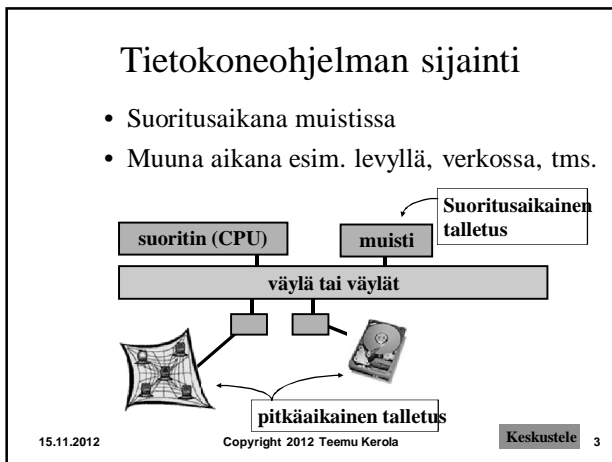
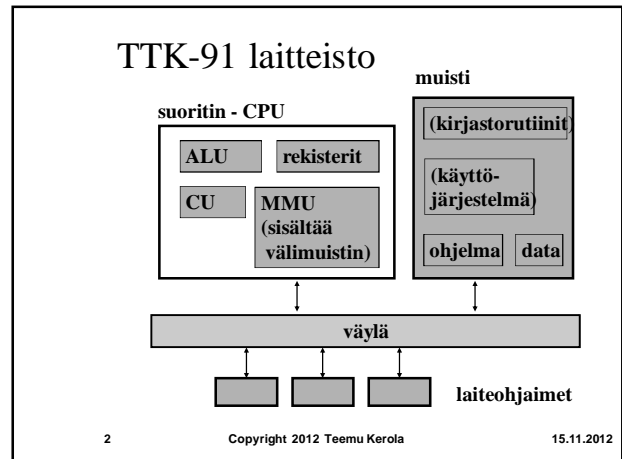


Luento 11
Kertaus ja yhteenveto



Nopeuserot: Teemun juustokakku

Rekisterien, välimuistin, muistin, levymuistin ja magneettinauhan nopeudet suhteutettuna juuston haku aikaan juustokakkuu tehdessä?

Europa (Jupiter)

The diagram compares speeds using a 'cheese cake' analogy. A stick figure represents the **käsi** (hand) with a speed of **0.5 sek (rekisteri)**. A desk represents the **pöytä** (desk) with a speed of **1 sek (välimuisti)**. A refrigerator represents the **jääkaappi** (refrigerator) with a speed of **10 sek (muisti)**. The moon represents **kuu** (moon) with a speed of **12 pv (levy)**. Europa represents **Europa (Jupiter)** with a speed of **4 v (verkko, ihminen)**.

2008: 0.5 ns? 10 ns? 4 ms? (50 pv?) 1 s? (n 65 v?)

15.11.2012 Copyright 2012 Teemu Kerola 4

Ohjelman esitysmuoto: symbolinen konekieli

- Usein symbolisella konekielellä
 - käsky jaettu osiin (kenttiin)
 - joidenkin kenttien arvot kuvattu symboleilla
 - helpompi ihmisten lukea ja kirjoittaa

Symb. konekieli	Konekielinen käsky
LOAD R2, =100	0000 0010 000 00 010 0000 0000 0110 0100
LOAD R1, 100	0000 0010 001 01 000 0000 0000 0110 0100
DIV R1, R2	0001 0100 001 00 010 0000 0000 0000 0000
JZER 6	0010 0010 000 00 000 0000 0000 0000 0110
STORE R1, 228	0000 0001 001 00 000 0000 0000 1110 0100
NOP	0000 0000 000 00 000 0000 0000 0000 0000

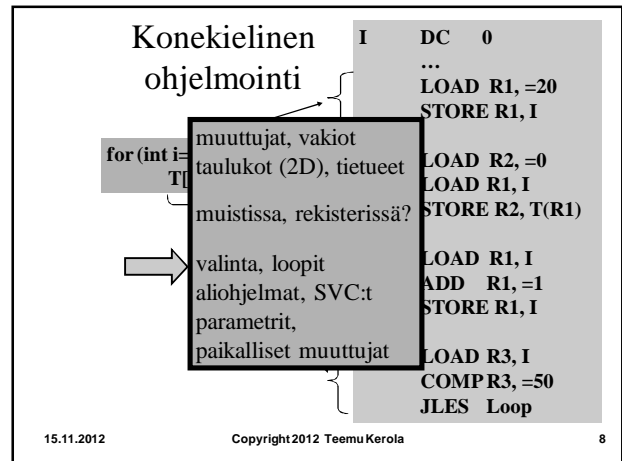
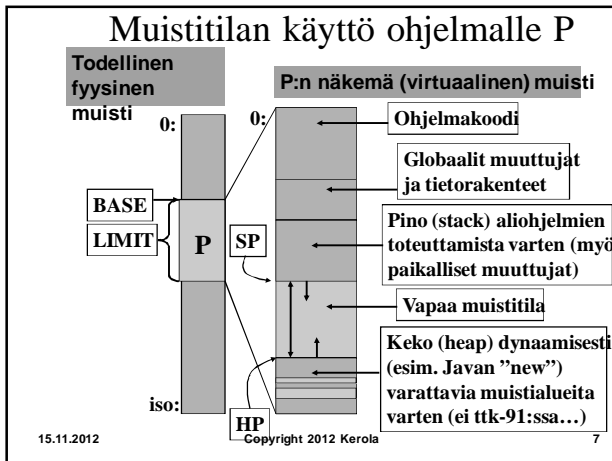
15.11.2012 Copyright 2012 Kerola 5

Tiedon sijainti suoritusajana

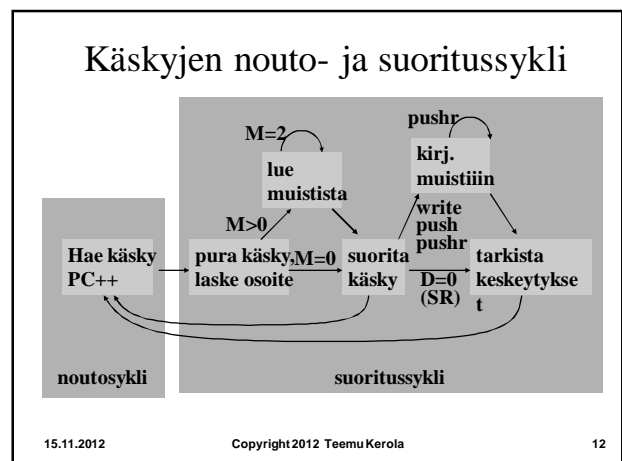
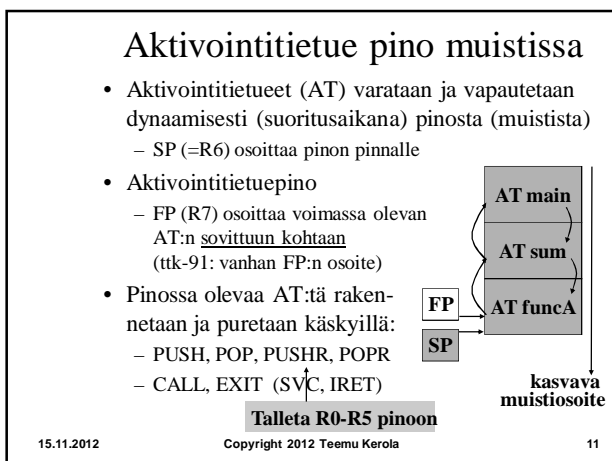
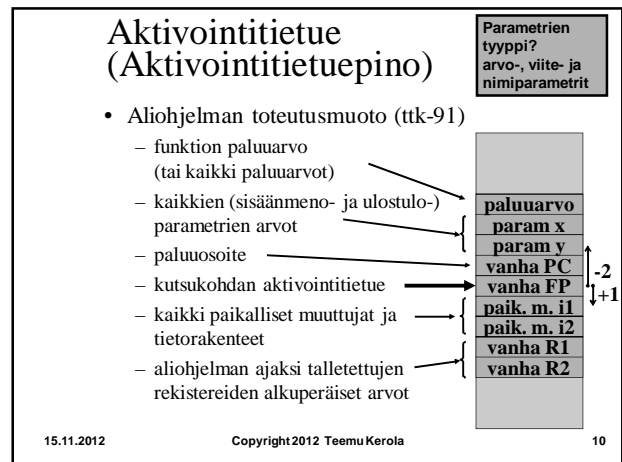
Miten tietoon viitataan eri paikoissa?

- Rekisteri (nopein)
 - kääntäjä päättää milloin muuttujan arvo on rekisterissä
- Välimuisti (nopea)
 - laitteisto hoitaa automaattisesti joillekin muistialueille
- Muisti (hidas)
 - kääntäjä/lataaja valitsee sijaintipaikan
 - globaali data ohjelman latauksen yhteydessä
 - vakiot konekäskyssä
 - ohjelma sijoittaa suoritusajana
 - aliohjelmien paikalliset muuttujat, parametrit
 - käyttöjärjestelmä sijoittaa suoritusajana
 - dynaaminen data keossa suorituksen aikana
- Levy, levypalvelin (liian hidas, ei mahdollista)
 - vaatii käyttöjärjestelmän varusohjelmien apua

15.11.2012 Copyright 2012 Kerola 6



- ### Aliohjelmat, funktiot
- Parametrien ja paluuarvon tyytit
 - Parametrien ja paluuarvon välitys
 - Paikallisten muuttujien käyttö
 - Rekistereiden talletus ja arvon palautus
 - Kutsun ja paluun toteutus
 - Aktivointitietue, AT-pino
- 15.11.2012 Copyright 2012 Teemu Kerola 9



Suorittimen suoritustilat

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 (esim. clear_cache, iret)
 - voi viitata kaikkialle muistiin, myös käyttöjärjestelmän ytimeen (kernel)
 - voi käyttää (myös) suoria muistiosoitteita (PA, physical address)

Miten ja milloin tila vaihtuu

15.11.2012 Copyright 2012 Teemu Kerola 13

Tiedon esitysmuodot

“+”	“15”	“0.1875” = “0.0011”
sign	exponent	mantissa or significand

$1/8 = 0.1250$
 $1/16 = 0.0625$
 0.1875

että ...

mantissa eksponentti	0.0011 “15”
1000 “12”	
1000 “12”	
bitin mantissa!	

kokonaisluvut
 liukuluvut
 merkit (kuvat) (äänet)
 ei-standardoitu tieto?
 suorittimen ymmärtämä tieto?

15.11.2012 Copyright 2012 Teemu Kerola 14

Keskeytyksäsittelijä

- Osa käyttöjärjestelmää
- Ennen keskeytyksäsittelijään hyppäämistä asetetaan suoritin ja MMU etuoikeutettuun käyttöjärjestelmätilaan (supervisor state)
 - SR:n bitti P on päällä => etuoikeutettu tila eli (P = Priviledged) käyttöjärjestelmä tila
 - käyttöjärjestelmätilassa saa viitata mihin tahansa kohtaan muistia (MMU: BASE=0, LIMIT=“hyvin iso”)
 - käyttöjärjestelmätilassa saa käyttää kaikkia konekäskyjä (esim. IRET tai ClearCache)
- Käsittelijästä paluun yhteydessä MMU:n tila ja suorittimen tila (bitti P) asetetaan ennalleen

15.11.2012 Copyright 2012 Teemu Kerola 15

Tiedon tyypit

- Kommunikointi ihmisen kanssa
 - kuva, ääni, merkit, ...
- Laitteiston sisäinen talletus
 - kuvaformaattit, ääniformaattit, pakkausstandardit, ...
 - kokonaisluvut, liukuluvut, merkit, merkistöt
 - ohjelmat
- Suorittimen omana lajinaan ymmärtämät tyypit
 - on olemassa konekäskyjä tälle tietotyypille
 - kokonaisluvat
 - liukuluvut (useimmat suorittimet nykyään)
 - totuusarvot (jotkut suorittimet)
 - merkit (jotkut suorittimet)
 - konekäskyt

15.11.2012 Copyright 2012 Teemu Kerola 16

Big vs. Little Endian

- Miten monitavuiset arvot talletetaan?

0x1200:

0x1200 0x1201 0x1202 0x1203

Sanan osoite

talleta 0x11223344 ?? tavuosoitteet

Big-Endian: eniten merkittävä tavu **pienimpään** osoitteeseen

0x11	0x22	0x33	0x44
------	------	------	------

0x1200 0x1201 0x1202 0x1203

Little-Endian: vähiten merkittävä tavu **pienimpään** osoitteeseen

0x44	0x33	0x22	0x11
------	------	------	------

15.11.2012 Copyright 2012 Teemu Kerola 17

Negatiiviset kokonaisluvut

- Etumerkkibitti erikseen
- Yhden komplementtiesitys
- **Kahden** komplementtiesitys
- Vakioilisäys
 - Esim. lisää 127 (=2⁷ -1)
 - yleensä: 2^{bitilkm-1} -1
 - Talleta etumerkittömänä

arvo talletus

+57 = 0011 1001

sign bit = MSB = most significant bit

luku -57 = 1011 1001 talletusmuoto

-57 = 1100 0110

“sign” bit

+1

-57 = 1100 0111

“sign” bit

-57 = 0100 0110

-57 + 127 = 70

+57 = 1011 1000

+57 + 127 = 184

15.11.2012 Copyright 2012 Teemu Kerola 18

IEEE 32-bit FP Standard

“+” “15” “0.1875” = “0.0011”

sign exponent mantissa or significand

1/8 = 0.1250
1/16 = 0.0625
-0.1875

- 23 bittiä mantissalle siten, että ...

1) Binääripiste (.) on heti ensimmäisen bitin jälkeen

2) Mantissa on **normalisoitu**: vasemmanpuolimmainen bitti on 1

3) Vasemmanpuolimmaista (eniten merkitsevä) bittiä (1) ei talleteta (implied bit, piilobitti)

mantissa eksponentti

0.0011 “15”

1.1000 “12”

1000 “12”

24 bitin mantissa!

Miksi käytetään piilobittia?

15.11.2012 Copyright 2012 Teemu Kerola 19

Tiedon muuttumattomuus

- Virheitä tapahtuu
- Otetaan mukaan ylimääräisiä bittejä, joiden avulla virheitä voidaan havaita ja ehkä myös korjata
- Järjestelmä suorittaa tarkistukset automaattisesti joko laitteistotasolla tai ohjelmiston avulla

15.11.2012 Copyright 2012 Teemu Kerola 20

Virheen korjaava Hamming koodi

oikein virheellinen (parillinen pariteetti)

Data: 100 1100 → 1 0 1100

Bitti nro: 765 4321 → 765 4321

Pariteettibitti 1 tarkistaa bittejä 1, 3, 5, 7

Pariteettibitti 2 tarkistaa bittejä 2, 3, 6, 7

Pariteettibitti 4 tarkistaa bittejä 4, 5, 6, 7

Tapahtuu virhe: bitti 6 muuttuu (flips)

Pariteettibitti 2 tarkistaa bittejä 2, 3, 6, 7: VIRHE

Pariteettibitti 4 tarkistaa bittejä 4, 5, 6, 7: VIRHE

2+4 = 6 ⇒ korjaa bitti nro 6

1

1

11

1

1 1

11

111

15.11.2012 Copyright 2012 Teemu Kerola 21

Virheiden tarkistusmenetelmien käyttöalueet

- Mitä lähempänä suorittamista, sitä tärkeämpää tiedon oikeellisuus on
- Sisäinen väylä, muistiväylä
 - virheet lennossa korjaava Hamming koodi
- Paikallisverkko
 - uudelleenlähetyksen vaativa CRC
 - kun tulee virheitä, niin niitä tulee yleensä paljon
 - Hamming koodi ei riitä kuitenkaan
 - pariteettibitti päästää läpi (esim.) 2 virheen paketit

15.11.2012 Copyright 2012 Teemu Kerola 22

RAM:n kaksi eri teknologiaa

- DRAM: dynaaminen RAM, halvempi, hitaampi, tietoja pitää virkistää vähän väliä (esim. joka 2 ms)
 - tavallinen keskusmuisti (1975-..) useimmissa koneissa
 - toteutettu kondensaattoreilla, jotka ”vuotavat” ...
- SRAM: staattinen RAM, kalliimpi (~10-20x), nopeampi (~10-50x), vie tilaa enemmän, ei vaadi tietojen virkistämistä
 - välimuisti useimmissa koneissa
 - muisti superkoneissa (esim. Cray C-90)
 - toteutettu samanlaisilla logiikkaportteilla (gate) kuin prosessorikin

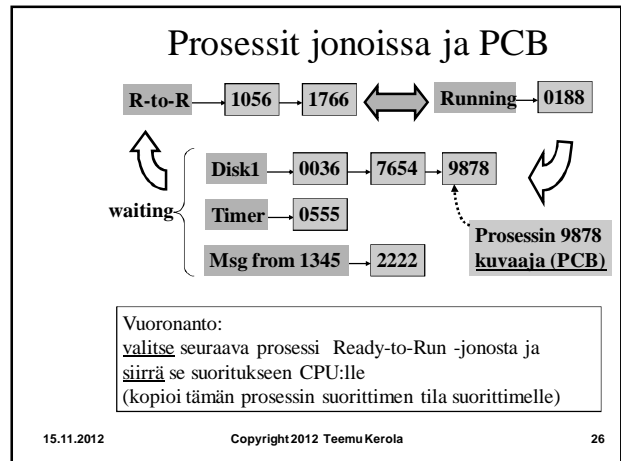
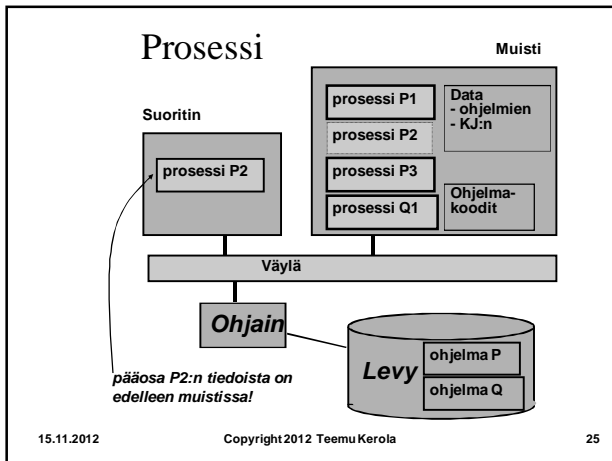
15.11.2012 Copyright 2012 Teemu Kerola 23

Muistihierarkia

- Flash?
- SSD?
- Levypalvelin?
- Pilvi?

Figure 4.1 The Memory Hierarchy

15.11.2012 Copyright 2012 Teemu Kerola 24



- ### Käyttöjärjestelmän rakenne
- Prosessien hallinta
 - Muistin hallinta
 - Tiedostojen ja laitteiden hallinta
 - Verkon hallinta
- 15.11.2012 Copyright 2012 Teemu Kerola 27

