftarrow \text{mem}(3)$
  - joskus keskeytykset on estetty (ttk-91:ssä SR:n bitti D)
  - paluu käsittelijästä ”return-from-interrupt-handler” käskyllä (esim. IRET, tms.)

# Suorittimen toiminta, tarkemmin

PC=0 alkuarvo  
(ttk-91:ssä)

start

PC = Program Counter  
IR = Instruction Register

Käskyn nouto  
IR := mem (PC)

Ohjelmanaskurin kasvatus  
PC := PC+1

Käskyjen nouto- ja suoritusyksi

Keskeytysten tarkastaminen  
(voi muuttaa PC:tä)

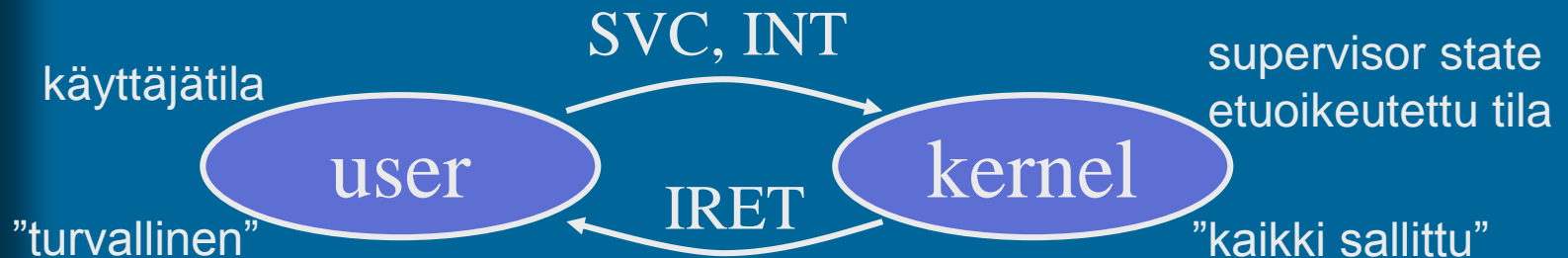
IR-käskyn suoritus  
(voi muuttaa PC:tä)  
(voi sisältää muistiviitteitä)

# Keskeytyskäsitteijä

- Tärkeä osa käyttöjärjestelmää
- Ennen keskeytyskäsitteijään siirtymistä asetetaan suoritin etuoikeutettuun tilaan (käyttöjärjestelmätilaan) (supervisor state, kernel 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, ClearCache, SetBASE, SetLIMIT, SetD, ReadSR, ...
- Käsitteijästä paluun yhteydessä MMU:n tila (BASE ja LIMIT) ja suorittimen tila (bitti P) 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

# Titokone - 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 kääntäjä, lataaja, debugger, kesk. käsittelijät
- Graafinen käyttöliittymä

Ks. Processor.java Titokoneen koodissa:

`titokone.jar\fi\hu\cs\titokone\Processor.java`

(<http://www.cs.helsinki.fi/group/nodes/kurssit/tito/Processor.java.txt> )

# TTK-91 käskyn suoritus sykli

hae käsky simuloidusta muistista

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

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

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

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)

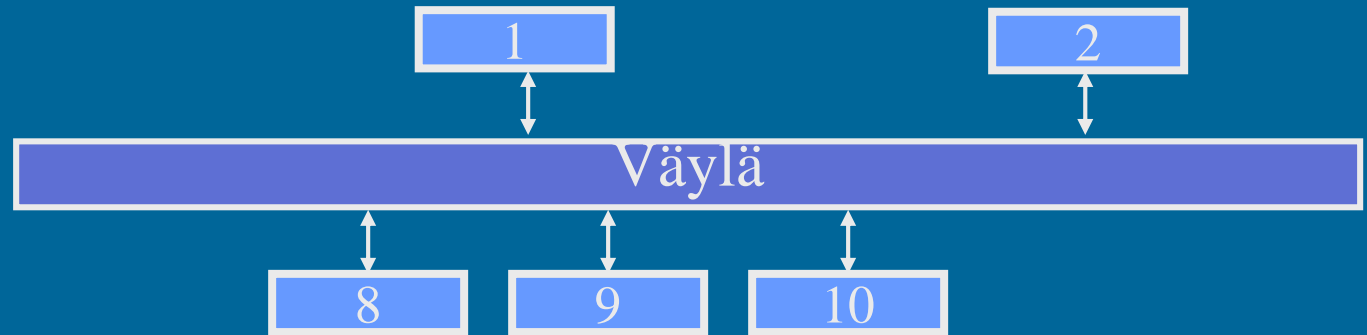
$\text{ADD } Rj, M \text{ ADDR}(Ri) \Rightarrow \text{reg}[Rj] += TR;$

lopetta suoritus jos SVC tai keskeytys

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

Simulaattori C:llä: <http://www.cs.helsinki.fi/group/nodes/kurssit/tito/simu/simu.c>

# Väylät



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

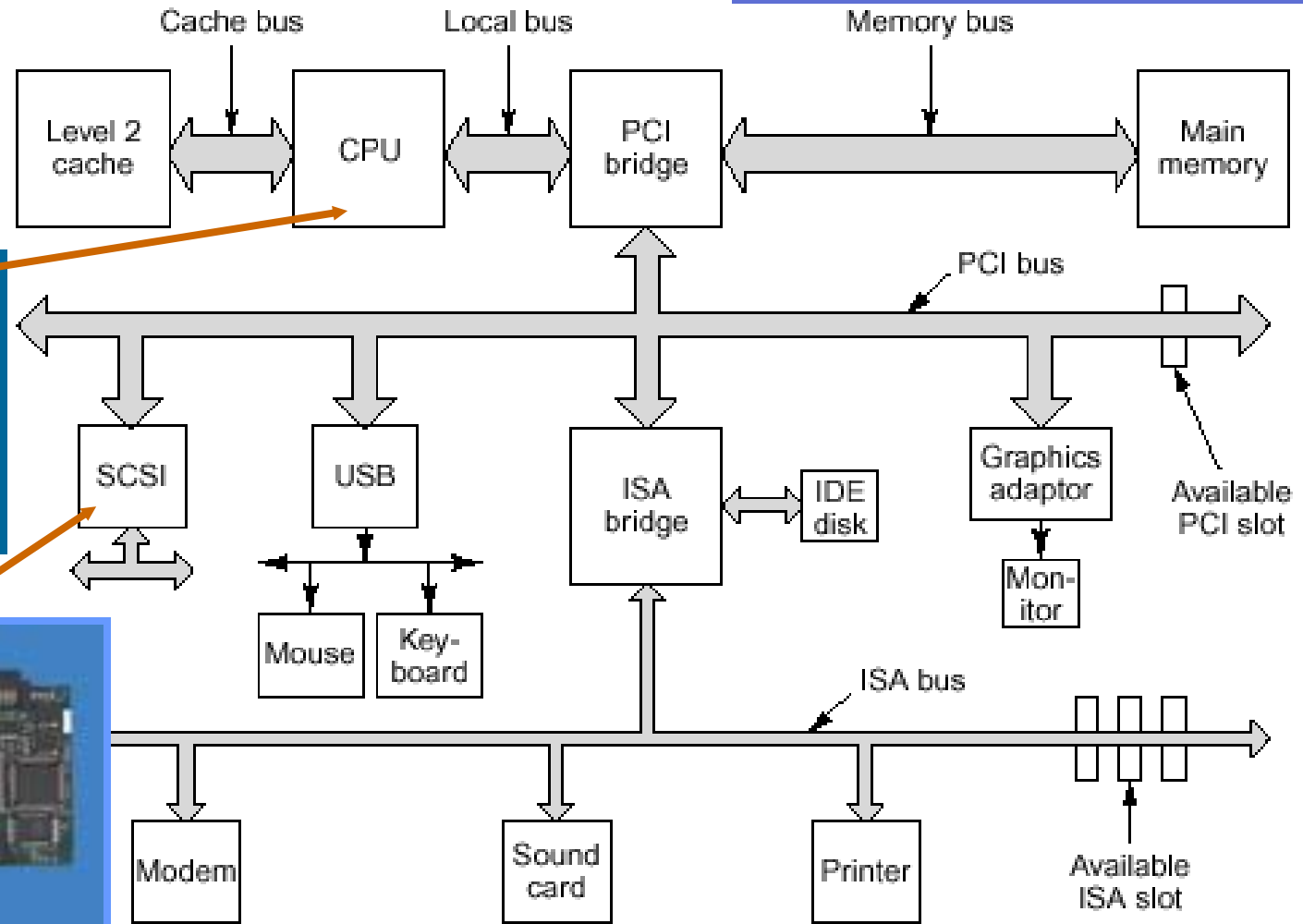
Lisää  
tietoa?



Tietokoneen  
rakenne-  
kurssi

# Väylähierarkia

Tyypillinen Pentium II  
systemin emolevy



omalla sirulla,  
tason 1  
välimuistin  
kanssa



PCI to SCSI bridge

Fig. 3-50 [Tane99]

# -- Loppu --

(Tekniikan museo)

ESKO, 1960, ensimmäinen Suomessa rakennettu ”tietokone”, vanhentunut jo valmistuessaan, 20 yhteenlaskua sekunnissa.



- Ohjelmakoodi luettiin reikänauhoilta (10 kpl)
  - Koodi ja data eivät ole samassa muistissa!
- Aliohjelmakutsu toteutettu kontrollin siirrolla toiseen reikänauhanlukijaan (jossa reikänauha silmukassa)