

FKEF.02.143

Arvuti arhitektuur

Computer Architecture and Organization



dotsent
Toomas Plank

©Toomas Plank, 2008

FKEF.02.143

Sisend/väljund

1. loeng




8. loeng,
18. aprill 2008

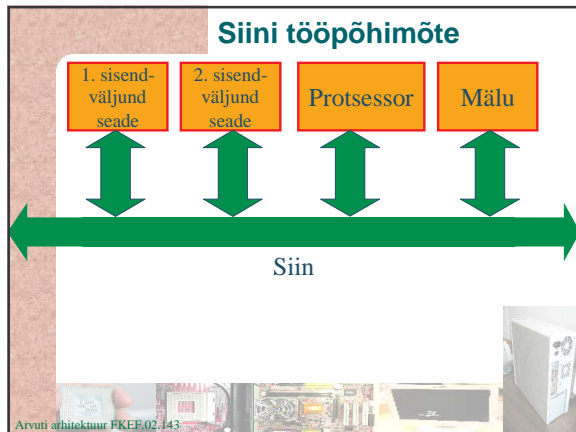
©Toomas Plank, 2008

Sissejuhatus

- Programmi poolt juhitud sisend-väljundoperatsioonid
- Katkestused (IRQ)
- Riistvara ja tarkvara, mida katkestuste korraldamiseks tarvitatakse



Arvuti arhitektuur FKEF.02.143



Seadmete adresseerimine

- Siinil on liinid
 - aadresside,
 - andmete
 ja
 - juhtsignaalide jaoks
- Igal I/O seadmel on oma unikaalne aadress
- Kui protsessor saadab aadressisiinile konkreetse seadme aadressi, siis seade, kellele info mõeldud on, hakkab täitma juhtsignaalide poolt etteõeldavat käsku
 - Read
 - Write
 - Seejärel saadetakse vastavad andmed andmesiinile

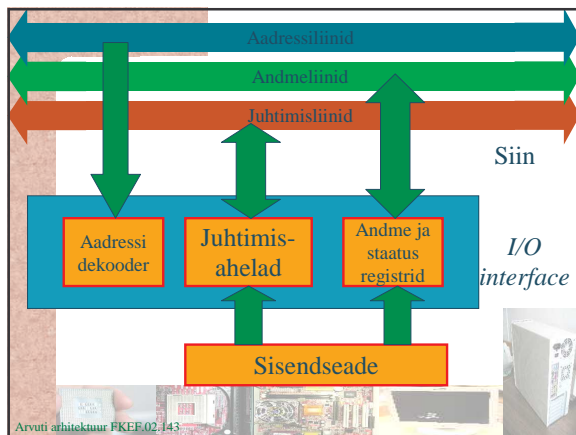
Memory-mapped I/O

- *Memory-mapped I/O* – nii mälu kui ka seadmed kasutavad sama aadressiruumi
- Kõik käsud, millega saab pöörduda mälu poole, sobivad siin ka seadmetele andmete saatmiseks või neist andmete lugemiseks.
- Move DATAIN,R1
- Move R1,DATAOUT
- Enamik arvutisüsteeme kasutabki *Memory-mapped I/O*'d

Eraldi aadressiväli sisend-väljundseadmete jaoks

- Teistel on spetsiaalsed I/O käsud ja eraldi 16-bitine aadressiväli I/O seadmete jaoks
- NB! Eraldi aadressiväli ei tähenda eraldi traate aadresside jaoks!!!
- Spetsiaalne signaal siinil näitab, et tegemist on I/O operatsiooniga
 - Mälu ignoreerib selliseid käskke
 - I/O seade uurib aadressivälja madalamaid bitte saamaks teada, kas käsk on mõeldud talle

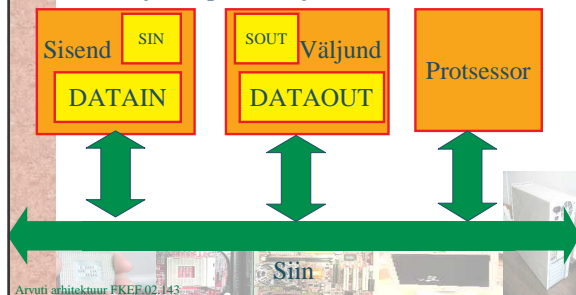
Arvuti arhitektuur FKEE.02.143



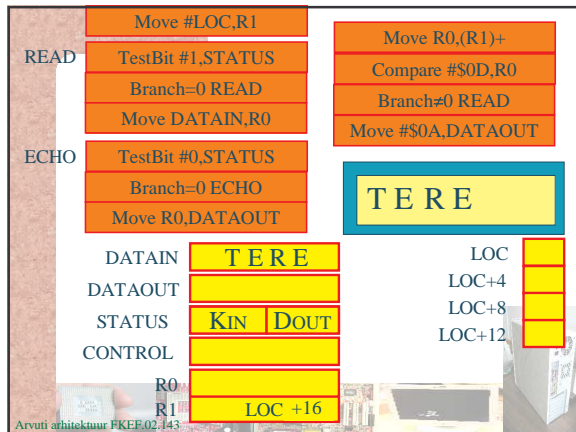
Arvuti arhitektuur FKEE.02.143

Sisend/väljund operatsioonid

Peame olema veendunud, et sisendandmeid loetakse just täpselt üks ja ainult üks kord



Arvuti arhitektuur FKEE.02.143



Program controlled I/O

- Eelmine programm illustreeris programmi poolt kontrollitavat sisend/väljundit (*program controlled I/O*)
 - protssessor kontrollib pidevalt staatustippe tagamaks sünkroniseeritud andmevahetuse sisend- ja väljundseadmega
 - Processor *polls* the device
- Sellise programmi korral toimetab protssessor suurema osa ajast ootamistsükliks
- Tohutu ressursi raiskamine L
- Mõistlik kasutada
 - katkestusi σ
 - või
 - otsemälupöördumist σ

Arvuti arhitektuur FKEE.02.143

IRQ ja DMA

- Katkestuse soovi (*Interrupt ReQuest*) korral saadab seade protssessorile spetsiaalse signaali millal iganes ta on valmis andmeid vahetama
- Otsemälupöördumist (*Direct Memory Access*) kasutatakse kiirete I/O seadmete puhul: seade kirjutab/loeb andmed mällu/mälust otse, ilma protssorit tülitamata

Arvuti arhitektuur FKEE.02.143

Katkestused

- Seade saadab protsessorile riistvaralise signaali (katkestus, *interrupt*), millega annab teada oma valmisolekust andmevahetuseks
- Tehniline teostus eeldab vähemalt ühte spetsiaalset liini selleks otstarbeks (*interrupt-request line*)
- Katkestuse teenindamise protseduur (*Interrupt-service routine*)
- Protsessor saab nüüd ootamise asemel tegeleda kasulike asjadega, ideaalvariandis saame ootamisperiodidest täielikult lahti

Arvuti arhitektuur FKEE.02.143



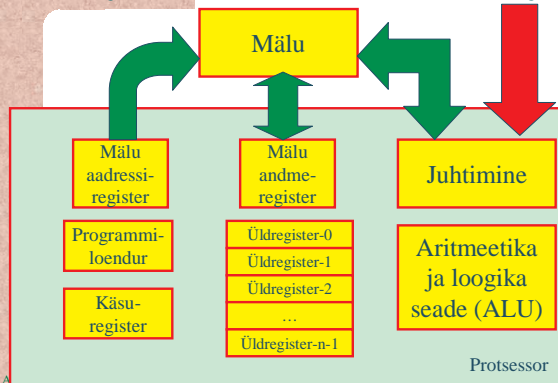
Katkestused

- Enne katkestuse täitmist tuleb salvestada kogu info, mis katkestuse täitmise käigus muutuda võib
- Enne katkestusest naasmist tuleb kogu info uuesti taastada
 - nii saab algne programm jätkata samast kohast kus enne töö pooleli jäi
 - ainus kaotus on ajaline viivis
- Salvestatakse
 - programmi loenduri sisu
 - seisundilipud
 - registreid, mida mõlemad (algne programm ja *interrupt-service routine*) kasutavad

Arvuti arhitektuur FKEE.02.143

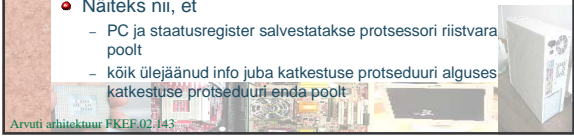


PC ja staatusinfo haldus katkestuste ajal



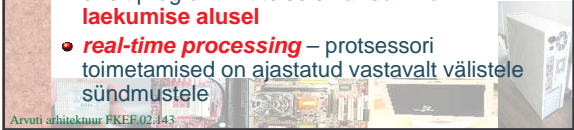
Katkestused

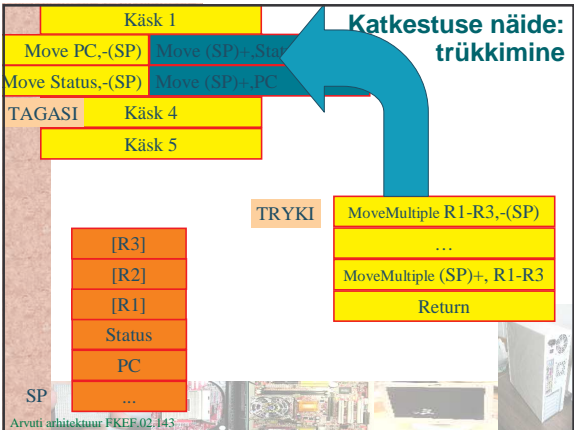
- Info salvestamine ja taastamine on raisatud aeg programmide töökiiruse mõttes
- Samuti pikeneb nii aeg, mil katkestuse nõue reaalsetl täidetud saab (*interrupt latency*)
- Mõnedel juhtudel pole pikk viivis katkestusesooivi saabumisest reaalse täitmiseni aktsepteeritav
- Seetõttu salvestavad tänapäeva protsessorid ainult nii palju infot kui on hädavajalik katkestusele eelnenud programmikohast jätkamiseks
- Näiteks nii, et
 - PC ja staatusregister salvestatakse protsessori riistvara poolt
 - kõik ülejäänud info juba katkestuse protseduuri alguses katkestuse protseduuri enda poolt



Katkestused

- Üks lahendus on kasutada erineva reaktsiooniajaga katkestusi
 - kiirete puhul EI SALVESTATA registreite sisu
 - aeglaste puhul SALVESTATAKSE
 - konkreetne seade saab valida, kumba parasjagu tarvitada
- Teine alternatiiv: topelt registrid protsessoris
 - nüüd pole salvestada üldse tarvis
- Katkestused võimaldavad üle anda juhtimise ühelt programmilt teisele **välise info laekumise alusel**
- **real-time processing** – protsessori toimetamised on ajastatud vastavalt väliste sündmustele





Katkestused

- Lisaks sellele, et seade paneb liinile püsti katkestuse soovi, peab protsessor seadet informeerima katkestusesoovi rahuldamisest
 - *interrupt-acknowledge* signaal
 - andmevahetuse alustamise käsu käivitaminekatkestuse teenindamise protseduuris võib anda seadmele sama info
- Erinevus alamprogrammi ja katkestuse protseduuri vahel: **katkestuse protseduur kutsutakse välja suvalisel hetkel** (käigusoleva programmi ajaarvamise mõttes)
 - käigusolev programm ja katkestuse väljakutsunud programm võivad olla lausa eri kasutajate omad

Arvuti arhitektuur FKFE.02.143

Avatud neeluga katkestuse-siini ekvivalentskeem



- Kui kõik lülitid on lahti, on siini olek kõrge (**mitteaktiivne**).
- Nii kui ühe lülitist sulgeme, on siin olek madal (**aktiivne**).

Arvuti arhitektuur FKFE.02.143

Katkestuste lubamine-keelamine

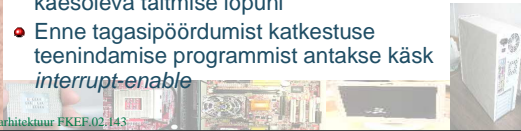
- Katkestuse lubamine suvalisel ajahetkel võib olla vastuolus programmeerija soovituga
- Seetõttu peab olema mehhanism katkestuste haldamiseks
 - *Interrupt-enable*
 - *Interrupt-disable*
- Soovitud katkestus tuleb täitmisel võtta üks ja ainult üks kord, s.t aktiivne katkestusesoovi signaal ei tohi põhjustada arvuti lõputusse tsükklisse sattumist
- Järgnevalt vaatame kolme võimalust

Arvuti arhitektuur FKFE.02.143

Katkestuste lubamine-keelamine: võimalus 1

- Protsessori riistvara ignoreerib katkestusesoovi signaali seni kuni katkestuse teenindamise protseduuri esimene käsk saab täidetud
- Nüüd saab programmeerija alustada (soovi korral) katkestuse teenindamise protseduuri käsuga *interrupt-disable*
- Nii välistame uued katkestused kuni käesoleva täitmise lõpuni
- Enne tagasipöördumist katkestuse teenindamise programmist antakse käsk *interrupt-enable*

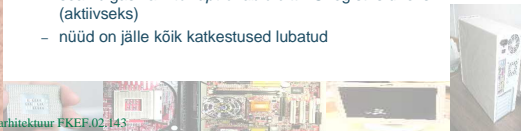
Arvuti arhitektuur FKFE02.143



Katkestuste lubamine-keelamine: võimalus 2

- Protsessor võib automaatselt blokeerida katkestused peale katkestuse teenindamise protseduuri algust
- Peale PC (*program counter*) ja PS (*processor status register*) sisu salvestamist pinus seatakse *interrupt-enable* bitt PS registris nulliks (passiivne) ja rohkem katkestusi ei aktsepteerita
- Katkestusest naasmisel kopeeritakse PC ja PS sisu tagasi
 - sealhulgas ka *interrupt-enable* bitt PS registris üheks (aktiivseks)
 - nüüd on jälle kõik katkestused lubatud

Arvuti arhitektuur FKFE02.143



Katkestuste lubamine-keelamine: võimalus 3

- Katkestuse soovi liinil kasutatakse frondilt käivituvaid trigereid, s.t katkestuse soovi arvestatakse nüüd ainult üks kord, sõltumata sellest kui pikalt katkestuse soovi signaal aktiivne on

Arvuti arhitektuur FKFE02.143



Kokkuvõte katkestuse täitmise stsenaariumist

1. Seade avaldab soovi katkestuseks
2. Protsessor katkestab jooksva programmi täitmise
3. Uued katkestused blokeeritakse kontrollbiti passiivseks seadmisega PS registris
4. Seadmele öeldakse, et tema katkestusesoov on aktsepteeritud
5. Seade võtab katkestusesoovi maha
6. Soovitud katkestuseprotseduur täidetakse
7. Lubatakse uued katkestused ja protsessor naaseb katkestatud programmi täitmise juurde

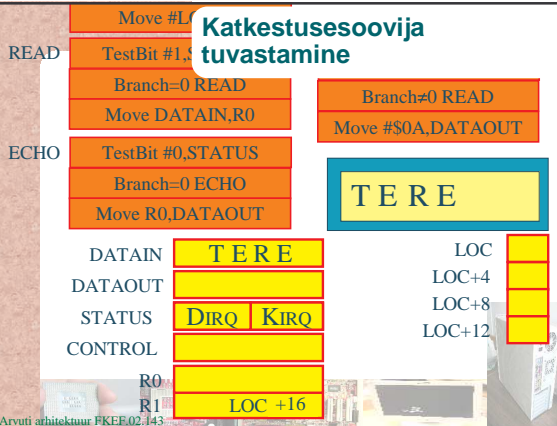
Arvuti arhitektuur FKKEE.02.143

Katkestused paljude seadmega süsteemides

- Kuidas protsessor tuvastab katkestusesoovija?
- Kui katkestuse protseduurid on eri seadmete jaoks erinevad, kust protsessor teab milliselt adressilt katkestuse protseduuri alustada?
- Mis juhtub siis, kui seade A soovib katkestust ajal kui protsessor juba täidab seadme B katkestusesoovi?
 - Kas esimese katkestuse täitmise ajal tuleks uut katkestust hakata kohe teenindama?
 - Või oodata esimese täitmist ja alles siis hakata teenindama uut?
- Mis juhtub siis, kui katkestusesoovid laekuvad üheaegselt?

Arvuti arhitektuur FKKEE.02.143

Katkestusesoovija tuvastamine



Arvuti arhitektuur FKKEE.02.143

Katkestuse vektor

- Katkestust sooviv seade võib ennast ka otseselt protsessorile tutvustada, saates protsessorile siini vahendusel spetsiaalse koodi.
- Selleks koodiks võib olla näiteks katkestuse teenindamise protseduuri aadress
- ... või osa sellest (4-8 bitti). Ülejäänu lisab protsessor ise.
- Kui sellesse kohta panna tegelikule katkestuse-programmile viitav hargnemise aadress, siis anname programmeerijale täiendavalt veidi lisavõimalusi koodi kirjutamisel.
- **Oluline on, et protsessor oleks valmis seda katkestuse vektori infot vastu võtma**

Arvuti arhitektuur FKFE02.143



Interrupt nesting

- Eelnevalt soovitasime, et ühe katkestuse täitmisel peame teiste katkestuste täitmisele võtmise blokeerima
- Alati pole see võimalik
 - näide reaalaja kellast
- Vajalik paika panna **prioriteedid**
 - katkestuse soov kõrgema prioriteediga seadmelt tuleks aktsepteerida isegi siis, kui parasjagu mõne madalama prioriteediga seadme katkestust täidetakse

Arvuti arhitektuur FKFE02.143



Prioriteedid

- Protsessori prioriteediks on tema poolt parasjagu täidetava programmi prioriteet
- Kui katkestust soovitakse kõrgema prioriteediga seadme poolt, siis katkestus võetakse täitmiseks
- Katkestuse täitmise ajaks saab protsessor nüüd katkestust soovinud seadme prioriteeditaseme
- Kui katkestust soovitakse madalama prioriteediga seadme poolt, siis katkestus ei võeta täitmiseks
- Prioriteedid kodeeritakse protsessori staatusregistri paari bitti

Arvuti arhitektuur FKFE02.143

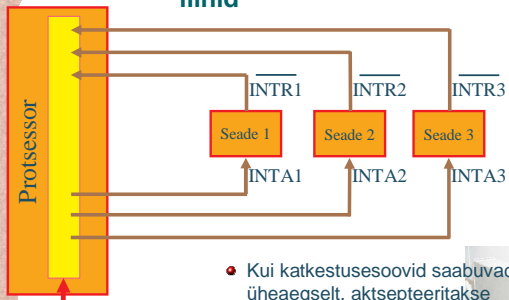


Prioriteedid

- Protsessori staatusregistrit saavad muuta käsud, mida antakse *supervisor* moodis
- Protsessor on selles moodis ainult operatsioonisüsteemi käskude käivitamisel
- Kasutajamoodil neid õigusi ei ole
- Rakendusprogrammi täitma asudes lülitub protsessor kasutajamoodi
 - seega ei saa rakendusprogramm juhuslikult ega tahtlikult prioriteete muuta
 - privilegeeritud käsu käivitamise katse kutsub välja spetsiaalse katkestuse *privilege exception*

Arvuti arhitektuur FKEE.02.143

Individaalsed katkestuse liinid



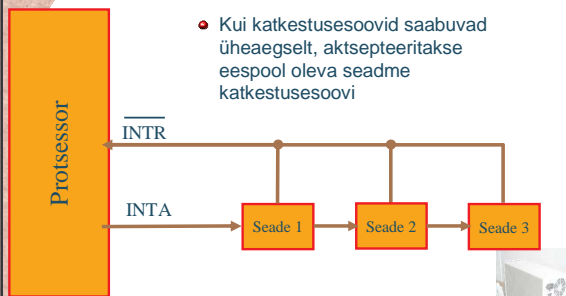
- Kui katkestusesoovid saavad üheaegselt, aktsepteeritakse kõrgema prioriteediga katkestusesoovi

Prioriteetide haldamise ahel

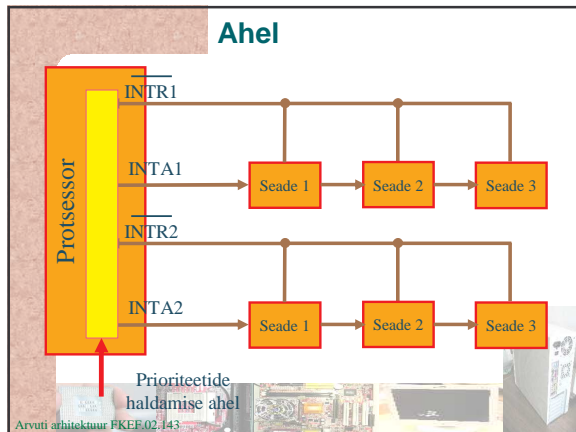
Arvuti arhitektuur FKEE.02.143

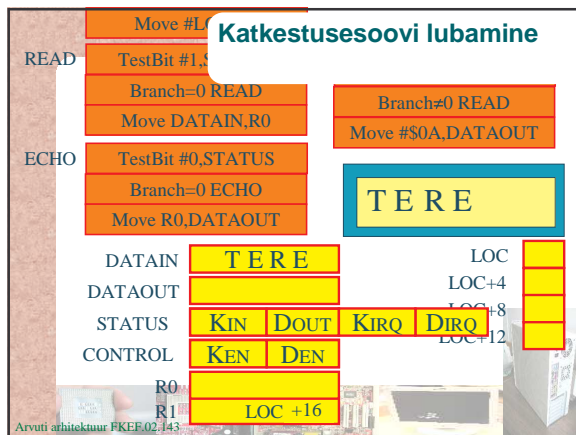
Ahel

- Kui katkestusesoovid saavad üheaegselt, aktsepteeritakse eespool oleva seadme katkestusesoovi



Arvuti arhitektuur FKEE.02.143

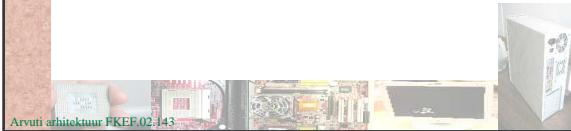




- ### Erandid (*Exceptions*)
- *Recovery from errors*
 - Näide: mida teha, kui programm käsib nulliga jagada?
 - *Debugging*
 - *trace*
 - *breakpoint*
 - *Priviledge exception*
- Arvuti arhitektuur FKEE02.143

Kasutatud kirjandus

- Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer organization 5th edition (2002) 805 p.



Arvuti arhitektuur FKKEE.02.143
