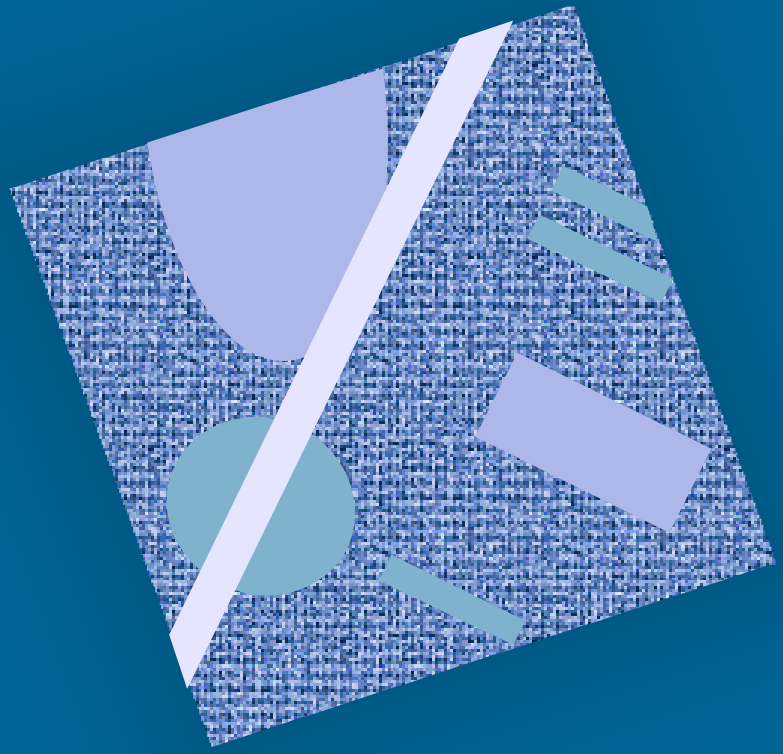
Jakso 5
Suoritin ja väylä



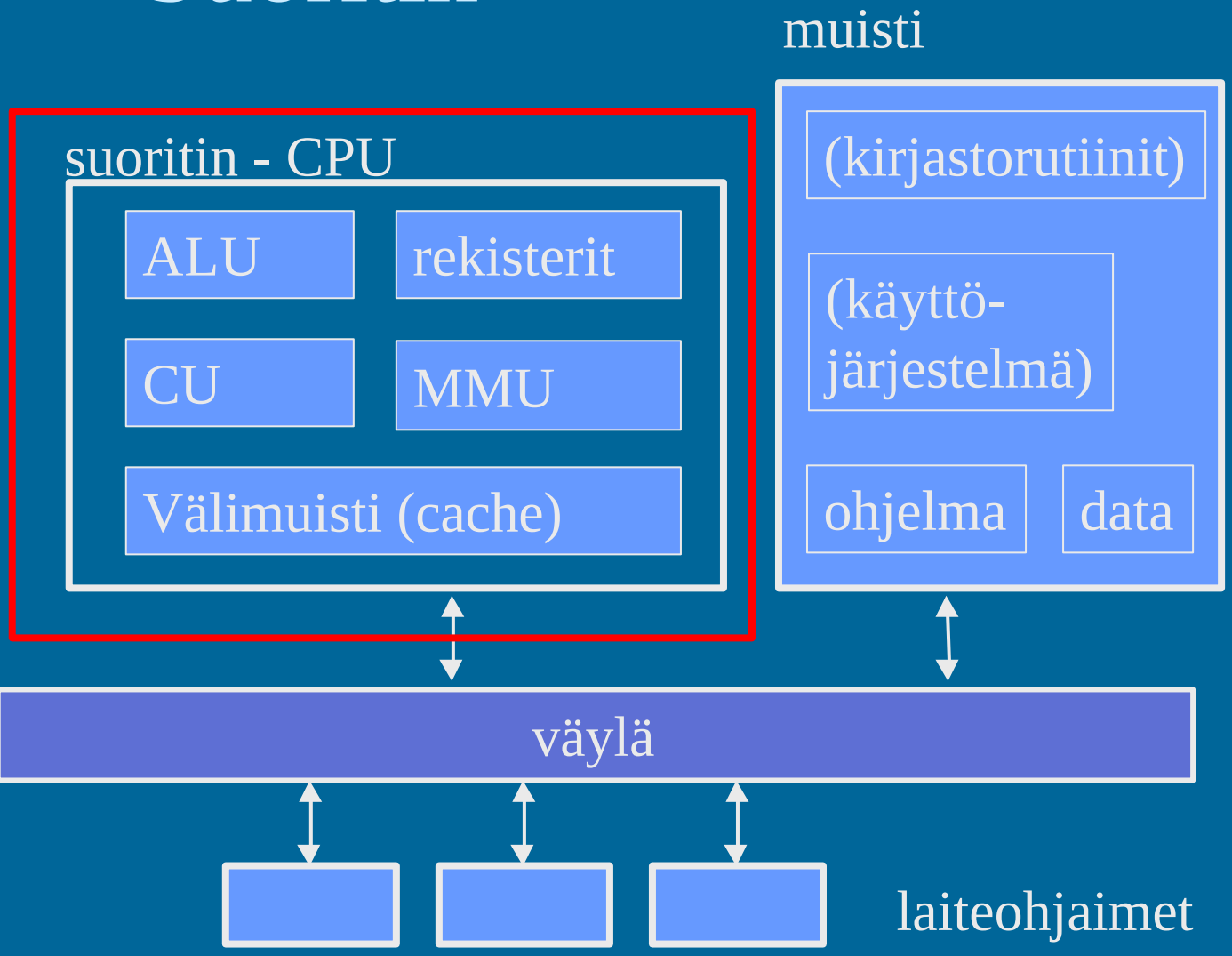
Suorittimen rakenne

Väylän rakenne

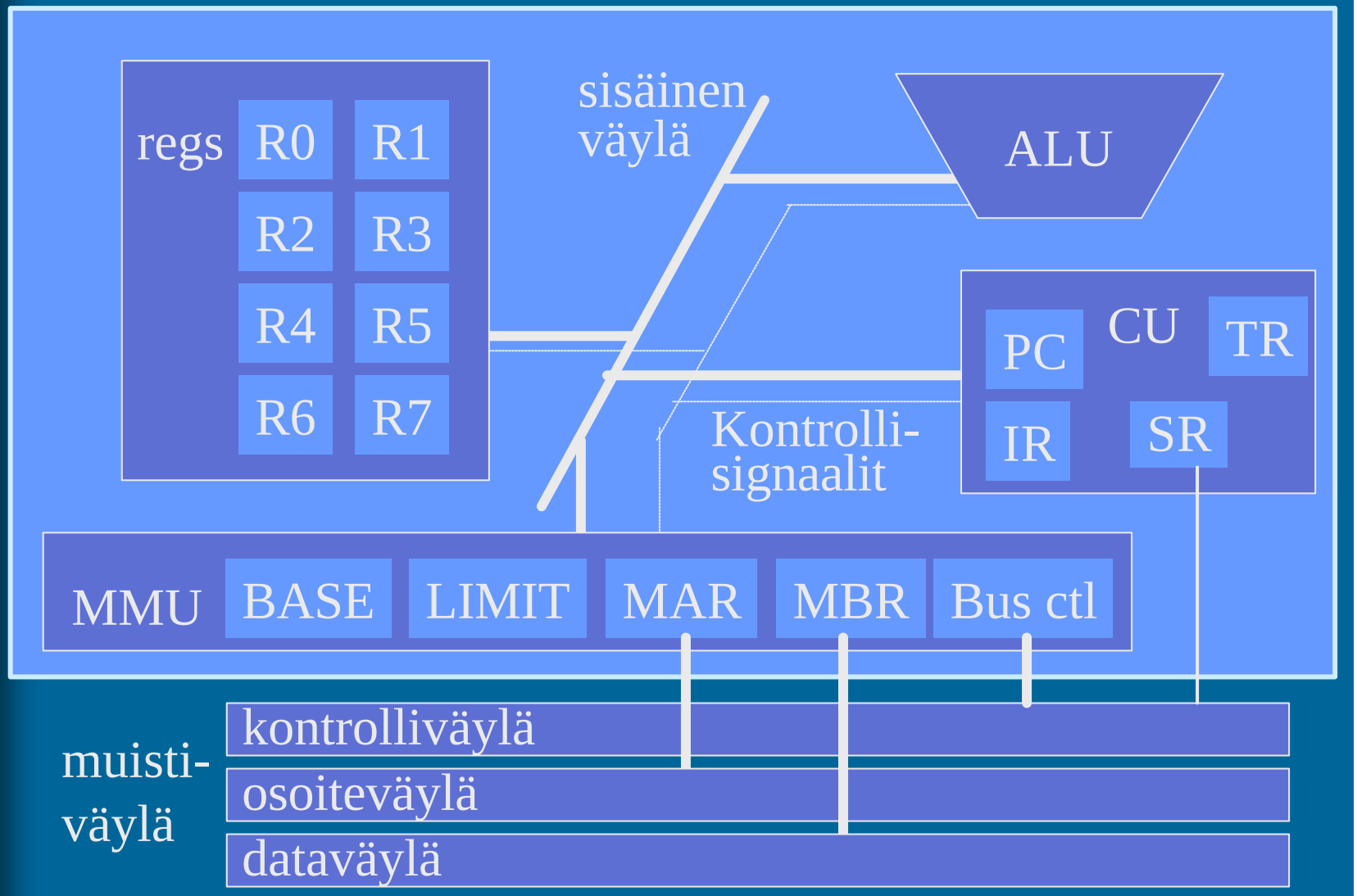
Käskyjen suoritussykli

Poikkeukset ja keskeytykset

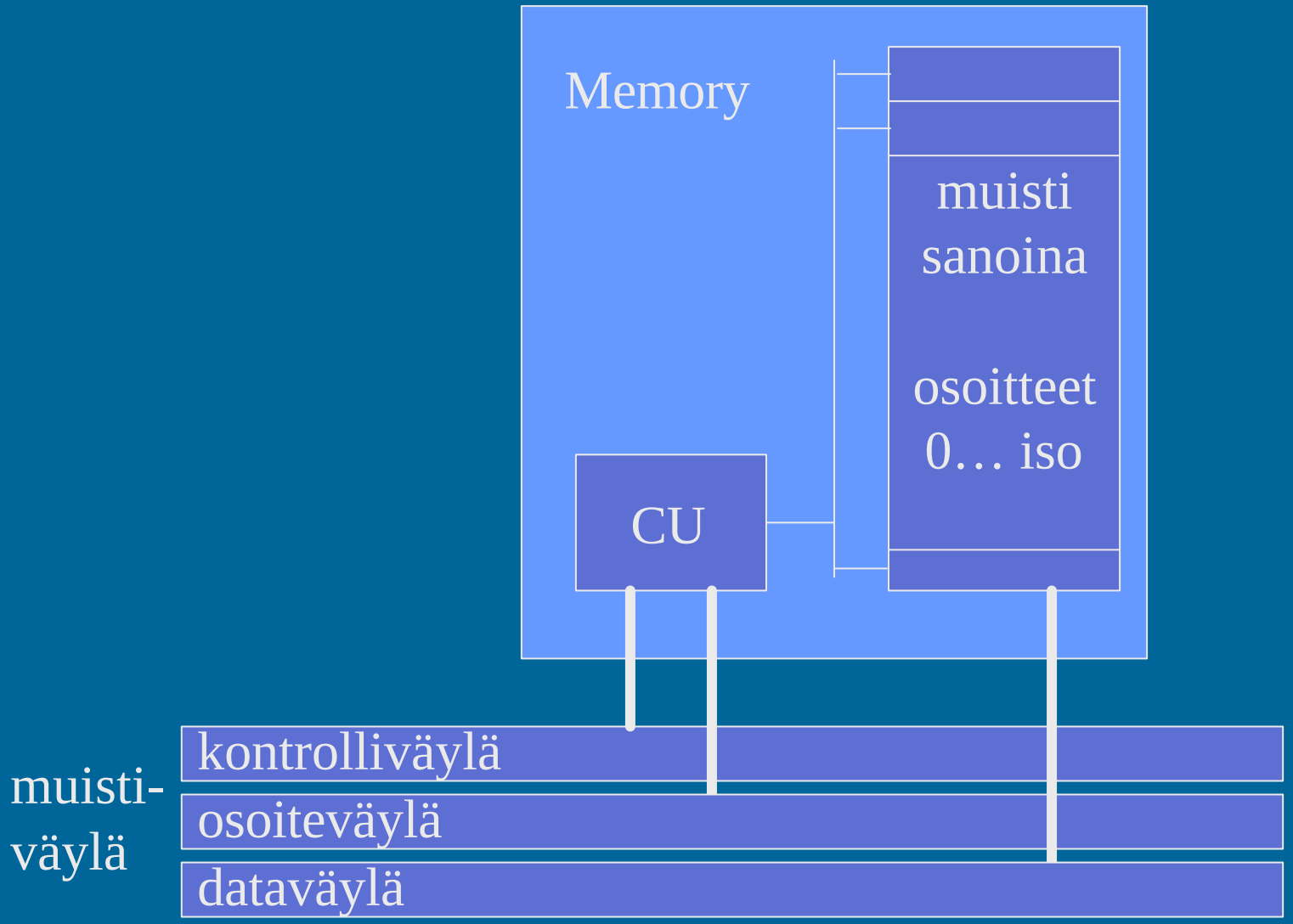
Suoritin


muisti
suoritin - CPU
ALU
rekisterit
CU
MMU
Välimuisti (cache)
(kirjastorutiinit)
(käyttö-
järjestelmä)
ohjelma
data
väylä
laiteohjaimet

TTK-91 suorittimen rakenne



TTK-91 muistin rakenne



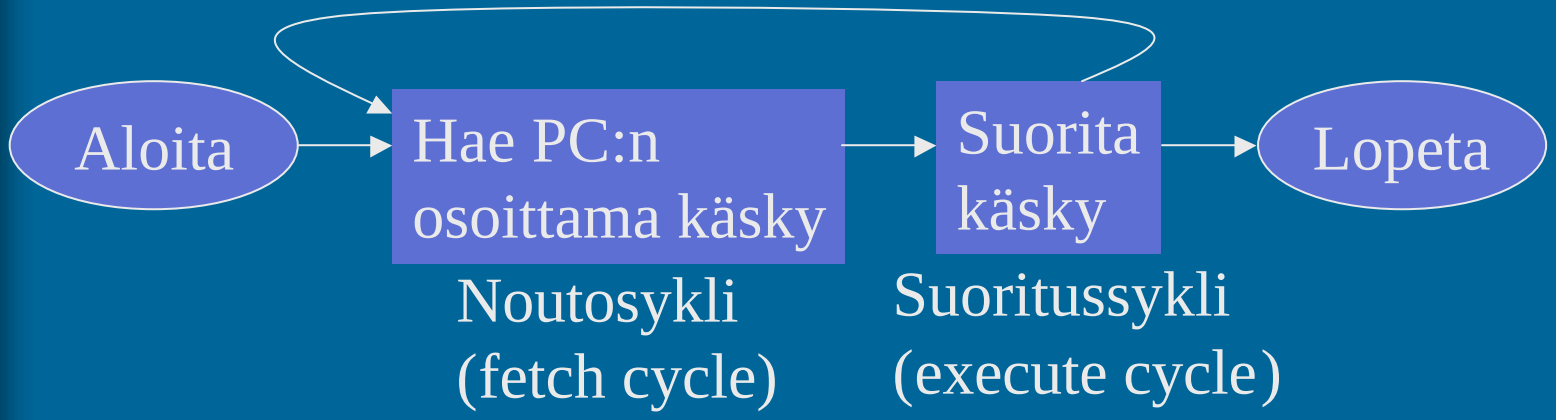
Käskyjen nouto- ja suoritussykli (5)

- Hae PC:n osoittama konekäsky muistista
  - lisää samalla PC:n arvoa yhdellä
- Suorita konekäsky
  - jos (ehdollinen) hyppykäsky, niin PC:n arvo voi vielä muuttua

Suoritin ei näe mitään suurempia kokonaisuuksia kuin konekäskyjä!

Suoritin ei tiedä mitään ohjelmista!

Nouto- ja suoritussykli



- Käskyn suoritus voi muuttaa systeemin tilaa
  - sisäiset ja ulkoiset rekisterit
  - muisti
  - laitteet

TTK-91 konekäskyn rakenne

- Käskyn esitys bittitasolla on aina:

muistiosoite tai (pienehkö) vakio

Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri (R0 ^{o} 0)

M = muistinoutojen määrä toiseen operandiin (addressing mode)
(ennen mahdollista muistiin talletusta)
00 eli 0 kpl, rekisteri tai välitön osoitus
01 eli 1 kpl, suora osoitus
10 eli 2 kpl, epäsuora osoitus
(11 eli 3 kpl, epäkelpo arvo → poikkeustilanne)

Nouto- ja suoritussykli tarkemmin (5)

- Noutovaihe
  - muistista MBR:n kautta IR:ään
  - Lisää 1 PC:hen
- Käskyn purku ja muistiosoitteen (EA) lasku
  - OPER, Rj, M, Ri, ADDR
  - TR ← (Ri) + ADDR (tai pelkkä ADDR)
- Operandin nouto
  - muistista MBR:n kautta TR:ään (0-2 krt ?)
- ALU operaatio
  - tulos rekisteriin R0-R7 tai TR:ään
- Muistiin talletus
  - muistiin MBR:n kautta

ks. TTK-91 suorittimen rakennekuva

Ei kaikilla käskyillä

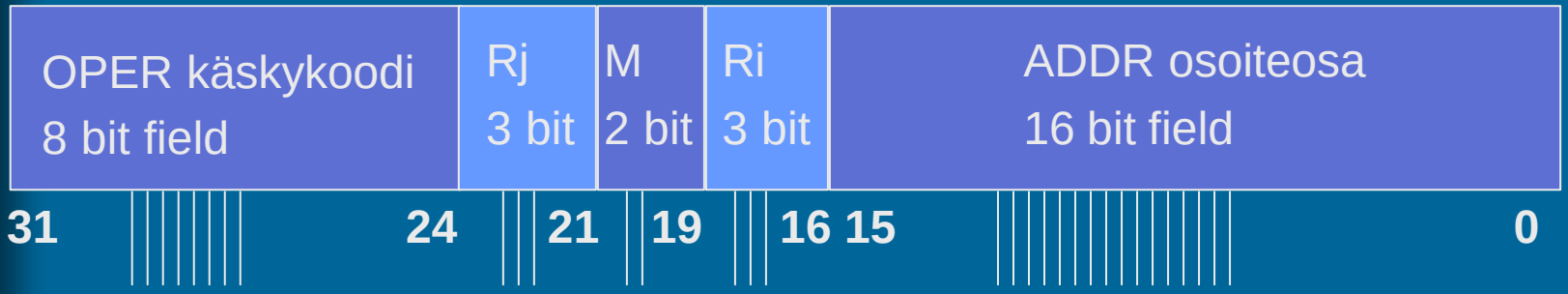
Ei kaikilla käskyillä

Käskyn noutovaihe (4)

ks. TTK-91 suorittimen rakennekuva

- Vie PC:n arvo MAR:iin
- Aseta muistin kontrollisignaalit väylälle asentoon ”lue”
- Odota kunnes muistiväylä vapautuu ja muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä konekäsky MBR:stä IR:ään

Käskyn purku ja muistiosoitteen laskemis -vaihe



- Purku automaattisesti langoitettuna IR:stä
- Muistiosoitteen lasku, tulos TR:ään
  - jos Ri=0, niin ADDR
  - muutoin laske (Ri)+ADDR
    - ALU suorittaa laskutoimituksen
  - Effective Address (EA)

Operandin luku vaihe (4)

ks. TTK-91 suorittimen rakennekuva

- Vie muistiosoite MAR:iin
- Aseta muistin kontrollisignaalit väylälle asentoon ”lue”
- Odota kunnes muistiväylä vapautuu ja muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä sana MBR:stä TR:ään
  - tai suoraan johonkin laiterekisteriin (R0-R7)

ALU operaatio -vaihe (10)

- Lähtötilanne
  - käsky haettu ja purettu osiin IR:ssä
  - 1. operandi rekisterissä (R0, …, R7)
  - 2. operandi TR:ssä
- Käskyn suoritus ALUssa
  - vie operandit sisäistä väylää pitkin ALU:un
  - anna ALU:lle sopiva ohjaussignaali
    - add, mul, copyLeft, comp, ...
  - odota, että tulos valmis
  - talleta tulos rekisteriin, MBR:ään, PC:hen ja/tai SR:ään

ks. TTK-91 suorittimen rakennekuva

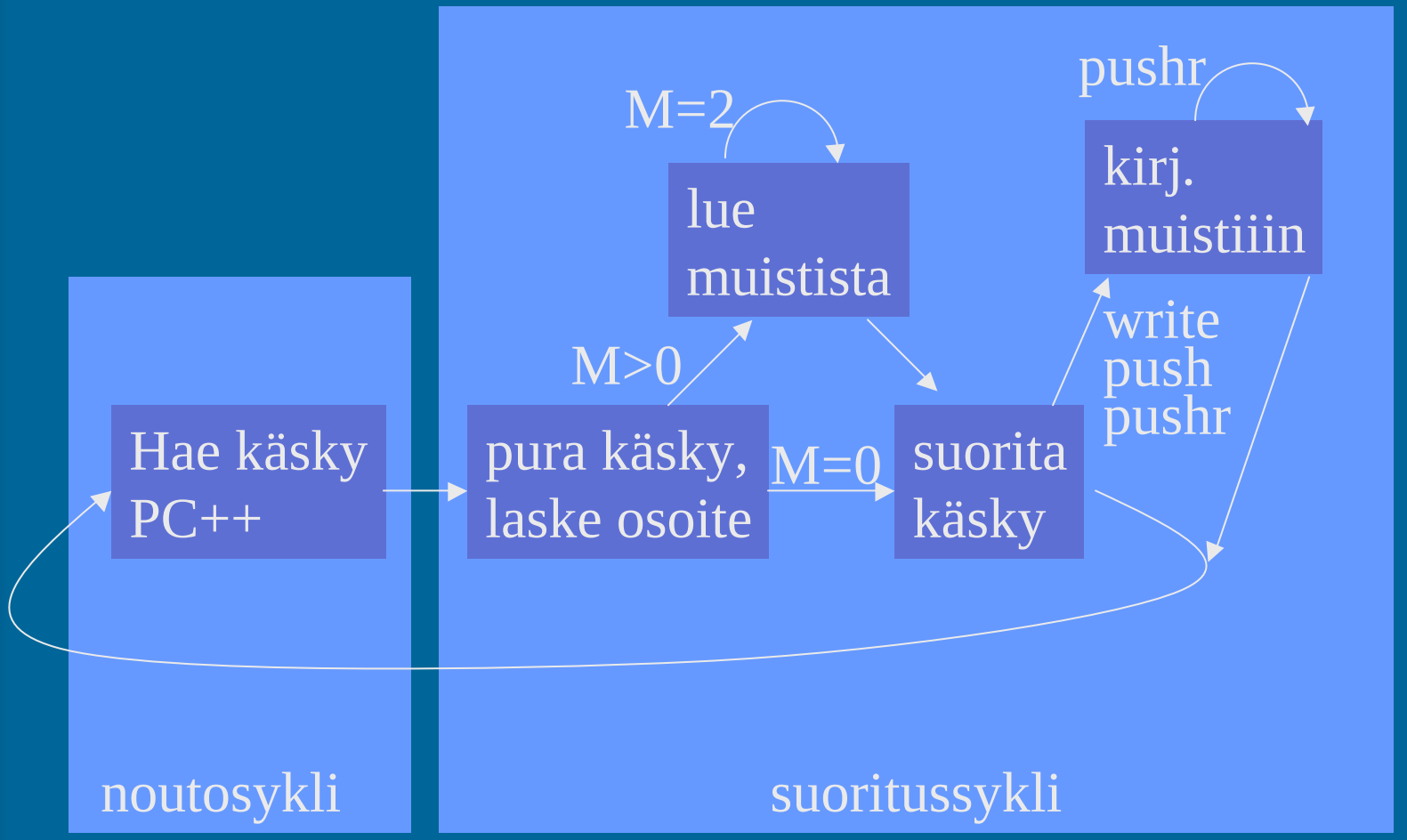
Tässä tapahtuu tietokoneen tekemä työ, kaikki muu on hallintoa

Tuloksen muistiin kirjoitus -vaihe (4)

ks. TTK-91 suorittimen rakennekuva

- Vie muistiosoite MAR:iin
- Vie kirjoitettava sana MBR:ään
- Aseta kontrollisignaalit väylälle asentoon ”kirjoita muistiin”
- Odota kunnes muistiväylä vapautuu, sana siirretään muistiin väylää pitkin, ja väylän kontrollisignaalit kertovat muistiinkirjoittamisen tapahtuneen

TTK-91 Nouto- ja suoritussykli vähän tarkemmin



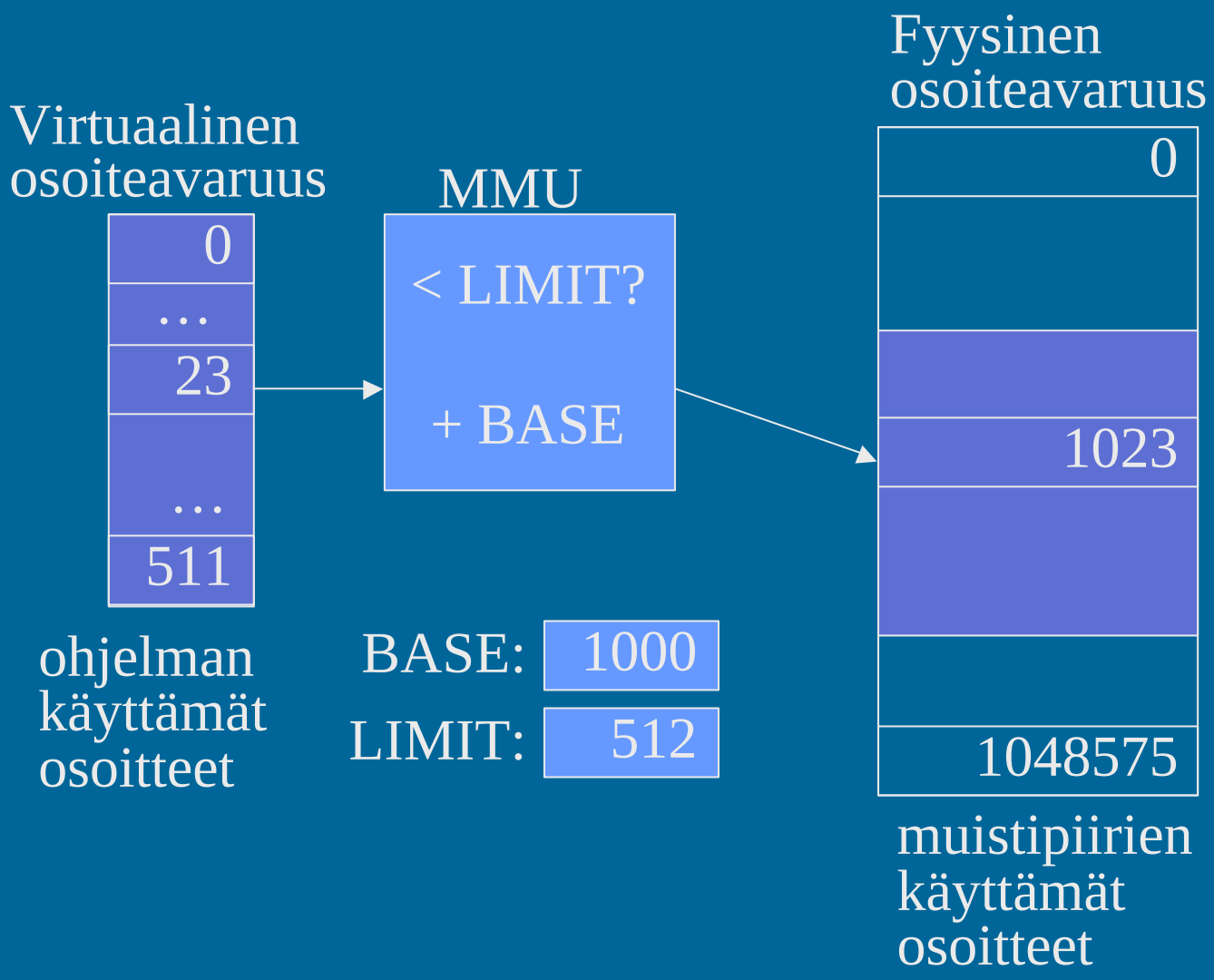
MMU:n toiminta (2)

ks. TTK-91 suorittimen rakennekuva

- Ohjelman käyttämät muistiosoitteet (VA) ovat näennäisiä, välillä 0 … LIMIT-1
  - ne eivät ole samoja osoitteita kuin keskusmuisti käyttää
- MAR:iin menevä arvoa VA ei käytetä suoraan, vaan se tarkistetaan ja muokataan ensin
  - Tarkista, onko VA ∈ [0, LIMIT-1].
    Jos ei ole, niin aseta bitti M SR:ssä päälle, ja lopeta käskyn suoritus
  - Lisää VA:han BASE ja laita tämä arvo (PA) MAR:iin

VA = virtual address, PA = physical address = BASE+VA

TTK-91 virtuaalimuisti



Virtuaalimuistin menetelmiä (4)

- Kanta- ja rajarekisteriin perustuva
  - base ja limit rekisterit
- 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

Keskeytystilanteet (3)

- Mikä tahansa tilanne, jonka käsittely vaatii poikkeuksen käskyjen normaaliin suoritusjärjestykseen
- Rakkaalla lapsella on monta nimeä:
  - poikkeus, keskeytys, virhetilanne, trappi, …
  - exception, interrupt, fault, trap, failure, ….
- Jatkossa yleisnimi keskeytys tarkoittaa kaikkia näitä eri tapauksia tai tyyppejä

Keskeytysten käsittely (4)

- Jokainen mahdollinen keskeytystyyppi on ennalta tunnettu
- Jokaiselle keskeytystyypille on oma käyttöjärjestelmän tuntema keskeytyskäsittelyrutiini

interrupt handler

- Käskyn suorituksen jälkeen tarkistetaan keskeytysten olemassaolo SR:stä ja haaraudutaan keskeytyskäsittelijään tarvittaessa
  - joskus keskeytykset on estetty (SR:n bitti D)
  - paluu käsittelijästä ”return-from-interrupt” käskyllä
- ”Yllättävä aliohjelmakutsu”

Keskeytystyyppejä (14)

- Käskyn aiheuttamat virhetilanteet
  - virheellinen käskyn tai datan osoite
  - tuntematon käsky (opcode)
  - nollalla jako
  - kokonaisluvun tai liukuluvun yli/alivuoto
  - käytetty osoite ei ole muistissa (MMU)
- Käskyn aiheuttamat muut poikkeustilanteet
  - SVC käsky
  - I/O konekäsky
  - trace keskeytys
- Ulkoapäin (muualta kuin CPU:lta) tulleet
  - kellolaitekeskeytys (esim. joka 10 ms)
  - laitekeskeytys (esim. levy I/O valmis)
  - laitteistovirhe (esim. virhe väylän tiedonsiirrossa)

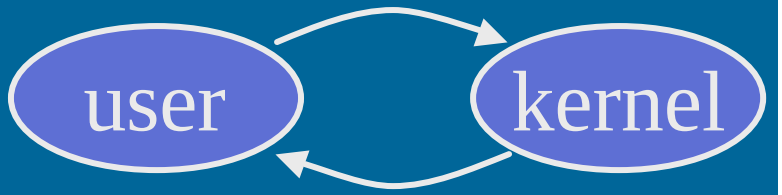
Keskeytyskäsittelijä

- Osa käyttöjärjestelmää
- Ennen käsittelijän aloittamista asetetaan suoritin (bitti P SR:ssä) ja MMU (BASE=0, LIMIT=”hyvin iso”) käyttöjärjestelmätilaan.

(supervisor state)

  - käyttöjärjestelmätilassa saa viitata mihin tahansa kohtaan muistia
  - käyttöjärjestelmätilassa saa käyttää kaikkia konekäskyjä
- Käsittelijästä paluun yhteydessä MMU:n tila ja prosessorin tila asetetaan ennalleen

Prosessorin tilat (6)



- Käyttäjätila (user mode, normal mode)
  - voi käyttää vain tavallisia käskyjä
  - voi viitata vain käyttäjän omaan muistiavaruuteen (MMU valvoo)
- Etuoikeutettu tila tai (KJ:n) ytimen tila (kernel mode, privileged mode)
  - voi käyttää kaikkia konekäskyjä, myös etuoikeutettuja
  - voi viitata kaikkialle muistiin, myös käyttöjärjestelmän ytimeen (kernel)
    - voi käyttää (myös) suoria muistiosoitteita (PA)

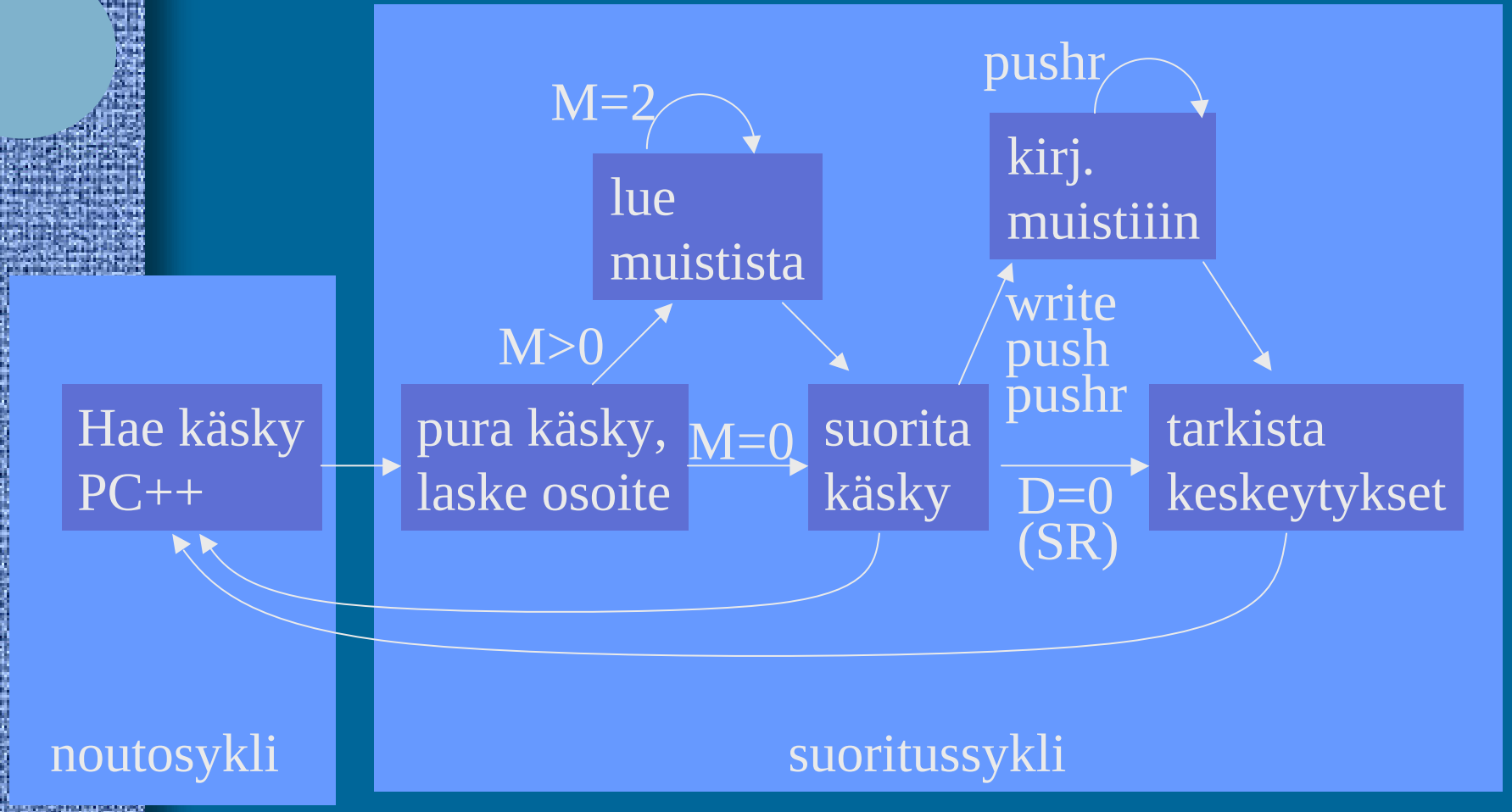
Prosessorin tilan muuttaminen (6)

- Käyttäjätila → etuoikeutettu tila
  - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
  - keskeytyskäsittelijä tarkistaa onko oikeutta tilan vaihtoon

(interrupt handler)

- Etuoikeutettu tila → käyttäjätila
  - etuoikeutettu konekäsky “return from interrupt handler” esim. IRET (Pentium II)
  - palauttaa kontrollin ja prosessorin tilan keskeytyneeseenkohtaan ja sen hetkiseen tilaan

TTK-91 Nouto- ja suoritussykli vielä vähän tarkemmin



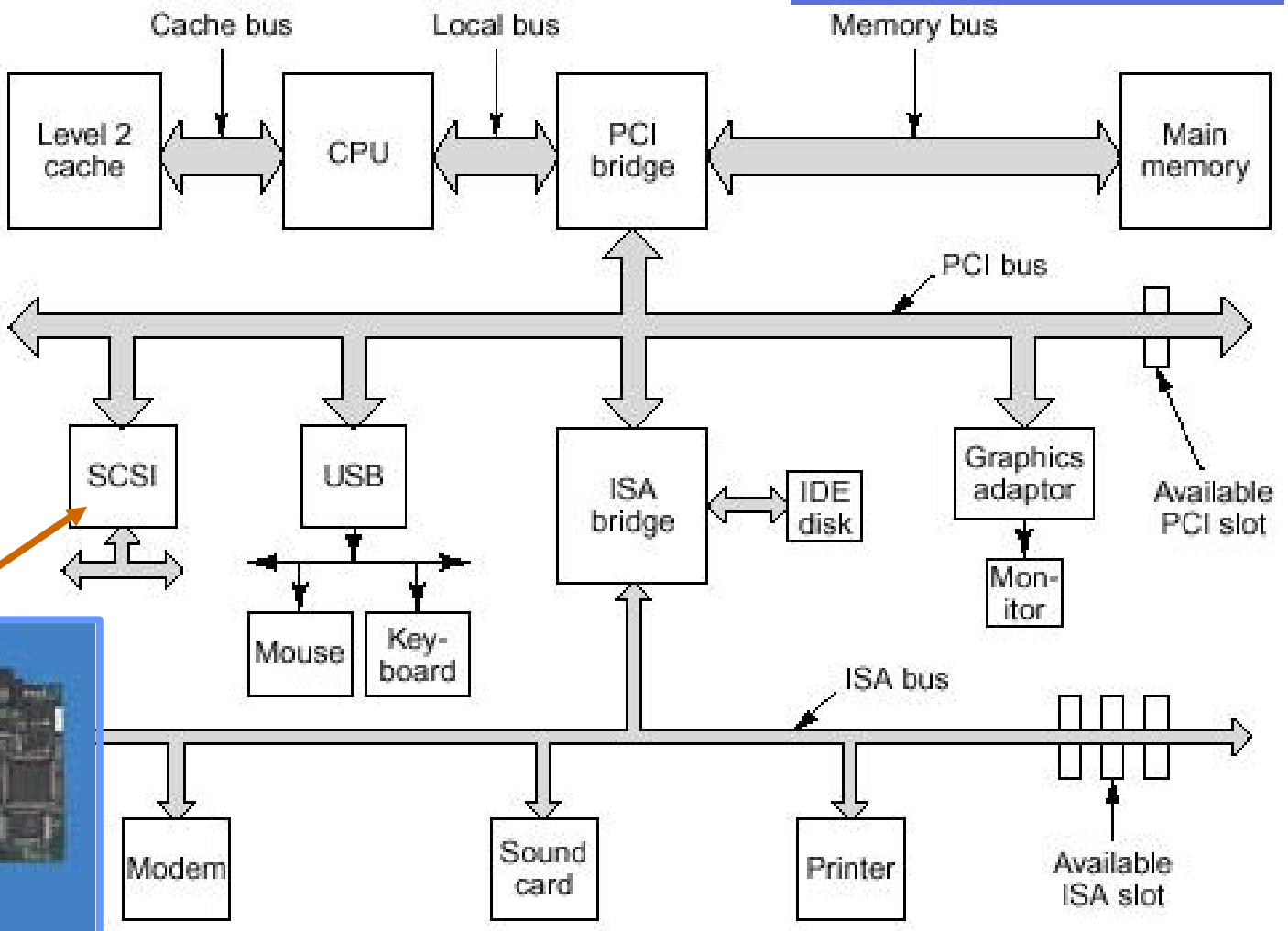
Väylät (5)

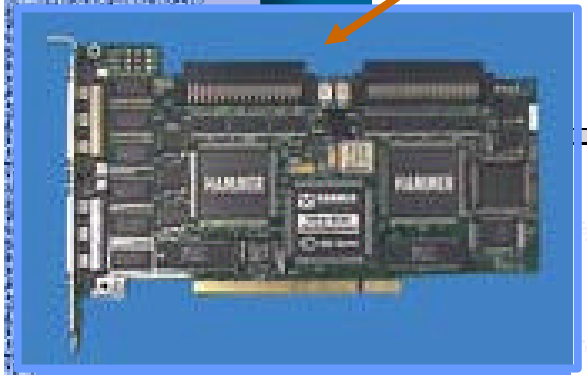
- Tiedon siirtoa varten laitteistossa
- Yksi käyttäjä kerrallaan
- Toteutettu johdinkimppuina

- Eri tasoilla
  - suorittimen sisällä ”sisäinen väylä” (internal bus)
  - muistiväylä suorittimen ja muistin välillä (memory bus)
  - I/O-väylä muistiväylän ja I/O-laitteiden välillä (I/O bus)
- Useita eri tapoja yhdistellä edellä olevia

Väylähierarkia

Typical Pentium II system

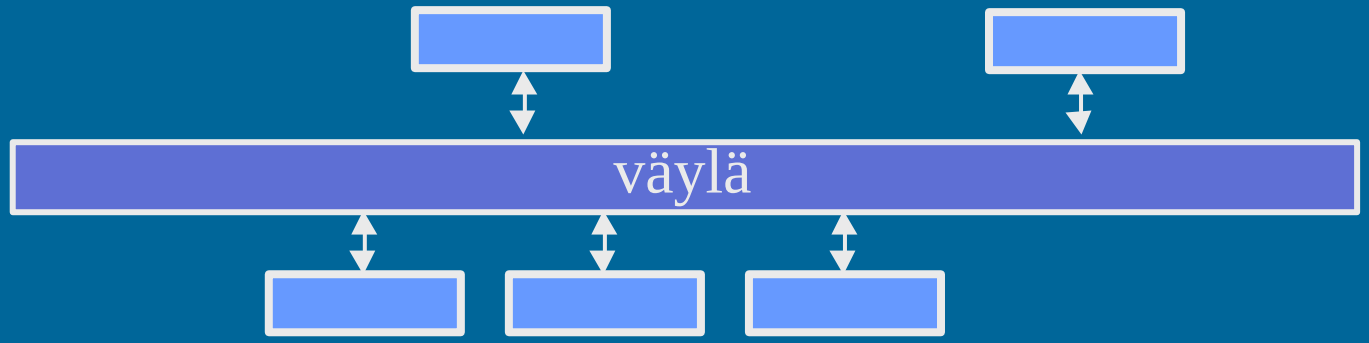




PCI to SCSI bridge

Fig. 3-50 [Tane99]

Väylät (5)



- Kullakin laitteella oma osoite
- Yksi lähettää, kaikki kuulevat
- Paljon erilaisia
- Lähellä prosessoria olevat ovat nopeampia

Lisää tietoa? → Tietokoneen rakenne kurssi

TTK-91 koneen KOKSI simulaattori (6)

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

TTK-91 käskyn suoritussykli (5)

hae käsky simuloidusta muistista

IR = mem[PC]

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

ADDR = IR % 65536

TR = regs[Ri] +ADDR

tee tarvittava määrä (M) operandin
hakuja muistista rekisteriin TR

TR = mem[TR]

valitse aliohjelma operaatiokoodin (OPER) perusteella

if (opcodeOK[OPER] = FALSE) then SR.U = 1;

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

ADD Ri, M ADDR(Rj) ⇒ regs[Ri] += TR;

lopeta suoritus jos SVC tai keskeytys

SR.O = …

-- Jakson 5 loppu --

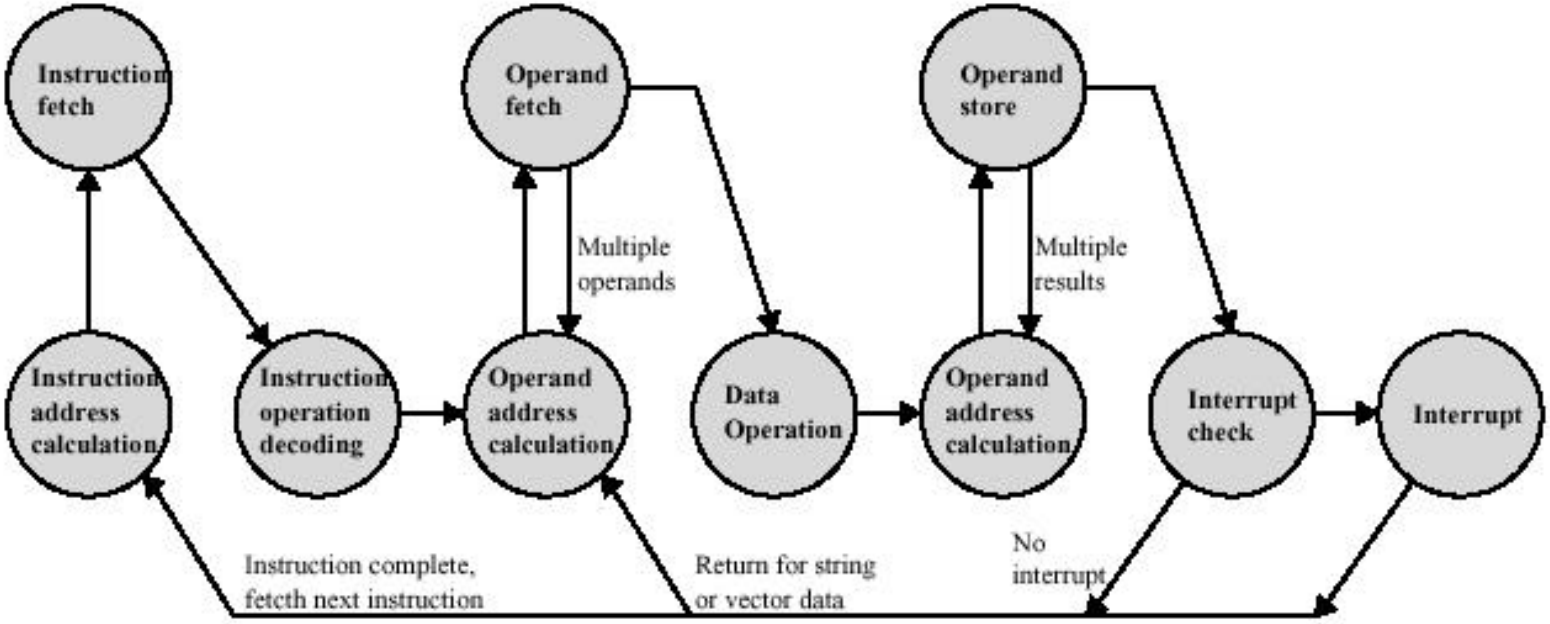


Figure 3.12 Instruction Cycle State Diagram, With Interrupts

[Stal99]