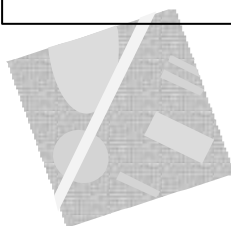


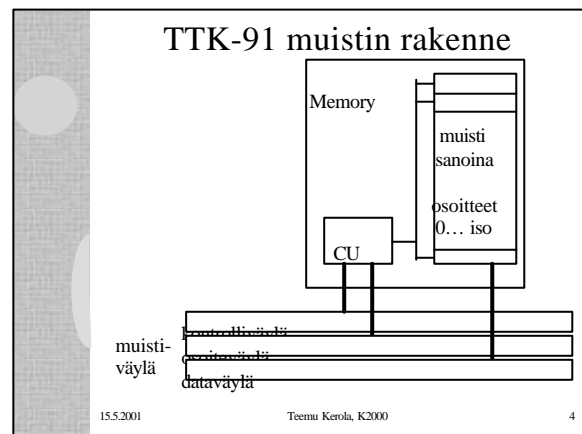
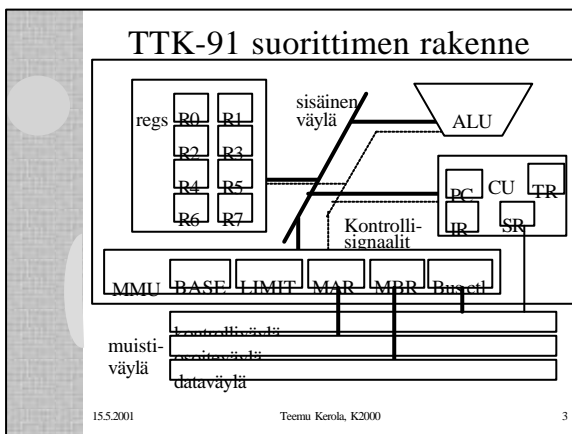
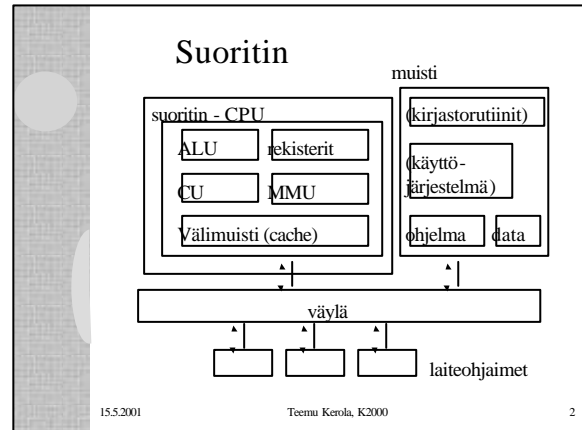
Jakso 5

Suoritin ja väylä



Suorittimen rakenne
 Väylän rakenne
 Käskyjen suoritusyksi
 Poikkeukset ja keskeytykset

15.5.2001 Teemu Kerola, K2000 1



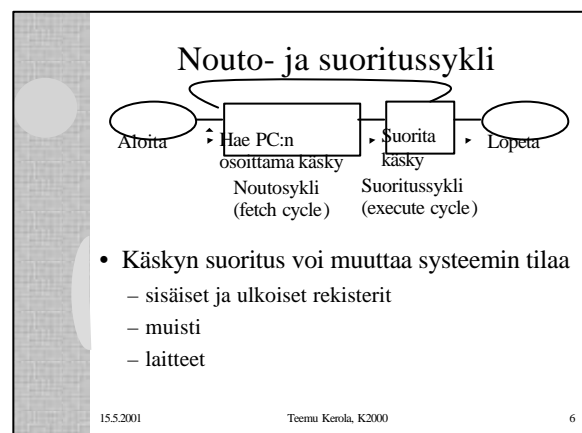
Käskyjen nouto- ja suoritusyksi (5)

- 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!

15.5.2001 Teemu Kerola, K2000 5



TTK-91 konekäskyn rakenne

- Käskyn esitys bittitasolla on aina:

OPER käskykoodi 8 bit field	Rj 3 bit	M 2 bit	Ri 3 bit	ADDR osoiteosa 16 bit field
31	24	21	19	16 15
				0

Rj = käskyn ensimmäinen operandi muistiosoite tai (pienehkö) vakio

Ri = indeksirekisteri (R0 - R7)

M = muistinoutojen määrä toiseen operandiin (ennen mahdollista muistiin talletusta) (addressing mode)

00 eli 0 kpl, rekisteri tai välitön osoitus
 01 eli 1 kpl, suora osoitus
 10 eli 2 kpl, epäsuora osoitus
 11 eli 3 kpl, epäkelvo arvo → poikkeustilanne

15.5.2001 Teemu Kerola, K2000 7

Nouto- ja suoritussykli tarkemmin ⁽⁵⁾

- Noutovaihe ks. TTK-91 suorittimen rakennekuva
 - muistista MBR:n kautta IR:ään
 - Lisää 1 PC:hen
- Käskyn purku ja muistiosoitteen (EA) lasku
 - OPER, Rj, M, Ri, ADDR
 - TR ← (Ri) + ADDR (tai pelkkä ADDR)
- Operandin nouto Ei kaikilla käskyillä
 - muistista MBR:n kautta TR:ään (0-2 krt ?)
- ALU operaatio
- Muistiin talletus Ei kaikilla käskyillä
 - muistiin MBR:n kautta

15.5.2001 Teemu Kerola, K2000 8

Käskyn noutovaihe ⁽⁴⁾

ks. TTK-91 suorittimen rakennekuva

- Vie PC:n arvo MAR:iin
- Aseta muistin kontrollisignaali väylälle asentoon "lue"
- Odot kunnes muistiväylä vapautuu ja muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä konekäsky MBR:stä IR:ään

15.5.2001 Teemu Kerola, K2000 9

Käskyn purku ja muistiosoitteen laskemis -vaihe

OPER käskykoodi 8 bit field	Rj 3 bit	M 2 bit	Ri 3 bit	ADDR osoiteosa 16 bit field
31	24	21	19	16 15
				0

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

15.5.2001 Teemu Kerola, K2000 10

Operandin luku vaihe ⁽⁴⁾

ks. TTK-91 suorittimen rakennekuva

- Vie muistiosoite MAR:iin
- Aseta muistin kontrollisignaali väylälle asentoon "lue"
- Odot kunnes muistiväylä vapautuu ja muistipiiri toimittaa väylän kautta uuden arvon MBR:ään
- Siirrä sana MBR:stä TR:ään
 - tai suoraan johonkin laiterekisteriin (R0-R7)

15.5.2001 Teemu Kerola, K2000 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:ssa
 - vie operandit sisäistä väylää pitkin ALU:un
 - anna ALU:lle sopiva ohjaussignaali
 - add, mul, copyLeft, comp, ...
 - odota, että tulos valmis
 - tulos siirretään MBR:ään ja/tai SR:ään

Tässä tapahtuu tietokoneen tekemä työ, kaikki muu on hallintoa

15.5.2001 Teemu Kerola, K2000 12

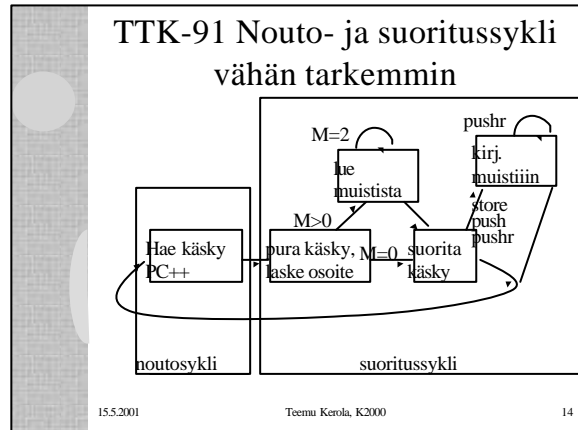
Tuloksen muistiin kirjoitus - vaihe ⁽⁴⁾

ks. TTK-91 suorittimen rakennekuva

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

Lisää tietoa? käyttöjärjestelmäkurssit

15.5.2001 Teemu Kerola, K2000 13



MMU:n toiminta ⁽²⁾

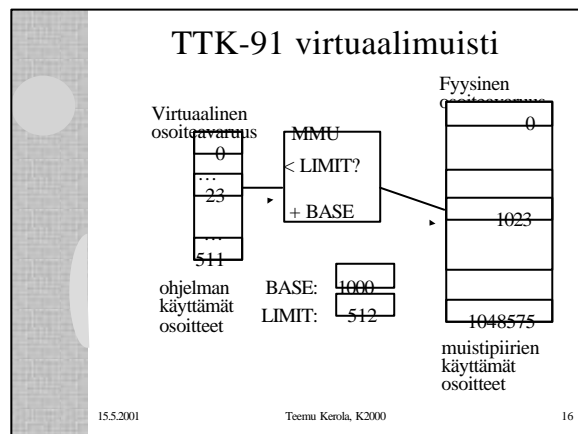
ks. TTK-91 suorittimen rakennekuva

- 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 ∈ [0, LIMIT-1]. Jos ei ole, niin aseta bitti MSR:ssä päälle, ja lopeta käskyn suoritus
 - Lisää VA:han BASE ja laita tämä arvo (PA) MAR:iin

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

Lisää tietoa? käyttöjärjestelmäkurssit

15.5.2001 Teemu Kerola, K2000 15



Virtuaalimuistin menetelmiä ⁽⁴⁾

- Kanta- ja rajarekisteriin perustuva
 - base ja limit rekisterit
- Sivuttava
 - sivutaulut
 - virtuaaliavaruus jaettu saman kokoisiiin sivuihin
- Segmentoiva
 - virtuaaliavaruus jaettu erillisiin eri kokoisiiin segmentteihin
 - koodi segmentti, data segmentti, ...

Lisää tietoa? käyttöjärjestelmäkurssit

15.5.2001 Teemu Kerola, K2000 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,
- Jatkossa yleisnimi keskeytys tarkoittaa kaikkia näitä eri tapauksia tai tyyppejä

15.5.2001 Teemu Kerola, K2000 18

Keskeytysten käsittely ⁽⁴⁾

- Jokainen mahdollinen keskeytystyyppi on ennalta tunnettu
- Jokaiselle keskeytystyypille on oma käyttöjärjestelmän tuntema keskeytyskäsittelyrutiini interrupt handler
- Käskyn suorituksen jälkeen tarkistetaan keskeytysten olemassaolo SR:stä ja haaraututaan keskeytyskäsittelijään tarvittaessa
 - joskus keskeytykset on estetty (SR:n bitti D)
 - paluu käsittelijästä "return-from-interrupt" käskyllä
- "Yllättävä aliohjelmakutsu"

15.5.2001 Teemu Kerola, K2000 19

Keskeytystyyppiä ⁽¹⁴⁾

- 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 kone käsky
 - trace keskeytys
- Ulkoapäin (muualta kuin CPU:lta) tulleet
 - kellolaitekeskeytys (esim. joka 10 ms)
 - laitekeskeytys (esim. levy I/O valmis)
 - laitteistovirhe (esim. virhe väylän tiedonsiirrossa)

15.5.2001 Teemu Kerola, K2000 20

Keskeytyskäsittelijä

- Osa käyttöjärjestelmästä
- Ennen käsittelijän aloittamista asetetaan asetetaan etuoikeutettu käyttöjärjestelmätilaan (supervisor state)
 - Asetetaan bitti P SR:ssä => etuoikeutettu (KJ) tila
 - käyttöjärjestelmätilassa saa viitata mihin tahansa kohtaan muistia (MMU: BASE=0, LIMIT="hyvin iso")
 - käyttöjärjestelmätilassa saa käyttää kaikkia konekäskyjä
- Käsittelijästä paluun yhteydessä MMU:n tila ja prosessorin tila asetetaan ennalleen

15.5.2001 Teemu Kerola, K2000 21

Prossessorin tilat ⁽⁶⁾

user

kernel

- 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
 - voi viitata kaikkialle muistiin, myös käyttöjärjestelmän ytimeen (kernel)
 - voi käyttää (myös) suoria muistiosoitteita (PA)

15.5.2001 Teemu Kerola, K2000 22

Prossessorin tilan muuttaminen ⁽⁶⁾

user

SVC, INT

↕

kernel

kernel

IRET


- Käyttäjätila → etuoikeutettu tila
 - keskeytys tai suora KJ:n palvelupyyntö (SVC käsky)
 - keskeytyskäsittelijä tarkistaa onko (interrupt handler) vaihtoon
- Etuoikeutettu tila → käyttäjätila
 - etuoikeutettu konekäsky "return from interrupt handler" esim. IRET (Pentium II)
 - palauttaa kontrollin ja prosessorin tilan keskeytyneeseen kohtaan ja sen hetkiseen tilaan

15.5.2001 Teemu Kerola, K2000 23

TTK-91 Nouto- ja suoritusyksi vielä vähän tarkemmin

15.5.2001 Teemu Kerola, K2000 24

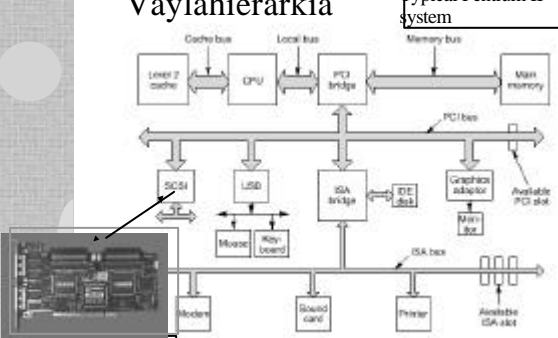
Väylät (5)

- Tiedon siirtoa varten laitteistossa
- Yksi käyttäjä 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

15.5.2001 Teemu Kerola, K2000 25

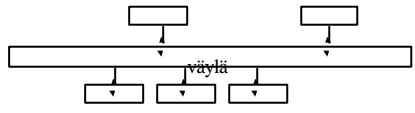
Väylähierarkia

Typical Pentium II system




15.5.2001 Teemu Kerola, K2000 26

Väylät (5)



- Kullakin laitteella oma osoite
- Yksi lähettää, kaikki kuulevat
- Paljon erilaisia
- Lähellä prosessoria olevat ovat nopeampia

Lisää tietoa?  Tietokoneen rakenne kurssi

15.5.2001 Teemu Kerola, K2000 27

TTK-91 koneen KOKSI simulaattori (6)

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

15.5.2001 Teemu Kerola, K2000 28

TTK-91 käskyn suoritusvykli (5)

hae käsky simuloidusta muistista $IR = mem[PC]$

pura käsky osiin (OPER, Rj, M, Ri, ADDR) ja laske osat $ADDR = IR \% 65536$ $TR = regs[Ri] + ADDR$

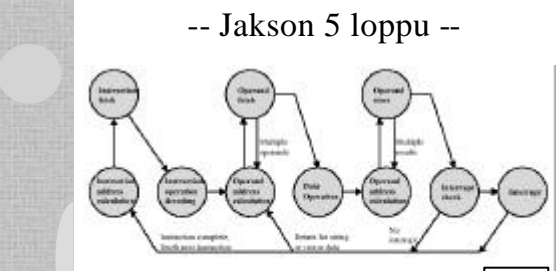
tee tarvittava määrä (M) operandin hakuja muistista rekisteriin TR $TR = mem[TR]$

valitse alku $if (opcodeOK[OPER] = FALSE) then SR.U = 1;$
simuloi konekäskyn suorituksen muutokset rekistereihin $ADDRi, M ADDR(Rj) \Rightarrow regs[Ri] += IR;$

lopetta suoritus jos SVC tai keskeytys $SR.O = \dots$

15.5.2001 Teemu Kerola, K2000 29

-- Jakson 5 loppu --



[Stal99]

15.5.2001 Teemu Kerola, K2000 30