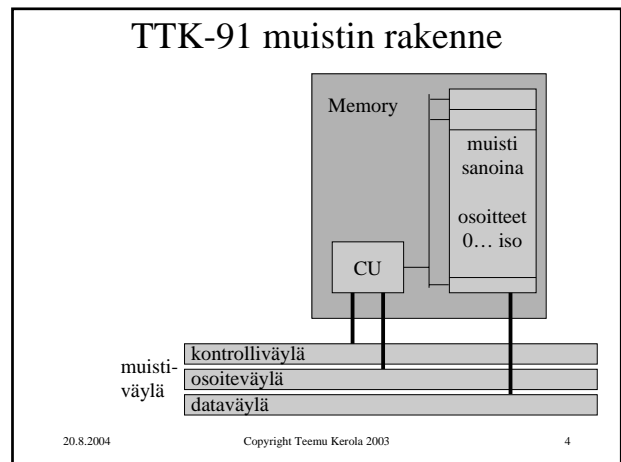
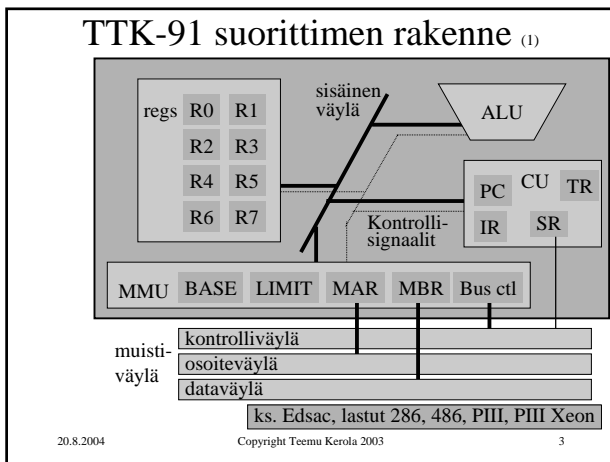
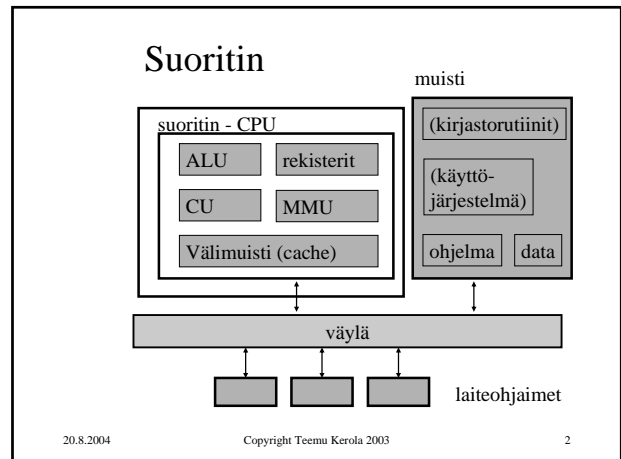


Luento 5 Suoritin ja väylä

Suorittimen rakenne
Väylän rakenne
Käskeyjen suoritusyksi
Suorittimen tilat
Poikkeukset ja keskeytykset
TTK-91:n ja sen simulaattorien
rakenne

20.8.2004 Copyright Teemu Kerola 2003 1



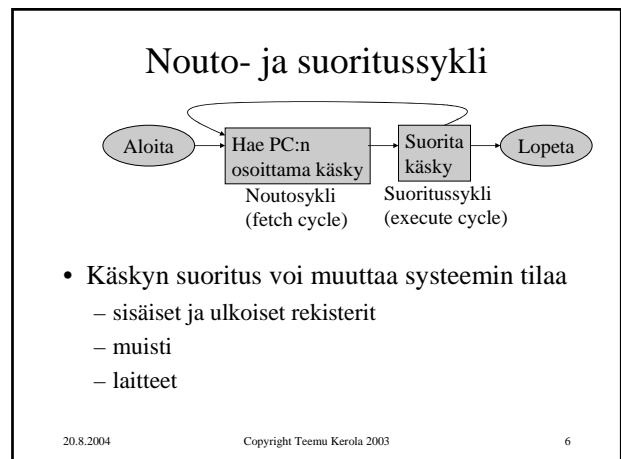
Käskeyjen nouto- ja suoritusyksi (5)

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

Suoritin ei näe mitään suurempia kokonaisuuksia kuin konekäskeyjä!

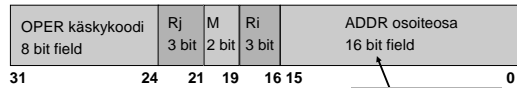
Suoritin ei tiedä mitään ohjelmista!

20.8.2004 Copyright Teemu Kerola 2003 5



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 osoit.)
- 10 eli 2 kpl, epäsuora osoitus (STORE: epäkelpo arvo)
- (11 eli 3 kpl, epäkelpo arvo → poikkeustilanne)

muistiosoite tai
(pienehkö) vakio

(addressing mode)

20.8.2004

Copyright Teemu Kerola 2003

7

Nouto- ja suoritusyksi 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 Ei kaikilla käskyillä
- ALU operaatio
 - tulos rekisteriin R0-R7 tai TR:ään (STORE, PUSH)
- Muistiin talletus Ei kaikilla käskyillä

ks. TTK-91
suorittimen
rakennekuva

20.8.2004

Copyright Teemu Kerola 2003

8

Käskyn noutovaihe ⁽⁴⁾

ks. TTK-91 suorittimen rakennekuva

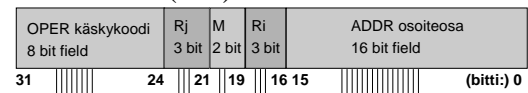
- Vie PC:n arvo MAR:iin
- Aseta muistin lukusignaali kontrolliväylälle asentoon "lue"
- Odota, kunnes muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä konekäsky MBR:stä IR:ään

20.8.2004

Copyright Teemu Kerola 2003

9

Käskyn purku ja tehollisen muistiosoitteen (EA) laskemisvaihe



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

20.8.2004

Copyright Teemu Kerola 2003

10

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)

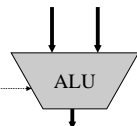
20.8.2004

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 ($R0, \dots, R7$)
 - 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



20.8.2004

Copyright Teemu Kerola 2003

12

Tuloksen muistiin kirjoitus -vaihe ⁽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

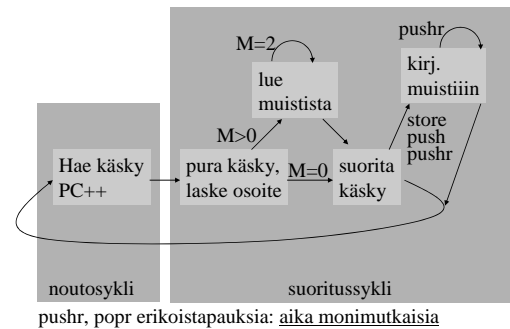
Lisää tietoa?  Tietokoneen rakenne-kurssi

20.8.2004

Copyright Teemu Kerola 2003

13

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



20.8.2004

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

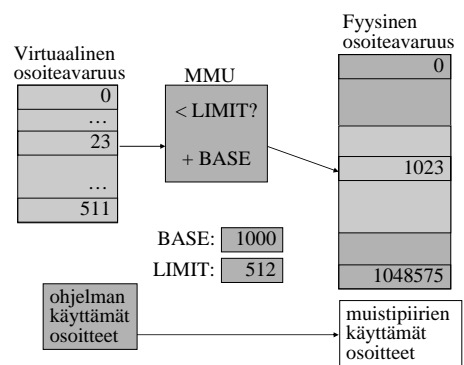
VA = virtual address, PA = physical address = BASE+VA

20.8.2004

Copyright Teemu Kerola 2003

15

TTK-91 virtuaalimuisti



20.8.2004

Copyright Teemu Kerola 2003

16

Virtuaalimuistin osoitteenmuunnosmenetelmiä ⁽³⁾

- Kanta- ja rajarekisteriin perustuva
 - base ja limit rekisterit (esim. ttk-91, 8086, ...)
- Sivuttava
 - sivutaulut
 - virtuaaliavaruus jaettu saman kokoisiiin sivuihin
- Segmentoiva
 - virtuaaliavaruus jaettu ohjelman mukaan erillisiin eri kokoisiiin segmentteihin
 - koodi segmentti, data segmentti, ...

Lisää tietoa?  käyttöjärjestelmäkurssit

20.8.2004

Copyright Teemu Kerola 2003

17

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)

20.8.2004

Copyright Teemu Kerola 2003

18

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?


20.8.2004

Copyright Teemu Kerola 2003

19

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

20.8.2004

Copyright Teemu Kerola 2003

20

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

20.8.2004

Copyright Teemu Kerola 2003

21

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 haaraututaan 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"

20.8.2004

Copyright Teemu Kerola 2003

22

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) tullessiin signaaleihin reagoiminen
 - kontrolliväylältä tuleva signaali

20.8.2004

Copyright Teemu Kerola 2003

23

Käskyn aiheuttamat virhetilanteet

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

20.8.2004

Copyright Teemu Kerola 2003

24

Käskyn aiheuttamat muut poikkeustilanteet

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

20.8.2004

Copyright Teemu Kerola 2003

25

Ulkoapäin (muualta kuin suorittimelta) tulleet keskeytykset

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

20.8.2004

Copyright Teemu Kerola 2003

26

Keskeytyksäsittelijä

- Osa käyttöjärjestelmää
- Ennen keskeytyksäsittelijään hyppäämistä asetetaan suoritin ja MMU etuoikeutettuun käyttöjärjestelmätilaan (supervisor state)
 - SR:n bitti P on päällä => etuoikeutettu tila eli (P = Priviledged) käyttöjärjestelmä tila
 - 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

20.8.2004

Copyright Teemu Kerola 2003

27

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)

20.8.2004

Copyright Teemu Kerola 2003

28

Suorittimen tilan muuttaminen (2)



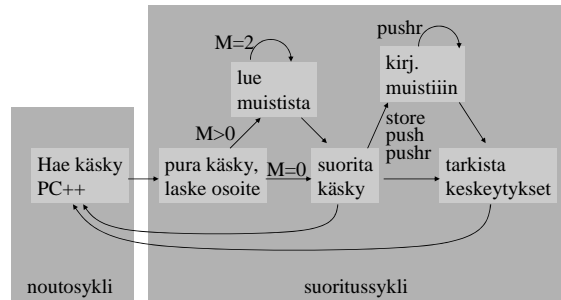
- Käyttäjätila → etuoikeutettu tila
 - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
 - keskeytyksäsittelijä tarkistaa onko (oliko) oikeutta tilan vaihtoon (interrupt handler)
- Etuoikeutettu tila → käyttäjätila
 - etuoikeutettu konekäsky "return from interrupt handler" esim. IRET (Pentium II)
 - palauttaa kontrollin keskeytyneeseen kohtaan ja suorittimen tilan keskeytystä edeltäneeseen tilaan

20.8.2004

Copyright Teemu Kerola 2003

29

TTK-91 Nouto- ja suoritusyksi vielä vähän tarkemmin



20.8.2004

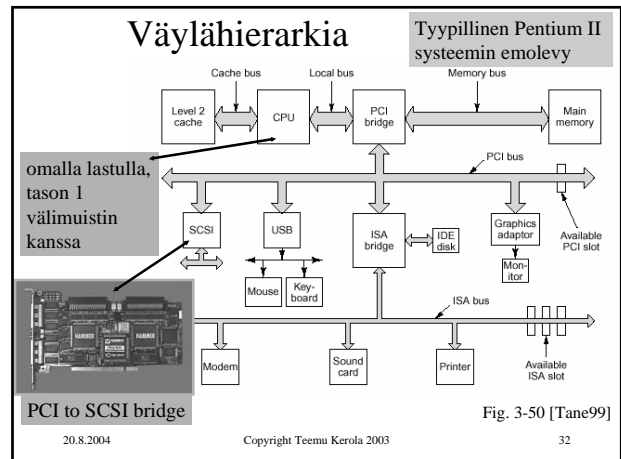
Copyright Teemu Kerola 2003

30

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

20.8.2004 Copyright Teemu Kerola 2003 31



Väylät (1)

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

Lisää tietoa? Tietokoneen rakenne-kurssi

20.8.2004 Copyright Teemu Kerola 2003 33

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
 - Tietokoneessa myös suorituksen animointi
- Toteuttaa myös TTK-91 koneen käyttöjärjestelmän osat tavallista ohjelmaa
 - assembler kääntäjä, lataaja, debugger, kesk. käsittelijät
- Graafinen käyttöliittymä

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

20.8.2004 Copyright Teemu Kerola 2003 34

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;

lopetta suoritus jos SVC tai keskeytys SR.O = ...

20.8.2004 Copyright Teemu Kerola 2003 35

