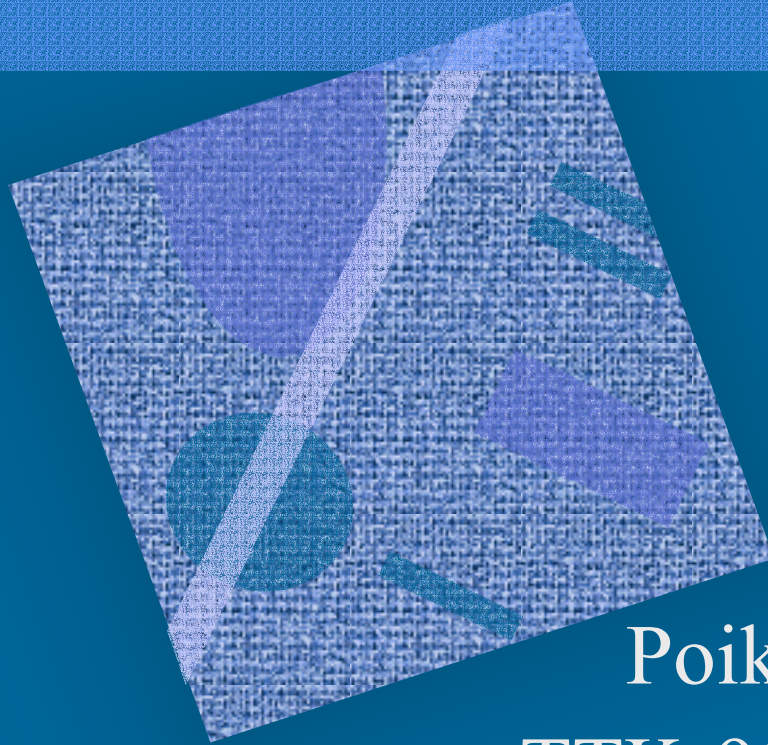


Luento 5

Suoritin ja väylä



Suorittimen rakenne

Väylän rakenne

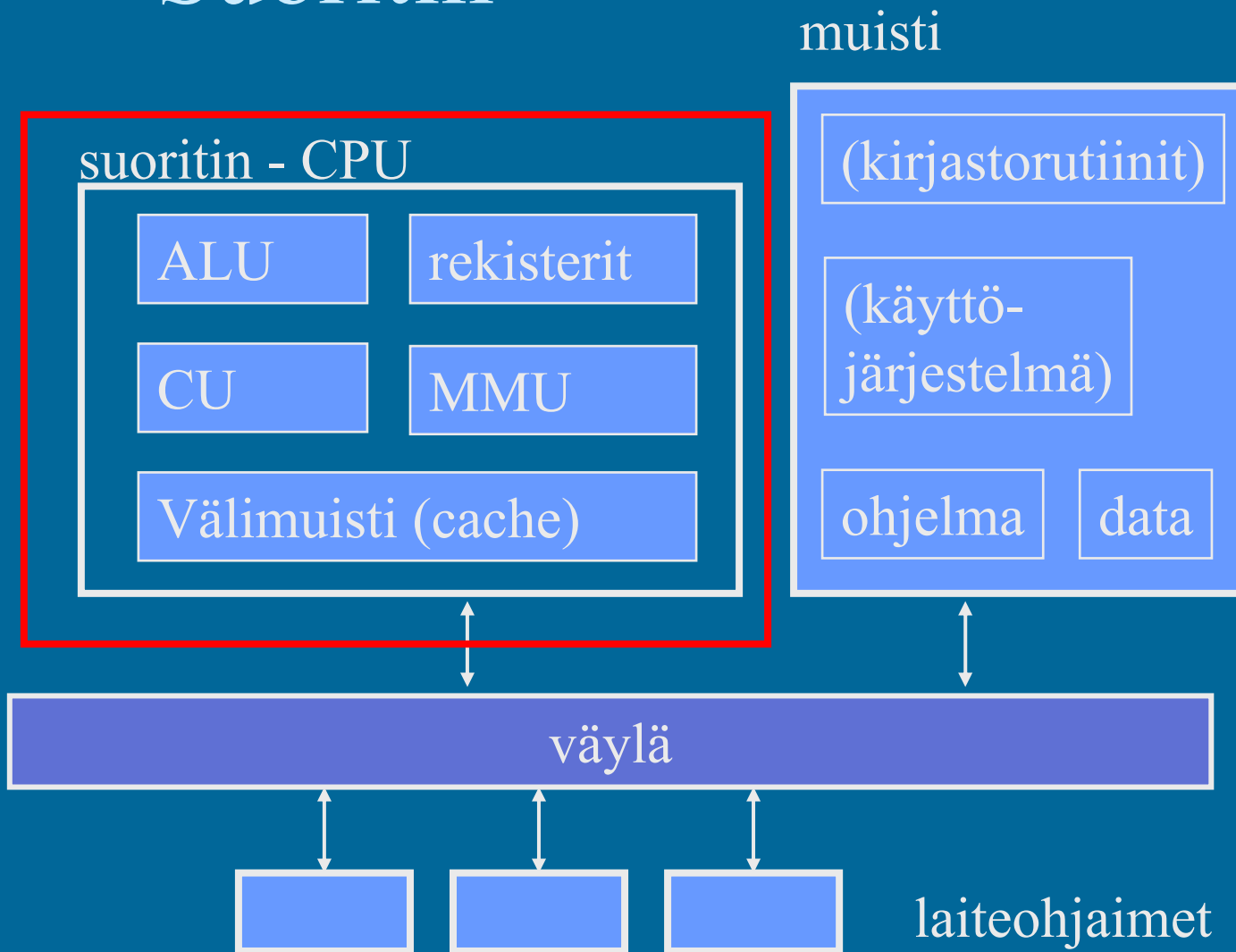
Käskyjen suoritussykli

Suorittimen tilat

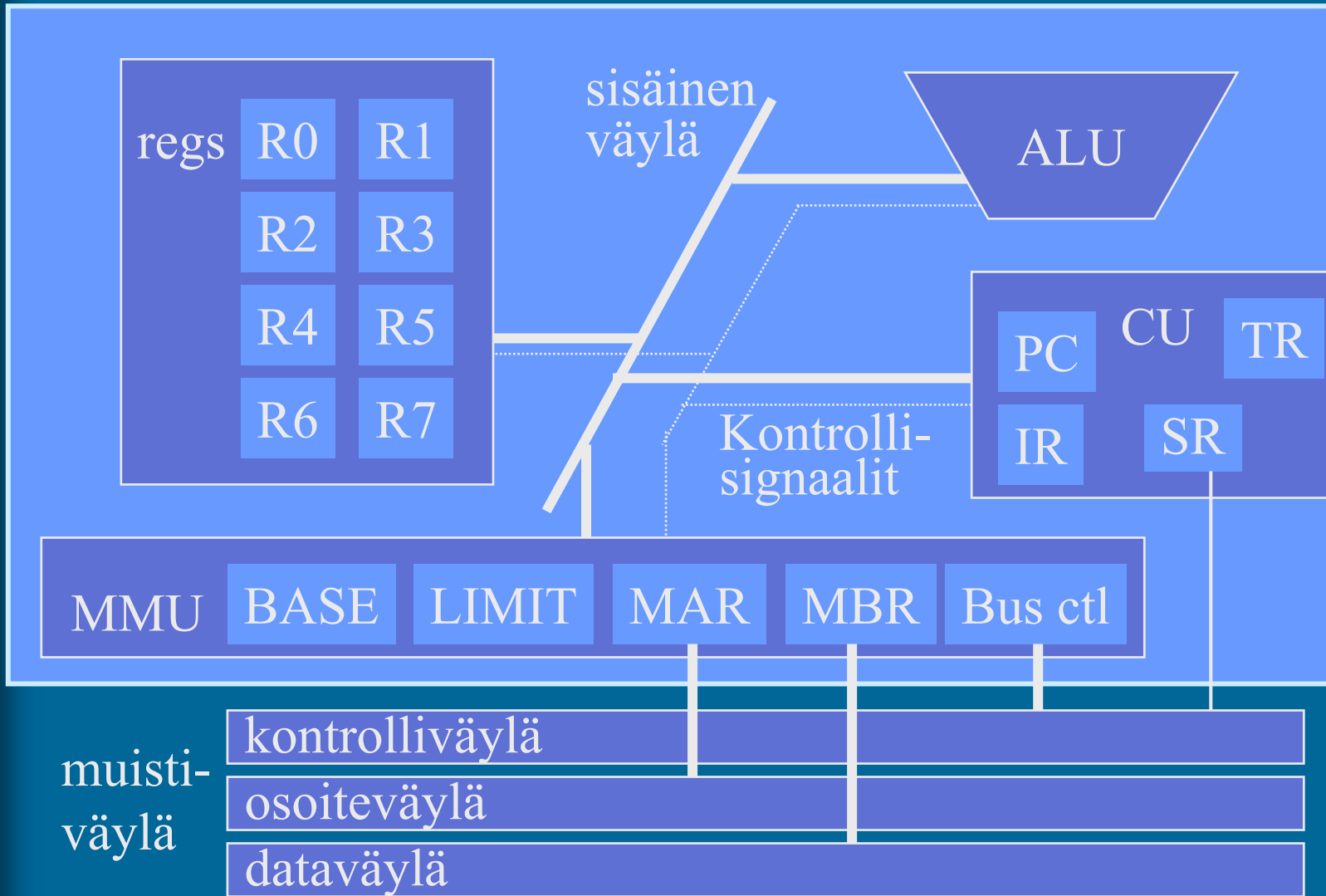
Poikkeukset ja keskeytykset

TTK-91:n ja KOKSI:n rakenne

Suoritin



TTK-91 suorittimen rakenne (1)



ks. Edsac, lastut 286, 486, PIII, PIII Xeon

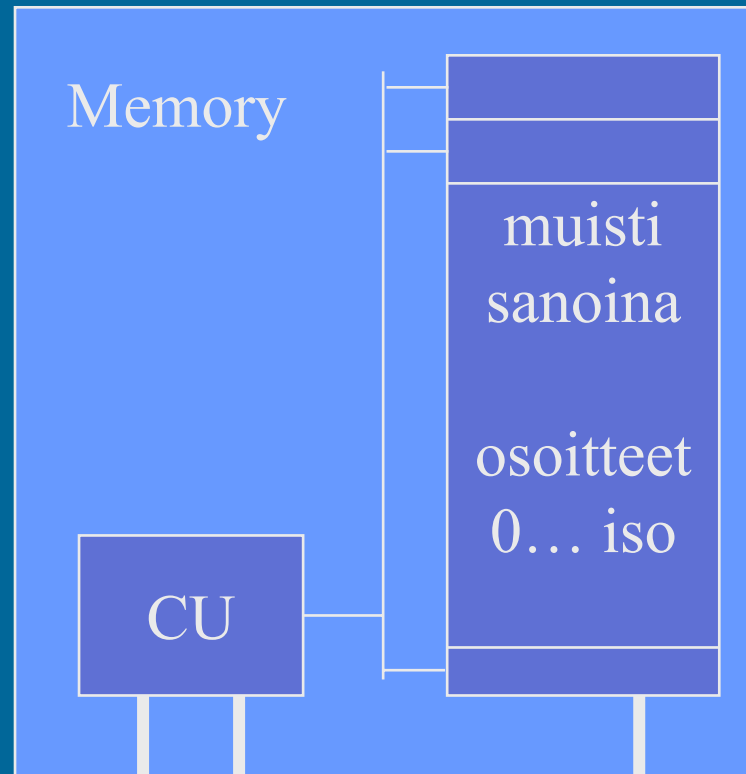
TTK-91 muistin rakenne

muisti-
väylä


kontrolliväylä

osoiteväylä

dataväylä



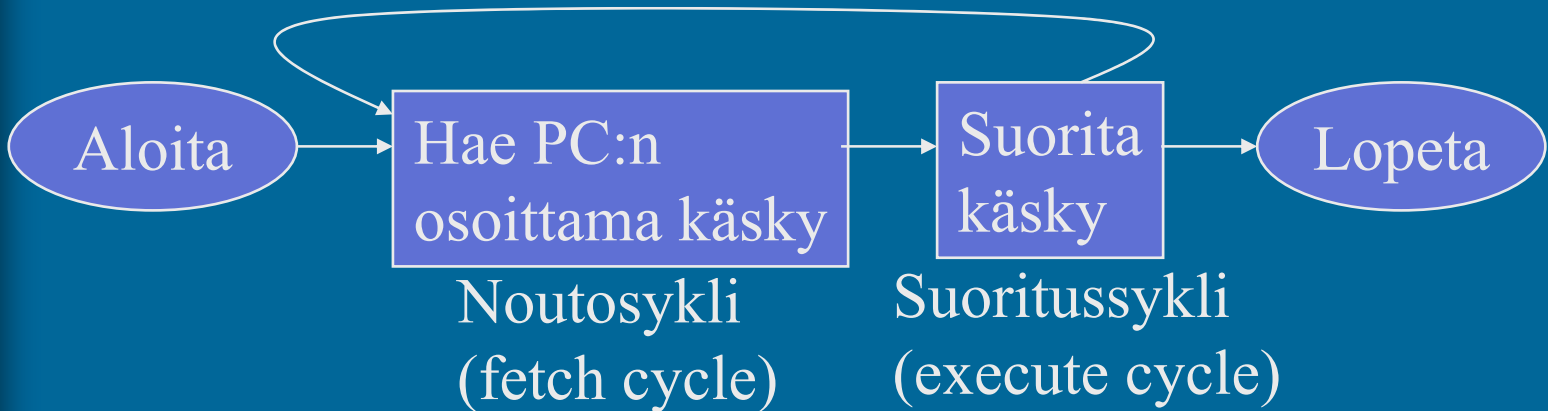
Käskyjen nouto- ja suoritus sykli (5)

- 
- 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 suoritusssykli



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

TTK-91 konekäskyn rakenne

- Käskyn esitys bittitasolla on aina:



Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri ($R0 \equiv 0$)

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 osoitus)

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

(11 eli 3 kpl, epäkelpo arvo → poikkeustilanne)

muistiosoite tai
(pienehkö) vakio

(addressing
mode)

Nouto- ja suoritus sykli tarkemmin (5)

- Noutovaihe
 - muistista MBR:n kautta IR:ään
 - Lisää 1 PC:hen
- Käskyn purku ja muistiosoitteen (EA) lasku
 - OPER, Rj, M, Ri, ADDR
 - $TR \leftarrow (Ri) + ADDR$ (pelkkä ADDR, jos $Ri=R0$)
- Operandin nouto
 - muistista MBR:n kautta TR:ään (0-2 krt ?)
- ALU operaatio
 - tulos rekisteriin R0-R7 tai TR:ään (STORE, PUSH)
- 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 lukusignaali kontrolliväylälle asentoon ”lue”
- Odota, kunnes muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä konekäsky MBR:stä IR:ään

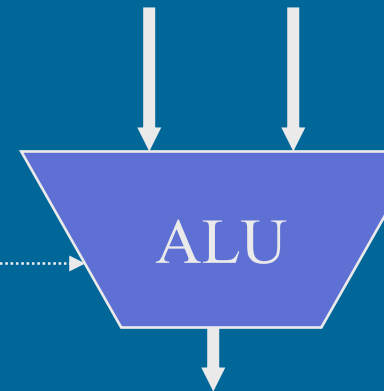
Operandin luku vaihe (4)

ks. TTK-91 suorittimen rakennekuva

- Vie muistiosoite MAR:iin
- Aseta muistin lukusignaali kontrolliväylälle asentoon ”lue”
- Odota kunnes 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 ks. TTK-91 suorittimen rakennekuva
 - käsky haettu ja purettu osiin IR:ssä
 - 1. operandi rekisterissä (R0, ..., R7)
 - 2. operandi TR:ssä
- Käskyn suoritus ALU:ssä
 - vie operandit sisäistä väylää pitkin yksi kerrallaan 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



Tässä tapahtuu tietokoneen tekemä työ,
kaikki muu on hallintoa

Tuloksen muistiin kirjoitus -vaihe (5)

ks. TTK-91 suorittimen rakennekuva

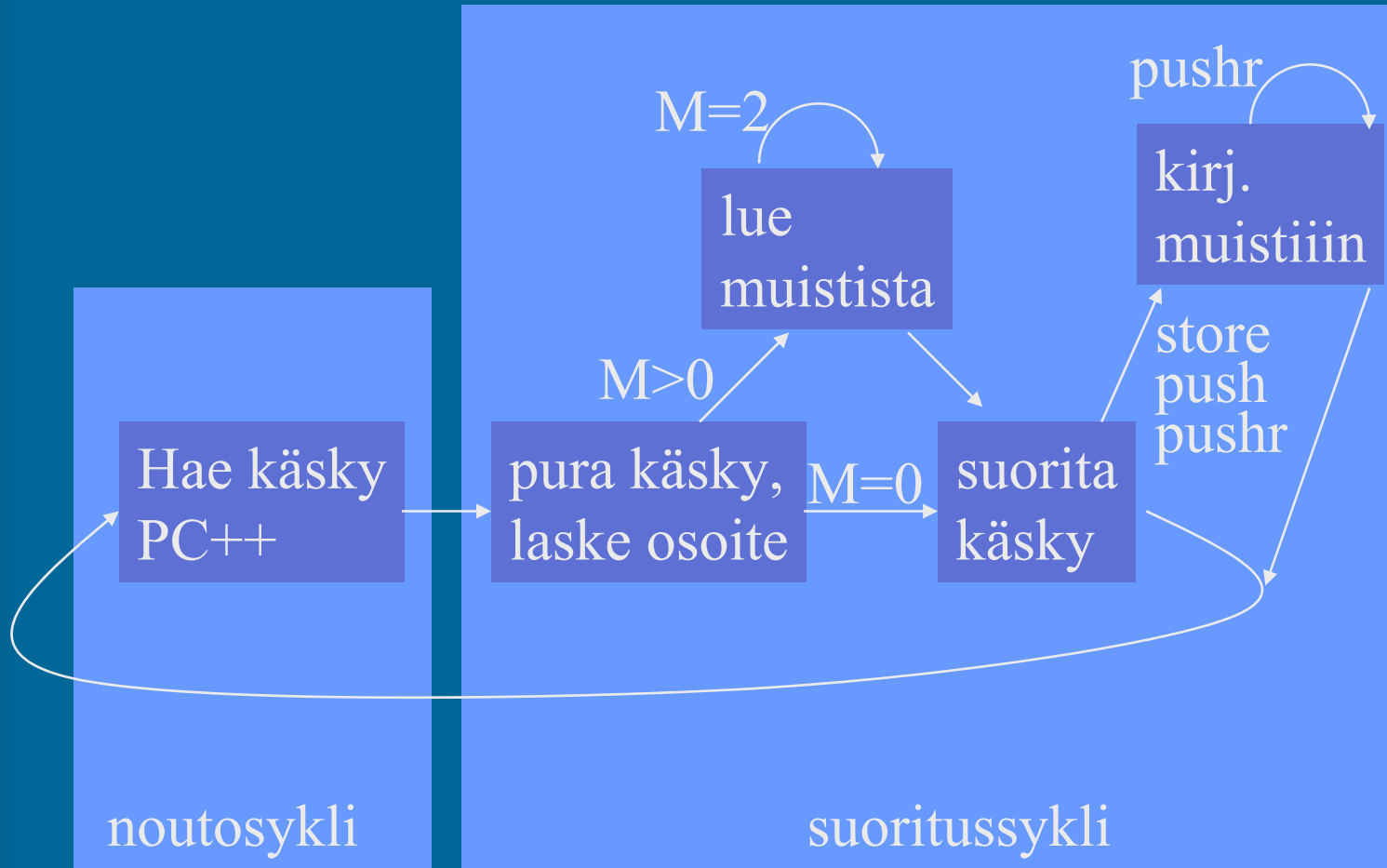
- Vie muistiosoite MAR:iin
- Vie kirjoitettava sana MBR:ään
- Aseta kirjoitussignaalit kontrolliväylälle asentoon ”kirjoita muistiin”
- Odota kunnes sana siirretään muistiin väylää pitkin ja väylän kontrollisignaalit kertovat muistiinkirjoittamisen tapahtuneen

Lisää
tietoa?



Tietokoneen
rakenne
-kurssi

TTK-91 Nouto- ja suoritussykli vähän tarkemmin (1)



pushr, popr erikoistapauksia: aika monimutkaisia

MMU:n toiminta (2)

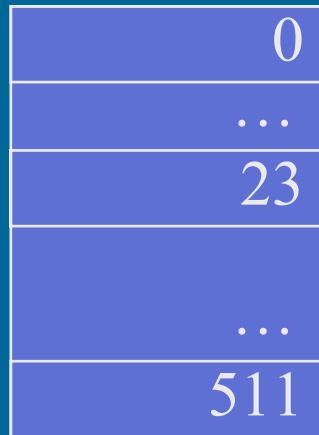
ks. TTK-91 suorittimen rakennekuva

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

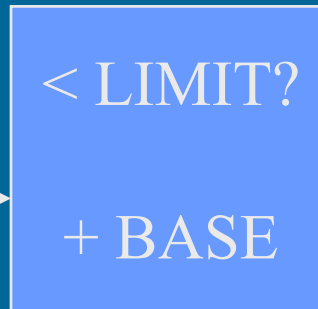
VA = virtual address, PA = physical address = BASE+VA

TTK-91 virtuaalimuisti

Virtuaalinen
osoiteavaruus



MMU



BASE: 1000

LIMIT: 512

Fyysinen
osoiteavaruus



ohjelman
käyttämät
osoitteet

muistipiirien
käyttämät
osoitteet

Virtuaalimuistin osoitteenmuunnos menetelmiä (4)

- 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

Keskeytystilanteet (3)

- Mikä tahansa tilanne, jonka käsittely vaatii poikkeuksen käskyjen normaaliin suorituserjestykseen
- Rakkaalla lapsella on monta nimeä:
 - poikkeus, keskeytys, virhetilanne, trappi, ...
 - exception, interrupt, fault, trap, failure, ...
 - SCV, KJ-kutsu, ...
- 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
- Jokaisen käskyn suorituksen jälkeen tarkistetaan keskeytysten olemassaolo SR:stä ja haaraudutaan keskeytyskäsittelijään tarvittaessa
 - joskus keskeytykset on estetty (ttk-91:ssä SR:n bitti D)
 - paluu käsittelijästä ”return-from-interrupt-handler” käskyllä (esim. IRET, tms)
- ”Yllättävä aliohjelmakutsu”

Keskeytystyyppejä (3)

- Käskyn aiheuttamat virhetilanteet
- Käskyn aiheuttamat muut poikkeustilanteet
 - kyseessä ei siis ole virhetilanne, vaan haluttu käyttäytyminen
 - tilanne vaatii erikoistoimenpiteen, jonka toteutus on tehty keskeytyskäsitteilyn kaltaiseksi
- Ulkoapäin (muualta kuin CPU:lta) tulleisiin signaaleihin reagoiminen

Käskyn aiheuttamat virhetilanteet (5)

- 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 (3)

- SVC käsky
- I/O konekäsky
- Trace keskeytys
- Käyttäjän määrittelemä keskeytys
 - esim. Javan throw/catch tai try/catch operaatioiden toteutus

Ulkoapäin (muualta kuin suorittimelta) tulleet keskeytykset ⁽³⁾

- Kellolaitekeskeytys (esim. joka 10 ms)
- Laitekeskeytys (esim. levy I/O valmis)
- Laitteistovirhe (esim. virhe väylän tiedonsiirrossa)

Keskeytyskäsitteijä

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

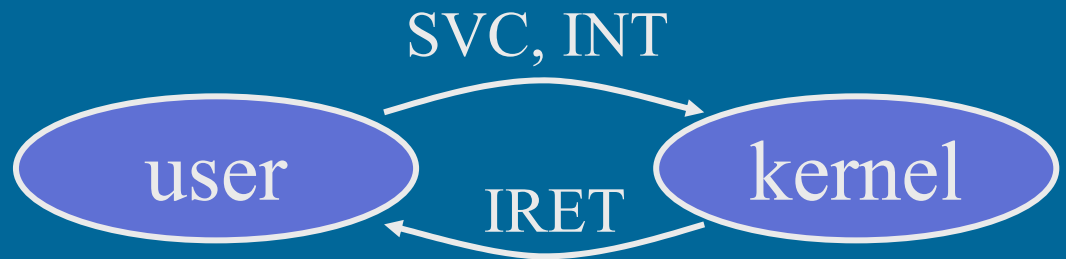
Suorittimen tilat ⁽⁶⁾

user

kernel

- 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 (esim. `clear_cache`, `iret`)
 - voi viitata kaikkialle muistiin, myös käyttöjärjestelmän ytimeen (kernel)
 - voi käyttää (myös) suoria muistiosoitteita (PA)

Suorittimen tilan muuttaminen (6)




- 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



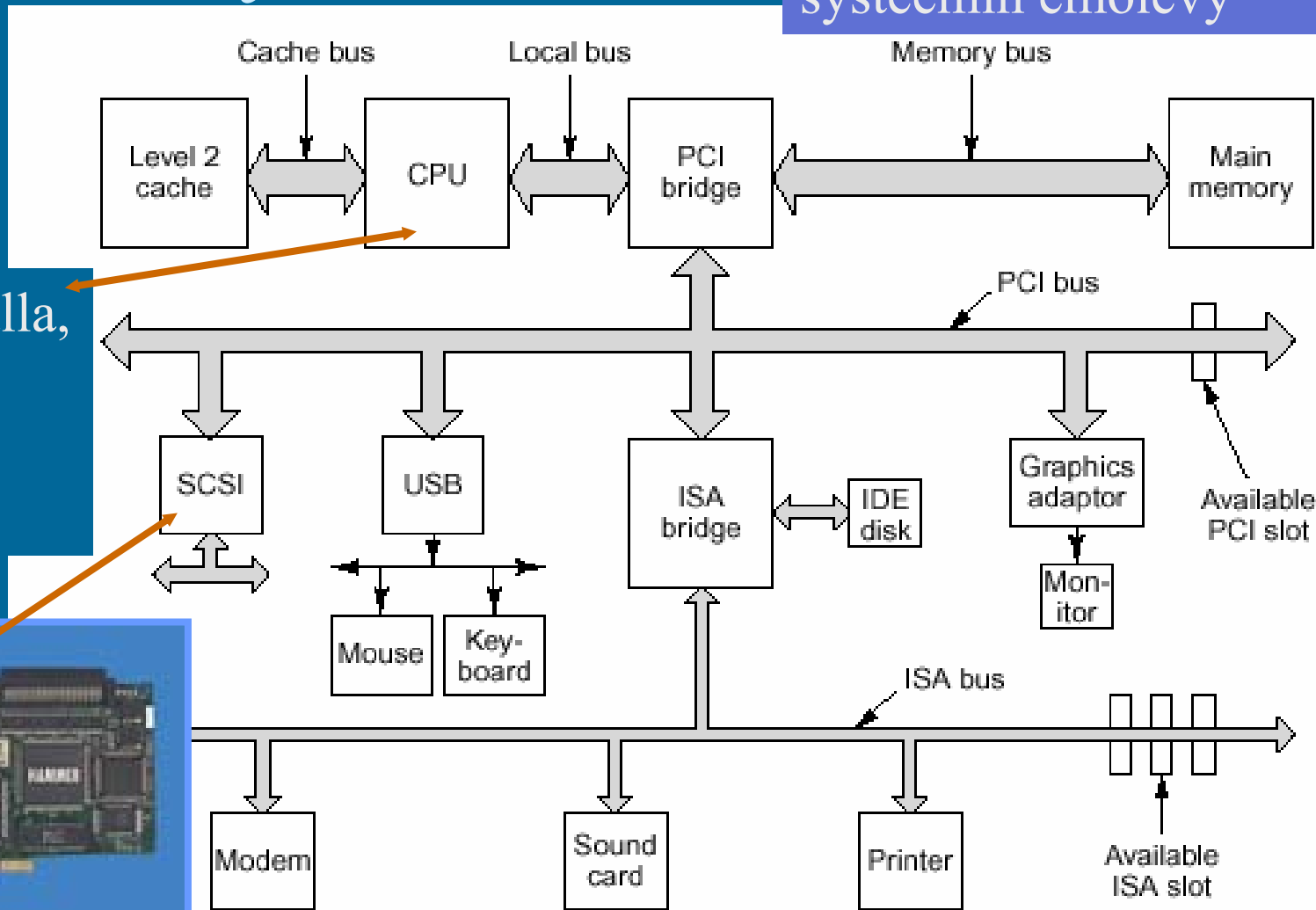
Väylät (5)

- Tiedon siirtoa varten laitteistossa
- Yksi kirjoittaja 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

Tyypillinen Pentium II
systemin emolevy

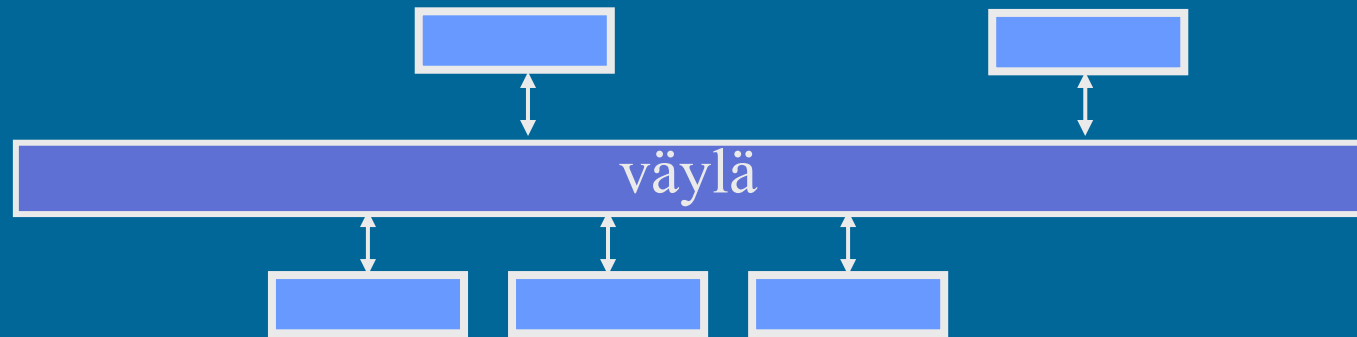
omalla lastulla,
tason 1
välimuistin
kanssa



PCI to SCSI bridge

Fig. 3-50 [Tane99]

Väylät (5)



- Kullakin laitteella oma osoite
- Yksi lähettää, kaikki kuulevat, vain ”oikea” laite vastaanottaa
- Paljon erilaisia
- Lähellä suoritinta 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 myös TTK-91 koneen käyttöjärjestelmän osat osana tavallista ohjelmaa
 - assembler kääntäjä, lataaja, debugger, kesk. käsittelijät
- Graafinen käyttöliittymä

ks. suoritussyklin toteutus Koksissa
(seur. kalvo + 6 kopiosivua)

TTK-91 käskyn suoritusssykli (5)

hae käsky simuloidusta muistista

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

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

$\text{ADDR} = IR \% 32768$

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

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

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

