

FKEF.02.143

Arvuti arhitektuur

Computer Architecture and Organization



dotsent
Toomas Plank

©Toomas Plank, 2008

FKEF.02.143

Sisend/väljund

2. loeng




9. loeng,
25. aprill 2008

©Toomas Plank, 2008

Sissejuhatus

- Mälu otsepöördumine (DMA)
- Siinid
 - sünkroonne andmeedastus
 - asünkroonne andmeedastus



Arvuti arhitektuur FKEF.02.143

Otsemälupöördumised

- Alternatiivne moodus suurte andmeblokkide kiireks edastamiseks
- Spetsiaalne juhtimisahel *DMA controller*
- Andmeblokk kantakse seadmest mällu (või vastupidi) ilma protsessori pideva vahelesegamiseta
- Siit ka nimi DMA (*Direct Memory Access*)
- Suurema osa protsessori tööst teeb siin ära DMA-kontroller
 - annab mälu jaoks aadressi
 - edastab siinile tarvilikud juhtsignaalid
 - oskab mälu aadressi automaatselt suurendada
 - peab arvet ülekantu kohta

Arvuti arhitektuur FKFE.02.143



Otsemälupöördumised

- Seejuures peab DMA kontrolleri tegevus olema protsessori poolt käivitatava programmi kontrolli all!
- Enne andmebloki saatmist annab protsessor
 - algusaadressi
 - sõnade arvu blokis
 - ülekande suuna
- Seejärel toimetab DMA kontrolleri vastavalt saadud korraldustele
- Kui kogu blokk on üle kantud, saadab DMA kontrolleri protsessorile katkestussoovi

Arvuti arhitektuur FKFE.02.143



Otsemälupöördumised

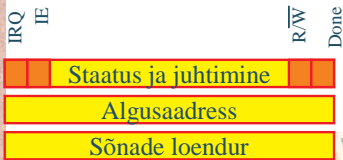
- DMA kanali kaudu andmete ülekandmise ajal ei saa väljakutsunud programm edasi toimetada – tema on blokeeritud seisus
- Küll aga saavad sel ajal toimetada teised programmid!
- Kui andmete ülekande on lõppenud, saab protsessor anda uuesti kontrolli DMA ülekande algatanud programmile (viib ta käivitatavasse seisundisse)
- Sisend/väljundoperatsioonid toimuvad operatsioonisüsteemi kontrolli all, viimane on vastutav ka programmide vahel toimetamisaja jagamisel

Arvuti arhitektuur FKFE.02.143



DMA seadme registrid

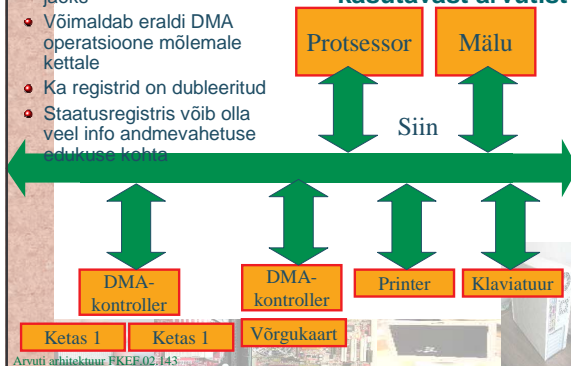
- Näites kasutatakse kolme registrit
- Bitt 1 näitab ära andmeedastuse suuna
- Bitt 0 seatakse kõrgeks, kui andmed on edastatud
- Kui bitt 30 on aktiivne, genereeritakse nüüd katkestussoov
- Peale katkestussoovi esitamist pannakse ka 31 bitt kõrgeks



Arvuti arhitektuur FKFE02.143

Näide DMA-kontrollereid kasutavast arvutist

- Kaks sõltumatut DMA kanalit, üks mõlema ketta jaoks
- Võimaldab eraldi DMA operatsioone mõlemale kettale
- Ka registrid on dubleeritud
- Staatuseregistris võib olla veel info andmevahetuse edukuse kohta



Arvuti arhitektuur FKFE02.143

Cycle stealing

- Protsessori ja DMA kontrolleri mälu poole pöördumised on põimunud: DMA kontrolleri soovid siini kasutada on alati protsessorist kõrgema prioriteediga!
- Seadmetevaheline prioriteedi jaotus
 - kettad, kiired võrgukaardid ja graafika on kõrgema prioriteediga seadmed
- Ülalkirjeldatud protseduuri kutsutakse ka *cycle stealing* protseduuriks, kuna DMA kontrolleri "varastab" protsessorilt mälu suhtlemise tsükleid

Arvuti arhitektuur FKFE02.143

block või burst mode

- Alternatiiv *cycle stealing* protseduurile, on *burst* mood
- Siin saab DMA kontrolleri eksklusiivse juurdepääsu mälule andmete ülekandmiseks
- Problem: mis saab siis, kui mitu seadet soovivad üheaegselt siini DMA tarvis kasutada?

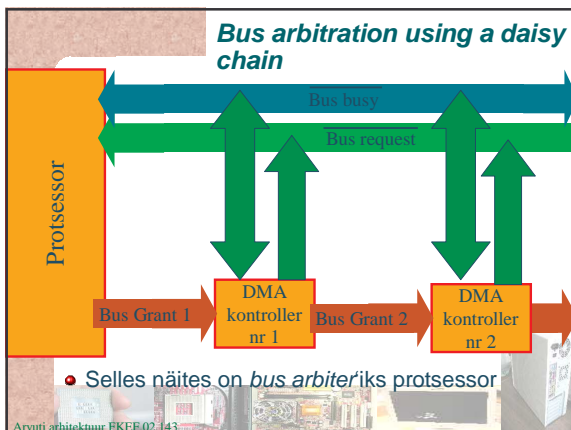
Arvuti arhitektuur FKEE02.143

Bus arbitration

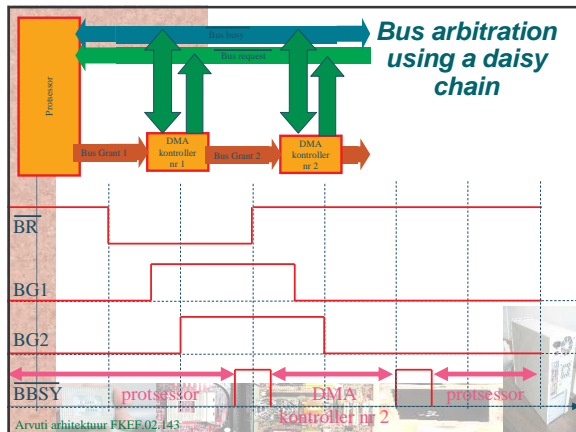
- *Bus master* – seade, mis saab igal ajal algatada andmevahetuse
- Need seadmed vahetuvad töö käigus
- *Centralized arbitration* – üks seade (*bus arbiter*) otsustab, kes on järgmine *bus master*
- *Distributed arbitration* – kõik seadmed osalevad uue *bus master*'i valimises

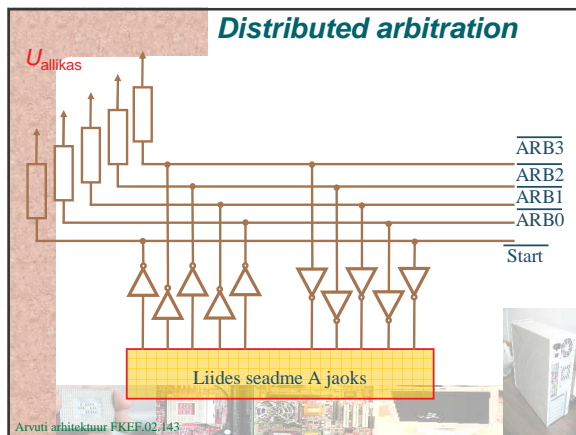
Arvuti arhitektuur FKEE02.143

Bus arbitration using a daisy chain



Arvuti arhitektuur FKEE02.143





Näide

- Olgu meil kaks seadet ID-numbritega 9 ja 10
 - s.t seade A saadab liinile info 1001
 - s.t seade B saadab liinile info 1010
- Liinil paistab kõigile seadmetele nüüd signaal 1011
- Mõlemad seadmed võrdlevad oma ID-numbrit signaaliga ja seavad esimesest mittekokkulangevusest alates bitid nulliks
 - s.t seade A parandab info 1001 à 1000
 - Siinil paistab nüüd signaal 1010, ehk teisisõnu, seade B võib toimetama hakata ☺
 - Ka seade B võib lühiajaliselt lülitada välja oma ARB0 liini draiveri (1010), aga nähes liinil ARB0 signaali 0 lülitab selle sisse tagasi
- Selline deentraliseeritud moodus on kõrge usaldusväärsusega, kuna ei sõltu ühestki konkreetselt seadmest

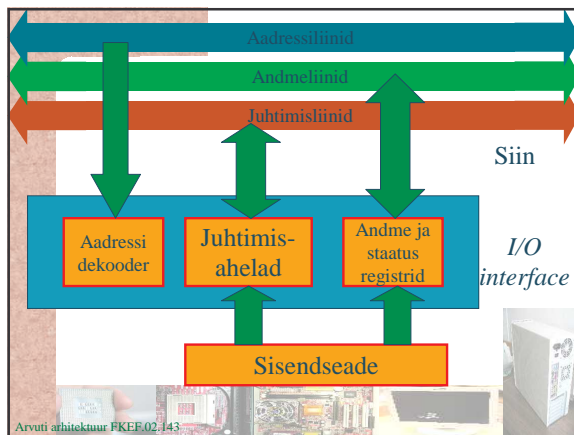
Arvuti arhitektuur FKFE.02.143

Siinid

- Protsessor, mälu ja I/O seadmed ühendatakse omavahel siini(de)ga.
- Siini esmaseks ülesandeks on andmete edastamiseks vajaliku tee loomine
- *Bus protocol* – reeglite kogum, mille alusel korraldatakse erinevate seadmete suhtlus siinil
 - millal saata infot siinile
 - siini kasutusjärjekorra jagamine
 - juhtsignaalidest arusaamine
- Siini liinid jagatakse kolme gruppi:
 - andmed
 - aadress
 - juhtimine

Arvuti arhitektuur FKFE02.143





Arvuti arhitektuur FKFE02.143



Juhtimisiinid

- Juhtimisiini vahendusel tuleb seadmele öelda, kas täitmisele läheb lugemise või kirjutamise käsk
 - seda saaks korraldada ühe R/W liini abil, mis olles seisus 1 viitaks lugemisele ja seisus 0 kirjutamisele
- Täpsustada on vaja andmete suurus (byte, word, long word jne)
- Ajastusinfo – millal on õige aeg seadmel andmeid siinile saata ja millal on siinil kvaliteetsed andmed lugemiseks
- *Master* – initsiaator
- *Slave* – sihtmärk, *target*

Arvuti arhitektuur FKFE02.143



Sünkroonne andmeedastus

- Kõik seadmed saavad ajastusinfo taktsignaalist
- Ajavahemik t_1-t_2 peab olema piisavalt pikk, et info jõuaks siini teise otsa kohale ja et seade jõuaks aadressi dekodeerida
- Slave ei tohiks enne ajahetke t_2 toimetama hakata kuna andmed siinil pole **veel usaldusväärsed!**
- Ajavahemik t_2-t_1 peab olema piisavalt pikk, et info jõuaks siini teise otsa kohale ja et puhver-register jõuaks andmed salvestada (Set-up time)

Arvuti arhitektuur FKKEE.02.143

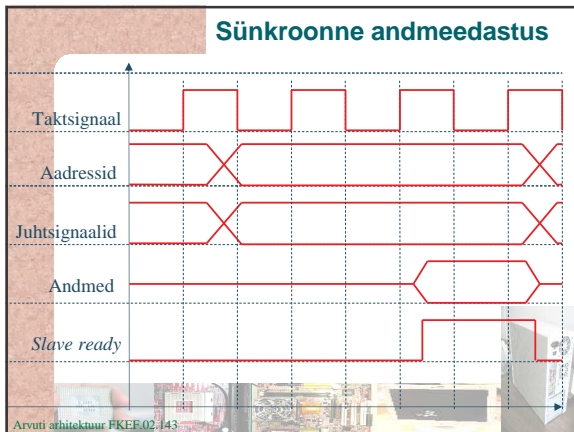
Eelmise slaidi pilt viiviseid arvestades

Arvuti arhitektuur FKKEE.02.143

Sünkroonne andmeülekanne mitme taktitsükli jooksul

- Eelmisel slaidil kujutat andmeedastus peab arvestama kõige aeglasema seadme vajadustega
- Teisisõnu: kõik seadmed suhtlevad aeglaseima seadme kiirusega
- Protsessor ei saa tagasisidet selle kohta, kas seade sai oma asjadega hakkama
- Lisatakse täiendav tagasisidesignaali
 - nüüd saab *master* teada, kui *slave* on aadressist aru saanud ja valmis osalema andmeedastuses
 - lisandub võimalus kohandada andmeedastusele kuluvat aega, kui seade pole valmis

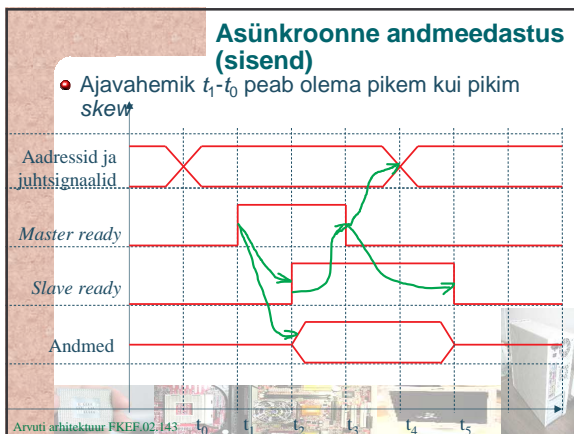
Arvuti arhitektuur FKKEE.02.143

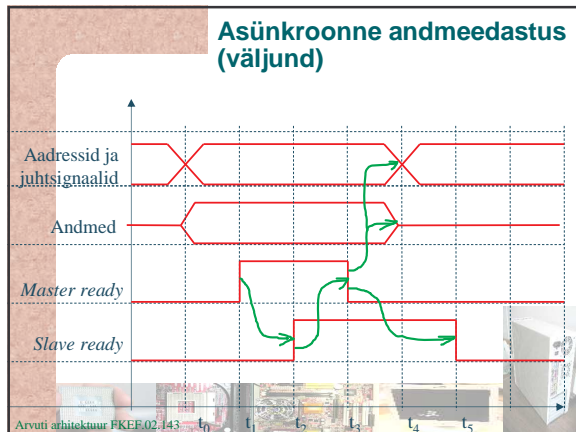


Asünkroonne andmeedastus

- Käepigistuse (*handshake*) kasutamine *master* ja *slave* seadme vahel
- Taktsignaali asendab siin kaks juhtsignaali
 - *master ready*
 - *slave ready*
- *Master ready* seatakse aktiivseks mõningase viivise järel peale andmete siiniletoomist
- Põhjus: samast allikast pärit signaalid jõuavad erinevaid ahelaid läbides sihtkohta eri aegadel
- Inglise keeles nimetatakse seda nähtust *skew*

Arvuti arhitektuur FKKEE.02.143





- ### Plussid-miinused
- Asünkroonne andmeedastuse eeliseks on
 - pole vaja kellasid sünkroniseerida
 - seega lihtsam ehitada
 - Asünkroonne andmeedastuse puudusteks on
 - topelt teekonnaga signaalid võtavad rohkem aega kui ühekordse kinnitusega signaalid
 - seega aeglasem kui sünkroonne andmeedastus
- Arvuti arhitektuur FKEF.02.143

- ### Kasutatud kirjandus
- Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer organization 5th edition (2002) 805 p.
- Arvuti arhitektuur FKEF.02.143
