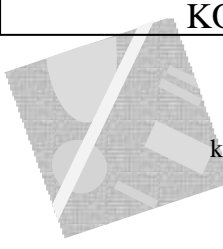


## Jakso 2

### TTK-91 -tietokone ja sen KOKSI -simulaattori

Miksi TTK-91?  
TTK-91:n rakenne ja  
käskykanta-arkkitehtuuri  
KOKSI-simulaattori



7.8.2000 Teemu Kerola, K2000 1

## Miksi konekieltä?

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

7.8.2000 Teemu Kerola, K2000 2

## 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

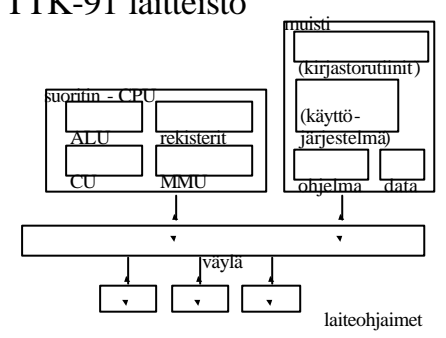
7.8.2000 Teemu Kerola, K2000 3

## Tietokone TTK-91

- 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

7.8.2000 Teemu Kerola, K2000 4

## TTK-91 laitteisto



7.8.2000 Teemu Kerola, K2000 5

## TTK-91 rekisterit

- 8 yleisrekisteriä ks. Kuva 4.1 [Hakk98]
  - vain näitä rekistereitä voi koskettaa (suoraan) konekäskyillä
  - R0 työrekisteri
    - indeksirekisterinä == 0 (tietyissä konekäskyissä R0 käyttö tarkoittaa lukua 0 rekisterin R0 sisällön asemesta)
  - R1-R5 työ- ja indeksirekistereitä
    - tyyppi riippuu konekäskystä
  - pino-osoitin SP (R6) Stack Pointer
  - ympäristöosoitin FP (R7) Frame Pointer

7.8.2000 Teemu Kerola, K2000 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

7.8.2000

Teemu Kerola, K2000

7

## TTK-91 Tilarekisteri SR <sup>(3)</sup>

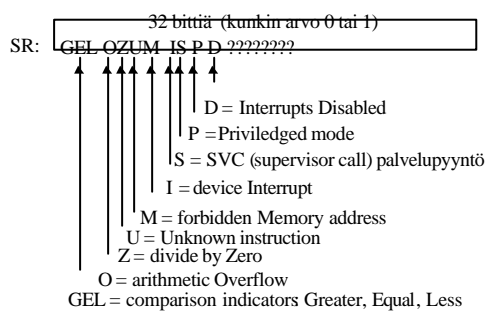
- Tilatietoa siitä, mitä suorittimella tapahtui edellisen käskyn suorituksessa
  - virhetilanteet, poikkeukset ks. Kuva 4.1 [Häkk98]
  - konekäsky olikin käyttäjärjestelmän palvelupyynnö
  - vertailun tulos
- Tilatietoa siitä, mitä systemissä tapahtui viime aikoina
  - käsittelemättömät laitteiden antamat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

7.8.2000

Teemu Kerola, K2000

8

## Tilarekisteri SR <sup>(9)</sup>



7.8.2000

Teemu Kerola, K2000

9

## TTK-91 Muistinhallintayksikkö (MMU)

- Muistiinviittausrekisterit ks. Kuva 4.1 [Häkk98]
  - 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
  - käyttäjärjestelmä asettaa ja valvoo

7.8.2000

Teemu Kerola, K2000

10

## TTK-91 Käskykanta

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

7.8.2000

Teemu Kerola, K2000

11

## TTK-91 tietotyypit <sup>(2)</sup>

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

7.8.2000

Teemu Kerola, K2000

12

### TTK-91 käskytyypit

- Aina 2 operandia itse käskyssä
  - aina ei molemmilla ole merkitystä
    - JUMP vain yksi operandi, Ri+ADDR
    - NOP ei operandeja lainkaan
- Käsky aina 32 bittiä
- Ensimmäinen operandi aina rekisterissä
- Toinen muistissa tai rekisterissä
  - käsittely rekisterissä on nopeampaa kuin muistista hakeminen

7.8.2000 Teemu Kerola, K2000 13

### Symbolinen konekieli <sup>(6)</sup>

**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 = käslyn symbolinen nimi, opcode  
 viite = käslyn (symbolinen) osoite

- Suora vastaavuus konekieleen
  - yksinkertainen assembler-käännös

7.8.2000 Teemu Kerola, K2000 14

### 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 muistisoitteet suhteellisia BASE-osoitteeseen, eli arvoalueella [0, LIMIT - 1]
- Osoitusmoodi: monimutkaisempi vastaavuus
  - konekielessä 3 moodia
  - symbolisessa konekielessä 8 moodia

7.8.2000 Teemu Kerola, K2000 15

### Symbolinen konekieli vs. konekieli <sup>(3)</sup>

7.8.2000 Teemu Kerola, K2000 16

### Operandin osoitusmuodot symbolisessa konekielessä

- 8 eri osoitusmoodia (vain 2. operandi)
- Tekstuaalisesti koodattu LOAD R1, @Field1(R3)
  - 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
    - indeksirekisteri R0 tai vakio 0

7.8.2000 Teemu Kerola, K2000 17

### TTK-91 muistin osoitusmuodit

ks. lista sivulla 50 [Hakkus]

R0:	104
R1:	10
R2:	
R3:	1
...	

SP=R6: 125  
 FP=R7: 125

200:	
201:	11

LIMIT: 300

symboli:	
Tbl:	200
X:	10
One:	1

```

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
    
```

7.8.2000 Teemu Kerola, K2000 18

### Indeksointi <sup>(2)</sup>

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 ← 201
  - tyhjä: 1 kpl R4 ← Mem[201] = 11
  - "@": 2 kpl R4 ← Mem[ Mem[201] ]  
= Mem[ 11 ] = 300

pelkkä rekisterin nro @-merkin jälkeen ⇒ 1 kpl  
STORE käsky ⇒ 1 kpl vähemmän noutoja

7.8.2000 Teemu Kerola, K2000 19

### Indeksoinnin käyttö <sup>(2)</sup>

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

LOAD R5, Tbl(R3)

LOAD R2, Salary(R5)

7.8.2000 Teemu Kerola, K2000 20

### TTK-91 operaatiot

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

7.8.2000 Teemu Kerola, K2000 21

### TTK-91 muistiinviittausoperaatiot

- LOAD
  - vain silloin kun viitataan muistiin
- STORE
- PUSH, POP, PUSHR, POPR
  - aliohjelmien toteuttamista varten
  - käsitellään myöhemmin

LOAD R1, X

LOAD R5, @tblX

STORE R2, X

STORE R3, Tbl(R4)

POP SP, R1 ; load ...

PUSH SP, R1 ; store ...

7.8.2000 Teemu Kerola, K2000 22

### TTK-91 I/O operaatiot

- IN
  - lue arvo (positiivinen kokonaisluku) rekisteriin annetulta laitteelta
- OUT
  - tulosta arvo (kokon. luku) rekisteristä annetulle laitteelle
- Laitteet?
  - KBD - näppäimistö, stdin
  - CRT - näyttö, stdout
  - ei muita!

IN R3, =KBD

OUT R2, =CRT

7.8.2000 Teemu Kerola, K2000 23

### TTK-91 kokonaislukuoperaatiot

- LOAD
- ADD, SUB
- MUL
- DIV, MOD

LOAD R3, R1 ; R3 ← R1

ADD R3, R1 ; R3 ← R3+R1

SUB R3, =1 ; R3 ← R3-1

MUL R3, Tbl(R1) ; R3 ← R3 \* Mem(Tbl+R1)

LOAD R1, =14

DIV R1, =3 ; R1 ← 4

LOAD R1, =14

MOD R1, =3 ; R1 ← 2

7.8.2000 Teemu Kerola, K2000 24

### TTK-91 loogiset operaatiot (4)

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

LOAD R1, =13 ; R1 = 000...000 1101

LOAD R2, =5 ; R2 = 000...000 0101

AND R1,R2	; R1 = 000...000 0101
OR R1,R2	; R1 = 000...000 1101
XOR R1,R2	; R1 = 000...000 1000
NOT R1	; R1 = 111...111 0010

7.8.2000
Teemu Kerola, K2000
25

### TTK-91 bittien siirtokäskyt

- SHL, SHR
  - siirrä bittijä vasemmalle tai oikealle
  - täytä nolilla

LOAD R1,=5 ;R1 = 000...000 00101 = 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 00101 = 5

\$SHR R1,=1 ;R1 = 000...000 00010 = 2

7.8.2000
Teemu Kerola, K2000
26

### TTK-91 kontrollin siirtokäskyt (6)

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

7.8.2000
Teemu Kerola, K2000
27

### TTK-91 muut käskyt

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

7.8.2000
Teemu Kerola, K2000
28

### TTK-91 assembler kääntäjän ohjaukset

- Eivät generoi lainkaan konekäskyjä Sata EQU 100
- EQU - Equal LOADR1,=Sata
  - antaa arvon symbolille symbolitauluun
- DC - data constant X DC 50
  - varaa yhden sanan tilaa muistista ja antaa sille arvon symbolille (symbolitauluun!) LOADR1,X
  - esim. muuttujan tai ison vakion määrittely
- DS - data segment Tb1 DS 200
  - varaa monia sanoja tilaa muistista, antaa arvon symbolille
  - alkuarvot ovat epämääräisiä! LOADR3,Tb1(R1)
  - Esim. taulukon tilan varaus

7.8.2000
Teemu Kerola, K2000
29

### TTK-91 symbolisia konekäskyesimerkkejä (10)

- Miten toimivat seuraavat käskyt?

```
LOAD R2, @100 ;R2 ← 200
ADD R2, 101 (R3) ;R2 ← R2 + 100 = 105
DIV R1, R3 ;R1 ← 0
LOAD R2, =100(R0) ;R2 ← 100
LOAD R0, @101(R3) ;R0 ← 101
```

regs		mem	
R0:	2	100:	101
R1:	1	101:	200
R2:	5	102:	101
R3:	2	103:	100

7.8.2000
Teemu Kerola, K2000
30

### TTK-91 symbolisia konekäskyesimerkkejä <sup>(10)</sup>

- Entä miten toimivat seuraavat käskyt?

```

LOAD R2, @Xptrptr ;R2 ← 200
ADD R2, Xptr (R3) ;R2 ← R2 +100 = 105
DIV R2, R3 ;R2 ← 2
LOAD R2, =Tbl(R1) ;R2 ← 101
LOAD R2, Sum(R4) ;R2 ← 101
  
```

	regs		mem		symbol
R1:	1	100:	101	Tbl =	100
R2:	5	101:	200	X =	200
R3:	2	102:	101	Xptr =	101
R4:	100	103:	100	Xptrptr =	100
				Sum =	2

7.8.2000 Teemu Kerola, K2000 31

### 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
  
```

7.8.2000 Teemu Kerola, K2000 32

### TTK-91 symbolinen konekieliohjelma

```

sum.k91
; sum - laske annetuja 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 rekisterissa?

     JUMP Sum

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

7.8.2000 Teemu Kerola, K2000 33

### KOKSI

#### TTK-91 -koneen simulaattori <sup>(7)</sup>

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

7.8.2000 Teemu Kerola, K2000 34

### KOKSI

#### TTK-91 -koneen simulaattori

- Käytettävissä (DOS, W95, W98, W-NT)
  - laitoksen koneis http://www.cs.Helsinki.FI/~kerola/
  - kotona
- Installoi itse kotihakemistoosi (n. 120 KB)
  - kopioi zip-tiedosto ja pura se koksi-hakemistoon
  - editoi kok C:\windows\command\edit.com
- Ohjelmatiedostojen (hello.k91 jne) tulee olla samassa hakemistossa kuin simulaattorin (koksi.exe)
  - käynnistä (esim.) klikkaamalla koksi.exe

7.8.2000 Teemu Kerola, K2000 35

### -- Jakson 2 loppu --

Some typical 80x86 instructions and their function

Instruction	Function
JM name	IF equal (ZF); EIP = name(); EIP = 229 * 0x00 + EIP + 12h
JMP name	EIP = name();
CALL name	SP = SP - 4; HI(PI) = EIP + 3; EIP = name();
MOVW REG, (REG + 4)	REG = R (REG + 4);
PUSH REG	SP = SP - 4; HI(PI) = REG;
POP REG	REG = HI(PI); SP = SP + 4;
ADD EAX, #75	EAX = EAX + 75h;
TEST REG, #12	Set condition codes (Zflag) with REG & 12;
MOVSI	HI(ESI) = HI(EDI); EHI = EHI + 4; ESI = ESI + 4;

Fig. 3.32 [PaHe98]

7.8.2000 Teemu Kerola, K2000 36