

# Järjestelmä, suoritin ja muisti



Ohjelma ja sen esitysmuodot

Laitteisto ja sen nopeus

Nouto- ja suoritussykli

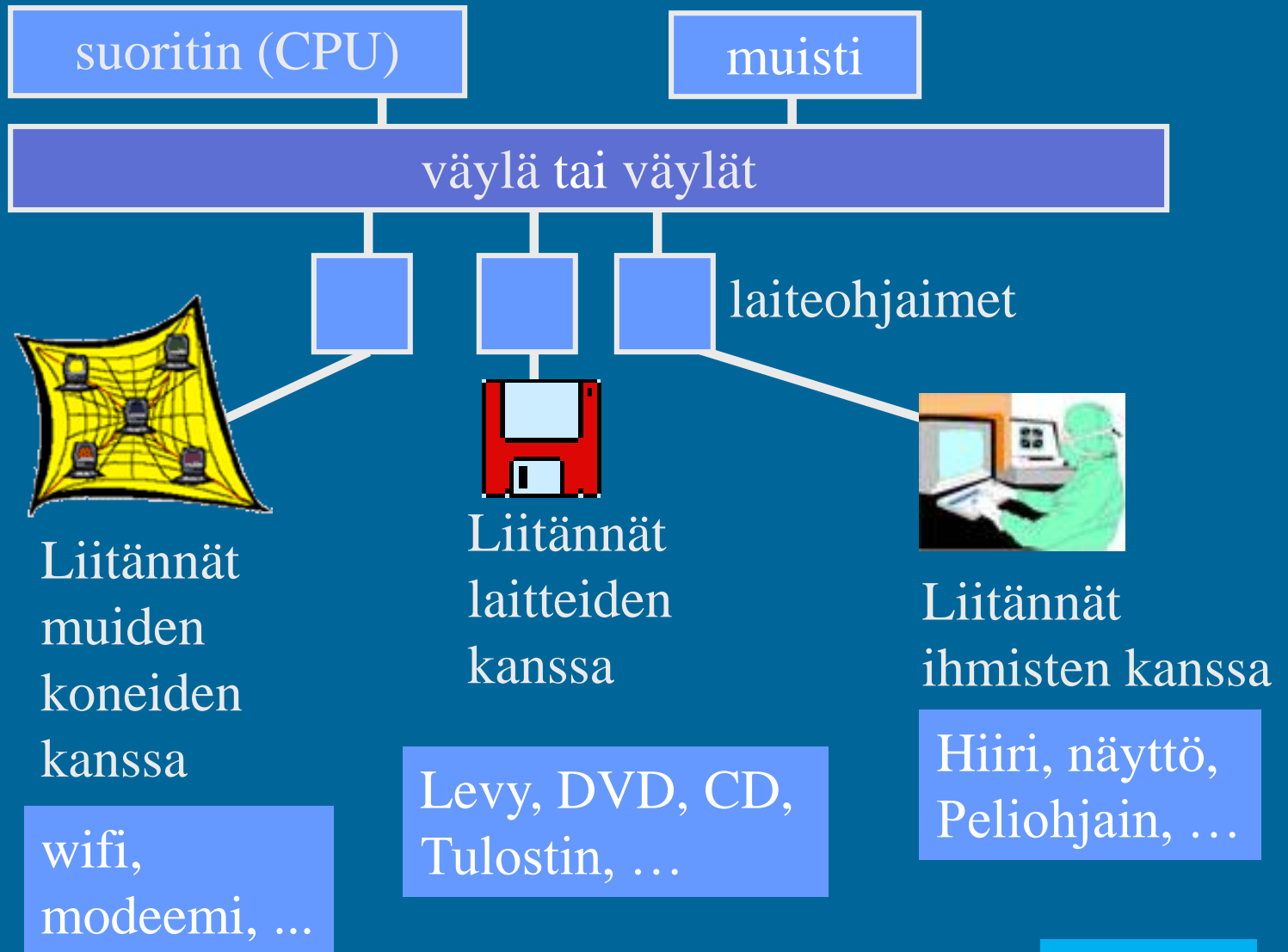
Keskeytykset

Etuoikeutettu suoritustila

Ttk-91 konekielen käskyt

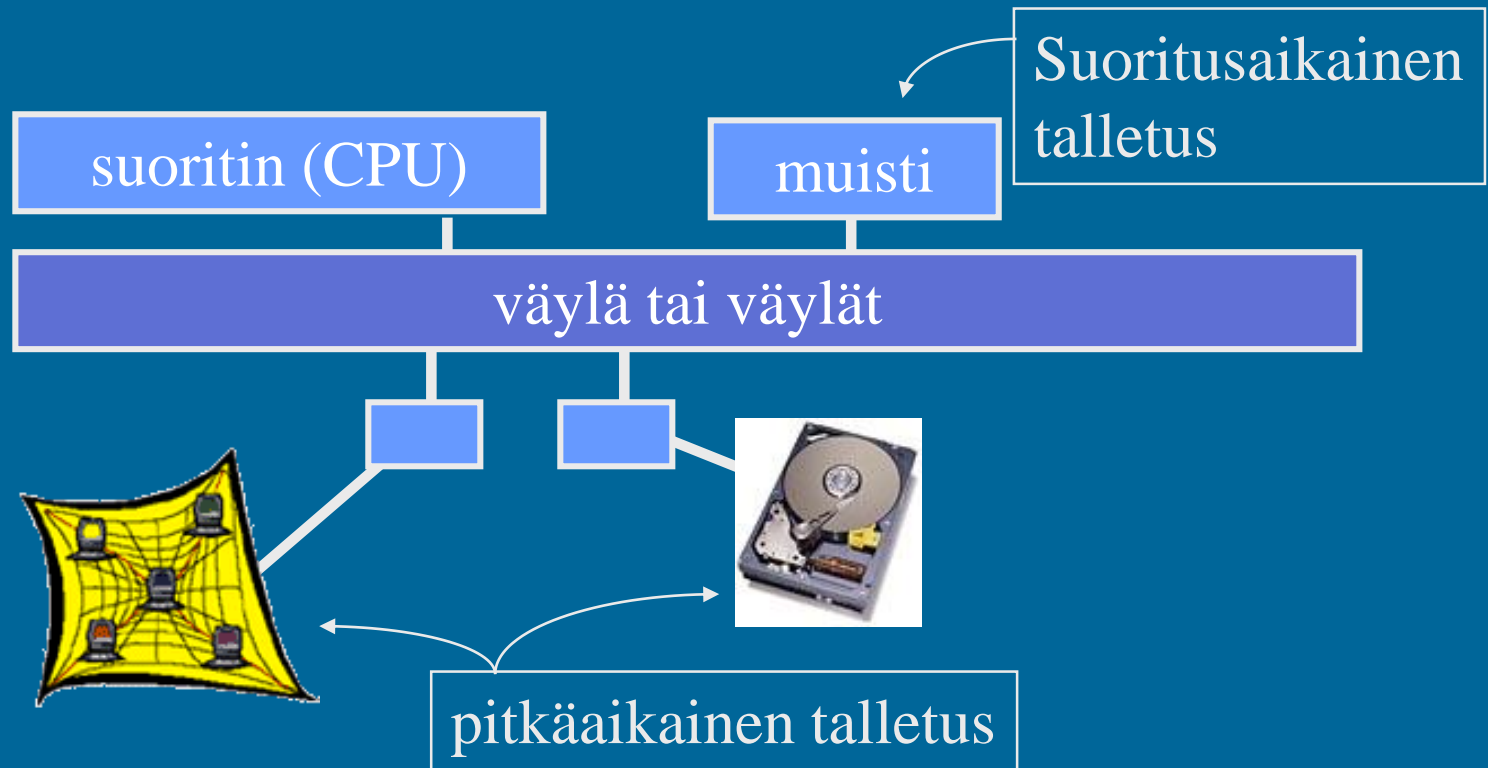
Keskusmuistin käyttö

# Tietokone

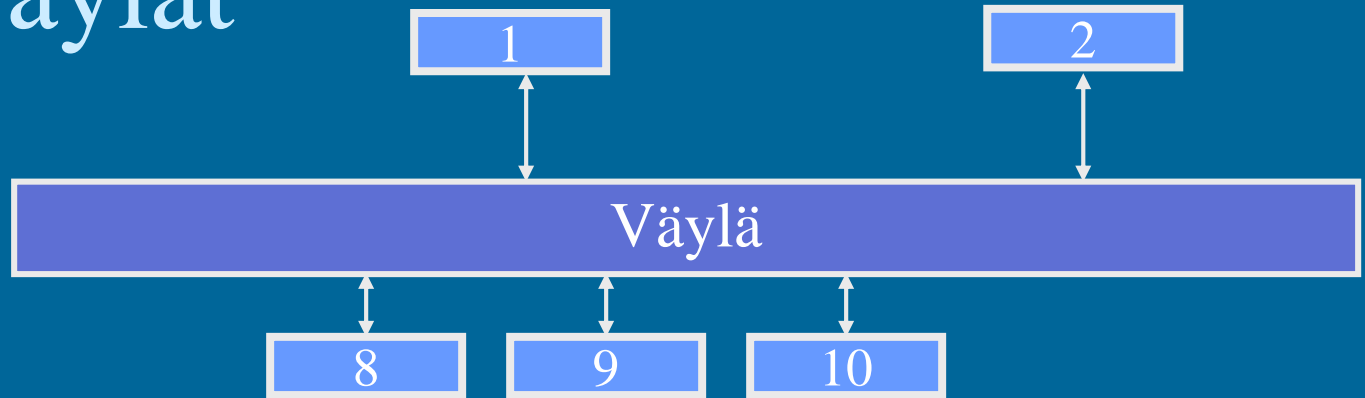


# Tietokoneohjelman sijainti

- Suoritusaikana muistissa
- Muuna aikana esim. levyllä, verkossa, tms.



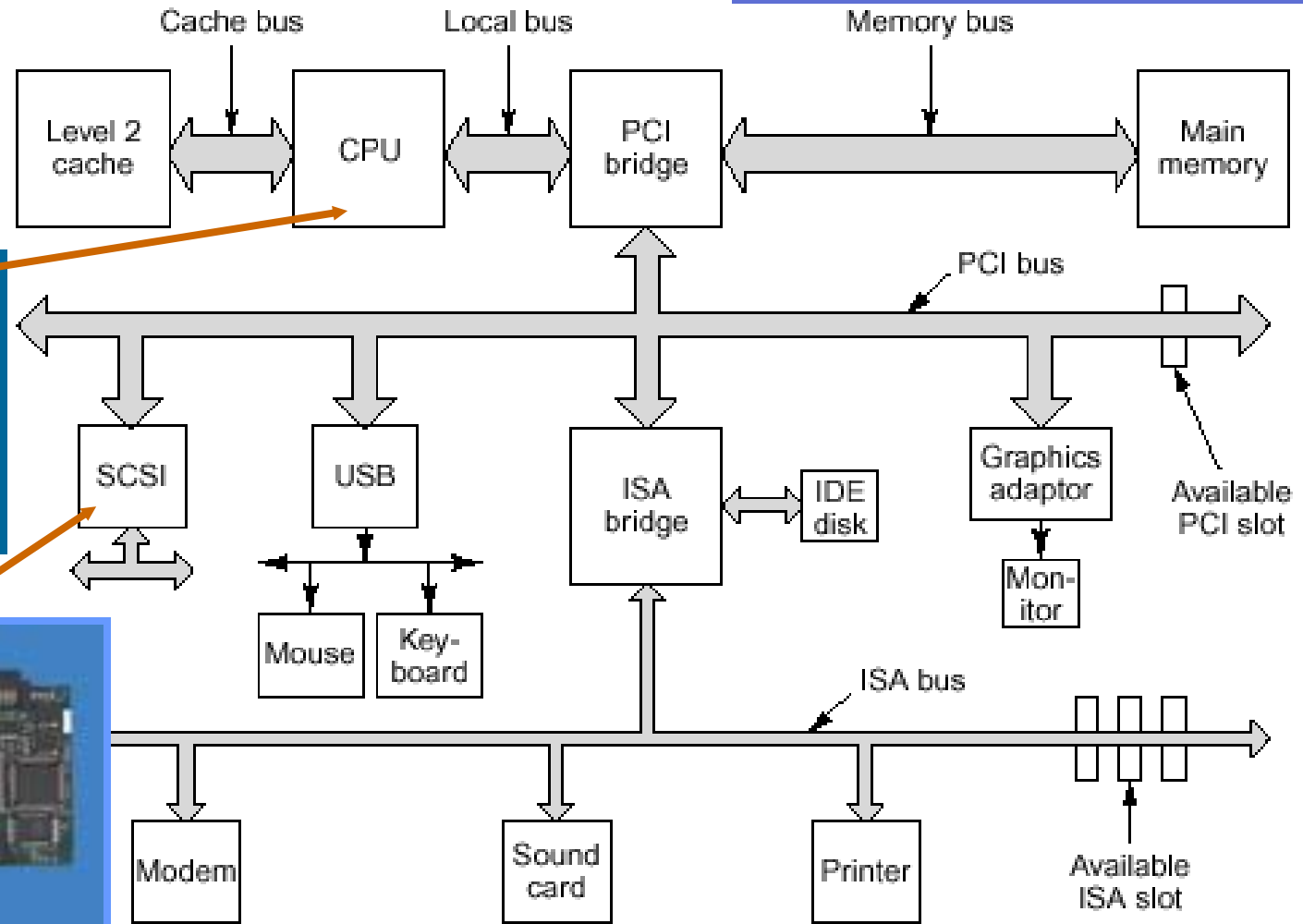
# Väylät



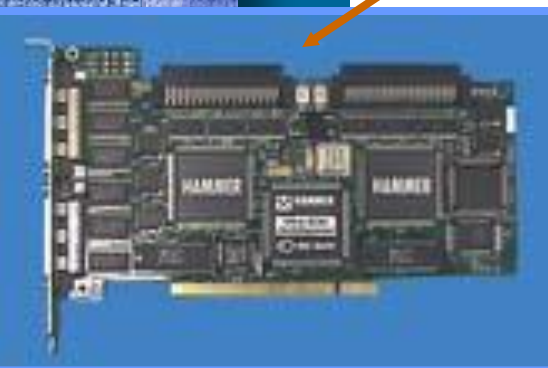
- Kullakin laitteella oma osoite
- Yksi lähettää kerrallaan, kaikki kuulevat, vain ”oikea” laite vastaanottaa ja reagoi
- Väylän kaikki laitteet toimivat samalla nopeudella
- Paljon erilaisia väyliä
- Lähellä suoritinta olevat väylät ovat nopeampia

# Väylähierarkia

Tyypillinen Pentium II  
systemin emolevy



omalla sirulla,  
tason 1  
välimuistin  
kanssa



PCI to SCSI bridge

Fig. 3-50 [Tane99]

# Symbolinen konekieli (assembler)

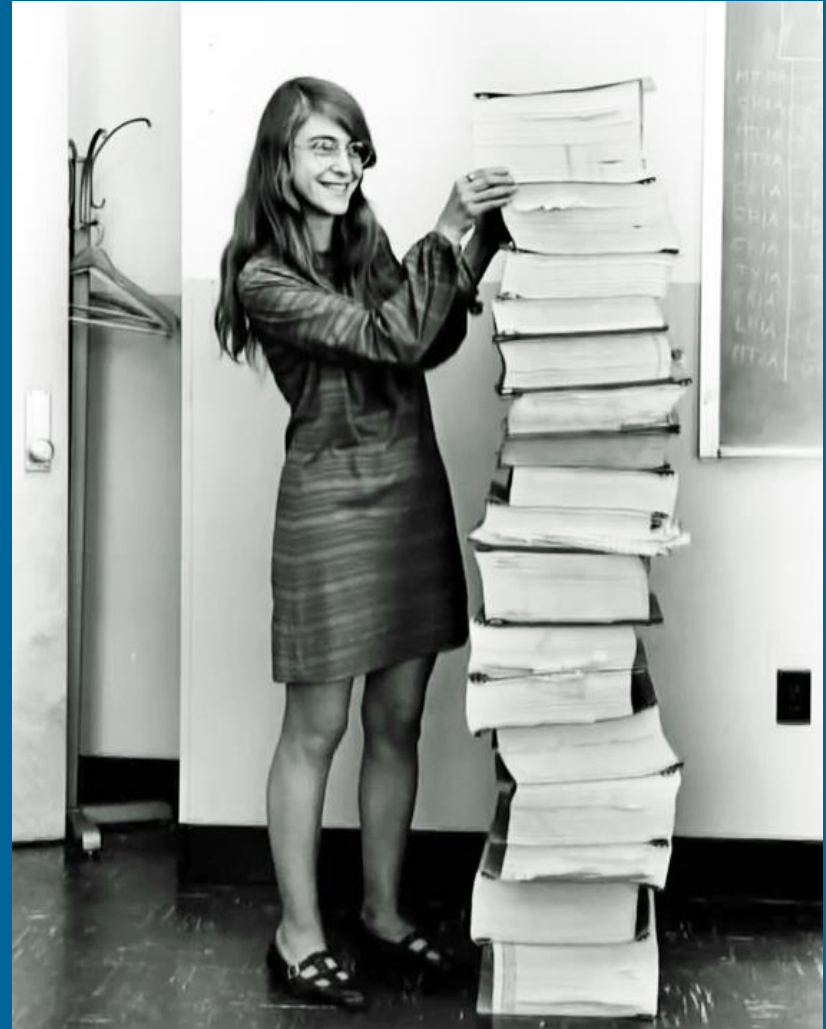
- Yleinen esitystapa konekielisille ohjelmille
  - luettavassa muodossa oleva konekieli
- Helppo muuttaa konekieleksi
  - suora vastaavuus konekieleeseen
  - usein mielletään (vähän väärin, muttei paljon):

symbolinen konekieli  $\approx$  konekieli

129543876	LOAD	R2, Summa	; R2 $\leftarrow$ Mem(Summa)
439874387	ADD	R2, =5	; R2 $\leftarrow$ R2 + 5
544399765	JUMP	Loop	; PC $\leftarrow$ Loop
	(koodi)		(; kommentti)

# Ohjelman toteutus symbolisella konekielellä (assemblerilla)

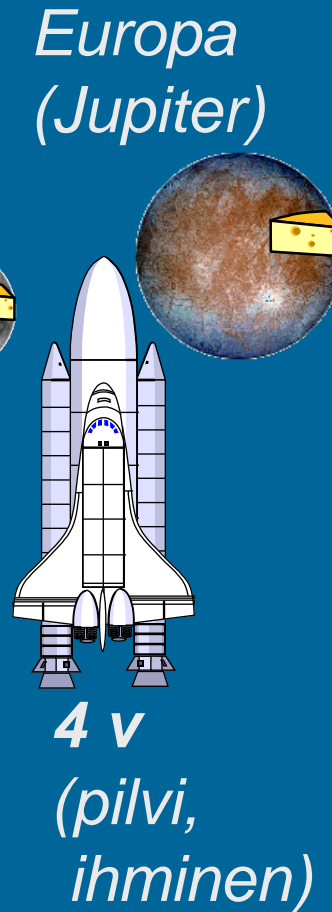
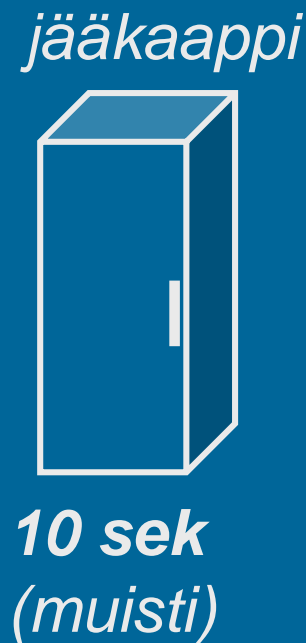
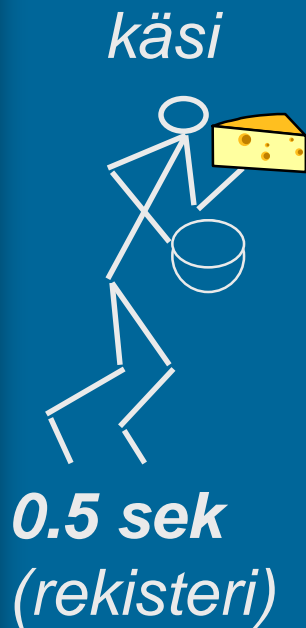
- Margaret Hamilton ja AGC:n (Apollo Guidance Computer) lähdekoodin listaus
- AGC ohjelmoitiin assemblerilla
- Margaret Hamilton
  - Otti käyttöön termin “ohjelmistotuotanto” käyttämiensä ohjelmointiperiaatteiden kuvaamiseen vuonna 1968
  - Augusta Ada Lovelace Award, 1986 (Ada Lovelace, 1815-1852, “ensimmäinen ohjelmoija”, Babbagen analyttinen kone)



[https://en.wikipedia.org/wiki/Apollo\\_Guidance\\_Computer](https://en.wikipedia.org/wiki/Apollo_Guidance_Computer)

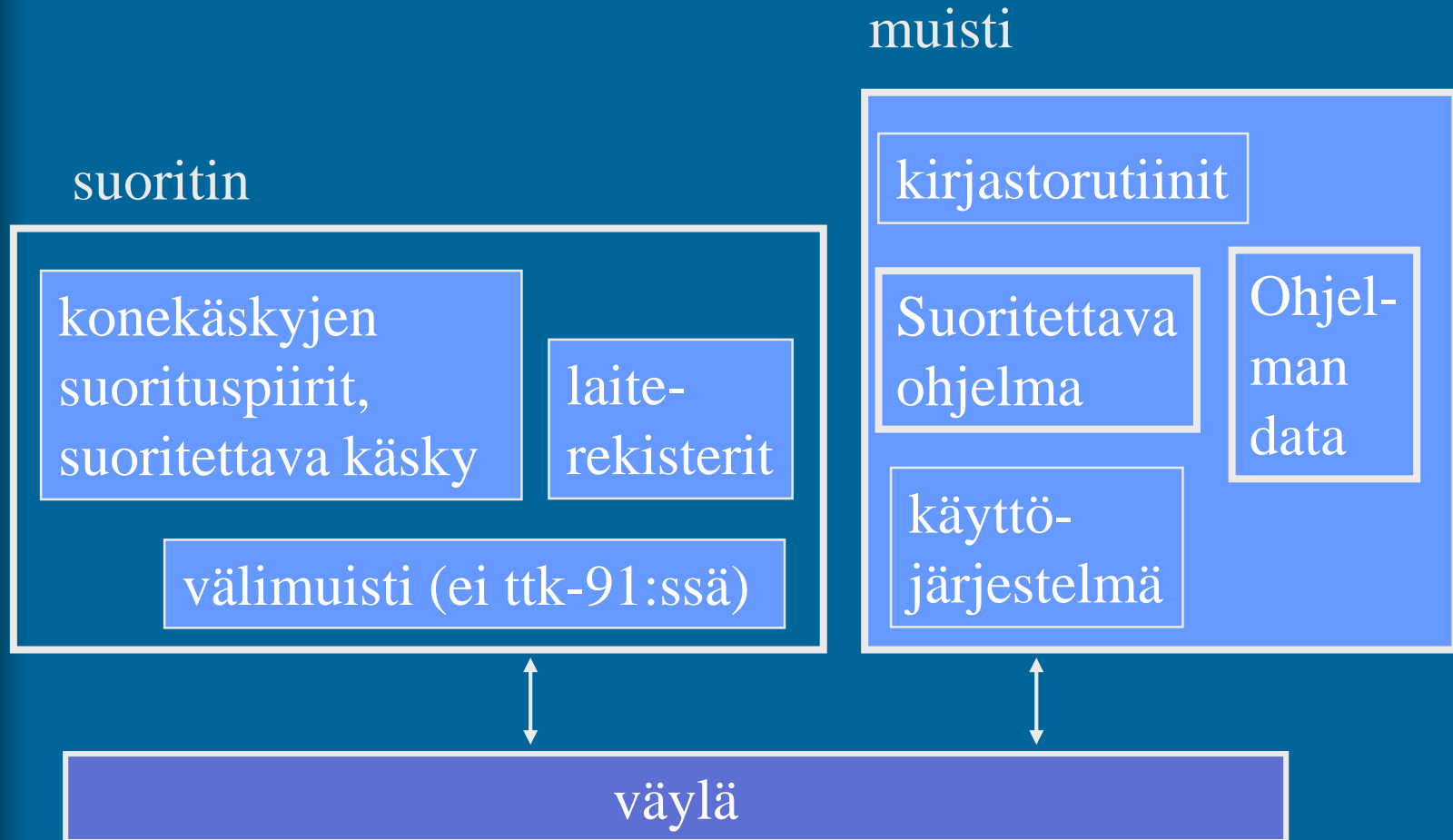
# Teemun juustokakku

Rekisterien, välimuistin, muistin, levymuistin ja magneettinauhan nopeudet suhteutettuna juuston hakuaikaan juustokakkuja tehdessä?

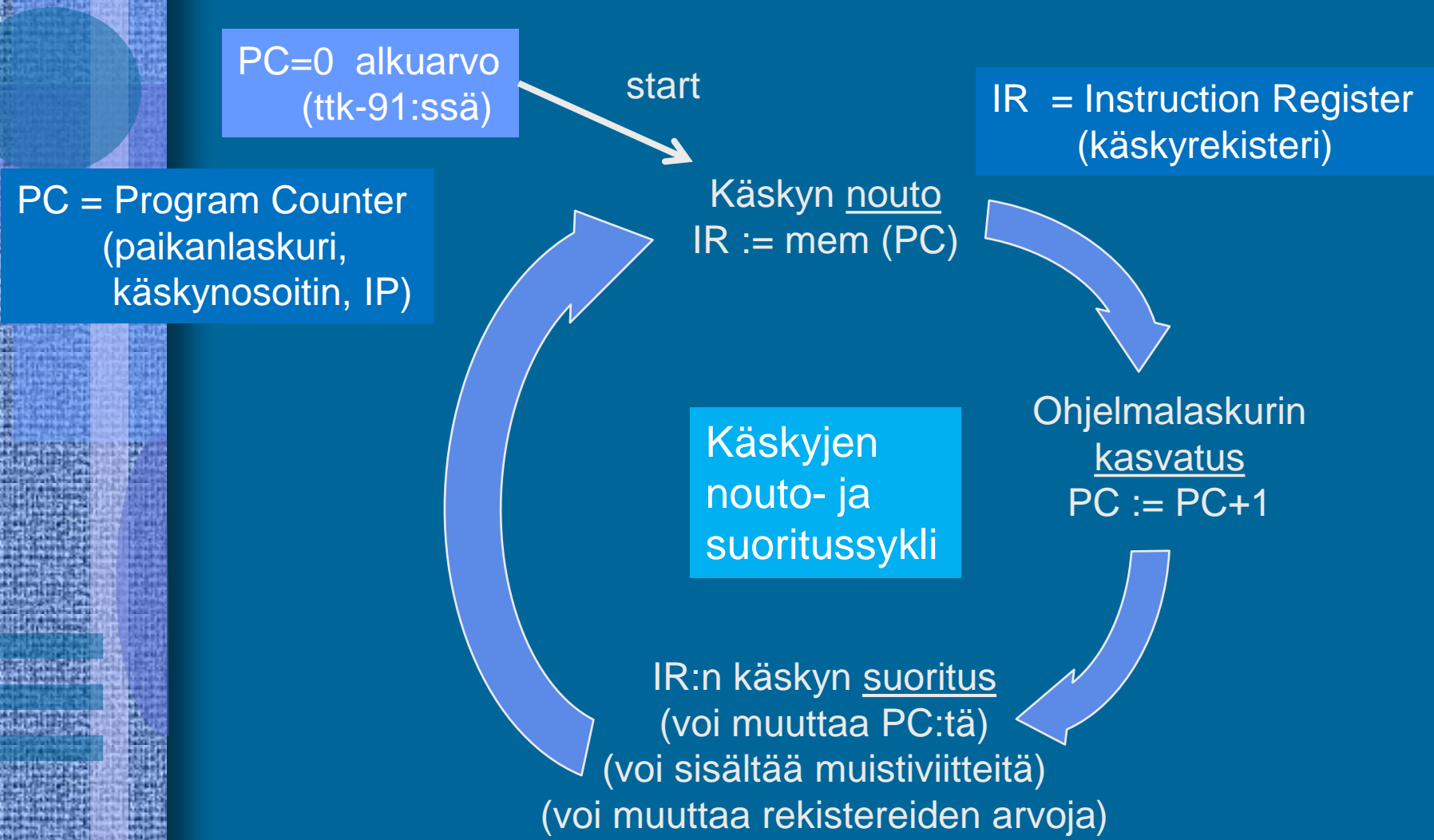




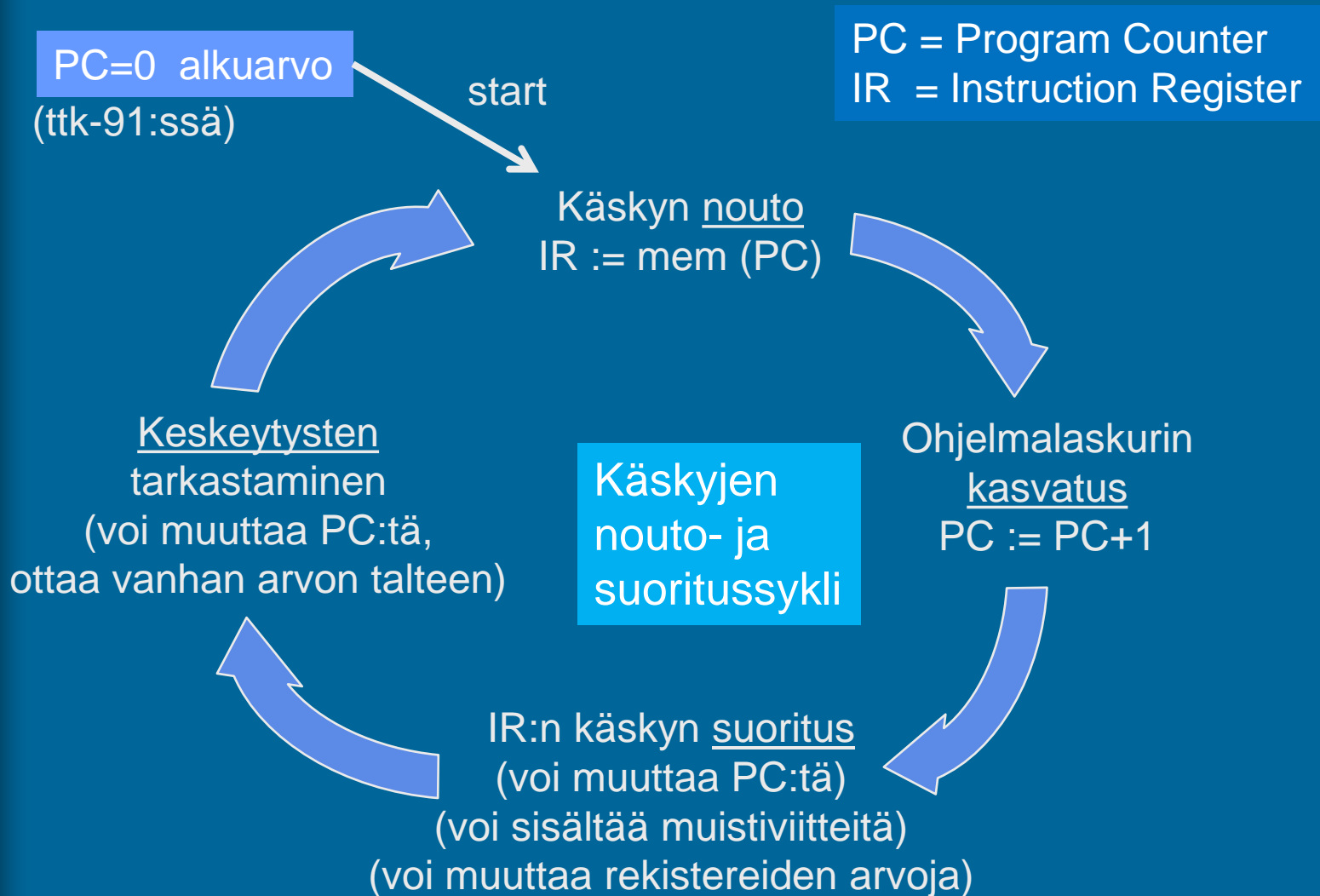
# Suoritusaikainen suorittimen ja muistin sisältö



# Suorittimen toiminta, perusidea



# Suorittimen toiminta, tarkemmin



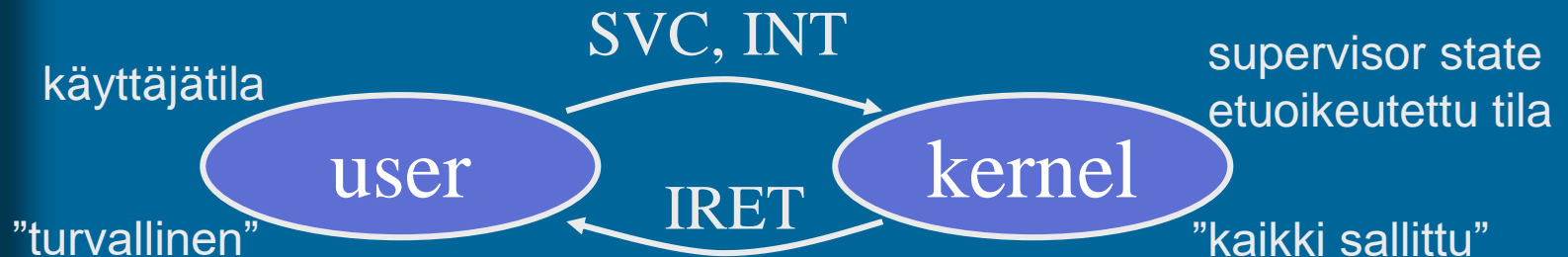
# Keskeytysten käsittely

- 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.)

# 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

# TTK-91 laitteisto

Central Processor Unit

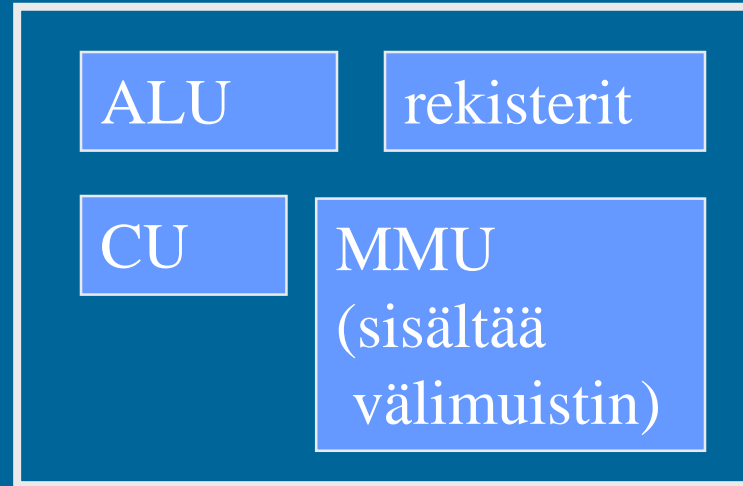
Arithmetic and Logic Unit

Control Unit

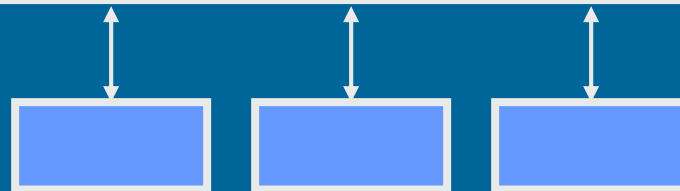
Memory Management Unit

Cache

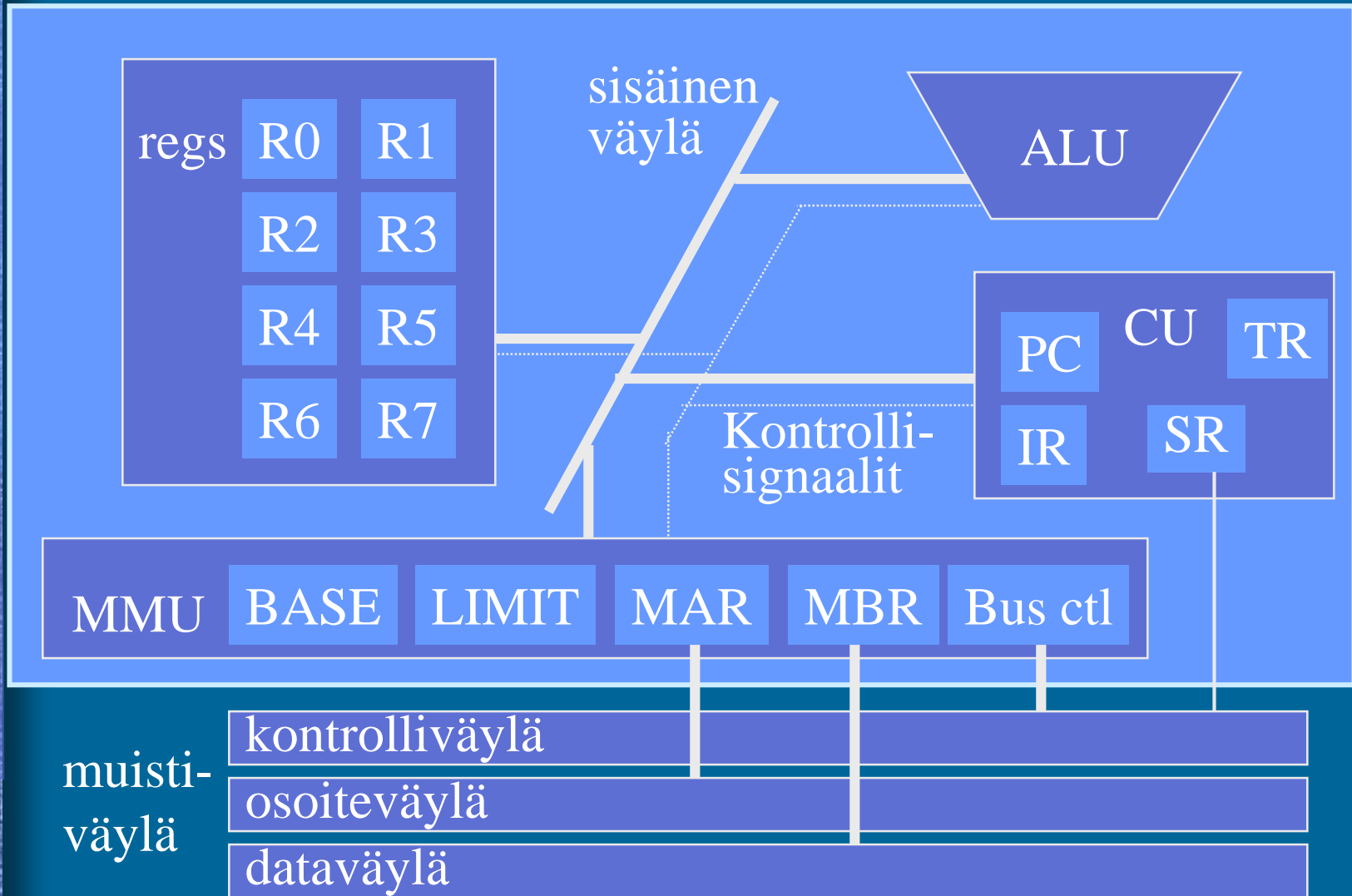
suoritin - CPU



muisti



# TTK-91 suorittimen rakenne





# 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)     | Stack Pointer |
|   |   | – ympäristöosoitin FP (R7) | Frame Pointer |

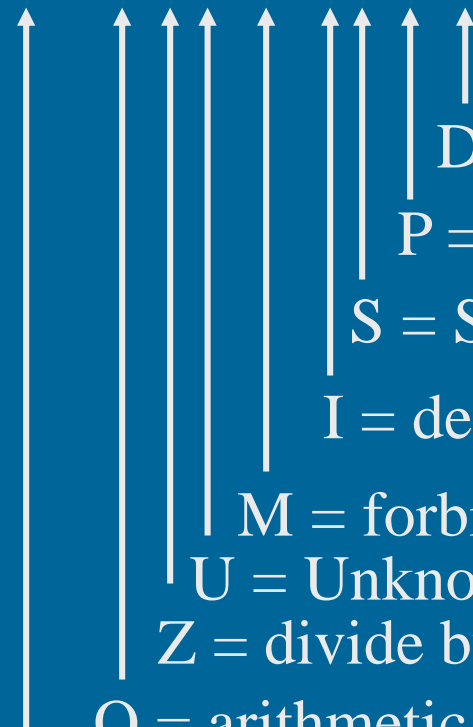
# TTK-91 Tilarekisteri SR

- 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?

# Ttk-91 Tilarekisterin SR kentät (9)

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

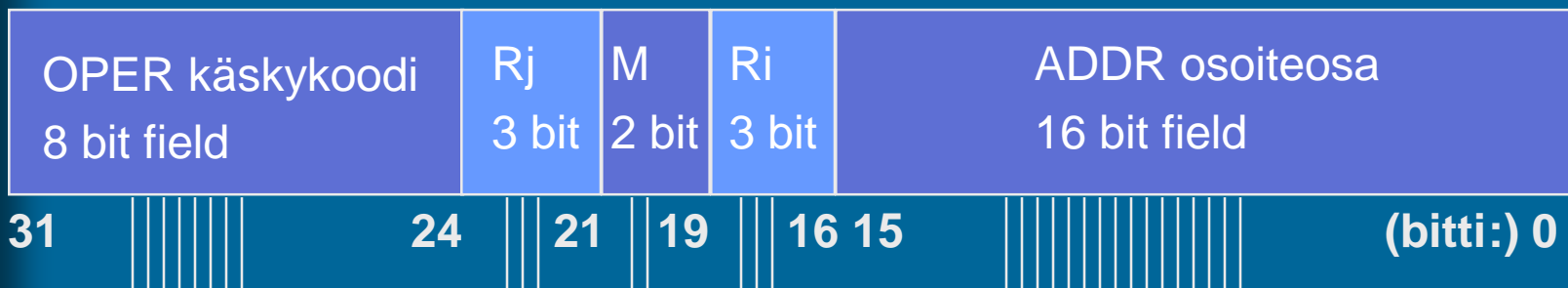
O = arithmetic Overflow

GEL = comparison indicators: Greater, Equal, Less

# 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

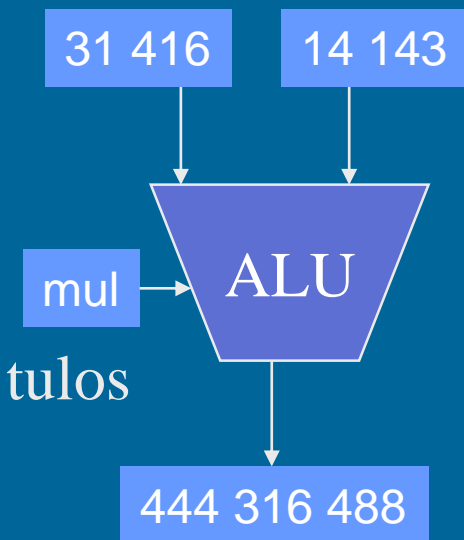
# Käskyn purku ja tehollisen muistiosoitteen (EA) laskemisvaihe



- 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

# Ttk-91 Aritmeettis-looginen yksikkö (ALU)

- Kaikkien aritmeettis-loogisten käskyjen toteutuspiirit
  - Add, sub, mul, div, and, or, xor, ...
  - Monimutkaisin operaatio: mul tai div
    - Esim.  $K = 31416 * 14143$
  - Syötä operandit, operaatio, odota, lue tulos
- Oikea suoritin
  - Monimutkaisin operaatio: fmul tai fdiv
    - Esim.  $X = 3.141592 * 1.4142857$



# Käskyn suorittaminen

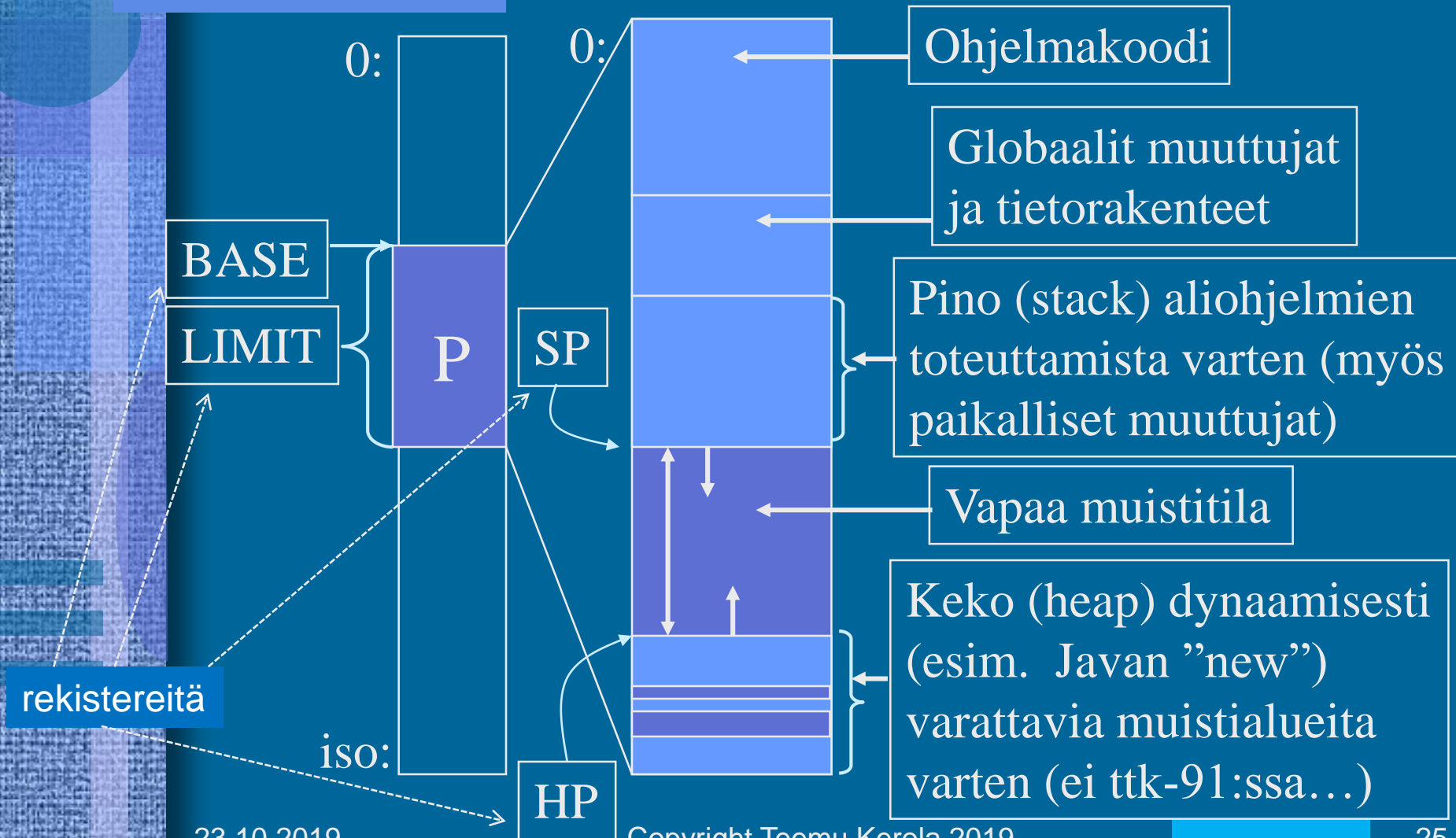
- Titokone näyttää vaiheet simuloimalla
- Store R4, @10(R1) ; R1 =20, R4=15
  - Käskyn nouto ja ohjelmanaskurin kasvatus:
    - PC → MAR, “bus read”, odota, mem → MBR
    - MBR → IR
    - PC+1 → PC
      - Esim: PC → ALU1, 1 → ALU2, +, odota, ALU → PC
      - Esim: PC → INCin, odota, INCout → PC (oma 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)



# 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?



käyttöjärjestelmäkurssi

Lisää tietoa?



käyttöjärjestelmäkurssi