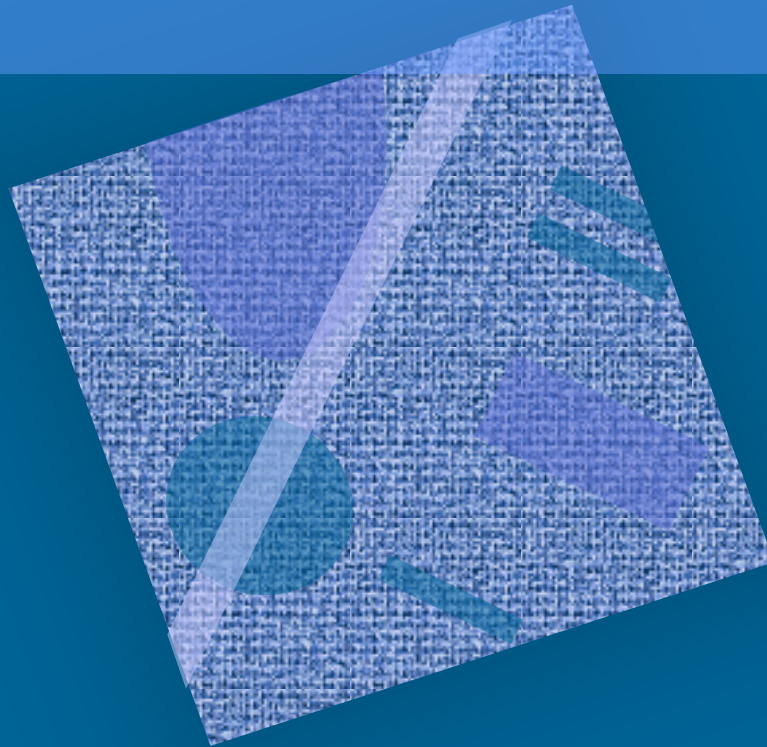


Luento 9 (verkkoluento 9)  
Järjestelmän ulkoinen muisti  
I/O

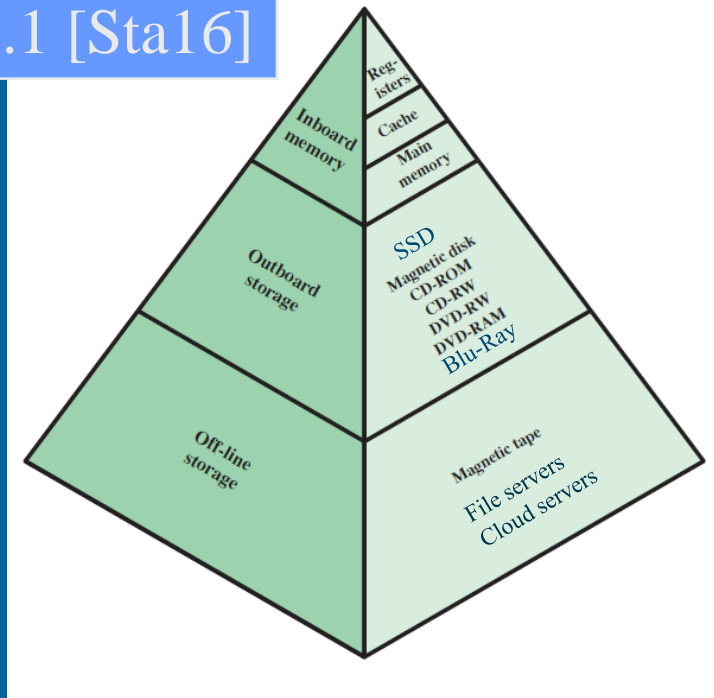


Muistihierarkia  
Tiedostojärjestelmä  
Kiintolevyt  
I/O:n toteutus laiteajurin ja  
laiteohjaimen yhteistyöllä

Fig 4.1 [Sta16]

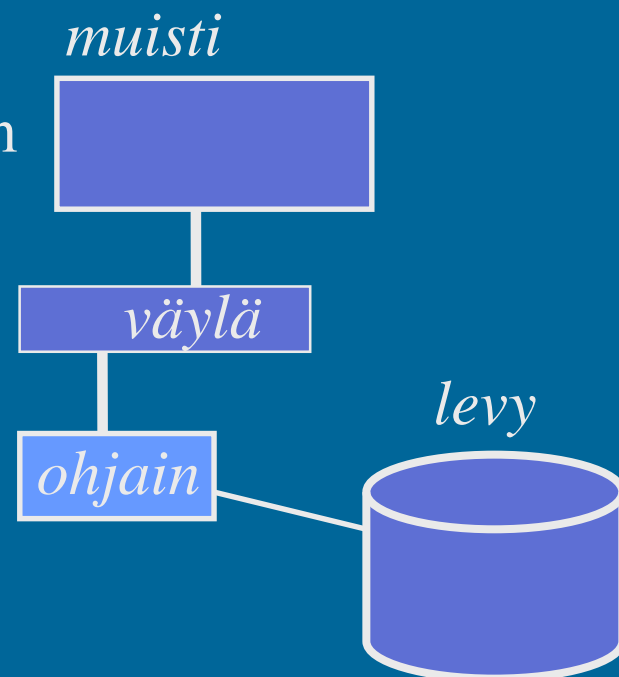
# Muistihierarkia

- Ulkoinen muisti (levymuisti) on halvempaa toteuttaa per tavu
- Ulkoinen muisti on paljon hitaampaa kuin sisäinen muisti
- Aika/tila optimointi
  - suuret tietomäärät täytyy (kannattaa) kustannussyistä pitää ulkoisessa muistissa
  - pienet tietomäärät täytyy (kannattaa) tehokkuussyistä pitää sisäisessä muistissa
- Kaiken viitatus tiedon tulee suoritusaikana olla sisäisessä muistissa tai vielä lähempänä suoritinta!
  - Suoritin ei voi odottaa viitattua tietoa kovin kauan aikaa



# Virtuaalimuisti

- Osa muistihierarkiaa
- Vastaus ongelmaan
  - Kuinka muisti, joka on ”yhtä suuri” kuin levymuisti (tai SSD) ja ”yhtä nopea” kuin keskusmuisti?
- Kaksitasoinen:
  - Keskusmuistissa on ”kulloinkin käytössä olevat” alueet
  - Levyllä on kaikki tiedot
  - Kopiointi tarvittaessa
    - Sivunpuutoskeskeytys (viitattu tieto ei ole keskusmuistissa)



# Virtuaalimuistin toteutus

- Toteutustavat
  - kanta- ja rajarekisterit
  - sivutus, (segmentointi ja sivuttava segmentointi)
- Pääosa toteutuksesta KJ:n ohjelmistotasolla
  - Osoitteenmuunnos: virtuaaliosoite → fyysinen muistiosoite
    - Hidas operaatio, usea konekäsky, muistiviitteitä?
  - Tietojen kopiointi: keskusmuisti ↔ levy
  - Muistitilan hallinta
- Laitteistotuki
  - MMU – muistinhallintayksikkö
  - Osoitteenmuutospuskurista (TLB) löytyy viimeksi tehdyt osoitteenmuutokset (vrt. välimuisti)
    - Fyysistä osoitetta ei tarvitse laskea usealla konekäskyllä, jos se löytyy TLB:stä

Lisää  
tietoa?



Tieto-  
koneen  
rakenne

Käyttö-  
järjes-  
telmät

# Tiedostojärjestelmä

read  
write  
execute

- KJ:n osa, hallitsee kaikkia tiedostoja
- Valvoo oikeuksia tiedostoa avattaessa
- Muuntaa tiedostonimet fyysisiksi osoitteiksi
- Ylläpitää (vain KJ:n käytössä olevia) taulukoita, joista näkee mitä kohtaa mistäkin tiedostosta kukin prosessi on käsittelemässä
- Tiedostojärjestelmä lukee ja kirjoittaa tiedostoja suurina kerralla käsiteltävinä lohkoina (0.5-8 KB?)
  - käyttäjätason prosessit käsittelevät tiedostoja tavuittain, eikä niiden tarvitse tietää tiedoston todellista fyysistä rakennetta (KJ:n laiteajuri huolehtii siitä)

# Tiedoston talletus levyllä

- Tiedosto koostuu useista lohkoista
  - lohko = 1 tai usea levyn sektori
- Levyn hakemisto
  - tiedoston lohkot
  - luetaan lohkot annetussa järjestyksessä

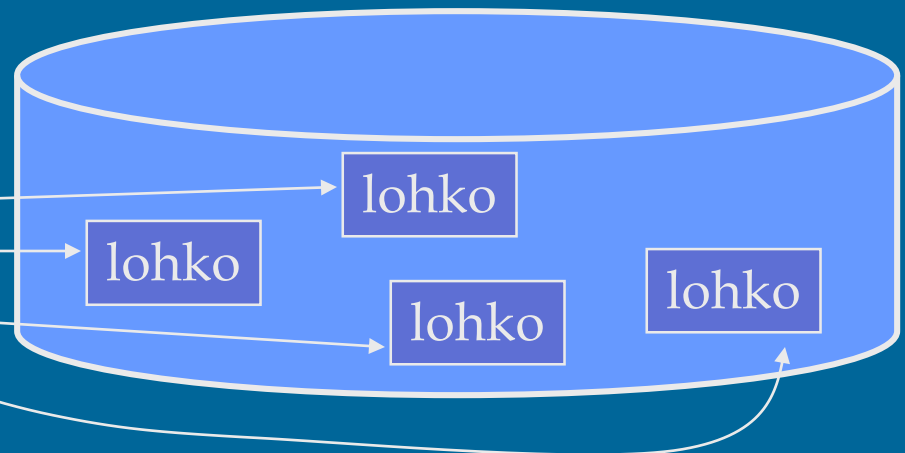
Pienin alue, jota voi lukea/kirjoittaa levyllä

hakemisto  
alkio

FileA

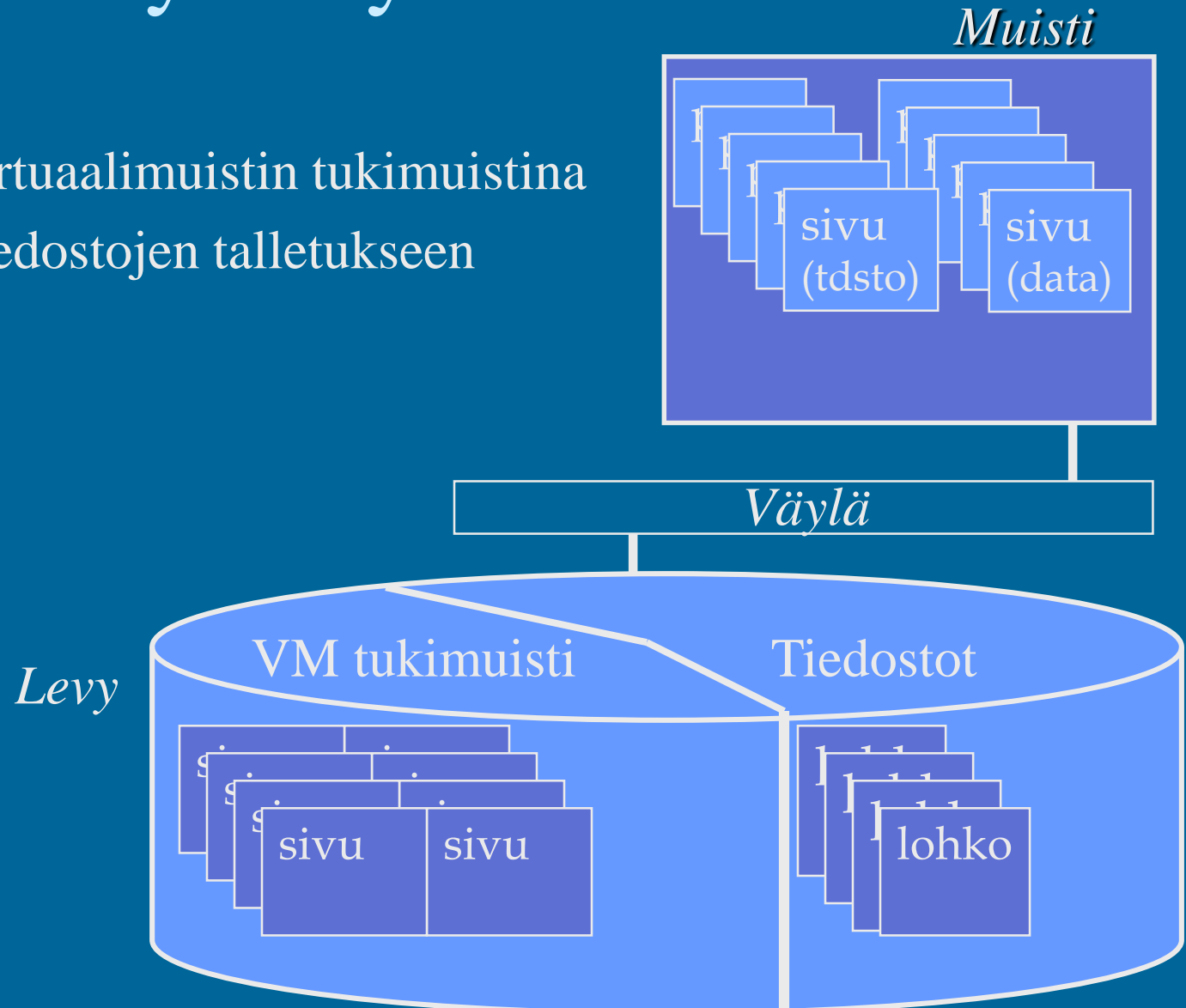
(unix)

indeksi  
lohko



# Levyn käyttö

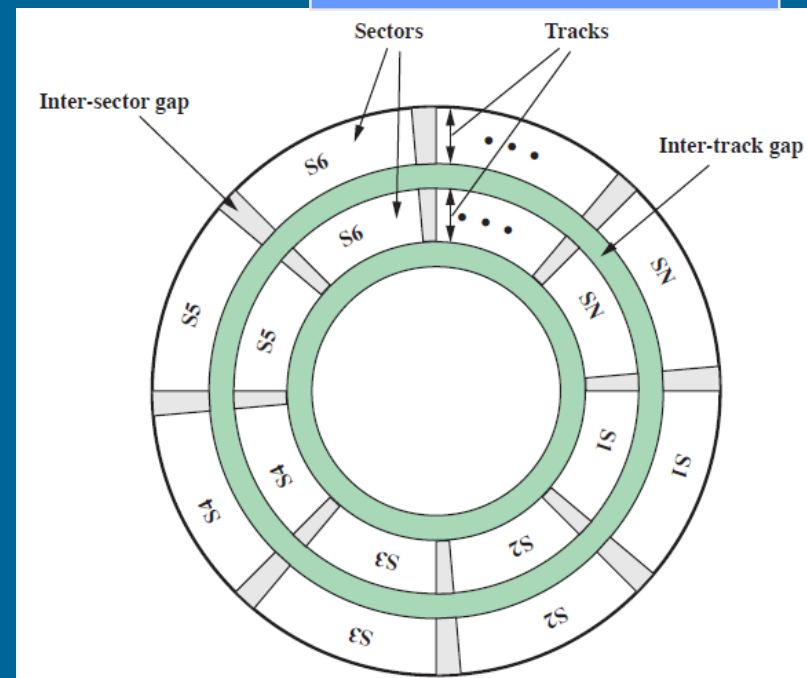
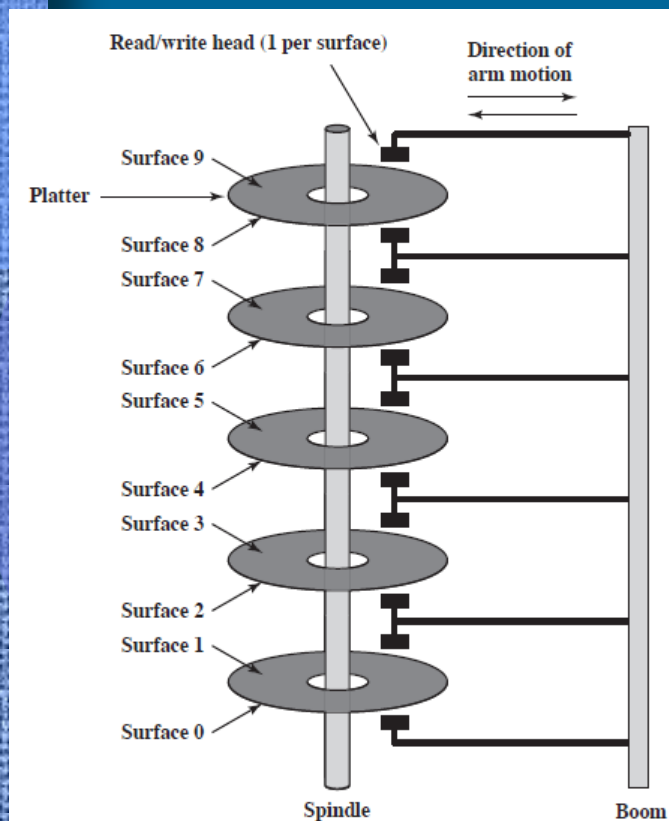
- Virtuaalimuistin tukimuistina
- Tiedostojen talletukseen



# Levymuistin saantiaika

- Tiedon (lohkon) osoite: levypinta + ura + sektori
  - laiteajuri etsii KJ-taulukoista loogisen osoitteen perusteella

Fig 6.2 [Sta16]



- Saantiaika

- Haku aika + pyör.viive + tiedon siirto



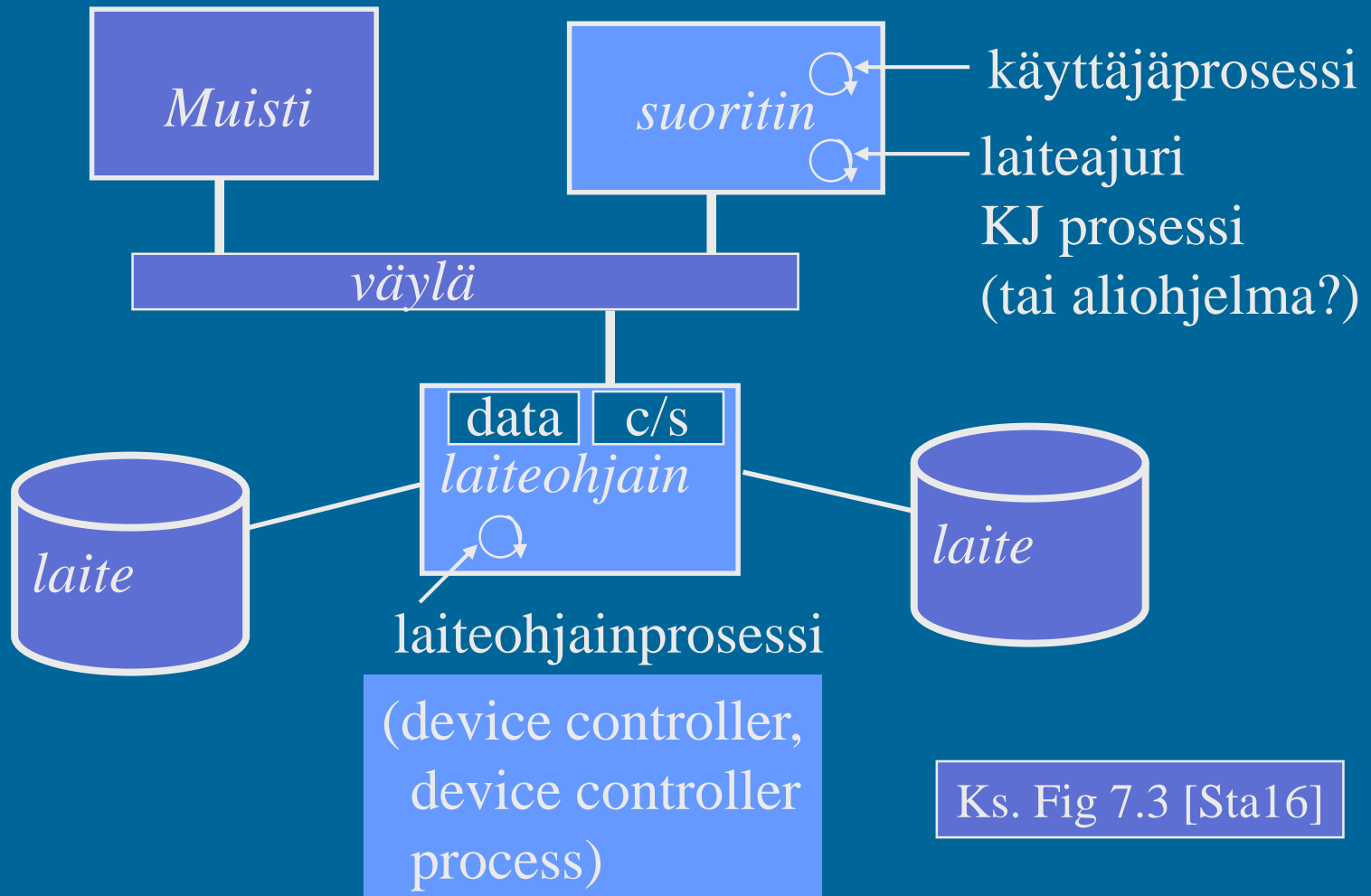
# SSD ja NVMe

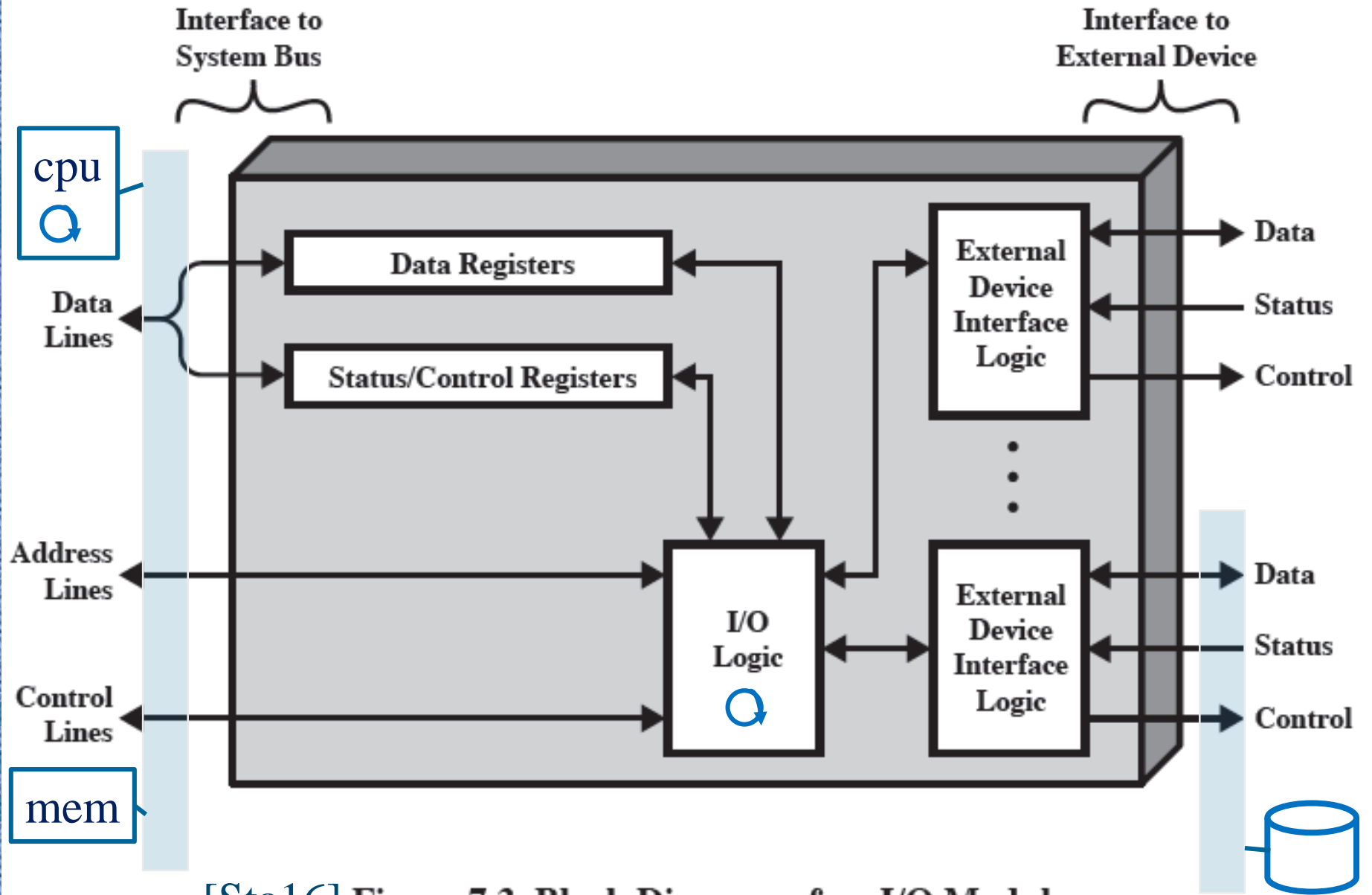
Solid State Disk

Non-Volatile Memory Express  
(NVMe, NVM Express)

- SSD
  - Nykyään yleensä Flash-muistipiiri (tms)
  - Yleensä: Käyttöjärjestelmä (KJ) näkee kovalevynä
  - NVMe: KJ näkee Flash-muistina
    - Nopeampi, osaa hyödyntää Flash'in samanaikaisuutta
- Lohkot ja sivut
  - Tiedostot (esim.) 4 KB sivuina (vrt. kovalevy)
  - Luku ja kirjoitus (esim.) 512 KB lohkoina
    - Koko lohko pitää tyhjentää ennen kirjoitusta
    - Kirjoitus ehkä uuteen lohkoon
    - Kullakin lohkolla raja kirjoituksille (esim. 100000)
      - Ylimääräisiä lohkoja piirillä

# Laiteohjain (I/O Moduuli)





[Sta16] Figure 7.3 Block Diagram of an I/O Module

# I/O:n toteutus konekäskyillä (i): Laiteohjaimen rekistereihin (muistiin) viittaaminen I/O konekäskyillä

- I/O-operaation tunnistaa operaatiokoodista
  - I/O laitteille on omat konekäskyt
  - I/O-käskyillä oma osoiteavaruus, eivät viittaa keskusmuistiin
- Käskyssä annetaan laiteohjaimen identifikaatio ja laiterekisterin nro (oma I/O osoiteavaruus)
- Vaikea laajentaa käyttöä uusiin laitteisiin, joilla ”laiterekisterit” voivat olla hyvinkin erilaisia
- Suorittimen konekäskyjä ei voi muuttaa

x86: IN, OUT  
INS, OUTS

Ttk-91:  
IN, OUT

# I/O:n toteutus konekäskyillä (ii): Laiteohjaimen rekistereihin (muistiin) viittaaminen muistin luku/kirjoitus käskyillä

- I/O-operaation tunnistaa viitatusta muistiosoitteesta
- Muistiinkuvattu I/O, memory-mapped I/O
- Laiteajuri lukee/kirjoittaa laiteohjaimella olevia rekistereitä (data, status/kontrolli) tavallisilla muistin luku/kirjoitus käskyillä
  - ei tarvita erillisiä I/O-konekäskyjä!

```
load R1,=DiskRd
store R1, DiskCtr
```
  - laiteohjaimella olevat ”laiterekisterit” ovat samanlaista viitattavaa muistia kuin ”normaali muisti”
  - muistiosoitteen ensimmäiset bitit (ei siis käskykoodi) valitsevat, mille laitteelle (vai tavallisen muistiin) viittaus kohdistuu
    - Osa muistiavaruudesta on varattu I/O-laitteille!

```
DiskCtr EQU 0x80000001
```

# I/O tyypit

- Suora I/O

Programmed I/O  
Direct I/O

- Laiteajuri koko ajan aktiivinen
- Laiteajuri odottaa busy-wait loopissa laiteohjainta

- Epäsuora I/O

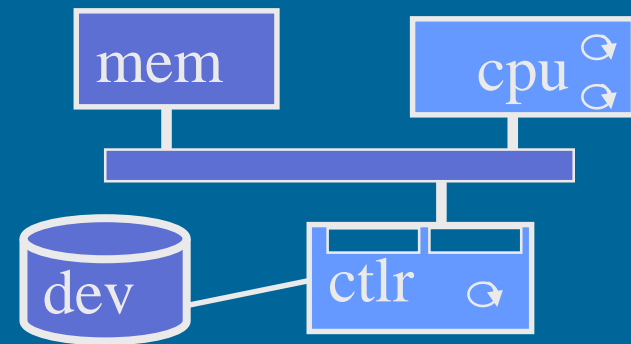
Indirect I/O, Interrupt driven I/O

- Laiteohjain osaa tehdä I/O-laitekeskeytyksiä
- Laiteajuri odottaa odotustilassa laiteohjainta

- DMA I/O

Direct Memory Access I/O

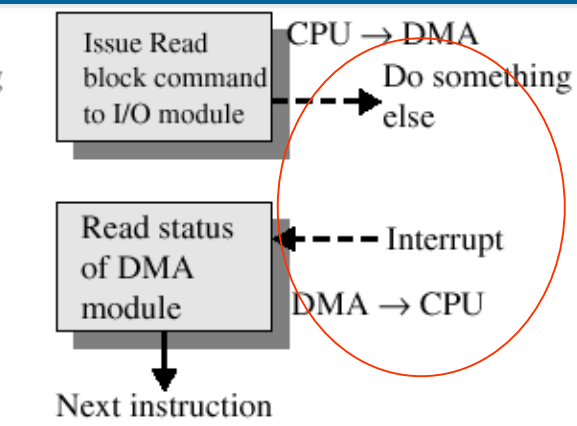
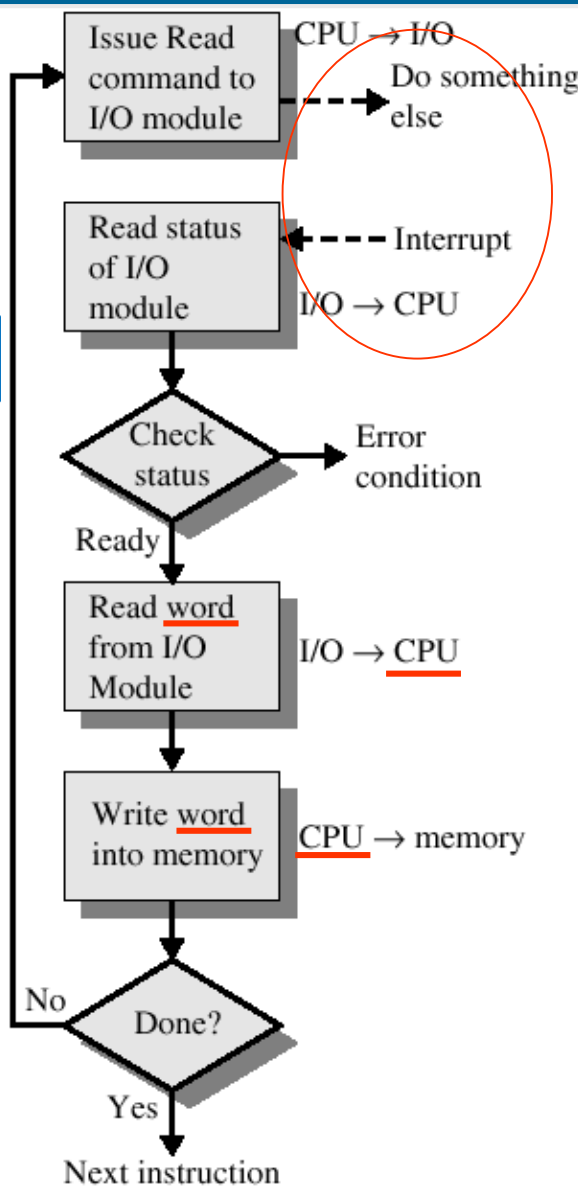
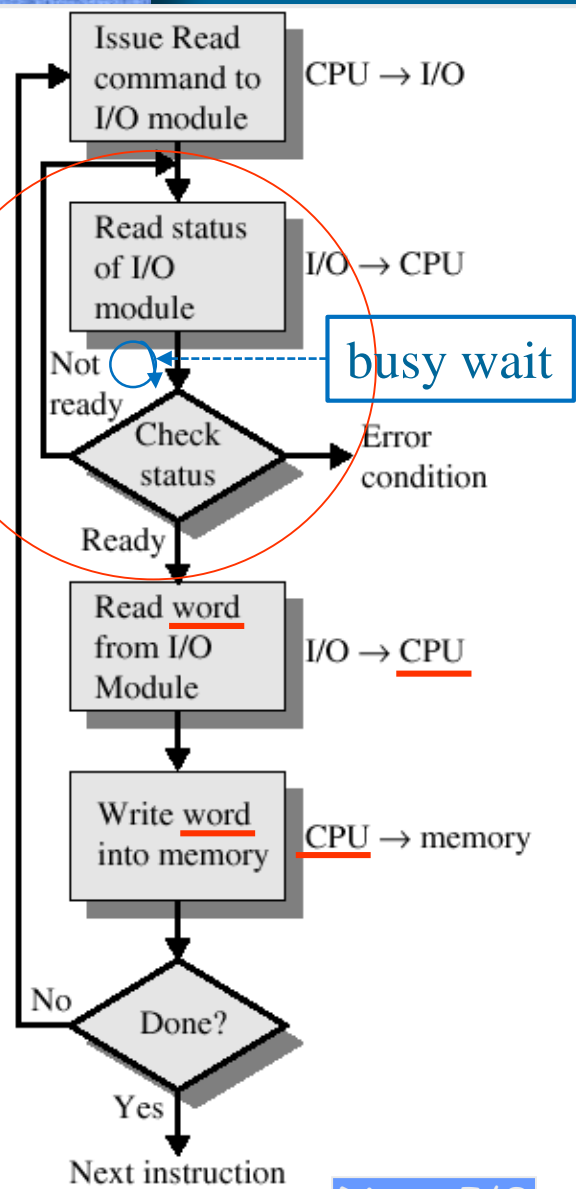
- Laiteohjain osaa myös viitata keskusmuistiin
  - Tieto kulkee väylän läpi vain yhden kerran
- Laiteohjaimelle annettavat tehtävät isompia
- Laiteajuri odottaa odotustilassa laiteohjainta



Suora I/O

Epäsuora I/O

DMA I/O



(c) Direct memory access

# Device driver (laiteajuri)

## Input of a Block of data

Fig 7.4 [Stal16]

(a) Programmed I/O **Direct I/O**

(b) Interrupt-driven I/O

# Esimerkki: kirjoittimen laiteajuri ttk-91 koneelle

- Laitteella voi tulostaa kokonaislukuja yksi kerrallaan
- Muistiinkuvattu I/O, suora I/O
- Laiteportti
  - kontrollirekisteri                      muistipaikka 1048576 = 0x80000
  - tilarekisteri                              muistipaikka 1048577 = 0x80001
  - datarekisteri                              muistipaikka 1048578 = 0x80002
- Laiteajuri Print toimii etuoikeutetussa tilassa
  - Voi viitata portin rekistereihin

- Kutsu:

```
PUSH SP, =0      ; space for return value
PUSH SP, X       ; parameter to print
SVC  SP, =Print  ; returns Success/Failure
POP  SP, R1
JNZER R1, TakeCareOfTrouble
```



# Esim: laiteajurin toteutus (12)

Solution with no timeout

```
ptrCtr   DC 1048576 ; control register address
ptrStat  DC 1048577 ; status register address
ptrData  DC 1048578 ; data register address
retVal   EQU -3
parData  EQU -2
```

```
Print   PUSHR SP           ;save regs
        LOAD  R1, parData(FP)
        STORE R1, @ptrData ; data to print


---


        LOAD  R1, =0
        STORE R1, @ptrStat ; init (clear) state register

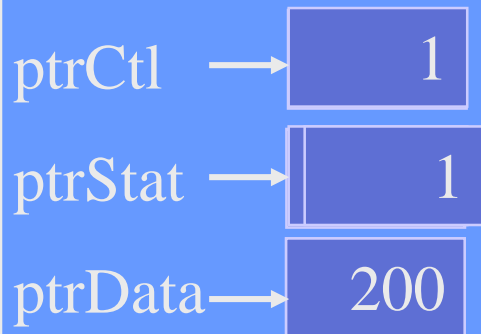

---


        LOAD  R1, =1
        STORE R1, @ptrCtr ; give command to print


---


```

Oleta: SVC:n ja IRET:n toteutus samalla tavalla kuin CALL ja EXIT



```
Wait   LOAD  R1, @PtrStat ; check state register
        JNZER R1, Done
        JUMP  Wait       ; wait until I/O done


---


```

```
Done   LOAD  R1, =0       ; return "Success"
        STORE R1, retVal(FP)
        POPR  SP           ; recover regs
        IRET  SP, =1
```

See: driver.k91

# -- Luennon loppu --

- Ferriittirengas (core) teknologia
  - 1952, Jay Forrester & Bob Everett, MIT (Whirlwind)
  - tieto säilyy ilman virtaa
  - ei häiriinny säteilystä (avaruus, sotilasteknologia)
  - 1955, valtaa markkinat Williams Tube'ltä
  - Käytössä vielä 1970-luvulla, nyt vain nimi jäljellä

