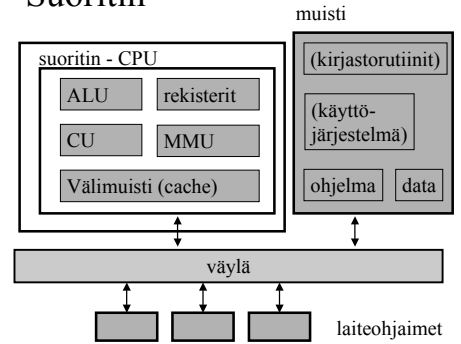


Luento 5

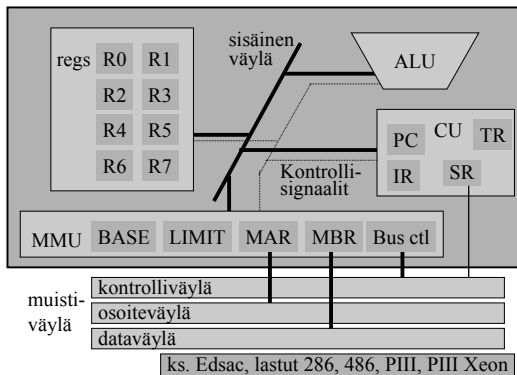
Suoritin ja väylä

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

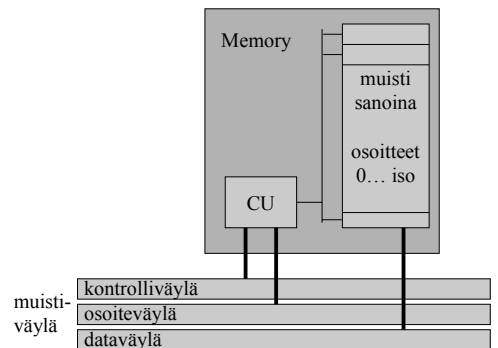
Suoritin



TTK-91 suorittimen rakenne (1)



TTK-91 muistin rakenne



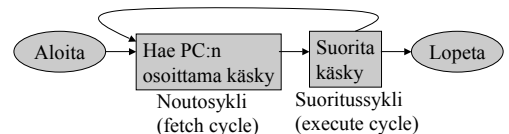
Käskeyjen nouto- ja suoritusyksi (5)

- 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!

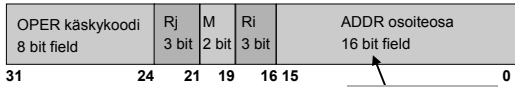
Nouto- ja suoritusyksi



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

muistiosoite tai (pienehkö) vakio

(addressing mode)

15/03/2002

Copyright Teemu Kerola 2002

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 $Ri=R0$)
- 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

15/03/2002

Copyright Teemu Kerola 2002

8

Käskyn noutovaihe ⁽⁴⁾

ks. TTK-91 suorittimen rakennekuva

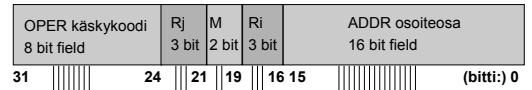
- Vie PC:n arvo MAR:iin
- Aseta muistin lukusignaali kontrolliväylälle asentoon ”lue”
- Odota, kunnes muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä konekäsky MBR:stä IR:ään

15/03/2002

Copyright Teemu Kerola 2002

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
 - Effective Address (EA) on nyt TR:ssä

15/03/2002

Copyright Teemu Kerola 2002

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)

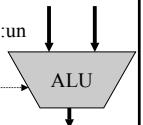
15/03/2002

Copyright Teemu Kerola 2002

11

ALU operaatio -vaihe ⁽¹⁰⁾

- Lähtötilanne
 - käsky haettu ja purettu osiin IR:ssä
 - 1. operandi rekisterissä ($R0, \dots, R7$)
 - 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



Tässä tapahtuu tietokoneen tekemä työ, kaikki muu on hallintoa

15/03/2002

Copyright Teemu Kerola 2002

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

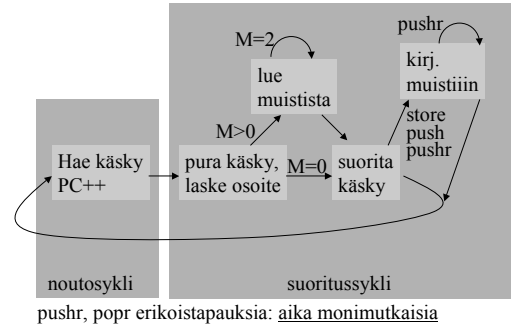


15/03/2002

Copyright Teemu Kerola 2002

13

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



15/03/2002

Copyright Teemu Kerola 2002

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 $VA \in [0, LIMIT-1]$. 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

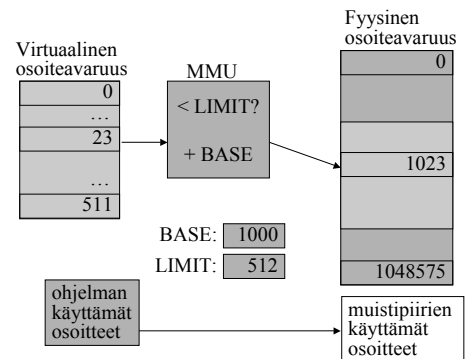
VA = virtual address, PA = physical address = BASE+VA

15/03/2002

Copyright Teemu Kerola 2002

15

TTK-91 virtuaalimuisti



15/03/2002

Copyright Teemu Kerola 2002

16

Virtuaalimuistin osoitteenmuunnos menetelmiä ⁽⁴⁾

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



15/03/2002

Copyright Teemu Kerola 2002

17

Keskeytystilanteet ⁽³⁾

- Mikä tahansa tilanne, jonka käsittely vaatii poikkeuksen käskyjen normaaliin suoritussykseen
- 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 tyyppisiä

15/03/2002

Copyright Teemu Kerola 2002

18

Keskeytysten käsittely ⁽⁴⁾

- Jokainen mahdollinen keskeytystyyppi on ennalta tunnettu
- Jokaiselle keskeytystyypille on oma käyttöjärjestelmän tuntema keskeytyskäsitteilyrutiini `interrupt handler`
- Jokaisen käskyn suorituksen jälkeen tarkistetaan keskeytysten olemassaolo SR:stä ja haaraututaan keskeytyskäsitteilyrutiiniin tarvittaessa
 - joskus keskeytykset on estetty (ttk-91:ssä SR:n bitti D)
 - paluu käsitteilyrutiinistä ”return-from-interrupt” käskyllä
- ”Yllättävä aliohjelmakutsu”

Keskeytystyyppejä ⁽³⁾

- Käskyn aiheuttamat virhetilanteet
- Käskyn aiheuttamat muut poikkeustilanteet
 - kyseessä ei siis ole virhetilanne
- Ulkoapäin (muualta kuin CPU:lta) tulleet

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 keskeytykset

Ulkoapäin (muualta kuin suorittimelta) tulleet keskeytykset ⁽³⁾

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

Keskeytyskäsitteilyrutiini

- Osa käyttöjärjestelmästä
- Ennen keskeytyskäsitteilyrutiinin 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
- Käsitteilyrutiinistä paluun yhteydessä MMU:n tila ja suorittimen tila asetetaan ennalleen

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

15/03/2002

Copyright Teemu Kerola 2002

25

Suorittimen tilan muuttaminen ⁽⁶⁾



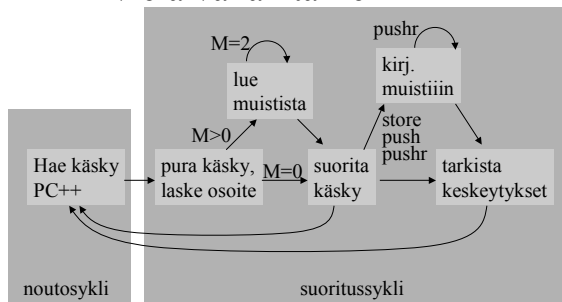
- Käyttäjätila → etuoikeutettu tila
 - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
 - keskeytyskäsitteittä tarkistaa onko (oliko) oikeutta tilan vaihtoon (interrupt handler)
- Etuoikeutettu tila → käyttäjätila
 - etuoikeutettu konekäsky “return from interrupt handler” esim. IRET (Pentium II)
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

15/03/2002

Copyright Teemu Kerola 2002

26

TTK-91 Nouto- ja suoritussykli vielä vähän tarkemmin



15/03/2002

Copyright Teemu Kerola 2002

27

Väylät ⁽⁵⁾

- Tiedon siirtoa varten laitteistossa
- Yksi kirjoittaja kerrallaan
- 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



(internal bus)

(memory bus)

(I/O bus)

15/03/2002

Copyright Teemu Kerola 2002

28

Väylähierarkia

Tyypillinen Pentium II systeemin emolevy

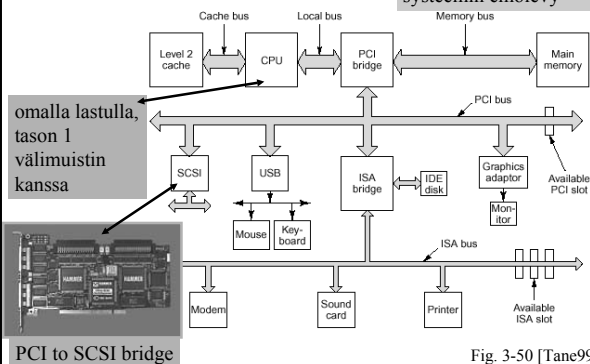


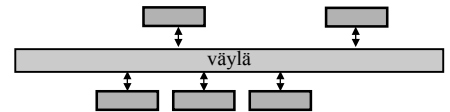
Fig. 3-50 [Tane99]

15/03/2002

Copyright Teemu Kerola 2002

29

Väylät ⁽⁵⁾



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

Lisää tietoa? Tietokoneen rakenne -kurssi

15/03/2002

Copyright Teemu Kerola 2002

30

TTK-91 koneen KOKSI simulaattori (6)

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

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

TTK-91 käsken suoritusyhti (5)

```

hae käsken simuloidusta muistista      IR = mem[PC]
pura käsken osiin (OPER, Rj, M, Ri, ADDR) ja
laske osoiteosan arvo TR (ADDR tai regs[Ri]+ADDR)
      ADDR = IR % 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 konekäsken 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 = ...
    
```

-- Luennon 5 loppu --

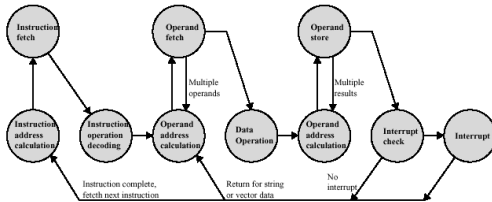


Figure 3.12 Instruction Cycle State Diagram, With Interrupts

[Stal99]