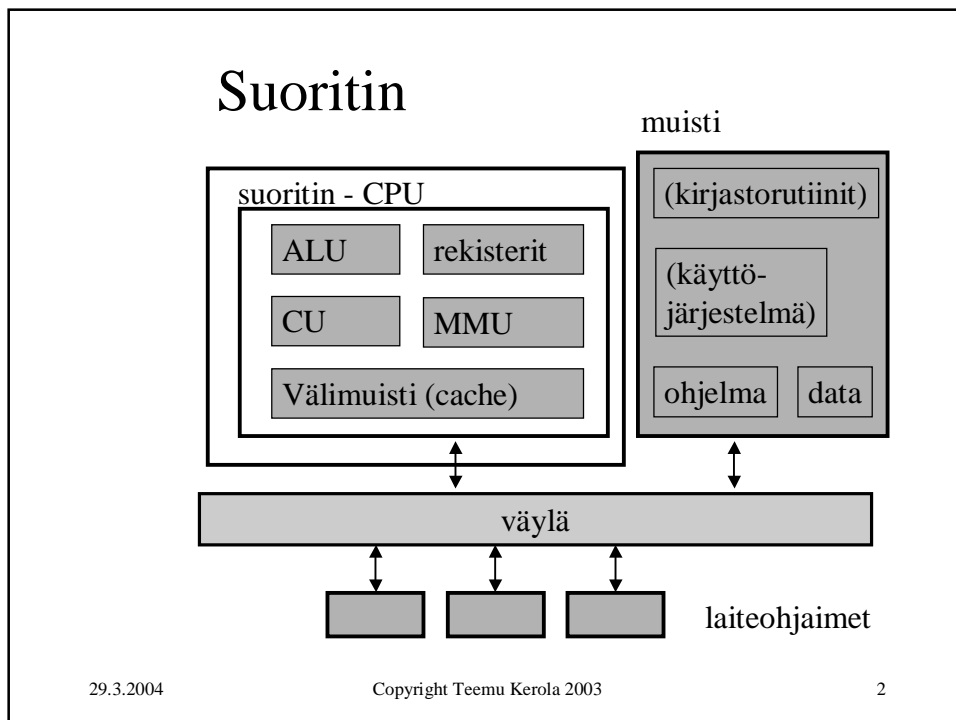


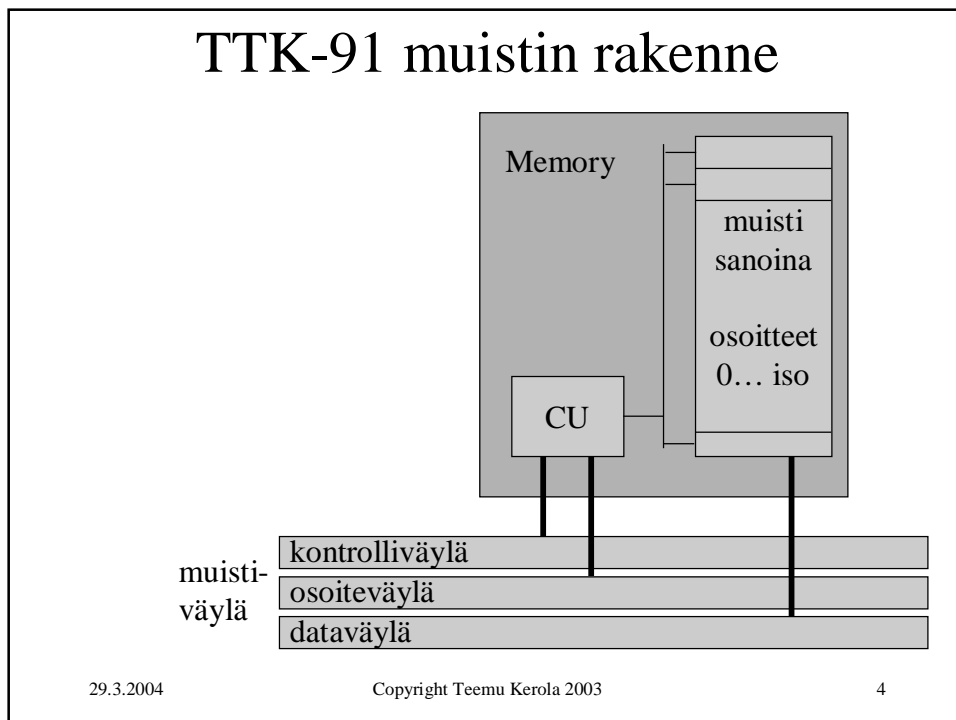
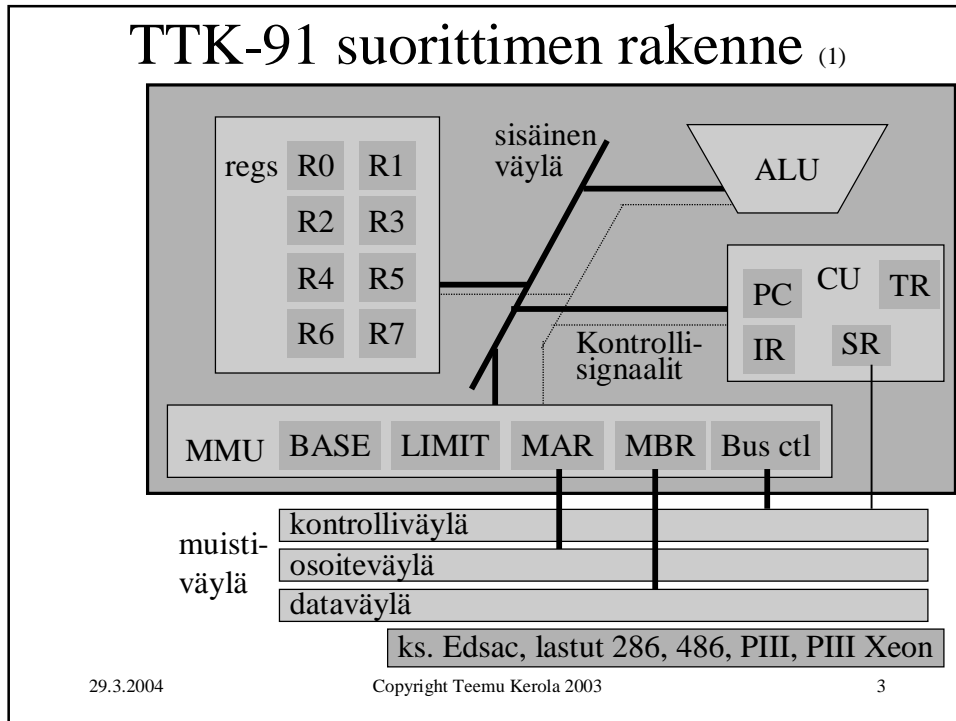
Luento 5

Suoritin ja väylä

Suorittimen rakenne
Väylän rakenne
Käskyjen suoritusyksi
Suorittimen tilat
Poikkeukset ja keskeytykset
TTK-91:n ja KOKSI:n rakenne

29.3.2004 Copyright Teemu Kerola 2003 1





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!

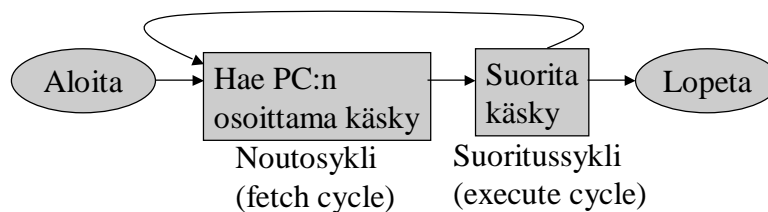
Suoritin ei tiedä mitään ohjelmista!

29.3.2004

Copyright Teemu Kerola 2003

5

Nouto- ja suoritussykli



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

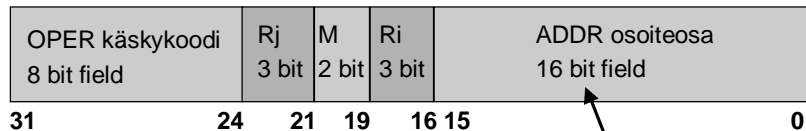
29.3.2004

Copyright Teemu Kerola 2003

6

TTK-91 konekäskyn rakenne

- Käskyn esitys bittitasolla on aina:



Rj = käskyn ensimmäinen operandi

Ri = indeksirekisteri ($R_0 \equiv 0$)

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 osoit.)

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

(11 eli 3 kpl, epäkelpo arvo → poikkeustilanne)

muistiosoite tai
(pienehkö) vakio

(addressing
mode)

29.3.2004

Copyright Teemu Kerola 2003

7

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
 - $TR \leftarrow (Ri) + ADDR$ (pelkkä ADDR, jos $Ri=R_0$)
- 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

29.3.2004

Copyright Teemu Kerola 2003

8

Käskyn noutovaihe ⁽⁴⁾

ks. TTK-91 suorittimen rakennekuva

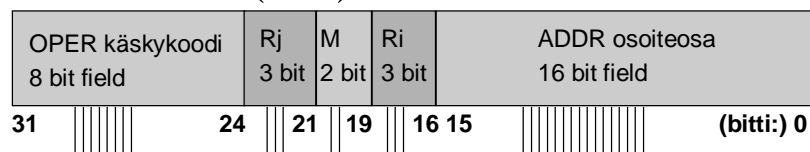
- Vie PC:n arvo MAR:iin
- Aseta muistin lukusignaali kontrolliväylälle asentoon ”lue”
- Odota, kunnes muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä konekäsky MBR:stä IR:ään

29.3.2004

Copyright Teemu Kerola 2003

9

Käskyn purku ja tehollisen muisti-osoitteen (EA) laskemisvaihe



- Purku automaattisesti langoitettuna IR:stä
- Muistiosoitteen lasku, tulos TR:ään
 - 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

29.3.2004

Copyright Teemu Kerola 2003

10

Operandin lukuvaihe ⁽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)

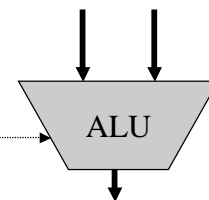
29.3.2004

Copyright Teemu Kerola 2003

11

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

29.3.2004

Copyright Teemu Kerola 2003

12

Tuloksen muistiinkirjoitusvaihe (5)

ks. TTK-91 suorittimen rakennekuva

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

Lisää
tietoa?



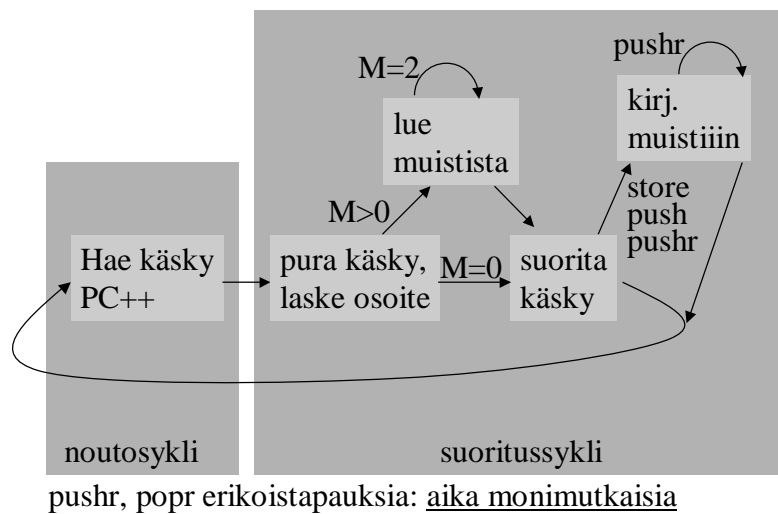
Tietokoneen
rakenne
-kurssi

29.3.2004

Copyright Teemu Kerola 2003

13

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



29.3.2004

Copyright Teemu Kerola 2003

14

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 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

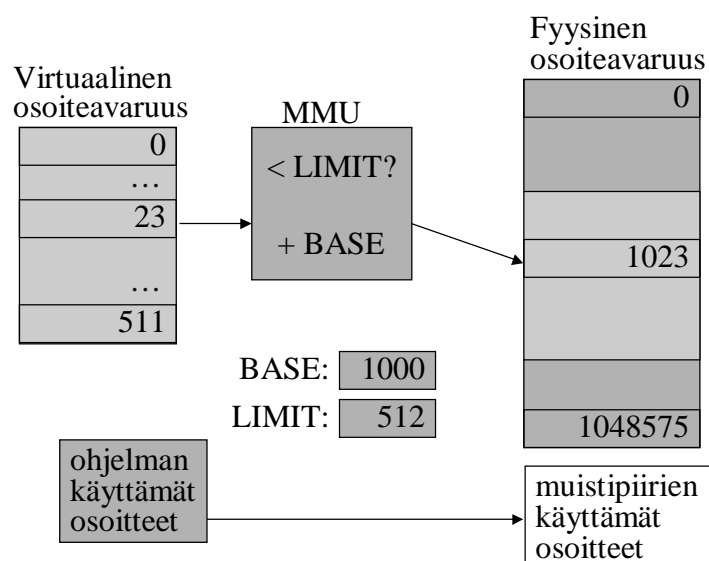
VA = virtual address, PA = physical address = BASE+VA

29.3.2004

Copyright Teemu Kerola 2003

15

TTK-91 virtuaalimuisti



29.3.2004

Copyright Teemu Kerola 2003

16

Virtuaalimuistin osoitteenmuunnosmenetelmiä ⁽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
 - koodisegmentti, datasegmentti, ...

Lisää tietoa?  käyttöjärjestelmäkurssit

Lisää tietoa?  käyttöjärjestelmäkurssit

29.3.2004

Copyright Teemu Kerola 2003

17

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

29.3.2004

Copyright Teemu Kerola 2003

18

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 aliohjelmautus”

29.3.2004

Copyright Teemu Kerola 2003

19

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

29.3.2004

Copyright Teemu Kerola 2003

20

Käskyn aiheuttamat virhetilanteet

- Virheellinen käskyn tai datan osoite
- Tuntematon käsky (opcode)
- Nollalla jako
- Kokonaisluvun tai liukuluvun yli/alivuoto
- Käytetty osoite ei ole muistissa (MMU)

29.3.2004

Copyright Teemu Kerola 2003

21

Käskyn aiheuttamat muut poikkeustilanteet

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

29.3.2004

Copyright Teemu Kerola 2003

22

Ulkoapäin (muualta kuin suorittimelta) tulleet keskeytykset

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

29.3.2004

Copyright Teemu Kerola 2003

23

Keskeytyskäsittelijä

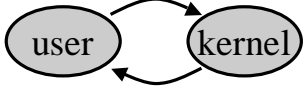
- Osa käyttöjärjestelmää
- Ennen keskeytyskäsittelijään hyppäämistä asetetaan suoritin ja MMU etuoikeutettuun käyttöjärjestelmätilaan (supervisor state)
 - SR:n bitti P on päällä => etuoikeutettu tila eli (P = Priviledged) käyttöjärjestelmätila
 - käyttöjärjestelmätilassa saa viitata mihin tahansa kohtaan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

29.3.2004

Copyright Teemu Kerola 2003

24

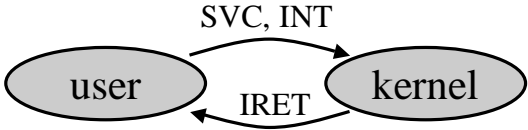
Suorittimen tilat ⁽²⁾



- 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)

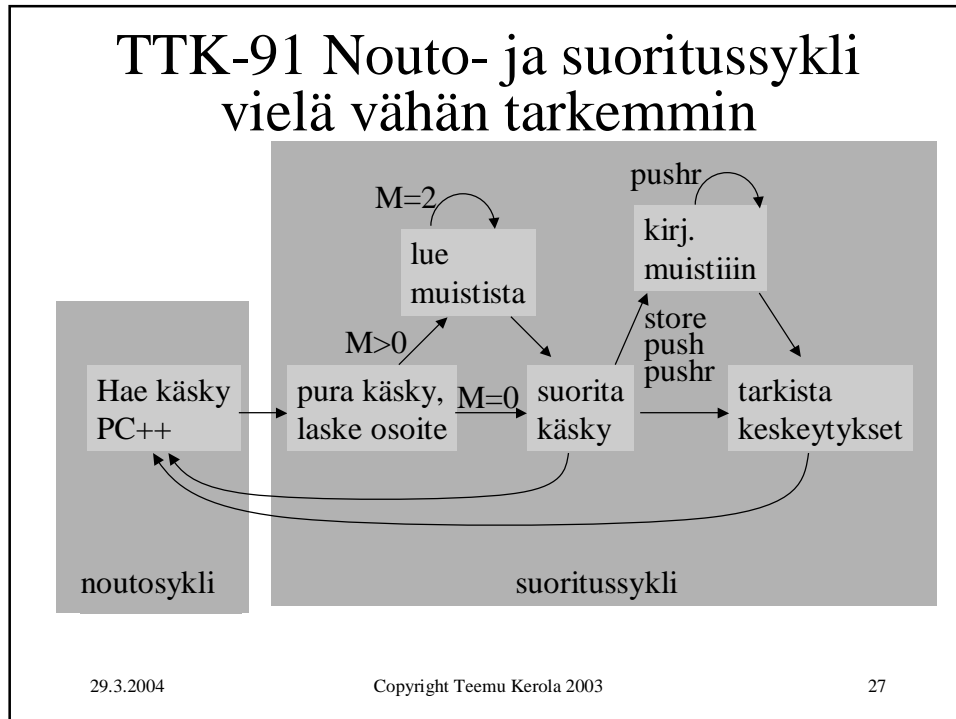
29.3.2004 Copyright Teemu Kerola 2003 25

Suorittimen tilan muuttaminen ⁽²⁾



- Käyttäjätila → etuoikeutettu tila
 - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
 - keskeytyskäsitteijä 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

29.3.2004 Copyright Teemu Kerola 2003 26

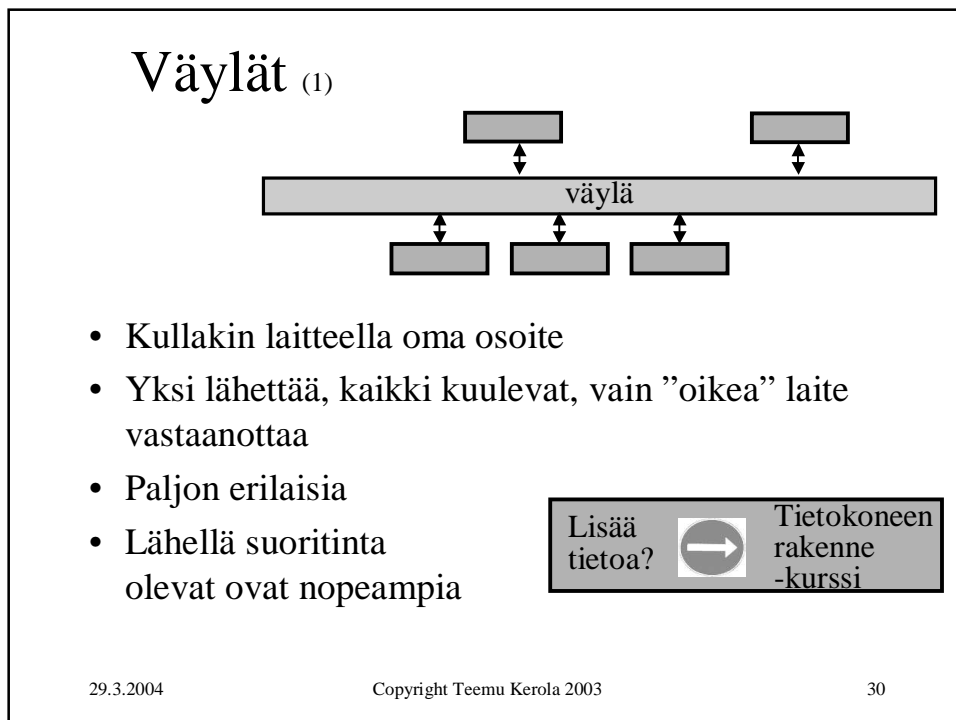
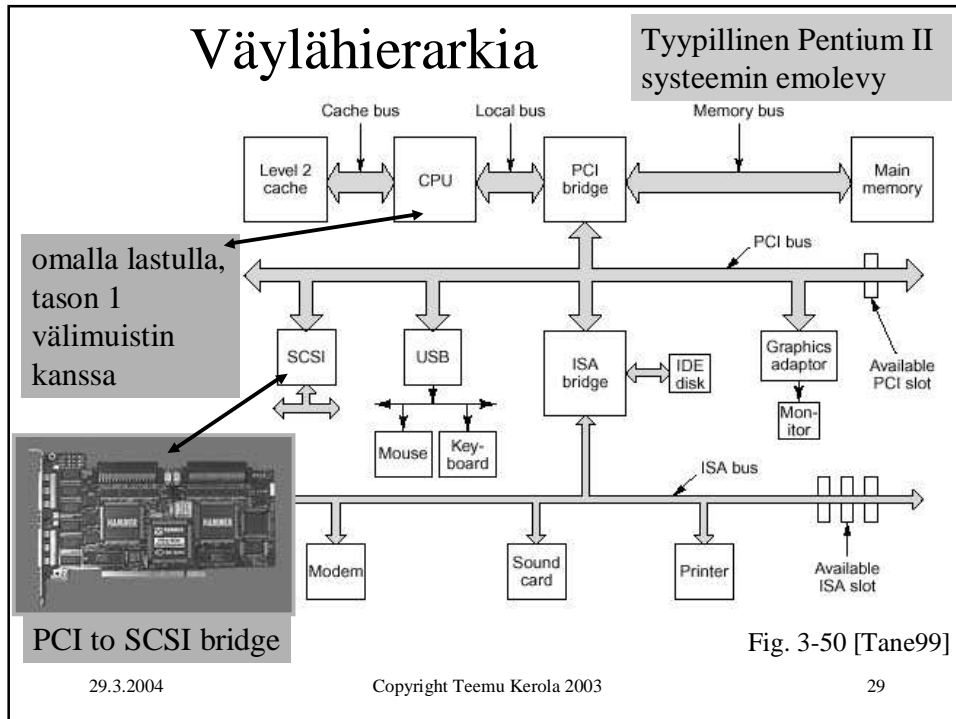


Väylät

- Tiedon siirtoa varten laitteistossa
- Yksi kirjoittaja kerrallaan (vain!)
- Toteutettu johdinkimppuina
- Eri tasoilla
 - suorittimen sisällä ”sisäinen väylä”
 - muistiväylä suorittimen ja muistin välillä
 - I/O-väylä muistiväylän ja I/O-laitteiden välillä
- Useita eri tapoja yhdistellä edellä olevia

The diagram shows three horizontal lines representing the **(internal bus)**. Below them are two more horizontal lines representing the **(memory bus)**. At the bottom are two more horizontal lines representing the **(I/O bus)**.

29.3.2004 Copyright Teemu Kerola 2003 28



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)

29.3.2004

Copyright Teemu Kerola 2003

31

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 = ...

29.3.2004

Copyright Teemu Kerola 2003

32

-- Luennon 5 loppu --

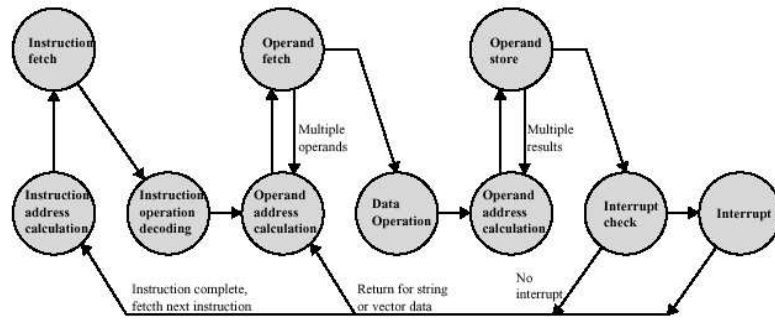


Figure 3.12 Instruction Cycle State Diagram, With Interrupts

[Stal99]