

# Luento 5

## Suoritin ja väylä

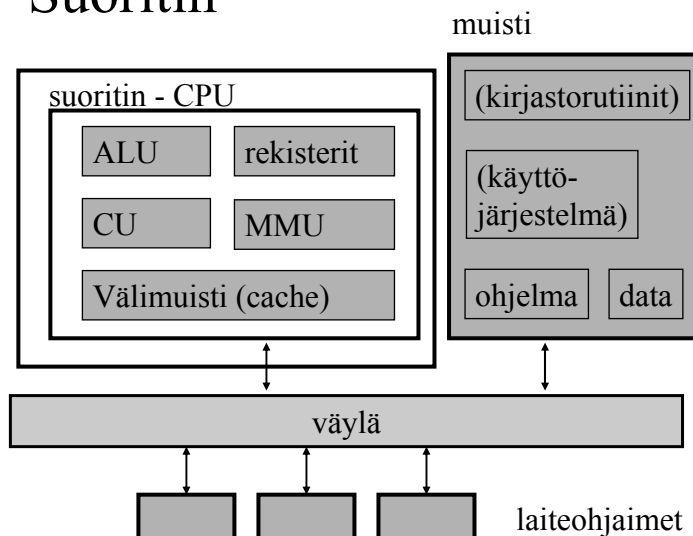
Suorittimen rakenne  
Väylän rakenne  
Käskeyjen suoritusykyli  
Suorittimen tilat  
Poikkeukset ja keskeytykset  
TTK-91:n ja KOKSI:n rakenne

14.1.2003

Copyright Teemu Kerola 2003

1

## Suoritin

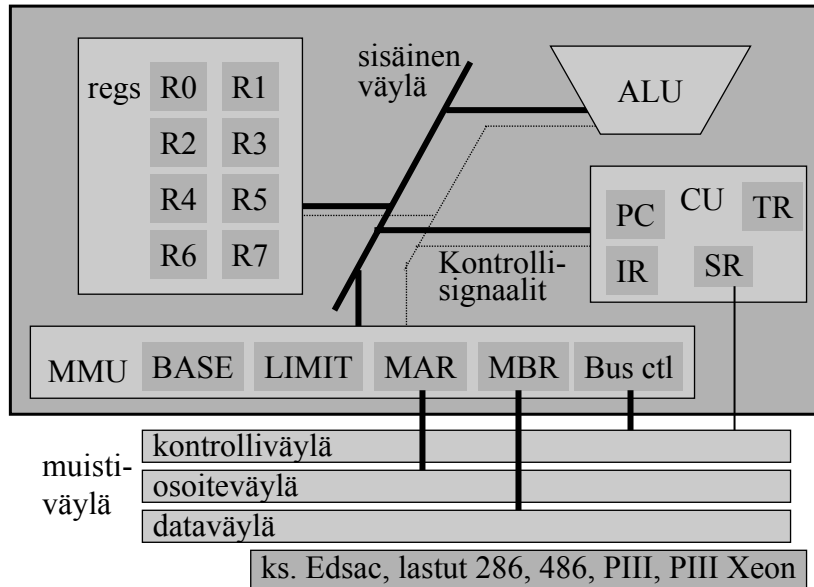


14.1.2003

Copyright Teemu Kerola 2003

2

# TTK-91 suorittimen rakenne (1)

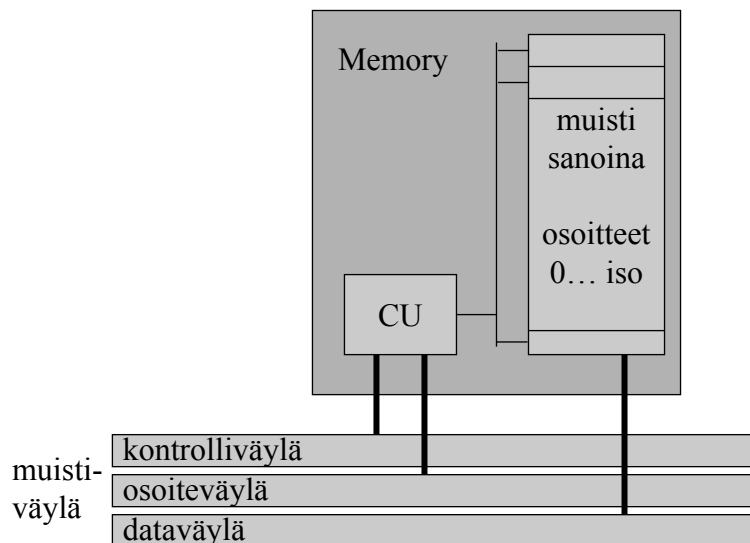


14.1.2003

Copyright Teemu Kerola 2003

3

# TTK-91 muistin rakenne



14.1.2003

Copyright Teemu Kerola 2003

4

# Käskyjen nouto- ja suoritusykli <sup>(5)</sup>

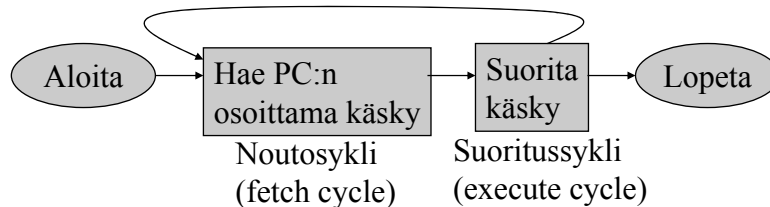


- Hae PC:n osoittama konekäsky muistista
  - lisää samalla PC:n arvoa yhdellä
- Suorita konekäsky
  - jos (ehdollinen) hyppykäsky, niin PC:n arvo voi vielä muuttua

Suoritin ei näe mitään suurempia kokonaisuuksia kuin konekäskyjä!

Suoritin ei tiedä mitään ohjelmista!

# Nouto- ja suoritusykli



- 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 osoit.)

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

(11 eli 3 kpl, epäkelpo arvo → poikkeustilanne)

muistiosoite tai  
(pienekkö) vakio

(addressing  
mode)

## Nouto- ja suoritussykli tarkemmin <sup>(5)</sup>

- 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



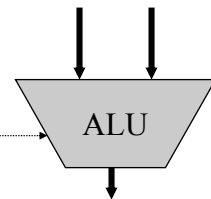
## Operandin luku vaihe <sup>(4)</sup>

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 <sup>(10)</sup>

- 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 kirjoitussignaali kontrolliväylälle asentoon ”kirjoita muistiin”
- Odota kunnes sana siirretään muistiin väylää pitkin ja väylän kontrollisignaali kertovat muistiinkirjoittamisen tapahtuneen

Lisää  
tietoa?



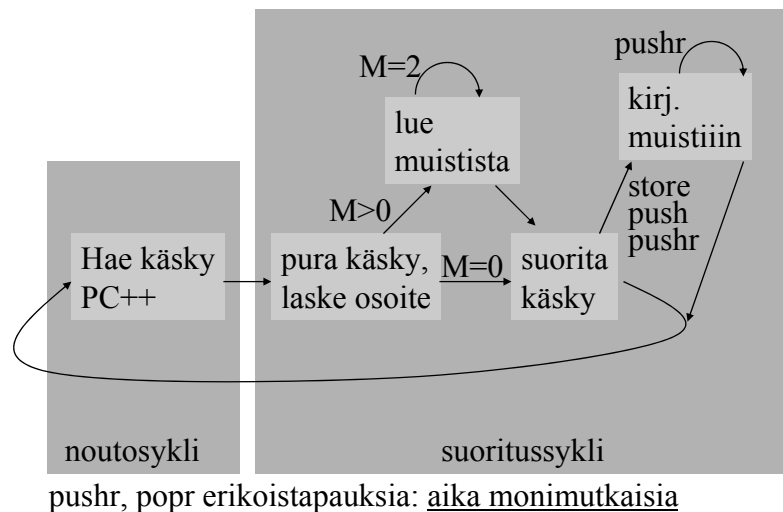
Tietokoneen  
rakenne  
-kurssi

14.1.2003

Copyright Teemu Kerola 2003

13

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



14.1.2003

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

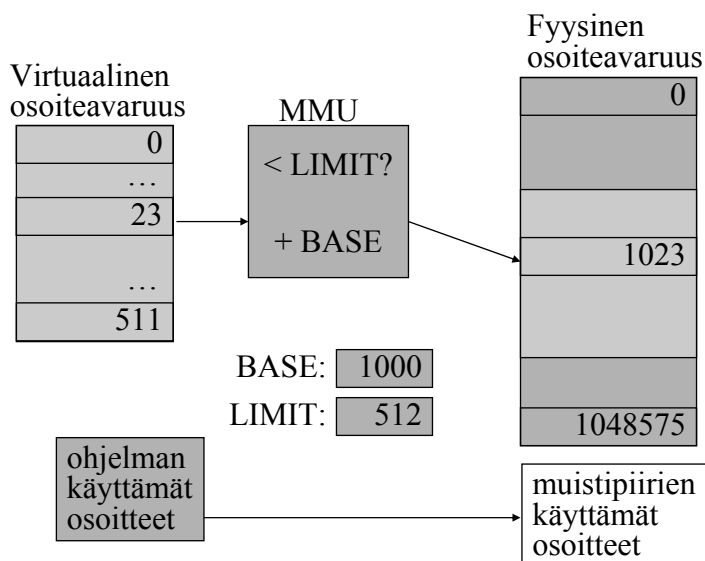
VA = virtual address, PA = physical address = BASE+VA

14.1.2003

Copyright Teemu Kerola 2003

15

## TTK-91 virtuaalimuisti



14.1.2003

Copyright Teemu Kerola 2003

16



## Virtuaalimuistin osoitteenmuunnos menetelmiä <sup>(4)</sup>

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



## Keskeytystilanteet <sup>(3)</sup>

- 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 <sup>(4)</sup>

- 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 <sup>(3)</sup>

- 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 CPU:lta) tulleisiin signaaleihin reagoiminen

## Käskyn aiheuttamat virhetilanteet <sup>(5)</sup>

- 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 <sup>(3)</sup>

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

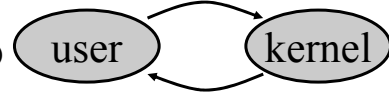
# Ulkoapäin (muualta kuin suorittimelta) tulleet keskeytykset <sup>(3)</sup>

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

## Keskeytyskäsittelijä

- Osa käyttöjärjestelmää
- Ennen keskeytyskäsittelijän 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 <sup>(6)</sup>



- 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)

14.1.2003

Copyright Teemu Kerola 2003

25

## Suorittimen tilan muuttaminen <sup>(6)</sup>



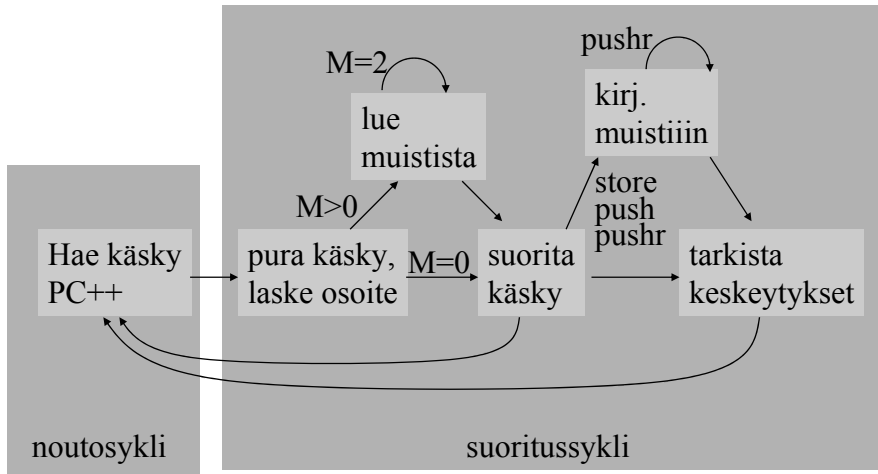
- Käyttäjätila → etuoikeutettu tila
  - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
  - keskeytyskäsittelijä tarkistaa onko (oliko) oikeutta tilan vaihtoon (interrupt handler)
- Etuoikeutettu tila → käyttäjätila
  - etuoikeutettu konekäsky “return from interrupt handler” esim. IRET (Pentium II)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

14.1.2003

Copyright Teemu Kerola 2003

26

# TTK-91 Nouto- ja suoritussykli vielä vähän tarkemmin

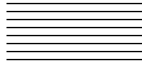


14.1.2003

Copyright Teemu Kerola 2003

27

## Väylät <sup>(5)</sup>

- 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

14.1.2003

Copyright Teemu Kerola 2003

28

# Väylähierarkia

Tyypillinen Pentium II  
systemin emolevy

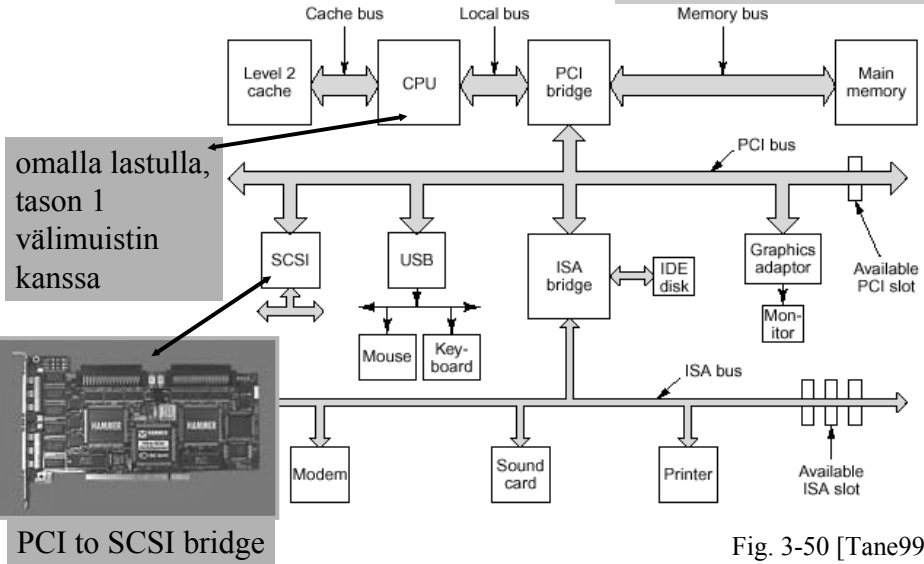


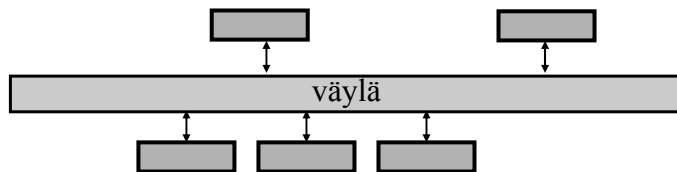
Fig. 3-50 [Tane99]

14.1.2003

Copyright Teemu Kerola 2003

29

## Väylät (5)



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

Lisää  
tietoa?



Tietokoneen  
rakenne-  
kurssi

14.1.2003

Copyright Teemu Kerola 2003

30

# TTK-91 koneen KOKSI simulaattori <sup>(6)</sup>

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

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

## TTK-91 käskyn suoritusyksi <sup>(5)</sup>

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)

$ADDR = IR \% 32768$      $TR = regs[Ri] + ADDR$

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)  $\Rightarrow$  regs[Rj] += TR;

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



## -- Luennon 5 loppu --

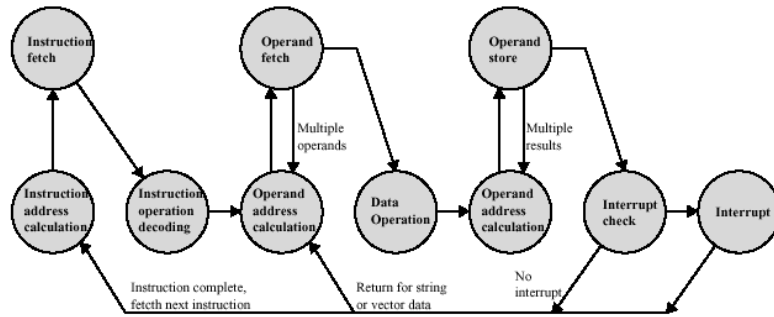


Figure 3.12 Instruction Cycle State Diagram, With Interrupts

[Sta199]