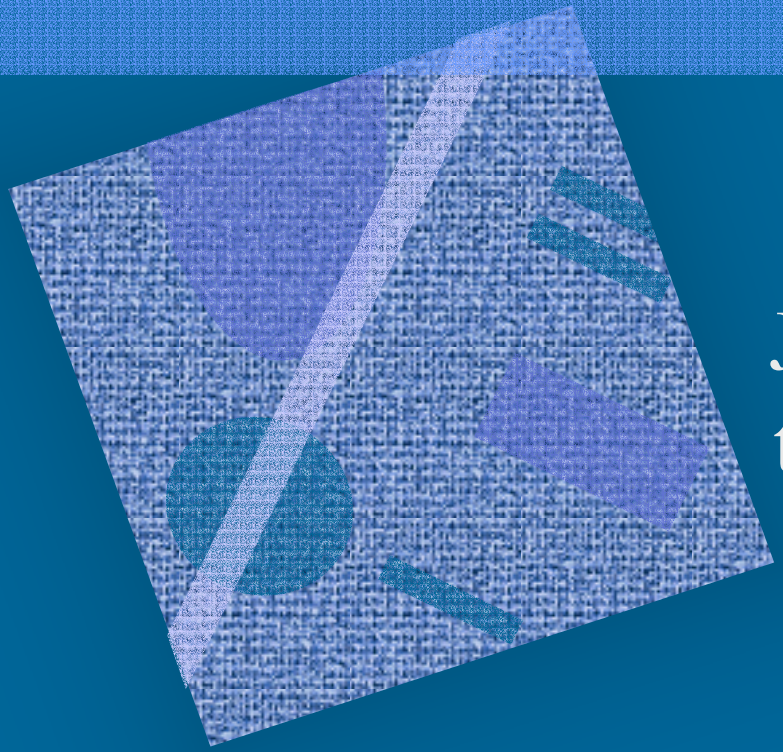
Jakso 11

Tulkinta ja emulointi

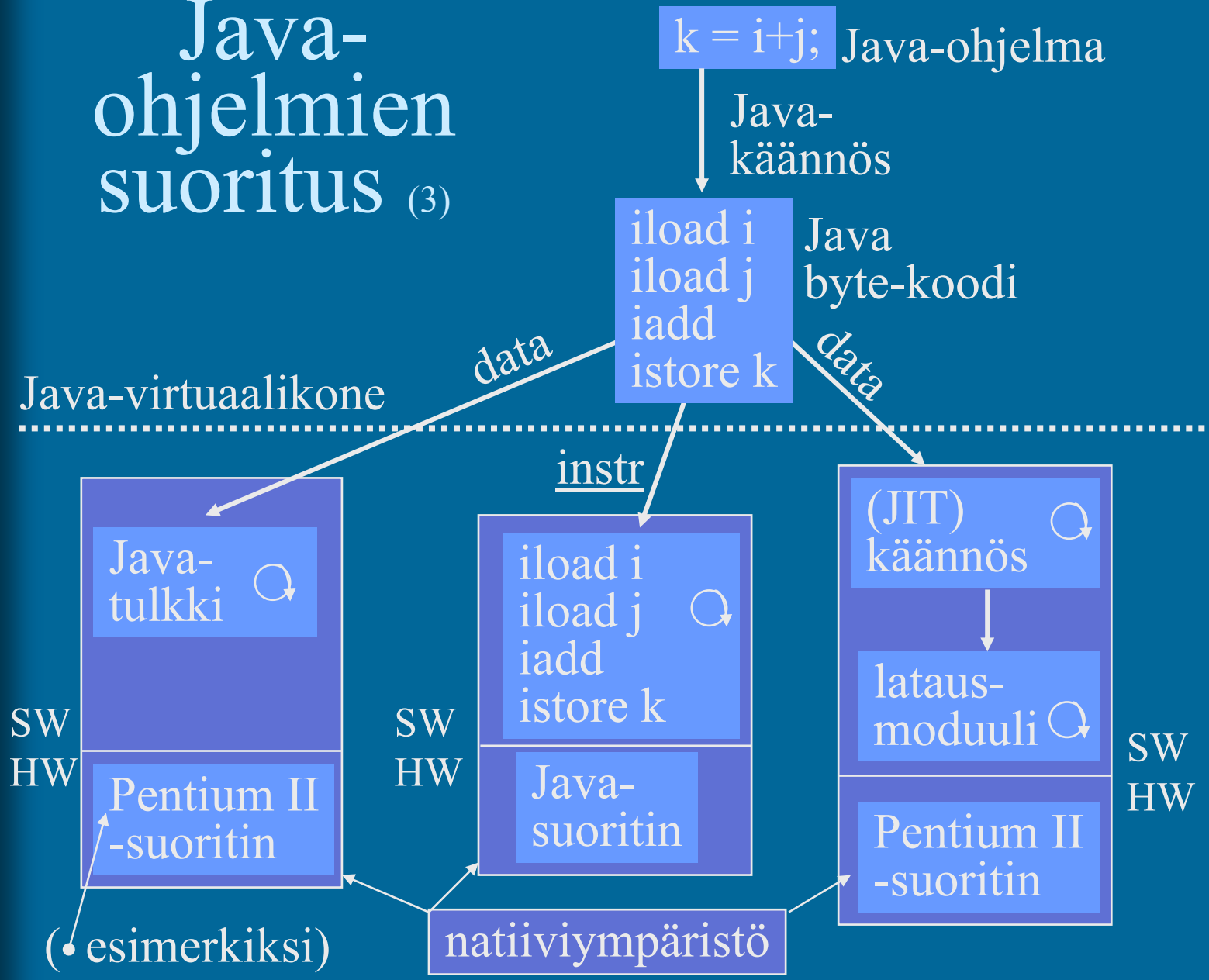


Tulkinta ja emulointi

Java-ohjelman suoritus, tulkinta ja kääntäminen

Suorittimen emulointi: ttk-91, Crusoe

Java-ohjelmien suoritus (3)



Java-virtuaalikone (JVM) (5)

- Hypoteettinen suoritin, toteutus eri tavoilla
- Geneerinen, sitä on ”helppo” simuloida kaikilla todellisilla suorittimilla
- Useita säikeitä (thread) voi olla ’samanaikaisesti’ suorituksessa
  - suorittimella mikroaikaskaalassa vain yksi kerrallaan
- Tietorakenteet
  - mm. virtuaalikoneen suorittimen ”rekisterit”
  - luodaan JVM:n käynnistämisen yhteydessä
- Käskyt
  - virtuaalikoneen suorittimen konekäskyt
  - 226 käskyä á 32 bittiä

JVM:n tietorakenteet (8)

- JVM-pino

ks. Fig. 4-10 (Tane99)

  - vähän kuten tavallinen AT-pino
  - koostuu useista *kehyksistä* (frames) (vrt. aktivointitietue) ja operandipinosta
  - käyttö: kehyksille ainoastaan push/pop-operaatiot, operandipinon alkioille myös push/pop
  - ei tarvita yhtenäistä muistialuetta
  - allokoidaan keosta (heap)
  - toteutuksesta riippuen rajallinen tai dynaamisesti laajennettavissa
  - tila loppu ⇒ StackOverflowError, OutOfMemoryError

http://java.sun.com/docs/books/vmspec/2nd-edition/html/VMSpecTOC.doc.html

JVM:n tietorakenteet (jatkuu) (6)

- JVM-keko (JVM heap)
  - yhteinen kaikille saman virtuaalikoneen säikeille
  - automaattinen roskienkeruu (garbage collector)
    - ei-käytössä oleva (eli vapautettu) muistialue palautetaan uusiokäyttöön (vapaaksi)
    - ei tarvita erikseen *free*-operaatiota Java-ohjelmassa
    - voi hidastaa suoritusta milloin vain
  - toteutuksesta riippuen kiinteän kokoinen tai dynaamisesti laajennettavissa
  - ei tarvitse muodostaa yhtenäistä muistialuetta natiivijärjestelmän kasassa
  - tila loppu $\Rightarrow$ OutOfMemoryError

JVM:n tietorakenteet (jatkuu) (6)

ks. Fig. 4-10 [Tane99]

- JVM-metodialue (JVM Method Area)
  - yhteinen kaikille JVM-säikeille
  - vastaa tavallista kääntäjän tuottamaa koodisegmenttiä
  - loogisesti osa JVM-kekoa
  - toteutuksesta riippuen kiinteän kokoinen tai dynaamisesti laajennettavissa
  - tila loppu ⇒ OutOfMemoryError

JVM:n tietorakenteet (jatkuu) (7)

ks. Fig. 4-10 [Tane99]

- Javan suoritusaikainen vakioallas (runtime constant pool)
  - joka luokalle (class) tai liittymälle (interface) omansa
  - suoritusaikainen esitystapa tiedoston *class* *constant_pool* -taulukolle
  - vastaa vähän tavallista symbolitaulua
  - useita erilaisia vakioita (käännösaikaiset literaalit, suoritusaikana ratkottavat attribuutit, …)
  - talletetaan JVM-metodialueelle
  - tila loppu ⇒ OutOfMemoryError

JVM:n tietorakenteet (jatkuu) (6)

- Natiivimetodien pinot (Native Method Stacks)
  - toteutus voi käyttää tavallisia pinoja (”C stacks”) sellaisten natiivimetodien tukena, jota ei ole kirjoitettu Javalla
  - käytetään myös Java-tulkin toteutuksessa
  - ei tarvita JVM-toteutuksissa, joissa ei ole natiivimetodeja
  - toteutuksesta riippuen kiinteän kokoinen tai dynaamisesti laajennettavissa
  - tila loppu $\Rightarrow$ StackOverflowError, OutOfMemoryError

JVM:n tietorakenteet (jatkuu) (6)

ks. Fig. 4-10 [Tane99]

- JVM-rekisterit
  - PC osoittaa johonkin JVM-metodialueelle
  - CPP osoittaa vakioaltaaseen
  - LV on paikallisten muuttujien kantaosoite (vähän kuten FP ttk-91:ssä)
  - SP osoittaa JVM-operandipinon huipulle
  - kaikki rekisterit implisiittisiä; niitä ei erikseen nimetä JVM -konekäskyissä

JVM:n tietorakenteet (jatkuu) (6)

ks. Fig. 4-10 [Tane99]

- JVM- kehys (frame, raami)
  - talletetaan JVM-pinoon, luodaan metodin kutsun yhteydessä, vapautetaan metodista poistuttaessa
  - paikalliset muuttujat
  - parametrit, paluuarvon ja välitulokset
  - dynaamisen linkityksen toteutusväline
  - keskeytysten toteutusväline

JVM-kehyksen data (8)

- Paikalliset muuttujat sisältävä taulukko

ks. Fig. 4-10 [Tane99]

  - viittaukset indeksoituna (0, 1, 2, …) rekisterin LV suhteen
  - indeksit sanoina
  - kaksi sanaa vaativa muuttuja (long, double) sijoitetaan kahteen peräkkäiseen (32 bittiseen) sanaan
  - big-endian talletus
- Parametrit, paluuarvon ja välitulokset sisältävä operandipino
  - SP osoittaa pinon huipulle
  - pinoarkkitehtuuri (vs. rekisteriarkkitehtuuri)

JVM:n tiedon osoitusmoodit (4)

- Välitön operandi

iINC 2 34

- Indeksoitu

iINC 2 34

tehollinen muistiosoite (LV) + 2

- Pino-osoitus

iADD

korvaa pinon kaksi päällä olevaa kokonaislukua niiden summalla

ks. Fig. 4-10 [Tane99]

- Taulukko-osoitus pinon kautta

iALOAD

Korvaa pinon pinnalla olevat taulukon alkuosoite ja indeksi k.o. taulukon alkiolla

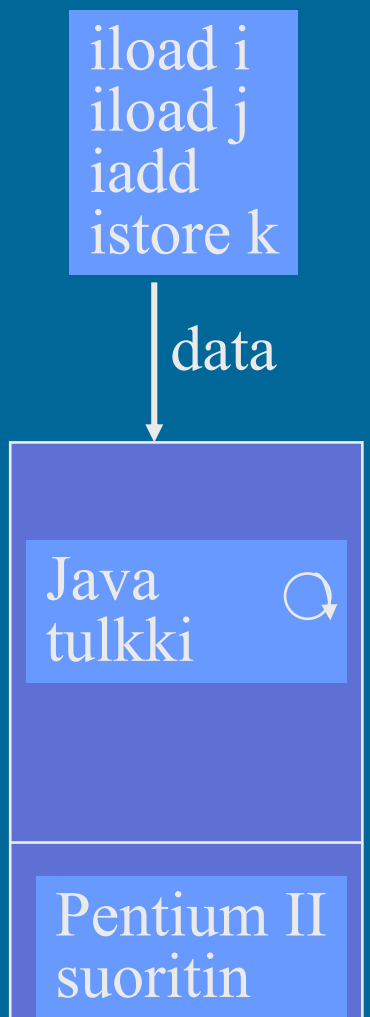
JVM-käskyt (7)

ks. Fig. 5-36 [Tane99]

- Peruslaskutoimitukset
  - add, sub, mul, div, rem, neg
- Boolean
  - and, or, xor, shl, shr, ushr
- Pinon hallinta
  - dup, pop, swap, tauluk. luonti, esitystavan muutokset
- Load/Store
  - load, aload, store, astore, push-käskyt
- Vertailut
- Kontrollinsiirrot
- Muut

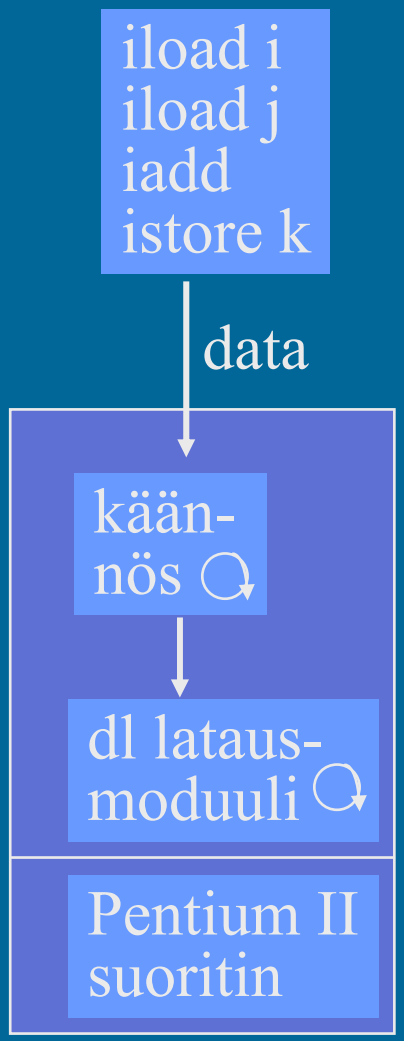
Java-tulkki (4)

- Emuloi JVM-konekielen käskyjä (byte-koodia)
- Yksi käsky kerrallaan
- JVM-rekisterit ja muistialueet emuloitu tulkin tietorakenteina
- Hidasta, mutta joustavaa



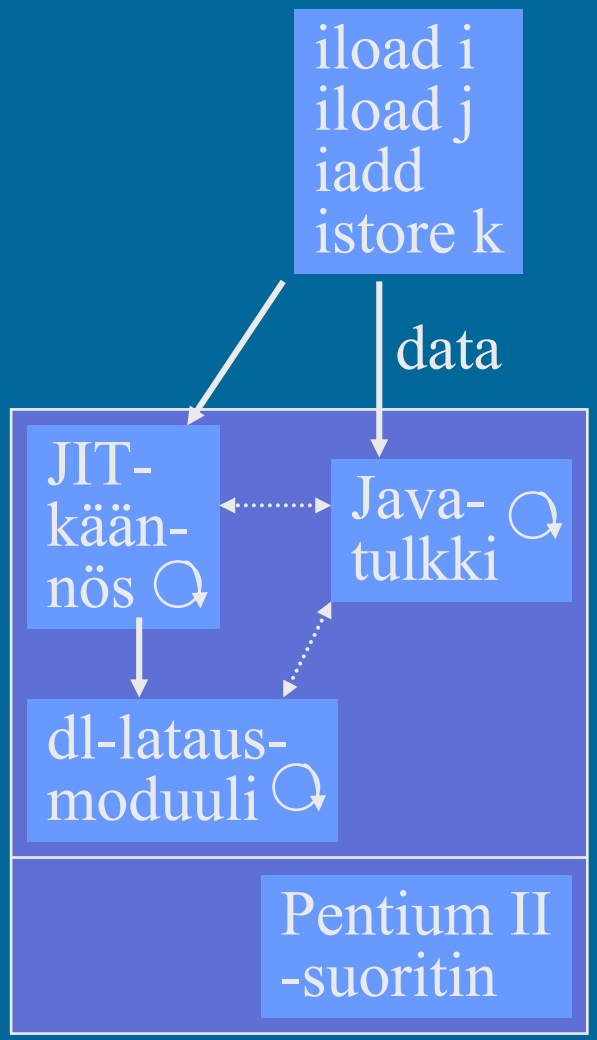
Käännös natiivikoneelle (3)

- (a) Käännetään tavukoodi natiivikoneen konekielelle, ja suoritetaan normaalin ohjelman tapaan
- (b)Käännetään tavukoodi korkean tason kielelle, joka sitten käännetään natiivikoneen konekielelle
  - alkuosa riitää tehdä kerran
  - loppuosa on jo valmiina yleensä
- ongelma: dynaaminen linkitys



Java JIT -käännös (6)

- JIT = Just-in-Time
- Emulointi ja/tai natiivikoneen suoritus tilanteesta riippuen
- Kääntää luokan natiivikonekielelle dynaamisesti linkitettäväksi moduuliksi, juuri ennen luokan metodin kutsua
- Tarvitsee paljon muistia
- Voi hidastaa suoritusta, jos käännökseen menee enemmän aikaa kuin tulkitsemiseen (käännös vasta 2. kutsukerralla?)
- JVM-rekisterit ja muistialueet emuloitu tulkin tietorakenteina, joita natiivikoodi myös käyttää

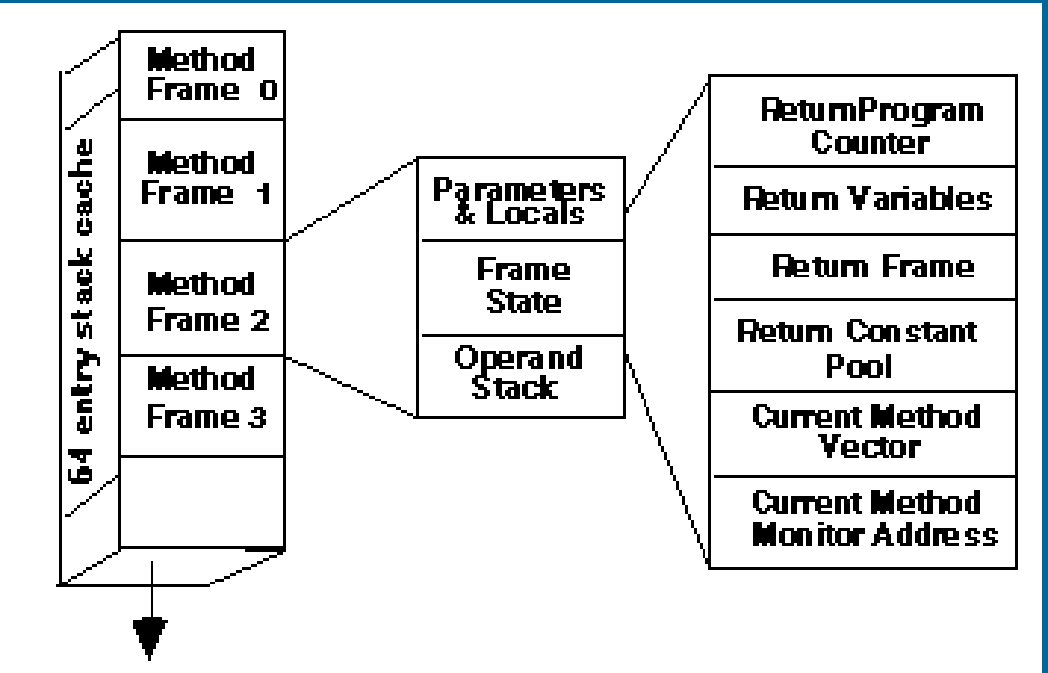


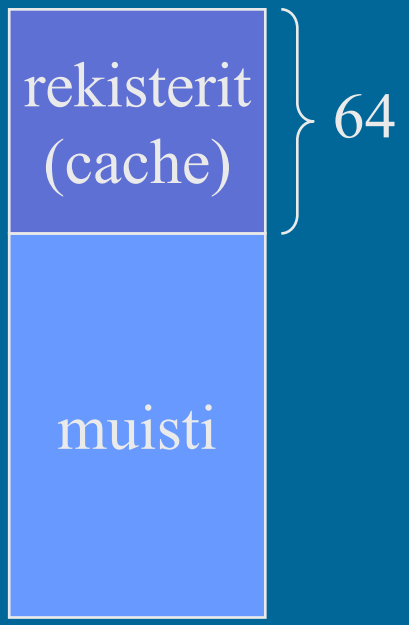
Java-suoritin: Sun PicoJAVA II (4)

- Suorittimen määrittely, jonka mukaisessa koneessa byte-koodi -muodossa olevia ohjelmia voidaan sellaisenaan suorittaa
- Valinnainen välimuisti ja liukulukusuoritin
- Kaikki 226 JVM-konekäskyä
  - jotkut käskyt toteutettu aliohjelmilla, jotka aktivoidaan keskeytyskäsittelymekanismin avulla
- Myös 115 muuta konekäskyä käyttöjärjestelmän j muiden ohjelmointikielten toteuttamiseksi

PicoJAVA II -pino

- 64 (välimuisti-) rekisteriä JVM-pinon huipun talletukseen
  - loput JVM-pinosta muistissa





Shawn Lauzon, Survey of the JavaChip

PicoJAVA II -rekisterit (11)

- 25 rekisteriä á 32 bittiä
  - PC, LV, CPP, SP (pino kasvaa alaspäin)
  - OPLIM alaraja SP:lle; alitus aiheuttaa keskeytyksen
  - FRAME osoittaa paikallisten muuttujataulukon jälkeen talletettuun metodista paluu osoitteeseen
  - PSW
  - rekisteri, joka kertoo pinon välimuistirekistereiden tämänhetkisen käytön
  - 4 rekisteriä keskeytysten ja break-point'ien käsittelyyn
  - 4 rekisteriä säikeiden hallintaan
  - 4 rekisteriä C ja C++ ohjelmien toteutukseen
  - 2 rajarekisteriä sallitun muistialueen rajoittamiseen
  - suorittimen version numero ja konfiguraatiorekisterit

PicoJAVA:n ylim. käskyt (5)

- Read/write ylimääräisille rekistereille
- Osoittimien manipulointikäskyt
  - mitä tahansa muistialuetta voidaan suoraan lukea/kirjoittaa
  - tarvitaan C/C++ varten
- C/C++ -aliohjelmien kutsu ja paluukäskyt
- Natiivi HW-manipulointi
  - tyhjennä välimuisti (osittain? kokonaan?), ...
- Muut käskyt
  - power on/off, ...

PicoJAVA-toteutuksia (2)

- Sun microJAVA 701
  - valinnainen välimuisti
  - oma muistiväylä
  - PCI-väylä muille laitteille
  - 16 ohjelmoitavaa I/O-johdinta
    - näppäimet, LEDit, …
  - 3 ohjelmoitavaa ajastinta ($\Rightarrow$ kellolaitekeskeytykset)
  - suunnattu halpoihin kannettaviin laitteisiin (kämmenmikro, PDA - Personal Digital Assistant)
- Sun ultraJAVA
  - nopeampi, parempi, kalliimpi, …
  - suunnattu grafiikka- ja multimediasovelluksiin

Muita Java-suorittimia

- JEM (Rockwell Collins)
- PSC1000 (Patriot Scientific)
  - dSys (Saksa), lääketieteellisiä laitteita
- MJ501 (LG Semicon)
  - TV, älykortit
- JSR001, Real-Time Specification for Java (Java Community Process, ”Sun Microsystems”)
  - aJile: aJ-80, aJ-100, älykkäät liikkuvat laitteet

Sun MAJC

- MAJC - Microprocessor Architecture for Java Computing
  - suoritinarkkitehtuurin määrittely
  - tavoitteena suuri nopeus Java-, C- ja C++-sovelluksille
  - suunnattu multimediasovelluksiin verkossa
  - tukee hyvin JIT-käännöstä

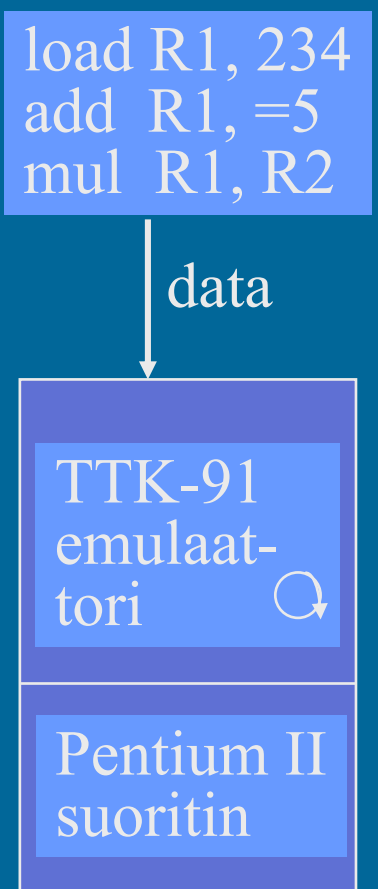
MAJC toteutus: MAJC 5200 (4)

- 1-4 suoritinta (2 suorittimen lastu v.1999)
- Useiden (peräkkäin kutsuttavien) metodien samanaikainen suoritus eri suorittimilla
  - ennakoiva (speculative) suoritukselle oma kasa
  - peruutus (rollback), jos ennakoitu suoritus meni pieleen
- 4 säiettä suorituksessa per suoritin
  - säikeen vaihto nopeampaa kuin muistista luku!
  - laiterekisterit 4:lle säikeelle!
  - välimuistin hudin aikana suoritetaan muita säikeitä
- Suunnattu interaktiiviseen TV:hen, virtuaalitodellisuussovelluksiin, ...

TTK-91-emulointi (5)

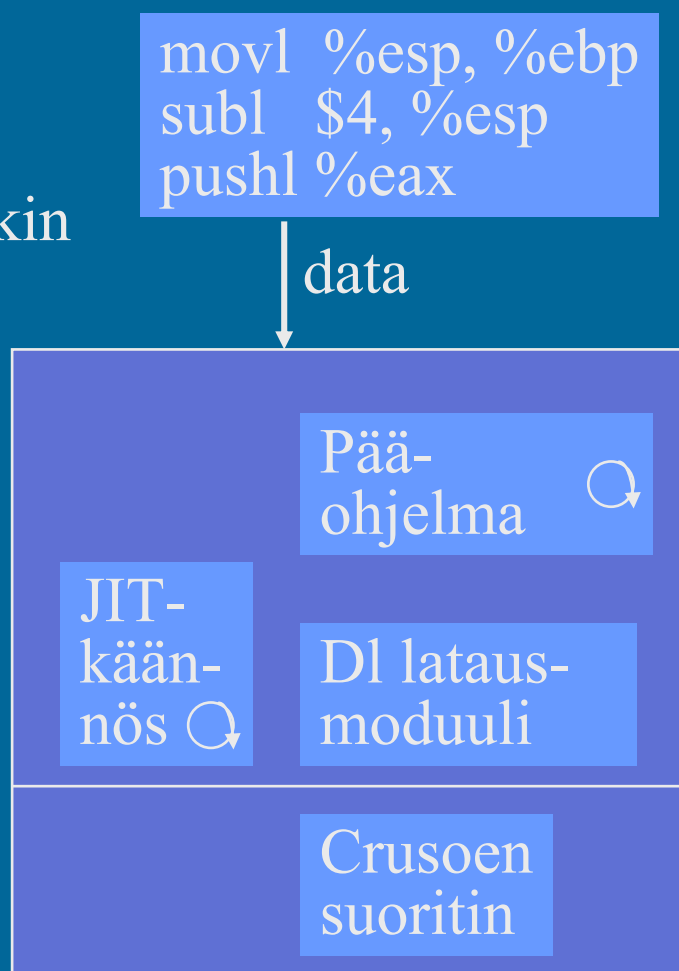
- TTK-91-konekielen emulointi
- KOKSI-simulaattorin osa
- Yksi käsky kerrallaan
- TTK-91-koneen rekisterit ja muisti emuloitu tulkin tietorakenteina

ks. simulaattorin koodi

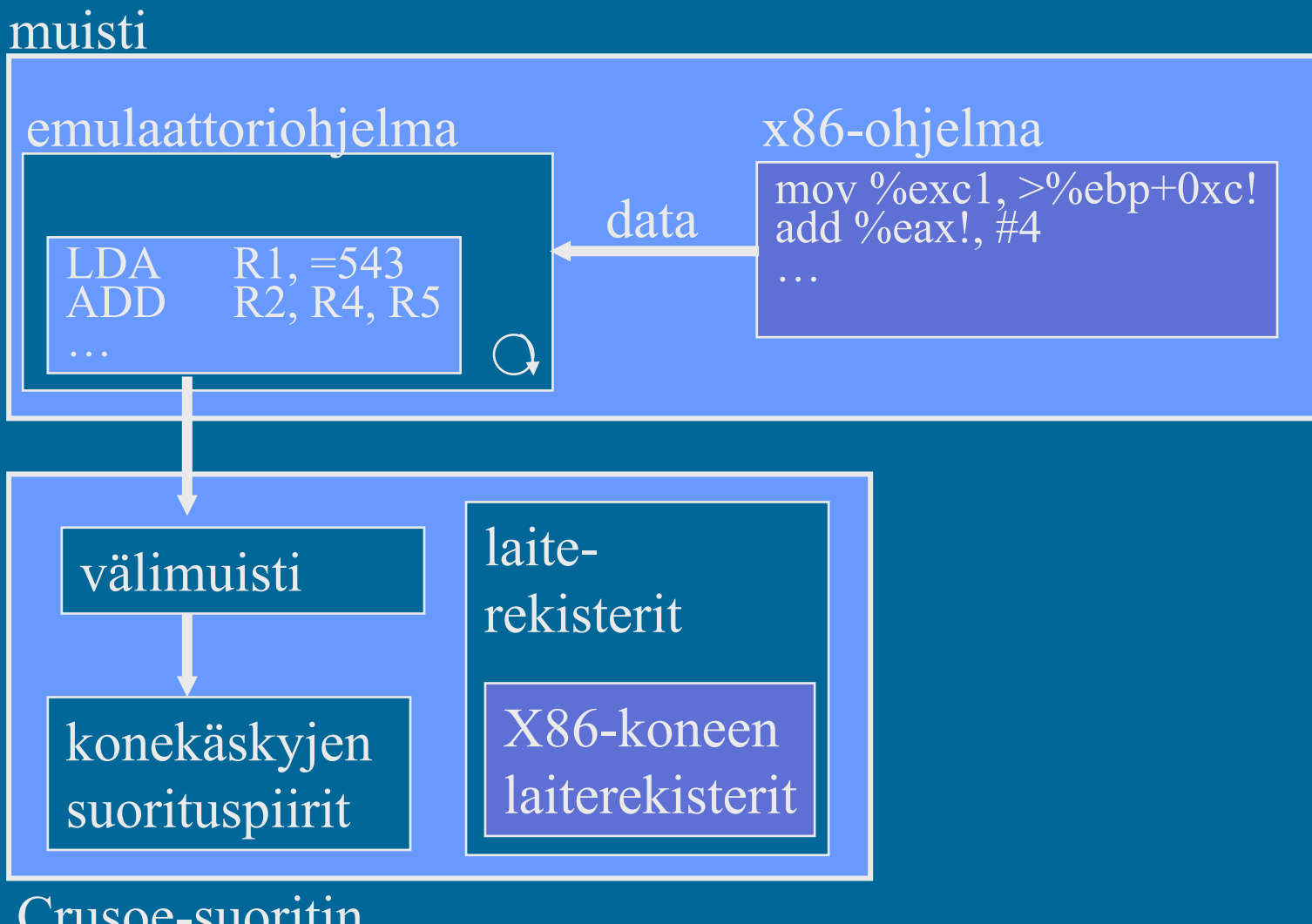


Transmetan Crusoe-suoritin (8)

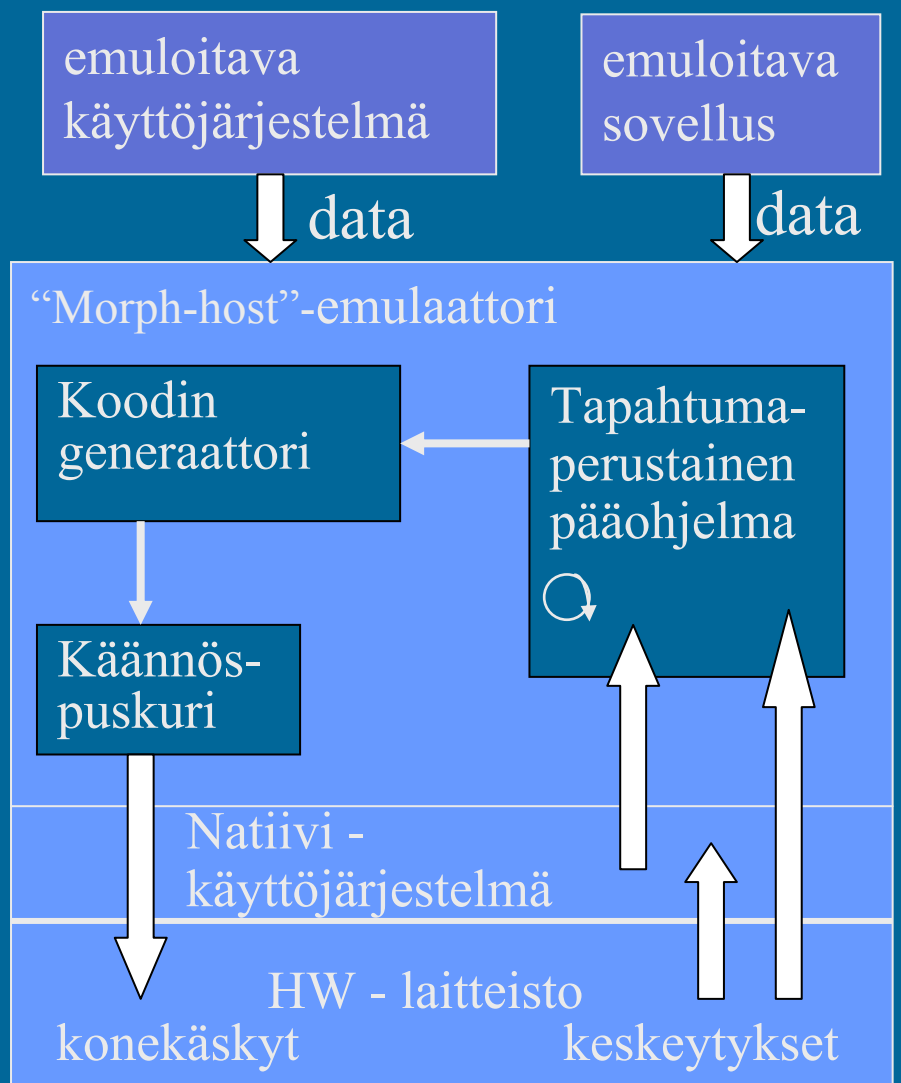
- x86-konekielen emulointi, JIT-käännös
- Natiivikäskykantaa ei ole julkistettu
- ”nopeampi, sama teknologia”?
- ”yhtä nopea, vähemmän virtaa”
- Monta x86-käskyä yhtäaikaa, sikin sokin emuloinnissa
- x86-rekisterit emuloitu natiivi-järjestelmän laiterekistereillä
- x86-muisti emuloitu suojattuna tietorakenteina
- Tarkat keskeytykset:
  - suorituksen peruutus
  - uusi, hidas JIT-käännös
  - hidas mutta tarkka emulointi



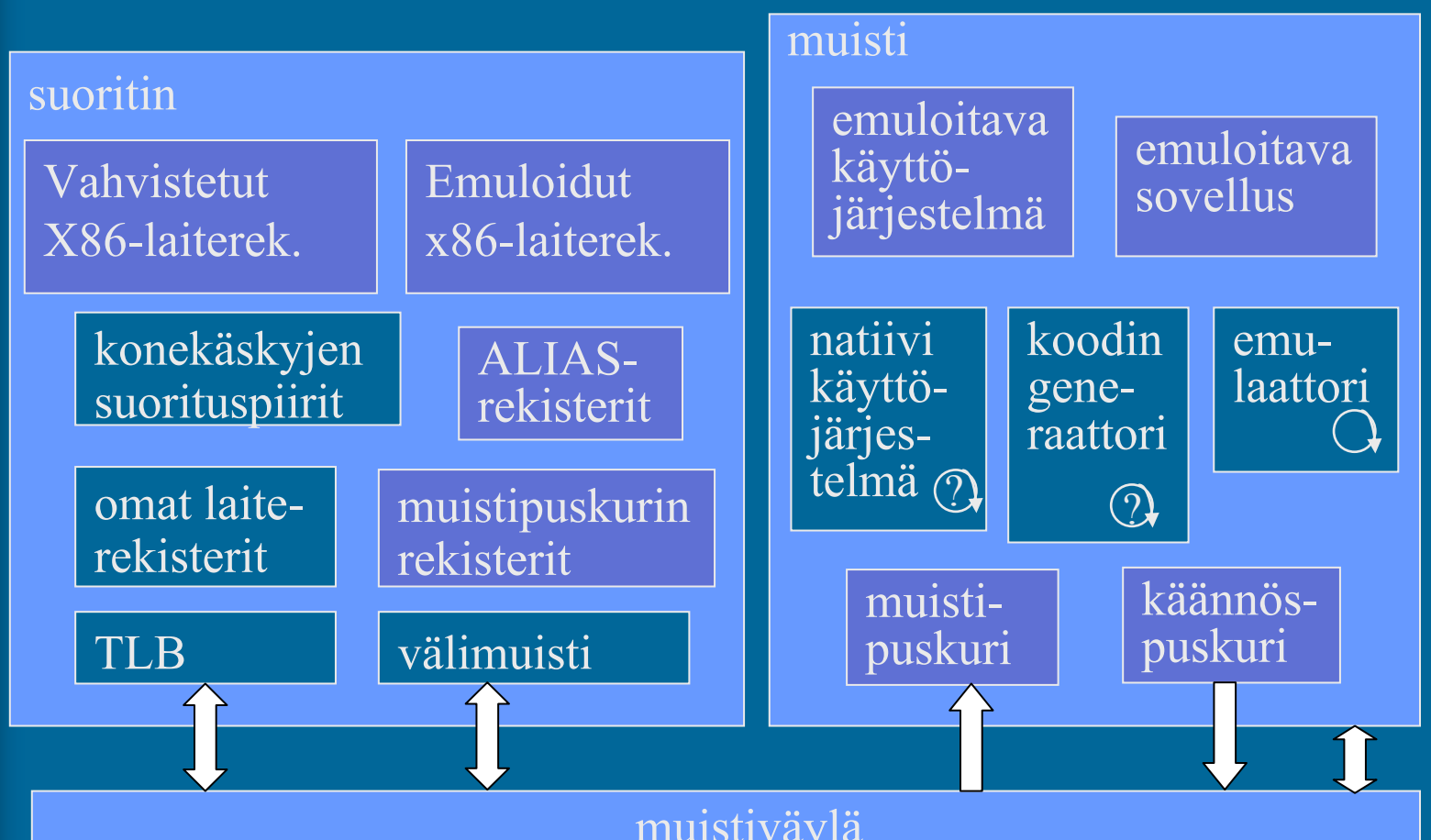
Crusoe-emulaattorin suoritus



Crusoe-suorittimen looginen rakenne



Crusoe-suorittimen fyysinen rakenne



-- Jakson 11 loppu --