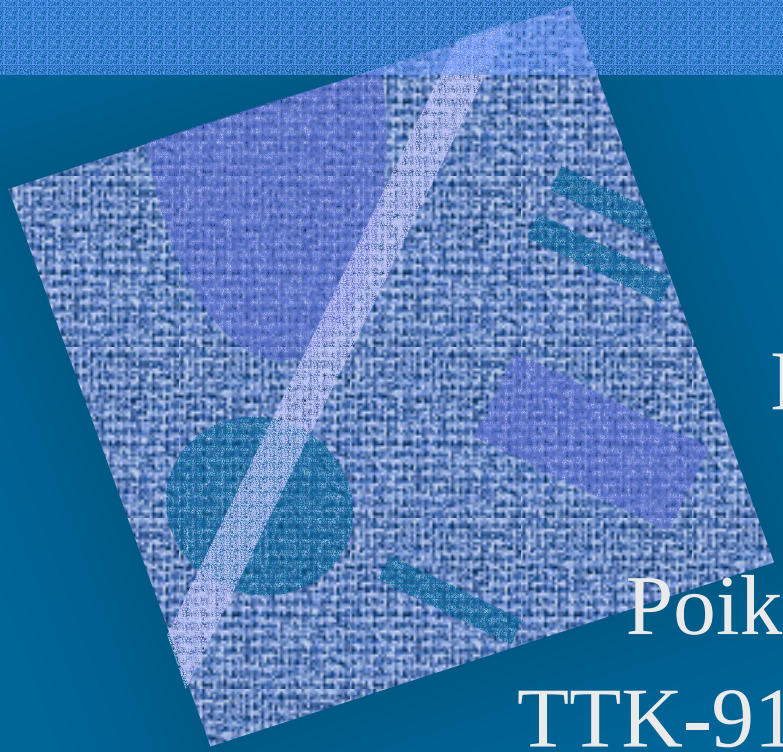
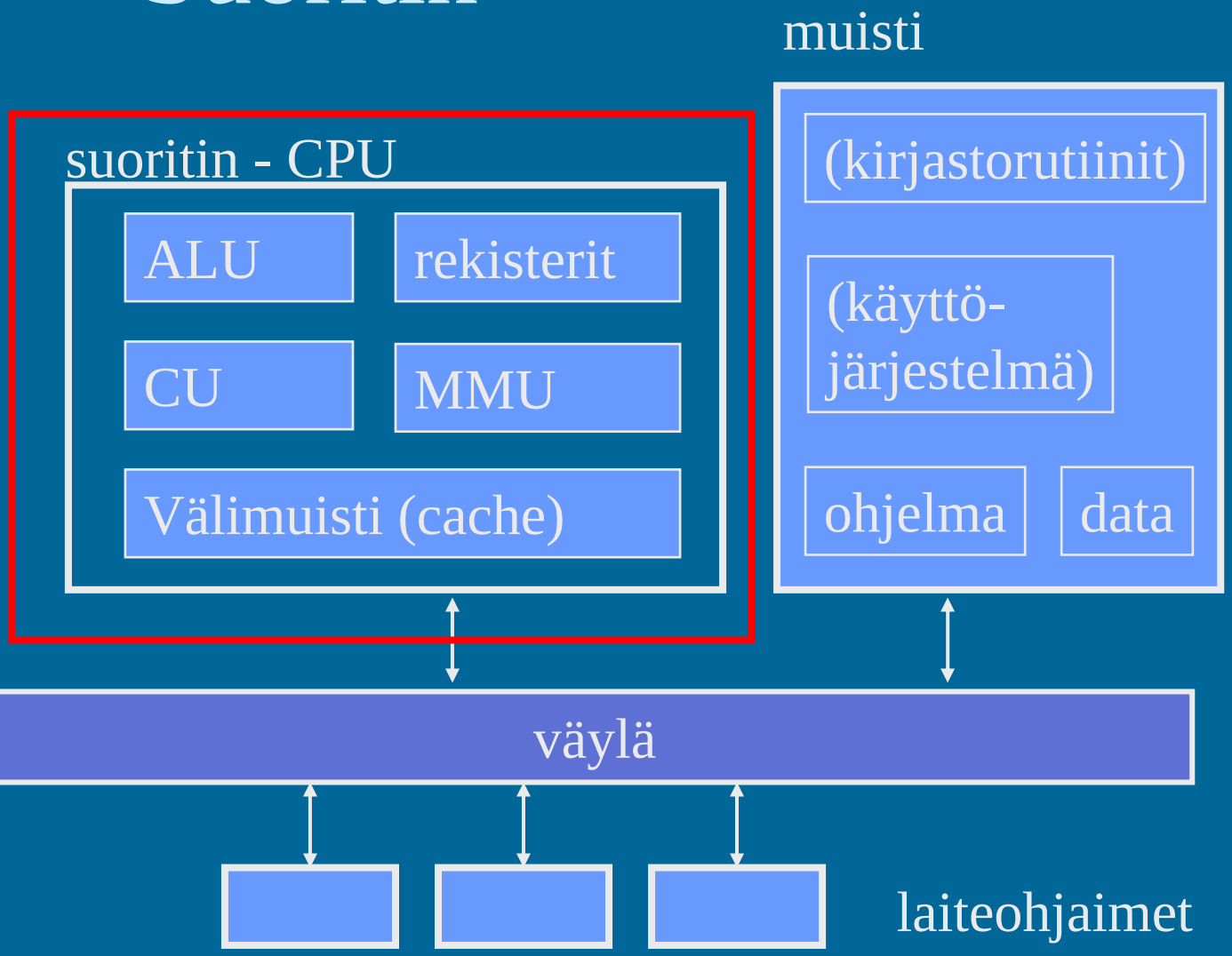
Luento 5
Suoritin ja väylä

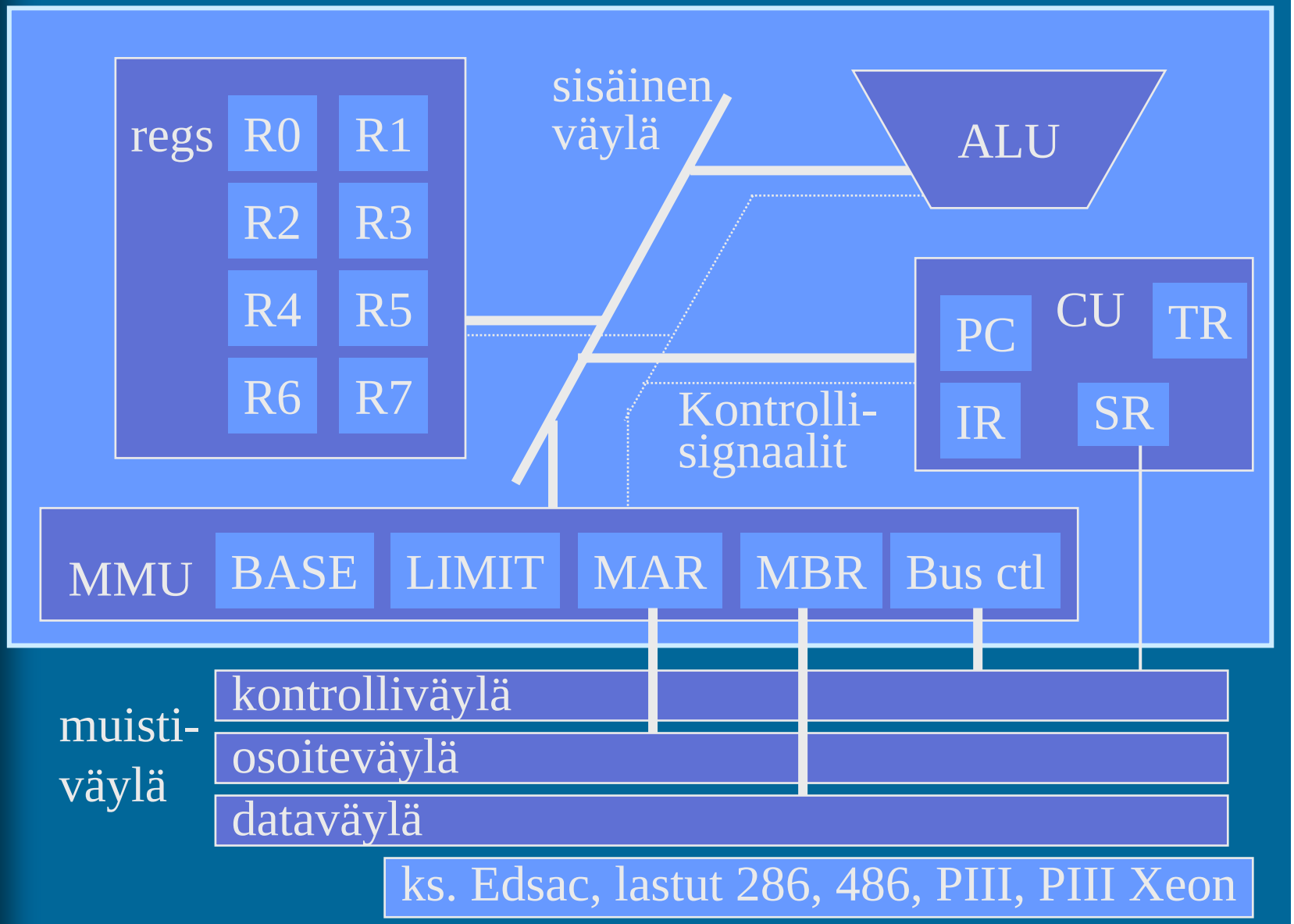


Suorittimen rakenne
Väylän rakenne
Käskyjen suoritussykli
Suorittimen tilat
Poikkeukset ja keskeytykset
TTK-91:n ja sen simulaattorien rakenne

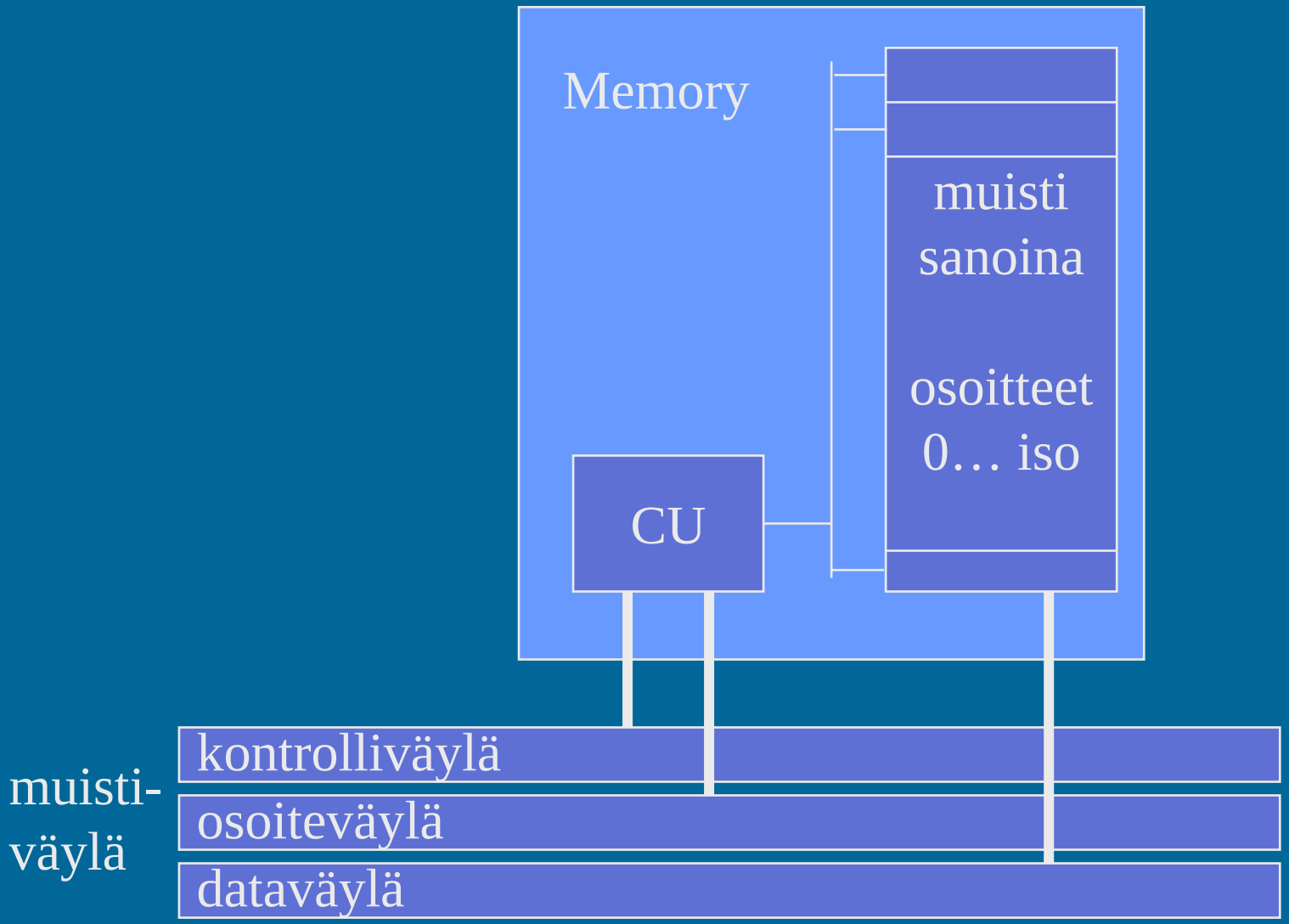
Suoritin



TTK-91 suorittimen rakenne (1)



TTK-91 muistin rakenne



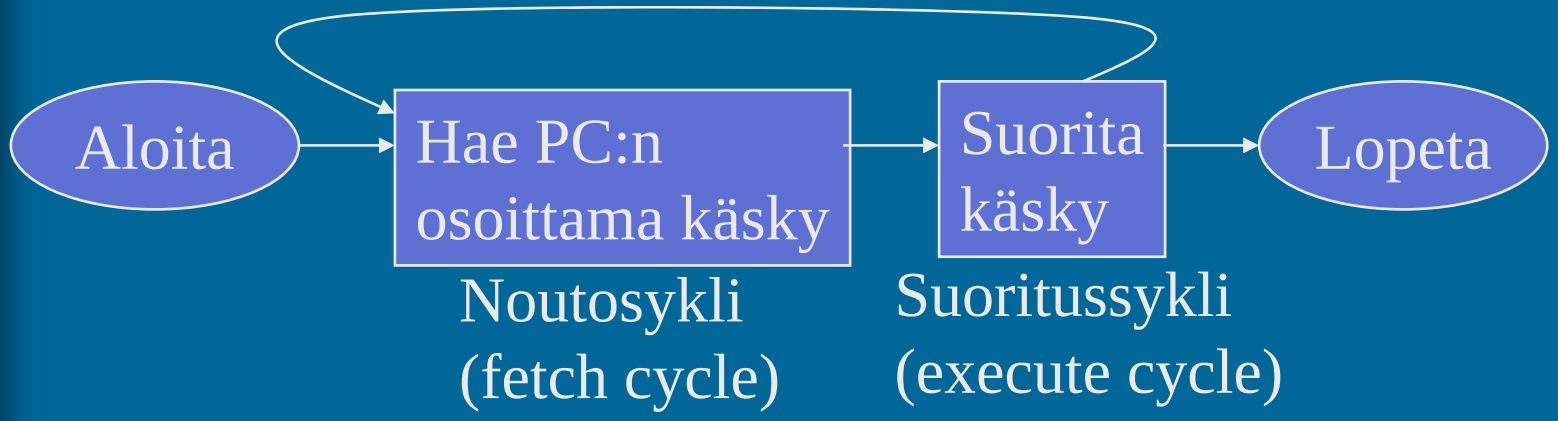
Käskyjen nouto- ja suoritussykli (5)

- 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

(addressing mode)

Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri (R0 ≡ 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 osoit.)
10 eli 2 kpl, epäsuora osoitus (STORE: epäkelpo arvo)
(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 (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

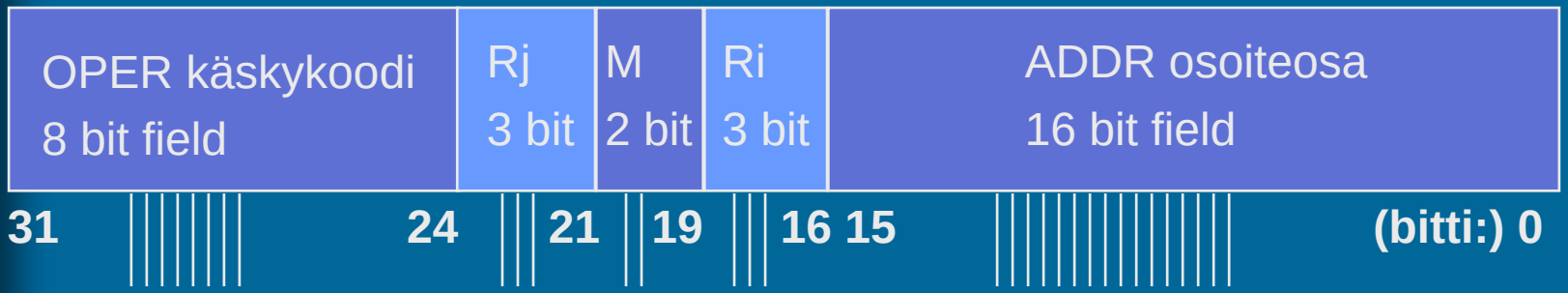
Ei kaikilla käskyillä

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

Käskyn purku ja tehollisen muisti-osoitteen (EA) laskemisvaihe



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

Operandin luku vaihe (4)

ks. TTK-91 suorittimen rakennekuva

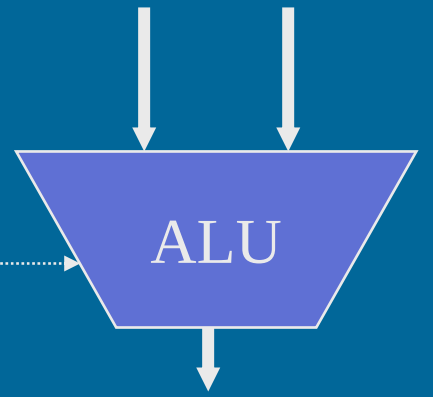
- Vie muistiosoite MAR:iin
- Aseta muistin lukusignaali kontrolliväylälle asentoon ”lue”
- Odota kunnes muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä sana MBR:stä TR:ään
  - (tai suoraan johonkin laiterekisteriin R0-R7)

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

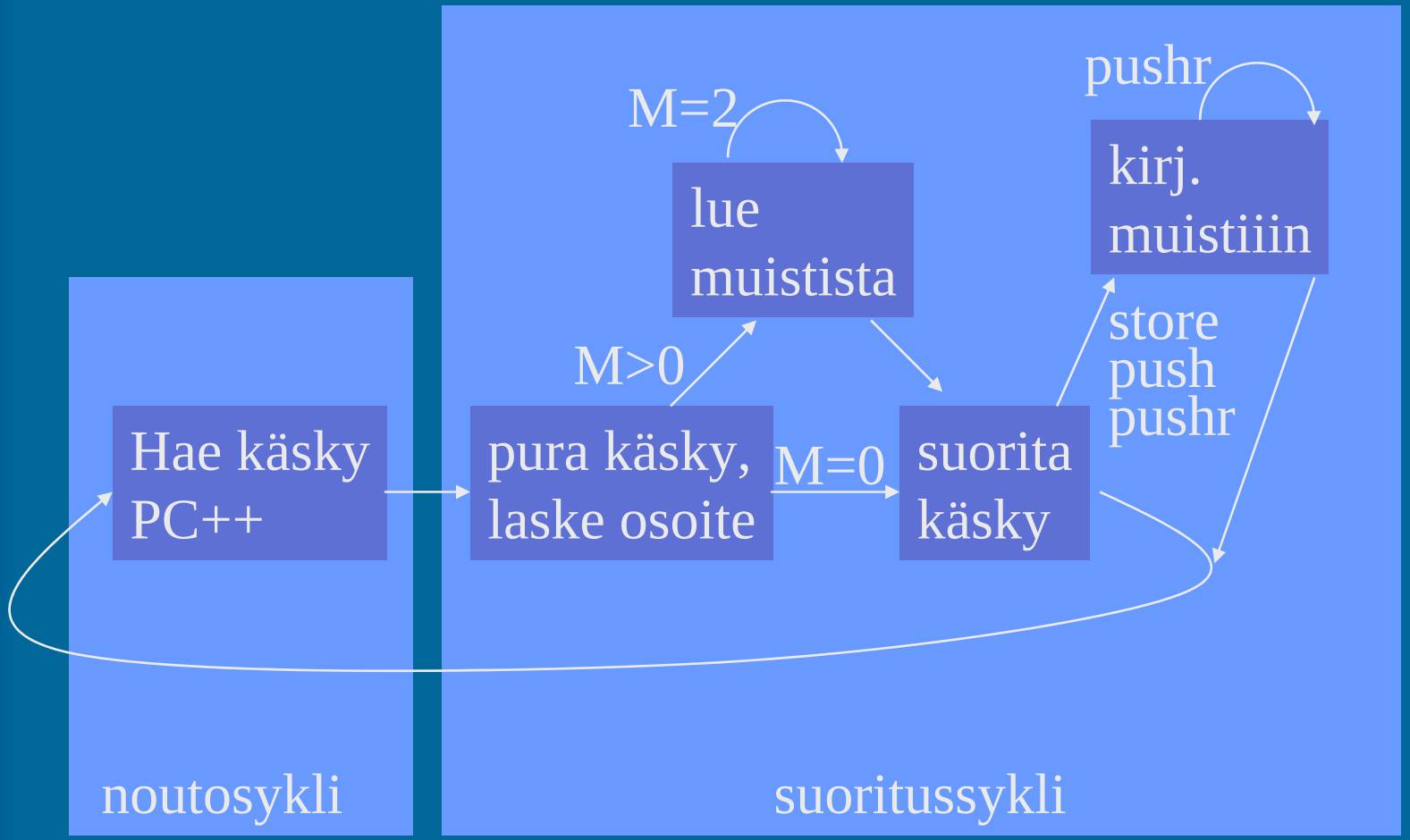
Tuloksen muistiin kirjoitus -vaihe (5)

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

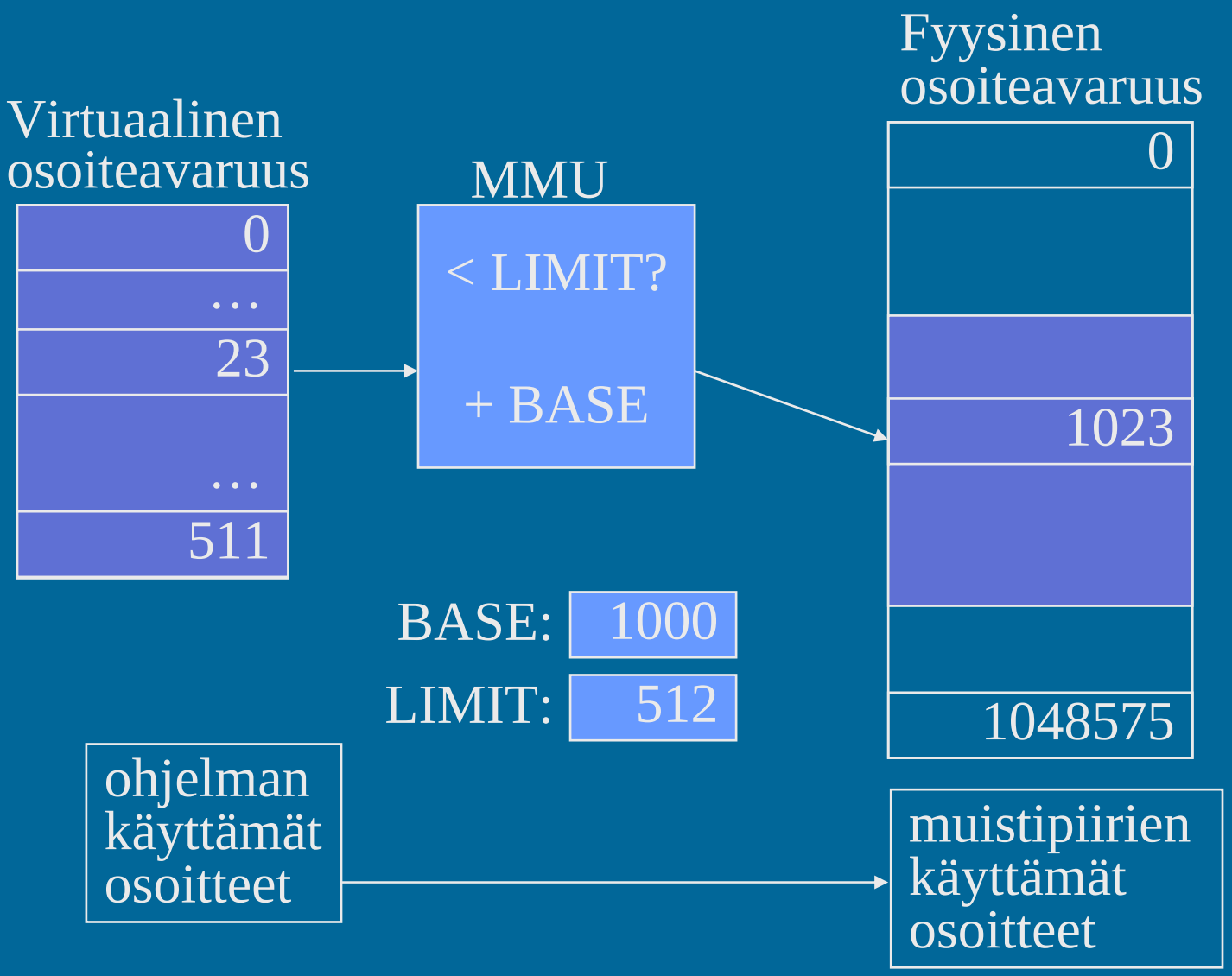
MMU:n toiminta (2)

ks. TTK-91 suorittimen rakennekuva

- Ohjelman käyttämät muistiosoitteet (VA) ovat näennäisiä, välillä 0 … 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 $\in$ [0, 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



Virtuaalimuistin osoitteenmuunnosmenetelmiä (3)

- 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

Sivuttava virtuaalimuisti

- Kaikki tieto ei sijaitse muistissa, loput on levyllä
  - erityisessä virtuaalimuistille varatussa partitiossa
- Muisti jaettu tasakokoisiin sivukehyksiin
  - mikä tahansa (levyllä oleva) virtuaalimuistin sivu voidaan sijoittaa mihin tahansa keskusmuistissa olevaan sivukehykseen
  - kirjanpito virtuaalimuistin sivutauluissa (isot taulukot muistissa)
- Osoitteenmuutosta nopeuttaa välimuistin kaltainen TLB (Translation Lookaside Buffer)
  - esim. 99.9% osoitteenmuutoksista TLB:stä
  - TLB osa muistinhallintayksikköä (MMU)

Virtuaalimuistin piirteitä

- Samalla ratkaistaan helposti muita ongelmia
  - kirjanpito eri ohjelmien muistin käytöstä
  - ohjelman muistialueiden suojaus muilta ohjelmilta
  - ohjelma tarvitsee enemmän muistitilaa kuin mitä on
  - yleinen muistinhallintaongelma
- Yleinen muistinhallintaongelma
  - miten paljon muistitilaa kullekin ohjelmalle?
  - missä päin muistia kunkin ohjelman muistitila on?
    - yhtenäinen alue vai paloittainen?
    - kiinteä sijainti koko ohjelman suorituksen ajan?
  - miten muistiin viitataan?

Virtuaalimuistin ongelma (1)

- Joka muistiviitteen yhteydessä täytyy tehdä aika monimutkainen kuvaus virtuaaliosoitteesta fyysiseen keskusmuistiosoitteeseen
  - osittainen ratkaisu: TLB
  - sivutaulujen koko silti ongelma! osa niistäkin levyllä!
- Aina joskus tulee viite sivuun, joka ei sijaitse keskusmuistissa
  - kustannus: peli seis, kunnes tiedot haettu levyltä!
  - lääke: niin iso keskusmuisti, että näitä ”sivunpuutoskeskeytyksiä” ei tule usein

Lisää tietoa?  tietokoneen rakenne ja käyttöjärjestelmäkurssit

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, …
  - SCV, KJ-kutsu, ...
- Jatkossa yleisnimi keskeytys tarkoittaa kaikkia näitä eri tapauksia tai tyyppejä

Keskeytysten käsittely (4)

- Jokainen mahdollinen keskeytystyyppi on ennalta tunnettu, eli mitään todella yllättävää ei tapahdu
- 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 keskeytyskäsittelyn kaltaiseksi
- Ulkoapäin (muualta kuin suorittimelta) tulleisiin signaaleihin reagoiminen
  - kontrolliväylältä tuleva signaali

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
- Käyttäjän määrittelemä keskeytys
  - esim. Javan throw/catch tai try/catch operaatioiden toteutus

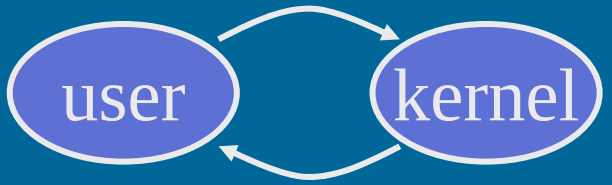
Ulkoapäin (muualta kuin suorittimelta) tulleet keskeytykset

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

Keskeytyskäsittelijä

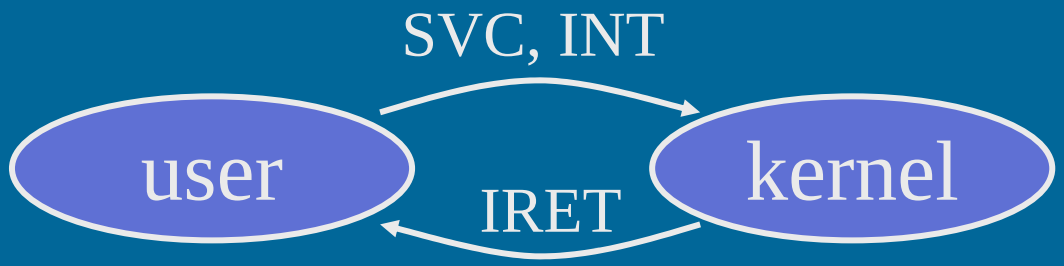
- Osa käyttöjärjestelmää
- Ennen keskeytyskäsittelijään hyppäämistä asetetaan suoritin ja MMU etuoikeutettuun käyttöjärjestelmätilaan (supervisor state)
  - SR:n bitti P on päällä => etuoikeutettu tila eli käyttöjärjestelmä tila (P = Priviledged)
  - käyttöjärjestelmätilassa saa viitata mihin tahansa kohtaan muistia (MMU: BASE=0, LIMIT="hyvin iso")
  - käyttöjärjestelmätilassa saa käyttää kaikkia konekäskyjä (esim. IRET tai ClearCache)
- Käsittelijästä paluun yhteydessä MMU:n tila ja suorittimen tila asetetaan ennalleen

Suorittimen tilat (2)



- 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 (supervisor state, 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 (2)

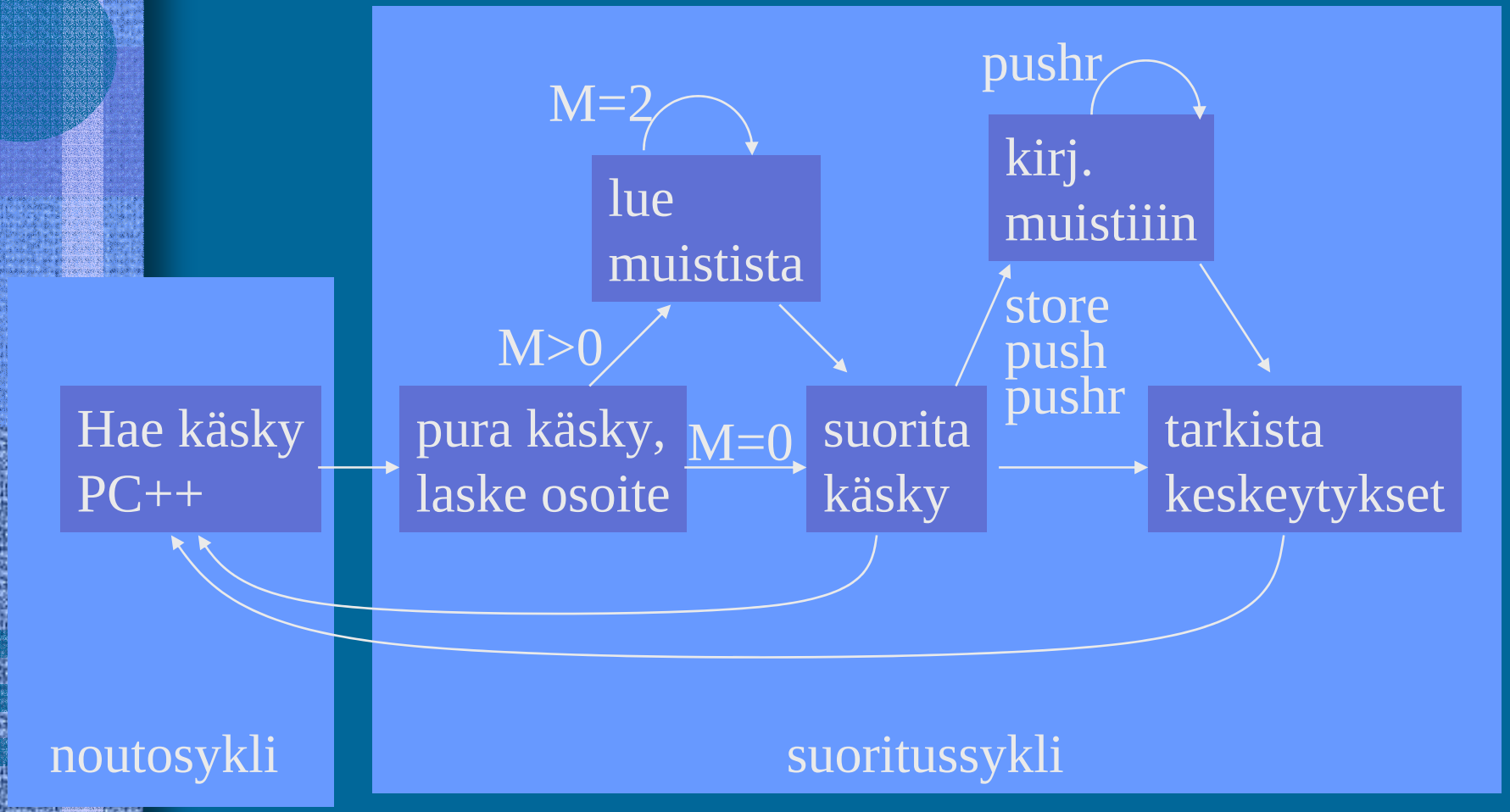


- Käyttäjätila → etuoikeutettu tila
  - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
  - keskeytyskäsittelijä tarkistaa onko (oliko) oikeutta tilan vaihtoon

(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



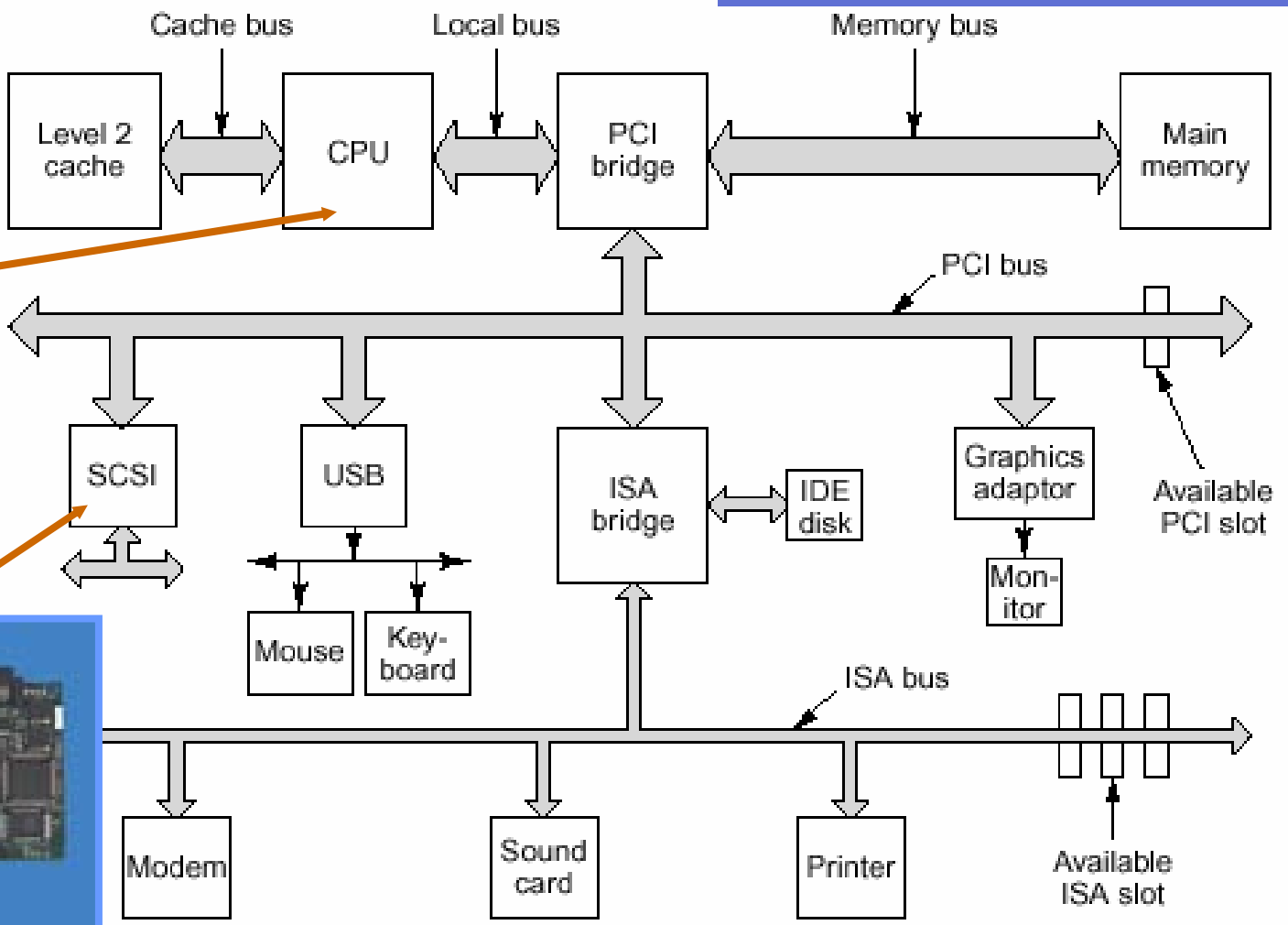
Väylät

- Tiedon siirtoa varten laitteistossa
- Yksi kirjoittaja kerrallaan (vain!)
- Toteutettu johdinkimppuina

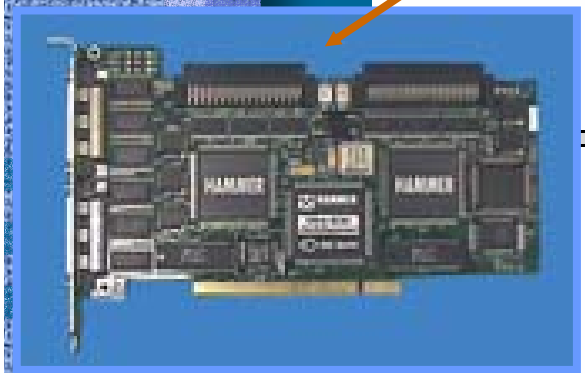
- Eri tasoilla
  - suorittimen sisällä ”sisäinen väylä” (internal bus)
  - muistiväylä suorittimen ja muistin välillä (memory bus)
  - I/O-väylä muistiväylän ja I/O-laitteiden välillä (I/O bus)
- Useita eri tapoja yhdistellä edellä olevia

Väylähierarkia

Tyypillinen Pentium II systeemin emolevy



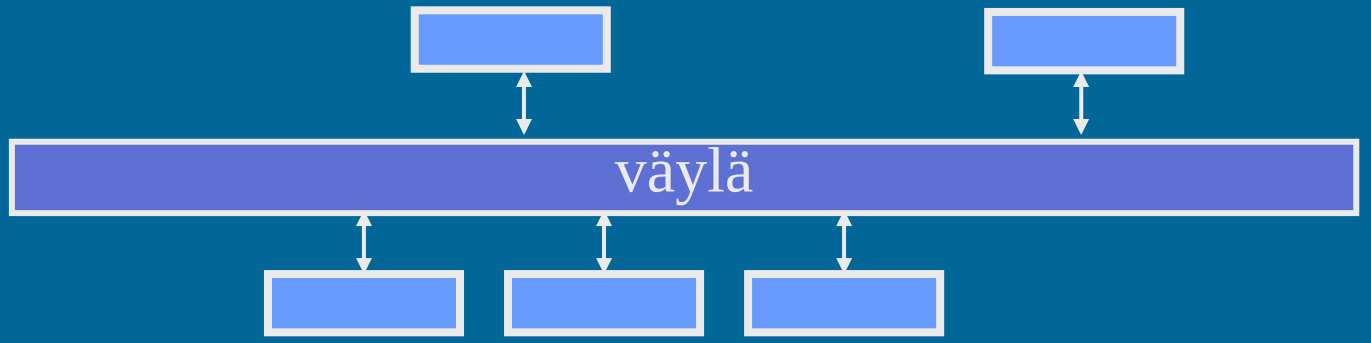
omalla lastulla, tason 1 välimuistin kanssa



PCI to SCSI bridge

Fig. 3-50 [Tane99]

Väylät (1)



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

Lisää tietoa? → Tietokoneen rakenne -kurssi

TTK-91 koneen simulaattori (6)

- Tavallinen Javalla tai Pascalilla kirjoitettu ohjelma
- TTK-91 koneen osat tietorakenteina
  - rekisterit, MMU, CU, muisti
- Simuloi käskyjen suoritussykliä käsky kerrallaan
  - Titokoneessa myös suorituksen animointi
- 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 suoritussykli

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 mod 32768

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 Rj, M ADDR(Ri) ⇒ regs[Rj] += TR;

lopeta suoritus jos SVC tai keskeytys

SR.O = …

-- Luennon 5 loppu --

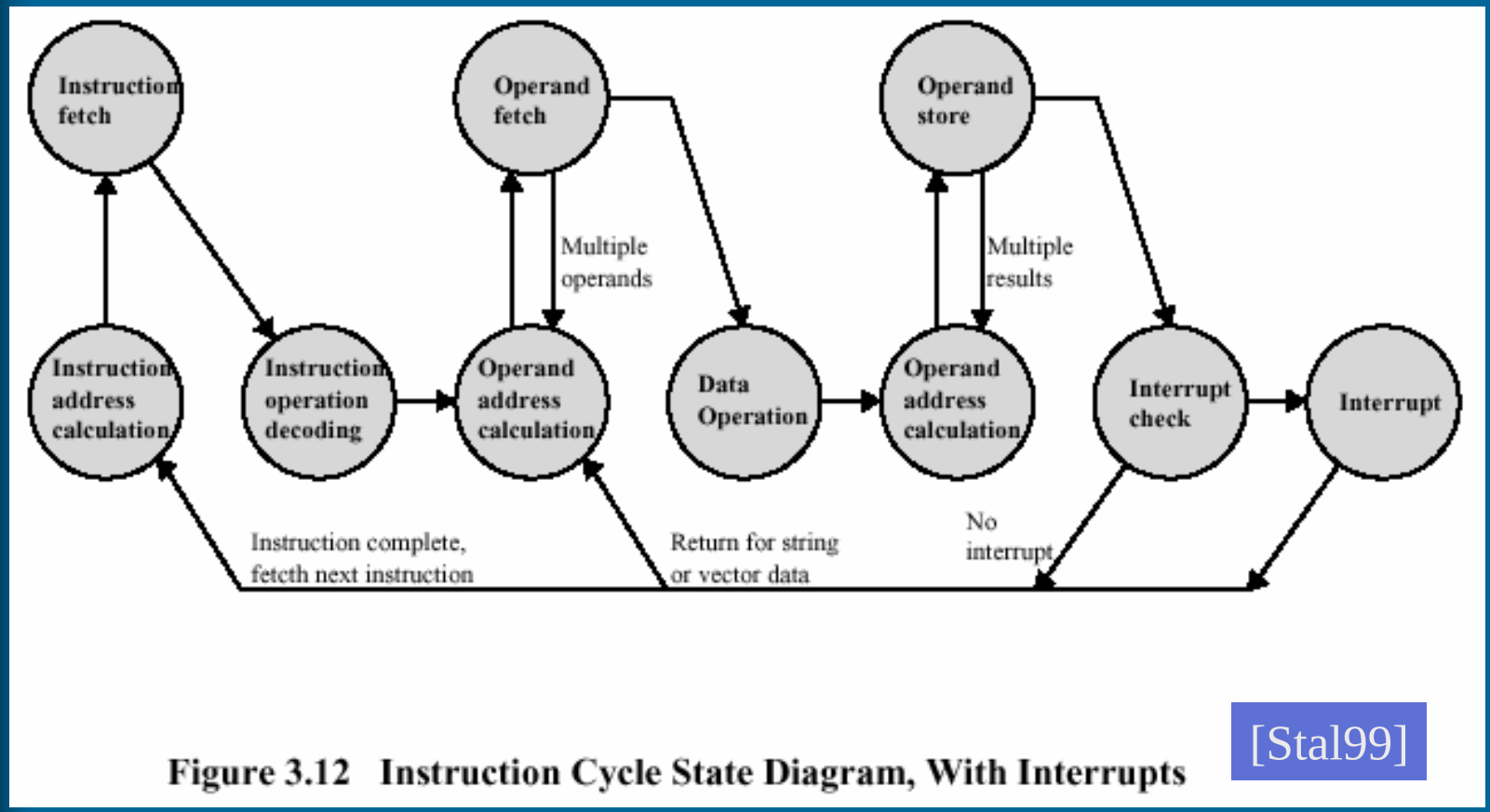


Figure 3.12 Instruction Cycle State Diagram, With Interrupts

[Stal99]