

Luento 2

TTK-91 tietokone ja sen KOKSI simulaattori

Miksi TTK-91?
TTK-91 rakenne ja
käskykanta-arkkitehtuuri
Mikä on simulaattori?
Miten TTK-91 ohjelmia
suoritetaan simulaattorissa?

5.3.2002

Copyright Teemu Kerola 2002

1

Tietokone TTK-91 ⁽⁴⁾

- Laitteisto, hardware (HW)
 - suoritin, muisti, väylät
 - oheislaitteiden liitännät
- Käskykanta - konekieliarkkitehtuuri
 - käyttöliittymä laitteistoon
 - konekäskyt, tiedon esitysmuodot, tietotyypit
- Symbolinen konekieli
 - luettavampi muoto konekielestä
 - kullakin symbolilla yksikäsitteiset arvot
- KOKSI simulaattori
 - ohjelma, joka simuloi TTK-91 koneen laitteistoa

5.3.2002

Copyright Teemu Kerola 2002

4

Miksi konekieltä?

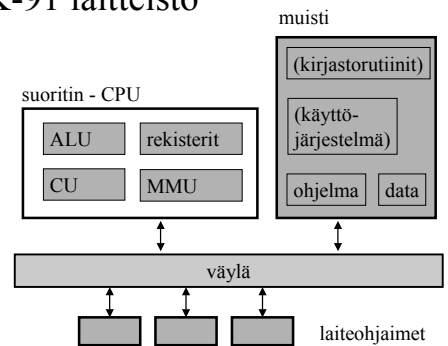
- Koneen toiminnan ymmärtäminen
- Oman ohjelman toiminnan ymmärtäminen
- Koneenläheinen ohjelmointi
- Kääntäjän tekeminen
 - kääntäjä kääntää konekielisen lausekielisen ohjelman
- Ohjelman tehokkuus
 - osia ohjelmasta ohjelmoidaan suoraan konekielillä

5.3.2002

Copyright Teemu Kerola 2002

2

TTK-91 laitteisto



5.3.2002

Copyright Teemu Kerola 2002

5

Miksi ei oikeaa konekieltä?

- Oikeat konekielet huomattavasti monimutkaisempia
 - niiden opetteluun tarvitaan oma kurssi
- Vaikeaa valita sopivinta
 - paljon erilaisia konekieliä
- Keskitytään vain opetuksen kannalta oleellisiin asioihin
 - tarvittaessa oikea konekieli 'helppo' oppia

5.3.2002

Copyright Teemu Kerola 2002

3

TTK-91 rekisterit

- 8 yleisrekisteriä ks. Kuva 4.1 [Häkk98]
 - vain näitä rekistereitä voi koskettaa (suoraan) konekäskyillä
 - kaikki laskenta tapahtuu rekistereiden avulla
 - R0 työrekisteri
 - indeksirekisterinä == 0 (tiettyssä tilanteessa R0:n käyttö tarkoittaa lukua 0 rekisterin R0 sisällön asemesta)
 - R1-R5 työ- ja indeksirekistereitä
 - tyyppi riippuu rekisterin käytöstä konekäskyssä
 - pino-osoitin SP (R6)
 - ympäristöosoitin FP (R7)

Stack Pointer

Frame Pointer

5.3.2002

Copyright Teemu Kerola 2002

6

TTK-91 Kontrolliyksikkö (CU)

ks. Kuva 4.1 [Häkk98]

- PC - Program Counter, käskyosoitin
 - seuraavaksi suoritettavan konekäskyn osoite
- IR - Instruction Register, käskyrekisteri
 - suorituksessa oleva konekäsky
- TR - Temporary Register, apurekisteri
 - tilapäinen talletuspaikka käskyn suoritusaikana
- SR - State Register, tilarekisteri
 - suorittimen tila ja rajoitukset tällä hetkellä

5.3.2002

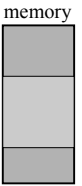
Copyright Teemu Kerola 2002

7

TTK-91 Muistinhallintayksikkö (MMU)

ks. Kuva 4.1 [Häkk98]

- Muistiinviittausrekisterit
 - MAR - Memory Address Register, muistiosoite
 - MBR - Memory Buffer Register, luettava/kirjoitettava arvo
- Käytössä oleva muistialue
 - vain tähän alueeseen voi nyt viitata
 - BASE - muistisegmentin alkuosoite
 - LIMIT - muistisegmentin koko
 - kaikki muistiosoitteet suhteellisia BASE rekisterin arvoon
 - käyttöjärjestelmä asettaa ja valvoo



5.3.2002

Copyright Teemu Kerola 2002

10

TTK-91 Tilarekisteri SR ⁽³⁾

- Tilatietoa siitä, mitä suorittimella tapahtui edellisen käskyn suorituksessa
 - virhetilanteet, poikkeukset ks. Kuva 4.1 [Häkk98]
 - konekäsky olikin käyttöjärjestelmän palvelupyyntö
 - vertailun tulos
- Tilatietoa siitä, mitä systemissä tapahtui viime aikoina
 - käsittelemättömät laitteiden antamat signaalit (laitekeskeytykset, device interrupts)
- Tilatietoa siitä, mitä suoritin saa tehdä jatkossa
 - etuoikeutettu tila: kaikki muistialueet, kaikki käskyt
 - poikkeukset ja keskeytykset sallittuja vai ei?

5.3.2002

Copyright Teemu Kerola 2002

8

TTK-91 Käskykanta

- Tietotyypit
- Konekäskyjen tyypit
- Konekäskyn rakenne
 - montako bittiiä, minkälainen sisäinen rakenne
- Muistissa olevan tiedon osoitustavat
 - konekielessä
 - symbolisessa konekielessä
- Operaatiot

5.3.2002

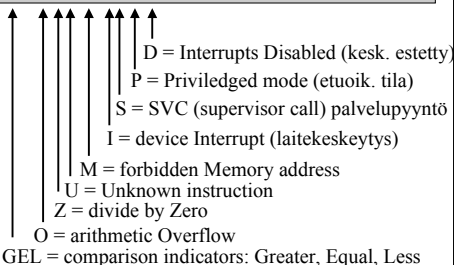
Copyright Teemu Kerola 2002

11

Tilarekisteri SR ⁽⁹⁾

32 bittiiä (kunkin arvo 0 tai 1)

SR: GEL OZUM IS P D ????????



5.3.2002

Copyright Teemu Kerola 2002

9

TTK-91 tietotyypit ⁽²⁾

- 32 bittinen kokonaisluku
 - noin 10 desimaalinumeroinen luku
- EI:
 - liukulukuja
 - merkkejä
 - totuusarvoja
 - ...

5.3.2002

Copyright Teemu Kerola 2002

12

TTK-91 käskytyypit

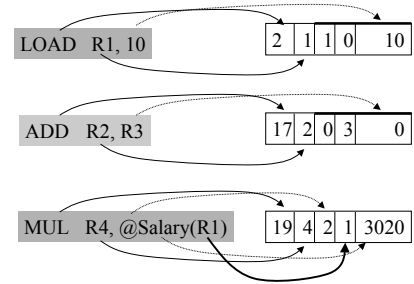
- Aina 2 operandia itse käskyssä
 - aina ei molemmilla ole merkitystä
 - JUMP
 - NOP
 - vain yksi operandi, Ri+ADDR
- Käsky aina 32 bitillä
 - ei operandeja lainkaan
- Ensimmäinen operandi aina rekisterissä
- Toinen operandi muistissa tai rekisterissä
 - luku rekisteristä on nopeampaa kuin muistista hakeminen
- ALU-operaatioiden tulos rekisteriin
 - korvaa 1. operandin arvon!

5.3.2002

Copyright Teemu Kerola 2002

13

Symbolinen konekieli vs. konekieli (3)



5.3.2002

Copyright Teemu Kerola 2002

16

Symbolinen konekieli (7)

```

LOOP: ADD R4, @TAULU(R1)
viite: OPER Rj, M ADDR(Ri)
    
```

Ri = indeksirekisteri
 ADDR = osoiteosa
 M = 2. operandin osoitusmoodi
 Rj = 1. operandina oleva rekisteri
 OPER = käskyn symbolinen nimi, opcode

viite = käskyn (symbolinen) osoite

- Suora vastaavuus konekieleen
 - yksinkertainen assembler-käännös

5.3.2002

Copyright Teemu Kerola 2002

14

Operandin osoitusmuodot symbolisessa konekielessä

- 8 eri osoitusmoodia (vain 2. operandille!)
- Tekstuaalisesti koodattuna
 - osoitusmoodi
 - = vakio [+ rekisterin arvo]
 - tyhjä arvo rekisterissä tai muistissa
 - @ epäsuora viite muistiin
 - sulkumerkit rekisterin ympärillä
 - ei sulkuja käytä rekisterin arvoa sellaisenaan
 - sulut käytä rekisterin osoittamaa muistipaikan arvoa
 - 0-arvoa ei kirjoiteta näkyviin
 - indeksirekisterinä R0 tai vakiona 0

LOAD R1, @Field1(R3)

5.3.2002

Copyright Teemu Kerola 2002

17

Symbolinen konekieli

- Symbolien vastaavuus 1:1 kaikkialla
 - viite: muistiosoite
 - operaatiokoodi, opcode: vakio
 - osoitekentän symboli: vakio tai muistiosoite
 - kenttään voi kirjoittaa joko symbolin tai arvon!

Kaikki muistiosoitteet suhteellisia BASE-osoitteeseen, eli arvoalueella [0, LIMIT-1]

- Osoitusmoodit: monimutkaisempi vastaavuus
 - konekielessä 2 moodia (plus vakion käyttö)
 - indeksoitu, epäsuora indeksoitu
 - symbolisessa konekielessä 8 moodia
 - helpottavat ohjelmointia

5.3.2002

Copyright Teemu Kerola 2002

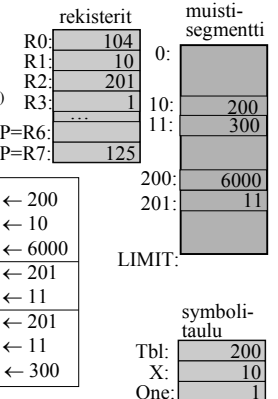
15

TTK-91 muistin osoitusmoodit (8)

ks. lista sivulla 50 [Häkk98]

```

LOAD R1, 10      ; R1 ← 200
LOAD R1, =10     ; R1 ← 10
LOAD R1, @10     ; R1 ← 6000
LOAD R4, R2      ; R4 ← 201
LOAD R4, @R2     ; R4 ← 11
LOAD R5, Tbl(R3) ; R5 ← 201
LOAD R5, Tbl(R3) ; R5 ← 11
LOAD R5, @Tbl(R3); R5 ← 300
    
```



5.3.2002

Copyright Teemu Kerola 2002

18

Indeksointi ⁽²⁾

LOAD R4,=Tbl(R3)

- Laske aina ensin tehollinen muistiosoite (effective address, EA): $EA = Tbl + (R3) = 201$
 - Sitten katso moodia ja tee niin monta muistinoutoa kun tarvitaan
 - ”=”: 0 kpl $R4 \leftarrow 201$ (vakion käyttö)
 - tyhjä: 1 kpl $R4 \leftarrow Mem[201] = 11$
 - ”@”: 2 kpl $R4 \leftarrow Mem[Mem[201]] = Mem[11] = 300$
- pelkkä rekisterin nro @-merkin jälkeen \Rightarrow 1 kpl
STORE käsky \Rightarrow 1 kpl vähemmän noutoja ja yksi tallennus

5.3.2002

Copyright Teemu Kerola 2002

19

TTK-91

muistiinviittausoperaatiot ⁽³⁾

- LOAD $LOAD R1, X$ $LOAD R5, @ptrX$
 - käskyä käytetään myös $LOAD R0, R5$ rekistereiden kopiointiin (Move operaatio)
- STORE $STORE R2, X$
 - tallettaa aina muistiin $STORE R3, Tbl(R4)$
- PUSH, POP, PUSHR, POPR $POP SP, R1 ; load \dots$
 - aliohjelmien toteuttamista varten $PUSH SP, R1 ; store \dots$
 - käsitellään myöhemmin

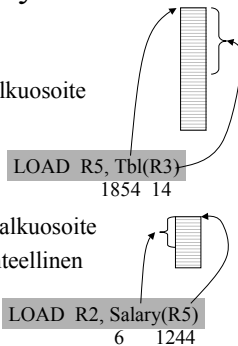
5.3.2002

Copyright Teemu Kerola 2002

22

Indeksoinnin käyttö ⁽²⁾

- Taulukot
 - Vakio (symboli) taulukon alkuosoite
 - indeksirekisterissä indeksi
- Tietueet
 - indeksirekisterissä tietueen alkuosoite
 - vakiona tietueen kentän suhteellinen osoite tietueen sisällä



5.3.2002

Copyright Teemu Kerola 2002

20

TTK-91 I/O operaatiot

- IN $IN R3, =KBD$
 - lue arvo (kokonaisluku) rekisteriin annetulta laitteelta
- OUT $OUT R2, =CRT$
 - tulosta arvo (kokonaisluku) rekisterin... annetulle laitteelle
- Laitteet?
 - KBD - näppäimistö, stdin
 - CRT - näyttö, stdout
 - ei muita!

5.3.2002

Copyright Teemu Kerola 2002

23

TTK-91 operaatiot

- Muistiinviittaukset
 - tavalliset: load & store
 - pino-operaatiot (aliohjelmien toteuttamista varten)
- I/O käskyt
- Kokonaislukuoperaatiot
- Loogiset operaatiot totuusarvoille
- Bittien siirtokäskyt (shift instructions)
- Kontrollin siirtokäskyt
 - mistä löytyy seuraavaksi suoritettava käsky?
- Muut käskyt

5.3.2002

Copyright Teemu Kerola 2002

21

TTK-91

kokonaislukuoperaatiot

- LOAD (”move”) $LOAD R3, R1 ; R3 \leftarrow R1$
- ADD, SUB $ADD R3, R1 ; R3 \leftarrow R3+R1$
 $SUB R3, =1 ; R3 \leftarrow R3-1$
- MUL $MUL R3, Tbl(R1) ; R3 \leftarrow R3 * Mem(Tbl+R1)$
- DIV, MOD $LOAD R1, =14$
 $DIV R1, =3 ; R1 \leftarrow 4$
 $LOAD R1, =14$
 $MOD R1, =3 ; R1 \leftarrow 2$

5.3.2002


Copyright Teemu Kerola 2002

24

TTK-91

loogiset operaatiot ⁽⁴⁾

- NOT, AND, OR, XOR
 - kaikille 32 bitille
 - yksi bitti kerrallaan



LOAD R1, =12	; R1 = 000...000 1100
LOAD R2, =5	; R2 = 000...000 0101
AND R1,R2	; R1 = 000...000 0100
OR R1,R2	; R1 = 000...000 1101
XOR R1,R2	; R1 = 000...000 1001
NOT R1	; R1 = 111...111 0011

5.3.2002

Copyright Teemu Kerola 2002

25

TTK-91 muut käskyt

- NOP NOP
 - Nop Operation, tyhjä käsky, älä tee mitään
 - varaa kuitenkin muistia yhden sanan (32 bittia)
 - suoritetaan samoin kuin muutkin käskyt

5.3.2002

Copyright Teemu Kerola 2002

28

TTK-91

bittien siirtokäskyt

- SHL, SHR
 - siirrä bittejä vasemmalle tai oikealle
 - täytät nolilla
- LOAD R1,=5 ;R1 = 000...000 0101 = 5
SHL R1,=1 ;R1 = 000...000 01010 = 10
- positiivisilla luvuilla yhden bitin siirto vasemmalle on sama kuin 2:lla kertominen!
– positiivisilla luvuilla yhden bitin siirto oikealle on sama kuin 2:lla jakaminen!

LOAD R1,=5 ;R1 = 000...000 0101 = 5
SHR R1,=1 ;R1 = 000...000 00010 = 2

5.3.2002

Copyright Teemu Kerola 2002

26

TTK-91 assembler

kääntäjän ohjaukset ⁽⁴⁾

- Eivät generoi lainkaan konekäskyjä Sata EQU 100
- EQU - Equal LOAD R1, =Sata
 - antaa arvon symbolille symbolitauluun
- DC - data constant X DC 50
 - varaa yhden sanan tilaa muistista, antaa sille alkuarvon ja antaa osoitteen symbolin arvoksi (symbolitauluun!)
 - esim. muuttujan tai ison vakion määrittely LOAD R1, X
- DS - data segment DS 200
 - varaa monta sanaa tilaa mu Tbl symbolille
 - alkuarvot ovat epämääräisiä!
 - esim. taulukon tai tietueen tilan varaus LOAD R3, Tbl(R1)

5.3.2002

Copyright Teemu Kerola 2002

29

TTK-91

kontrollin siirtokäskyt ⁽⁶⁾

- JUMP JUMP Loop
- COMP COMP R3, =27 COMP R2, X
 - asettaa tilarekisteriin SR vertailun tuloksen: L, E tai G
- JLES, JEQU, JGRE, JNLE, JNEQU, JNGRE
 - perustuu tilarekisterin tietoon eli viimeksi suoritettuun COMP-käskyyn JGRE Loop
- JNEG, JZER, JPOS, JNEG, JNZER, JNPOS
 - perustuu annetun rekisterin arvoon
- CALL, EXIT JPOS R1, Loop
- SVC (käsitellään myöhemmin)
SVC SP, =HALT ;ohjelman suoritus päättyy

5.3.2002

Copyright Teemu Kerola 2002

27

TTK-91 symbolinen konekieliohjelma

```
hello.k91 X DC 13
          Y DC 15

MAIN LOAD R1, X
      ADD R1, Y
      OUT R1, =CRT
      SVC SP, =HALT
```

5.3.2002

Copyright Teemu Kerola 2002

30

TTK-91 symbolinen konekieliohjelma

sum.k91

```
; sum - laske annettuja lukuja yhteen, luku 0 on loppumerkki
Luku DC 0 ; nykyinen luku, alkuarvo 0
Summa DC 0 ; nykyinen summa, alkuarvo 0

Sum IN R1, =KBD ; ohjelma Sum alkaa käskystä 0
STORE R1, Luku
JZER R1, Done ; luvut loppu?

LOAD R1, Summa ; Summa <- Summa+Luku
ADD R1, Luku
STORE R1, Summa ; summa muuttujassa, ei rekisterissä?

JUMP Sum

Done LOAD R1, Summa ; tulosta summa ja lopeta
OUT R1, =CRT
SVC SP, =HALT
```

5.3.2002

Copyright Teemu Kerola 2002

31

-- Luennon 2 loppu --

Some typical 80x86 instructions and their function

Instruction	Function
JE name	If equal (CC) EIP = name; EIP = 128 \$ name < EIP + 128
JMP name	EIP = NAME;
CALL name	SP = SP - 4; M[SP] = EIP + 5; EIP = name;
MOVW EBX,[EDI + 45]	EBX = M [EDI + 45]
PUSH ESI	SP = SP - 4; M[SP] = ESI
POP EDI	EDI = M[SP]; SP = SP + 4
ADD EAX,#6785	EAX = EAX + 6785
TEST EDX,#42	Set condition codes (flags) with EDX & 42
MOVSL	M[EDI] = M[ESI]; EDI = EDI + 4; ESI = ESI + 4

Fig. 3.32 [PaHe98]

5.3.2002

Copyright Teemu Kerola 2002

34

KOKSI

TTK-91 -koneen simulaattori (7)

- Toimii kuten oikea kone toimisi
- Graafinen käyttöliittymä
- I/O vain käyttöliittymän kautta
- Ohjelmien valinta ("lataus"), käännös ja suoritus
- Ohjelmien editointi
 - myös mikä tahansa tekstieditori kelpaa ks. sum.k91
- Käsky kerrallaan suoritus mahdollinen
- Käsky kerrallaan, kommentoinnin kera

5.3.2002

Copyright Teemu Kerola 2002

32

KOKSI

TTK-91 -koneen simulaattori

- Käytettävissä (DOS, W95, W98, W-NT, W2000)
 - laitoksen koneissa
 - kotona <http://www.cs.Helsinki.FI/u/kerola/tito/>
- Installoi itse kotihakemistoosi (n. 120 KB)
 - kopioi zip-tiedosto ja pura se koksi-hakemistoon
 - editoi koksi.cfg tiedostoon editorin polku
- Ohjelmatte. Esim: c:\windows\command\edit.com
 samassa hakemistossa kuin simulaattorin (koksi.exe)
 - käynnistä (esim.) klikkaamalla koksi.exe

5.3.2002

Copyright Teemu Kerola 2002

33