

FKEF.02.143

Arvuti arhitektuur

Computer Architecture and Organization



dotsent
Toomas Plank

©Toomas Plank, 2008

FKEF.02.143

Sisend/väljund

3. loeng







12. loeng,
16. mai 2008

©Toomas Plank, 2008

Sissejuhatus

- I/O ahelate liideste disain
- Siinistandardid
 - PCI
 - SCSI
 - USB



Arvuti arhitektuur FKEF.02.143

Liidese ahelad

- Liidese ühes otsas on siini signaalid
 - aadress
 - andmed
 - juhtsignaalid
- Liidese teises otsas on andmete ja juhtsignaalide I/O-seadmele edastamiseks vajalik
- Seda teist poolt kutsutakse pordiks (väratiks)
 - rööpvärat edastab 8/16 bitti üheaegselt
 - jadavärat edastab andmeid bitthaaval
 - siinipoolne ots on mõlemal liidesel ühesugune

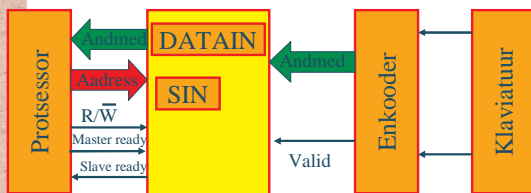
Arvuti arhitektuur FKFE.02.143

I/O liidese ülesanded

- Salvestuspuhver vähemalt ühe sõna pikkuste andmete salvestamiseks
- Staatuslipud, mida suudab protsessor lugeda
 - sisendlipp kõrge siis, kui andmed sisendis
 - väljundlipp kõrge siis, kui seade on valmis andmeid vastu võtma
- Aadressi dekodeerimise ahelad, saamaks aru kui tema poole pöördatakse
- Siinile juhtsignaalide genereerimise ahelad
- Vajadusel teeb teisendused (näiteks jadaà rööp ja vastupidi)

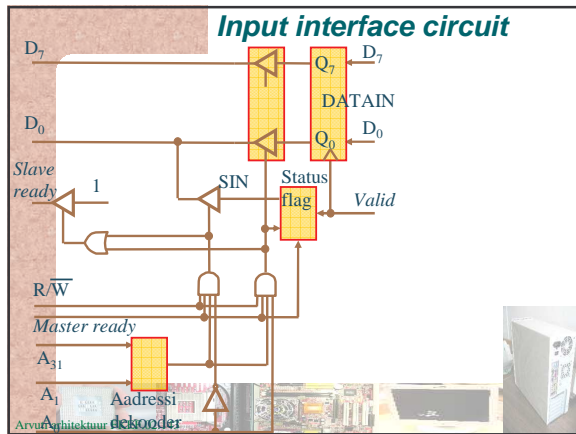
Arvuti arhitektuur FKFE.02.143

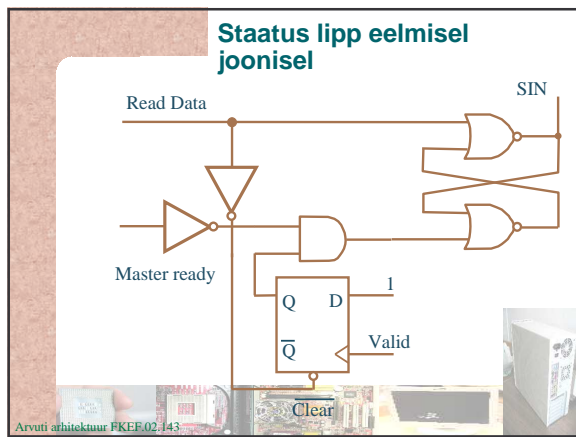
Klaviatuur koos liidesega

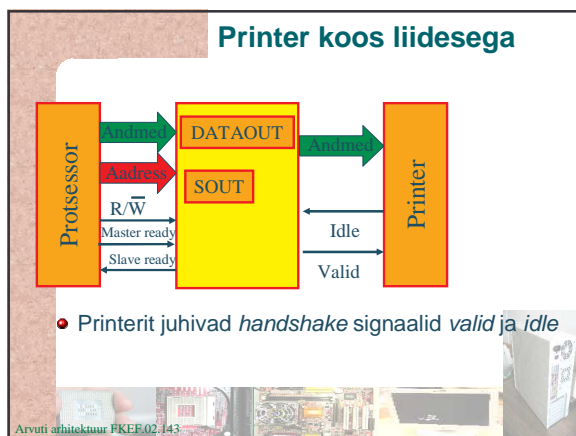


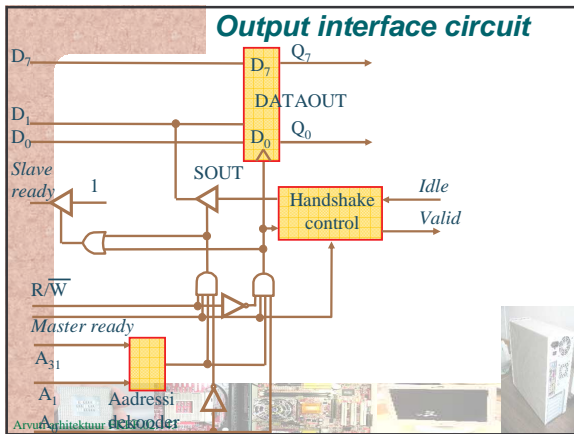
- Klaviatuuril on lülitid, mille vajutamisel tekib elektriline ühendus
- Võnkumiste summutamise ahel
- Signaal kodeeritakse ASCII koodiks ja edastatakse liidesele

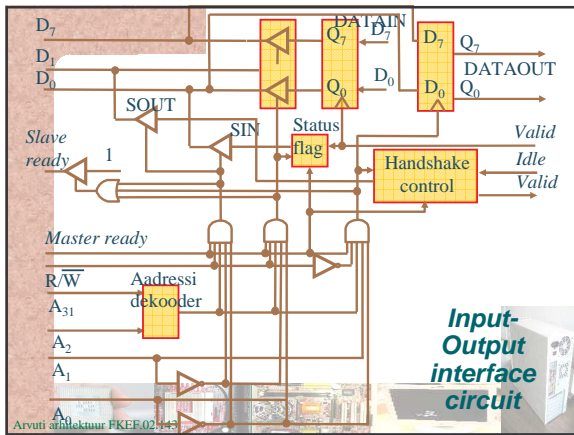
Arvuti arhitektuur FKFE.02.143











Standard I/O interfaces

- Eelnevast nägime, et siini kujundamisel on mitmeid alternatiive
- Põhimõtteliselt võiks oma liides olla iga I/O seadme ja protsessori kombinatsiooni jaoks...
- Kasulik oleks kokkuleppele jõuda ja standardiseeritud lahendusi kasutada
- Protsessoriga on otse ühendatud ainult mõned väga kiired seadmed (näiteks põhimälu)
- Elektrilistel põhjustel on teised seadmed ühendatud teise siiniga, laiendussiiniga, mis on protsessori siiniga ühenduses silla (*bridge*) abil

Sild (Bridge)

- Sild ühendab laiendussiinil paiknevad seadmed protsessori siiniga nii, et need paistavad seal nagu nad oleksid ühendatud protsessori siinile
- Sild tõlgib ühe siini signaalid ja protokollid teise siini signaalideks ja protokollideks
- Ainsaks erinevuseks on väike täiendav viivis signaalide liikumisel
- Protsessori siini ehitus on tihedalt seotud protsessori ehitusega, laiendussiinil selliseid piiranguid ei ole.
 - Seega saab kasutada standardiseeritud signaalide-skemi

Arvuti arhitektuur FKEE.02.143

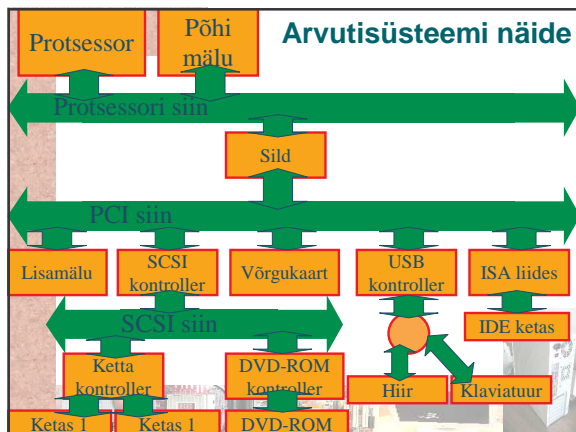
Siinistandardid

- ISA (*Industry Standard Architecture*)
 - IBM PC AT'ga koos loodud standard
- PCI (*Peripheral Component Interconnect*)
- SCSI (*Small Computer System Interface*)
- USB (*Universal Serial Bus*)

Kuidas need standardid ühes arvutis koos töötavad, illustreerib järgnev joonis

- PCI – laiendussiin emaplaadi peal
- SCSI ja USB lisaseadmete ühendamiseks nii arvuti korpuse sees kui sellest väljaspool

Arvuti arhitektuur FKEE.02.143



PCI (Peripheral Component Interconnect)

- Tutvustati esimest korda aastal 1992
- Omadused:
 - odav
 - protsessori tüübist sõltumatu standard
 - PCI-siini seadmed paistavad protsessorile, nagu nad oleksid ühendatud protsessori enda siinile
 - Nende aadressid on kirjeldatud protsessori mälu aadressiruumis
 - *plug-and-play*

Arvuti arhitektuur FKFE02.143



PCI (Peripheral Component Interconnect)

- PCI on põhiliselt mõeldud suhtluseks, kus mälusse tuleb kirjutada või mälust lugeda palju andmesõnu järjest
- Siin toetab kolme sõltumatut aadressivälja
 - mälu, I/O ja konfiguratsiooni aadressiväli
 - viimast kasutatakse PCI *plug-and-play* funktsionaalsuse andmiseks
 - Aadressiga kaasnev 4-bitine käsk annab teada, millist aadressivälja parasjagu kasutatakse

Arvuti arhitektuur FKFE02.143



PCI (Peripheral Component Interconnect)

- Andmevahetus käib üsna sarnaselt sünkroonse andmevahetuse peatükis räägituga.
Peamised erinevused
 - Aadressiinfot hoitakse siinil ainult ühe takti vältel
 - *Slave*-seade salvestab selle oma sisemises puhvris
 - Järgmiste tsüklike ajal saab neid liine kasutada juba andmete edastamiseks!
 - Kasutatakse nii *slave-ready* kui ka *master-ready* signaali analoogi
- Siini *master*'iks on kas protsessor või DMA kontroller. PCI terminoloogias **initsiaator**
- Adreseeeritav seade on **märklaud**
- Signaalid on kirjeldatud järgmise slaidi tabelis
- # nimetuse lõpus näitab, et signaal on aktiivne madalapingelises olekus

Arvuti arhitektuur FKFE02.143

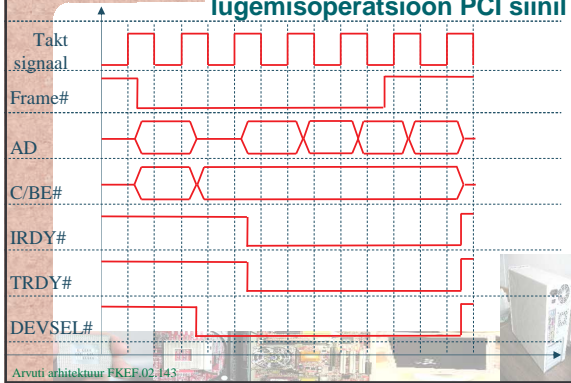


PCI signaalid

Nimi	Ülesanne
CLK	33 MHz või 66 MHz taktisignaali
FRAME#	Initsiaatori poolt saadetakse signaal, mis näitab ülekande kestust
AD	32(64) aadressi/andme liini
C/BE#	4(8) käsu/Byte-enable liini
IRDY#	Initsiaator valmis signaal
TRDY#	Sihtmärk valmis signaal
DEVSEL#	Seadme vastus, mis näitab, et ta on andmevahetusesoojust aru saanud ja andmevahetuseks valmis
IDSEL#	Initialization Device Select

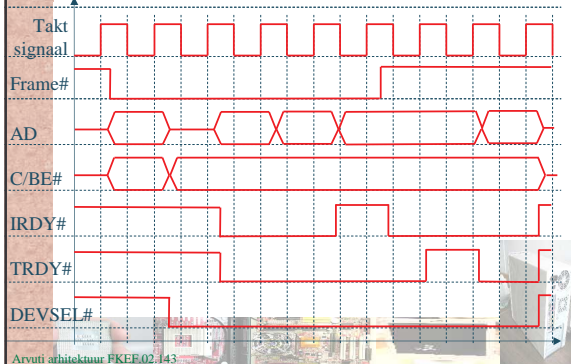
Arvuti arhitektuur FKEE.02.143

Näide: nelja 32-bitise sõna lugemisoperatsioon PCI siinil



Arvuti arhitektuur FKEE.02.143

Lugemisoperatsioon PCI siinil, kui seadmed ei saa nii kiiresti hakkama kui eelmisel slaidil



Arvuti arhitektuur FKEE.02.143

PCI seadme konfigureerimine

Seadme esmakordsel ühendamisel arvutiga on vaja teha terve rida toiminguid

- ISA kaardi puhul pidi kasutaja seadme silluseid ja/või lüliteid kaardil ja terve rida parameetreid seadme juhtprogrammis
- PCI seadmetesse on juba sisse ehitatud konfigureerimise ROM-kiip, kus hoitakse infot selle seadme kohta
- Kõigi seadmete konfiguratsiooni ROM'id on nähtavad PCI konfiguratsiooni aadressi ruumis
- PCI initsialiseerimisetarkvara loeb seda alati, kui süsteem sisse lülitatakse või algladimise teeb
 - saab teada, mis seadmega on tegemist
 - saab teada seadme valikväärtused ja karakteristikud

Arvuti arhitektuur FKFE.02.143

PCI seadme konfigureerimine (2)

- Seadmele omistatakse initsialiseerimise käigus aadress
 - esmase pöördumise jaoks tuleb seega tarvitada muud moodust
 - kasutatakse IDSEL# signaali
- Igal seadmel on signaal *Input Device Select*
- Emaplaadil on seadme IDSEL# viik ühenduses ühega ülemisest 21 aadressist AD11 ... AD31
- Seega saab seadme valida seades selle konkreetse viigu aktiivseks (ülejäanud 20 aadressi passiivseks) ja väljastades konfigureerimise käsu
- Alumisi aadressiliine AD0 ... AD11 kasutatakse operatsiooni tüübi kirjeldamiseks ja konfiguratsiooni ROM sisu edastamiseks
- PCI-seadmete arv on seega maksimaalselt 21

Arvuti arhitektuur FKFE.02.143

PCI seadme konfigureerimine (3)

- PCI konfigureerimise tarkvara skaneerib läbi kõik 21 võimalikku asukohta, selgitamaks välja millised seadmed on reaalselt olemas
- Iga seade võib küsida aadressi I/O aadressiruumis või mälu aadressiruumis – see info kirjutatakse seadme registrisse
- Ühtlasi määratakse ka katkestuse prioriteet ja millist neljast võimalikust PCI-katkestusest seade kasutada tohib
- Kui seade vajab initsialiseerimist, kirjutatakse initsialiseerimise kood seadme (teise) ROM'i.
- PCI tarkvara loeb seda koodi, käivitab ja teeb vajaliku initsialiseeringu
- Kasutaja hooleks jääb ainult seade õigesse pesasse pista ja pingestada

Arvuti arhitektuur FKFE.02.143

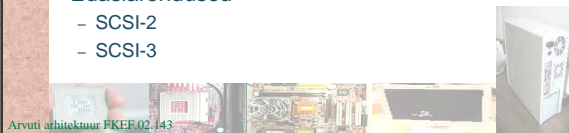
PCI elektrilised parameetrid

- PCI võib töötada nii 5V kui ka 3,3V pingel
- Emaplaadid võivad olla mõeldud töötamiseks nii ühe kui teisega



Small Computer System Interface (SCSI)

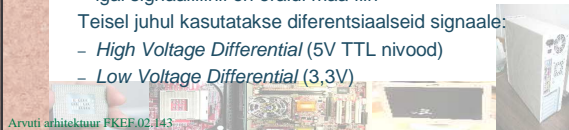
- SCSI viitab Ameerika Rahvusliku Standardiseerimise Instituudi (ANSI) standardiga kirjeldatud (paraleel) siinile
- Originaalspetsifikatsiooni järgne SCSI siin
 - võimaldas ühendada seadmeid 50-traadise kaabli abil,
 - kaabli pikkuseks kuni 25 meetrit
 - andmeedastuskiirus kuni 5 Mb/s
- Edasiarendused
 - SCSI-2
 - SCSI-3



SCSI (2)

Tehnilised üksikasjad võivad olla lahendatud mitut moodi:

- Andmeliinide arv
 - 8 liini (*narrow bus*)
 - 16 liini (*wide bus*)
 - Elektriline skeem
 - signaaliiniidel on ühine maa-liin (*single-ended transmission*)
 - igal signaaliiniil on eraldi maa-liin
- Teisel juhul kasutatakse diferentsiaalseid signaale:
- *High Voltage Differential* (5V TTL nivood)
 - *Low Voltage Differential* (3,3V)



SCSI (3)

- Kitsa siini (*narrow bus*) korral saab ühendada kuni 8 seadet
- Laia siini (*wide bus*) korral saab ühendada kuni 16 seadet
- Andmeedastuskiirused
 - 5 – 160 Mb/s
 - 320 Mb/s (uuemad SCSI versioonid)
 - 640 Mb/s
- Maksimaalne reaalselt saavutatav kiirus on sõltuvalt
 - kaabli pikkusest
 - ja
 - seadmete arvust siinil
- Maksimaalse andmeedastuskiiruse saamiseks piiratakse kaabli pikkust
 - 1,6 m SE signaalide korral
 - 12 m LVD signaalide korral
- Liinide arv kaablis ja kontaktide arv pistikus
 - 50
 - 68
 - 80

Arvuti arhitektuur FKFE.02.143



SCSI (4)

- SCSI seadmete aadressiruum protsessori siinile ei paista
- SCSI seadmed ühendatakse SCSI-siiniga
- SCSI siin on protsessori siiniga ühenduses SCSI kontrolleri abil
- SCSI seadme ja mälu vaheline suhtlus käib SCSI kontrolleri DMA kanali kaudu
- Pakett võib sisaldada
 - andmeblokki
 - protsessoripoolseid käskse seadmele
 - seadme staatusinfot

Arvuti arhitektuur FKFE.02.143



SCSI (5)

- SCSI siinile ühendatud kontrolleri võib olla kas
 - initsiaator
 - või
 - märklaud
- Initsiaator suudab välja valida sihtmärgi ja jagada sellele korraldusi (selline seade on näiteks SCSI kontrolleri)
- Sihtmärk täidab initsiaatorilt saadud korraldusi (selline seade on näiteks ketta kontrolleri)
- Initsiaator loob sihtmärgiga loogilise ühenduse, mida saab vastavalt vajadusele peatada ja uuesti aktiveerida.
- Peatatud ühenduse korral saavad siini kasutada teised seadmed
- Andmevahetust kontrollib SCSI-siinil alati **sihtmärk**

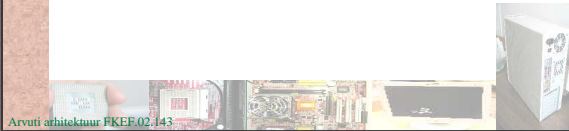
Arvuti arhitektuur FKFE.02.143



SCSI andmevahetus

- Andmevahetust kontrollib SCSI-siinil alati **sihtmärk**
 - esmalt võtab initsiaator kontrolli siinil endale
 - peale arbitreerimise võitmist valib välja sihtmärgi
 - seejärel annab initsiaator kontrolli üle sihtmärgile
 - sihtmärk alustab andmevahetust saamaks initsiaatorilt teada täitmisele tulevat käsku

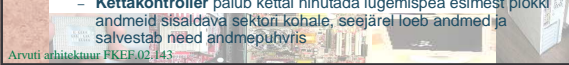
Arvuti arhitektuur FKFE.02.143



Näide: lugemisoperatsioon SCSI kettalt (1)

- Oletame, et protsessor tahab lugeda kettalt kaks plokki andmeid, mis paiknevad kahes teineteisest kaugel paiknevas sektoris
- Protsessor saadab SCSI-kontrollerile käsu, millele järgnevad alljärgnevad tegevused:
 - esmalt võtab **SCSI kontroller** (initsiaator) kontrolli siinil endale
 - peale arbitreerimise võitmist valib välja sihtmärgina **kettakontrolleri**
 - seejärel annab initsiaator kontrolli üle **kettakontrollerile**
 - **kettakontroller** alustab andmevahetust ja **SCSI kontroller** saadab talle lugemiskäsu koos soovivate andmete kirjeldusega
 - **Kettakontroller**, mõistes vajadust esmalt kettaga suhelda, peatab andmevahetuse SCSI-siinil (võimaldades nii teistel seadmetel omavahele suhelda)
 - **Kettakontroller** palub kettal nihutada lugemispea esimest plokki andmeid sisaldava sektori kohale, seejärel loeb andmed ja salvestab need andmepuhvis

Arvuti arhitektuur FKFE.02.143



Näide: lugemisoperatsioon SCSI kettalt (2)

- ...
 - **Kettakontroller** palub kettal nihutada lugemispea esimest plokki andmeid sisaldava sektori kohale, seejärel loeb andmed ja salvestab need andmepuhvis
 - **Kettakontroller** (sihtmärki) küsib kontrolli siini üle tagasi. Peale arbitreerimise võitmist taastab ühenduse **SCSI-kontrolleriga** (initsiaatoriga)
 - Nüüd edastab **kettakontroller** SCSI-kontrollerile loetud andmed ja peatab uuesti ühenduse siinil
 - **Kettakontroller** palub kettal nihutada lugemispea teist plokki andmeid sisaldava sektori kohale, seejärel loeb andmed ja salvestab need andmepuhvis
 - **Kettakontroller** (sihtmärki) küsib kontrolli siini üle tagasi. Peale arbitreerimise võitmist taastab ühenduse **SCSI-kontrolleriga** (initsiaatoriga)
 - Nüüd edastab **kettakontroller** SCSI-kontrollerile teisest sektorist loetud andmed. Peale seda lõpetab **kettakontroller** ühenduse siinil
 - **SCSI-kontroller** salvestab andmed mälus (vastavalt andmete laekumisele), kasutades DMA-kanalit
 - **SCSI-kontroller** saadab protsessorile katkestussoovi, informeerimaks teda lugemiskäsu edukast täitmisest

Arvuti arhitektuur FKFE.02.143



SCSI-siini signaalid		
Kategooria	Nimi	Funktsionaalsus
Andmed	-DB(0) ...	Üks bait andmeid või seadme valimise info
	-DB(7)	
	-DB(P)	
Faas	-BSY	Aktiivne, kui siin pole vaba
	-SEL	Aktiivne, siini valimisel ja uuestivalimisel
Info tüüp	-C/D	Aktiivne, kui edastatakse juhtimisinfot
	-MSG	Aktiivne, kui edastatakse teadet
Käepigistus	-REQ	Aktiivne, kui märklaud küsib luba andmeedastuseks
	-ACK	Aktiivne, kui initsiaator lõpetas admeedastusoperatsiooni
Ülekande suund	-I/O	Aktiivne, kui initsiaatori mäta otsast vaadates on tegemist sisendoperatsiooniga
Muud	-ATN	Aktiivne, kui initsiaator tahab saata teadet sihtmärgile
	-RST	Sunnib kõiki seadmeid ennast siinilt lahti ühendama ja minema start-up seisu

SCSI-siini signaalid

Kommentaariid eelmise tabeli juurde

- Miinusmärk signaali tähise ees viitab sellele, et aktiivne olek (loogiline 1) vastab madalapingelisele olekule
- Siinil aadressiliine ei ole, kasutatakse andmeliine
- Iga seadme jaoks on üks "oma" andmeliin, mida kasutatakse selle seadme adresseerimiseks
- Seega saab adresseerida niipalju seadmeid, kui on andmeliine
- Kontroller saadab oma aadressi või sihtmärgi aadressi siinile, aktiveerides vastava liini (ehk siis korraga saavad siinil aktiivsed olla mitme seadme aadressid)
- Peale ühenduse loomist pole aadressiinfot enam tarvis, seega saab siini kasutada andmete edastamiseks

SCSI-siini suhtlusfaasid

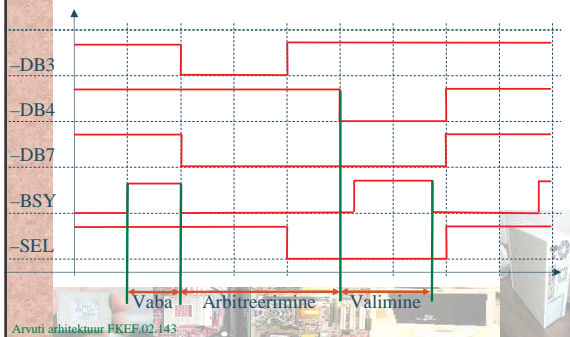
- Arbitreerimine (*arbitration*)
- Valimine (*selection*)
- Info ülekanne (*information transfer*)
- Taasvalimine (*reselection*)

SCSI-siini arbitreerimine

- Seni kuni -BSY-signaal on kõrges olekus (mitteaktiivne), on siin vaba
- Suhtluse alustamiseks viib kontrolleri -BSY alla (aktiivseks) ja andmesüsteemile saadetava aadressiinfoga teatab, kes ta selline on
- Siin kasutab jagatud arbitreerimise meetodit, kus kontrolleri 7 on kõrgeima prioriteediga
- Kõrgeima prioriteediga seade võidab arbitreerimise, kõik teised ühendavad end siinilt lahti ja ootavad -BSY mitteaktiivseks minekut
- Võitnud seade saadab siinile sihtmärgi aadressi ja alustab sellega valiku-faasi

Arvuti arhitektuur FKKEE.02.143

SCSI-siini arbitreerimine



SCSI-siini valik

- ...
- Võitnud seade hoiab jätkuvalt aktiivsetena nii enda aadressi kui ka -BSY
- Oma soovi sihtmärgiga suhelda näitab initsiaator nüüd -SEL-signaali aktiivseks viimisega ja sihtmärgi aadressi aktiveerimisega
- Kõik teised seadmed peavad -SEL-signaali aktiveerumise järel oma aadressiinfo siinilt eemaldama (kui nad seda juba teinud pole)
- Peale sihtmärgi aadressi seadmist deaktiveerib võitja -BSY signaali ja seadmed uurivad aadressi
- Sihtmärk saab nüüd aru, et temaga tahetakse suhelda ja arusaamise kinnituseks aktiveerib ise -BSY
- Sellega on valikuprotseduur lõppenud ja sihtmärgil vaba voli andmeid saada/vastu võtta (tema kontrollib siini)

Arvuti arhitektuur FKKEE.02.143

SCSI-siini andmeedastus

... sihtmärgil on vaba voli andmeid saada/vastu võtta (tema kontrollib siini).

- Info koosneb käsust (initsiaatorilt sihtmärgile), staatusinfost (sihtmärgilt initsiaatorile) ja andmevoost (suund sõltub käsust).
- Käepigistussignaali kontrollivad info ülekannet (analoogiliselt asünkroonse andmevahetuse peatükis räägituga).
- Siini *master*iks on praegu sihtmärk, *master-ready* ja *slave-ready* signaale asendavad *-REQ* ja *-ACK*.
- Sihtmärk aktiveerib *-I/O*, kui andmeid saadetakse sihtmärgilt initsiaatorile.
- Sihtmärk aktiveerib *-C/D*, kui andmeid saadetakse juhtinfot või staatusinfot. Ja andmeedastuse korral deaktiveerib selle liini.
- Andmeedastuse lõpus laseb sihtmärk *-BSY* vabaks, andmaks teistele seadmetele võimaluse suhtluseks.
- Vajadusel saab ühenduse taastada Taasvaliku protseduuriga.

Arvuti arhitektuur FKFE02.143

SCSI-siini andmeedastus *high-speed* moodis

- *High-speed* moodis kasutatakse *double-edge docking* või *double transitions* tehnikat
 - tavalise käepigistuse juures oodatakse käepigistussignaali(de) muutumist kõrgest->madalaks ja sellele järgnevat muutumist madalast->kõrgeks
 - siin piisab ühest nimetatud muutumisest
 - sellega saavutatakse kahekordne andmeedastuskiirus

Arvuti arhitektuur FKFE02.143

SCSI-siini taasvalimine

Kui loogiline ühendus on peatatud ja sihtmärk on valmis selle taastama, siis

- Sihtmärk alustab arbitreerimistsükliga
- Peale selle võitmist valib initsiaatori analoogiliselt eelpoolkirjeldatuga
- Erinevuseks on see, et *-BSY* signaali aktiveerib nüüd initsiaator, mitte sihtmärk.
- Andmevahetuse alustamiseks peab initsiaator andma kontrolli tagasi sihtmärgile.
- Selleks aktiveerib *-BSY* signaali ka sihtmärk ja mõningase aja möödudes initsiaator lõpetab omapoolse *-BSY* signaali aktiivsena hoidmise.
- Sellega on ühendus taastatud ja andmevahetus võib jätkuda

Arvuti arhitektuur FKFE02.143

Kokkuvõtte (SCSI protokollist)

Eelpoolkirjeldatud signaali-skeem pakub mehhanismi kahe seadme vaheliseks suhtluseks ja teadete vahetamiseks

- Ühendust saab peatada ja taaslustada suvalisel ajal
- SCSI standard kirjeldab struktuuri ja hulga pakettide sisu, mida kontrolleri eri situatsioonides kasutavad
- Initsiaator kasutab neid protsessori käskude saatmiseks sihtmärgile
- Sihtmärk vastab staatusinfoga ja andmete ülekandmisega
- Viimast kontrollib **sihtmärk**, kuna tema teab, millal on andmed olemas ja millal on tarvis ühendust peatada või taaslustada

Arvuti arhitektuur FKEF.02.143



Universal Serial Bus (USB)

- Sündis firmade (Compaq, HP, Intel, Lucent, Microsoft, Nortel Networks, Philips) koostöös
- Loodi selleks, et saada
 - lihtne,
 - odavja
 - lihtsalt kasutatav süsteem
- Peab kasutaja vabastama lõpliku arvu portide probleemist
- Peab võimaldama nii kiiret kui ka aeglast andmeedastust
 - 1,5 Mb/s (*low speed*)
 - 12 Mb/s (*full speed*)
 - 480 Mb/s (USB 2.0: *high speed*)
- Peab olema kasutajasõbralik (*plug-and-play*)

Arvuti arhitektuur FKEF.02.143



Pordi piirangud

- Jada- ja rööpvärateid on arvutis lõplik hulk
- Uue värti lisamiseks tuleb arvuti välja lülitada ja kaan maha keerata
- Lisaks veel konfigureerimise probleemid
- USB peab võimaldama lisada seadmeid suvalisel hetkel, ilma vajaduseta kaan maha keerata

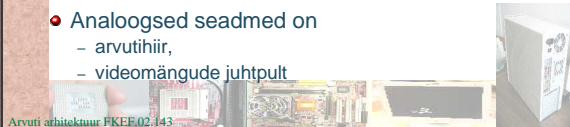
Arvuti arhitektuur FKEF.02.143



Seadme karakteristikud (1)

- Vajadus ühendada väga erinevaid seadmeid
- Klaviatuur:
 - üks klahvivajutus tähendab ühte baiti andmeid
 - klahvivajutust ei saa ette prognoosida (asünkroonne info)
 - kõik info peab adekvaatselt ka protsessorini jõudma
 - andmevoog madal, max 100 baiti sekundis ehk alla 1000 biti sekundis
- Analoogsed seadmed on
 - arvutihiir,
 - videomängude juhtpult

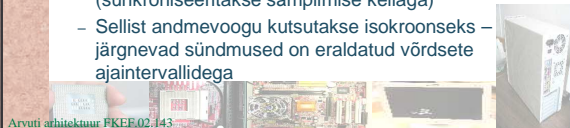
Arvuti arhitektuur FKEF.02.143



Seadme karakteristikud (2)

- Mikrofon:
 - mikrofonis tekib helist analoogsignaali
 - enne arvutisse edastamist see digitaliseeritakse analoog-digitaal muundi abil
 - kõlaritesse saatmisel toimub vastupidine protsess – digitaalsest infost tehakse digitaal-analoog muundi abil analoogsignaali
 - Sämplimise käigus tekib pidev andmevoog, mis saabub arvutisse regulaarsete ajaintervallide järel (sünkroniseeritakse sämplimise kellaga)
 - Sellist andmevoogu kutsutakse isokroonseks – järgnevad sündmused on eraldatud võrdsete ajaintervallidega

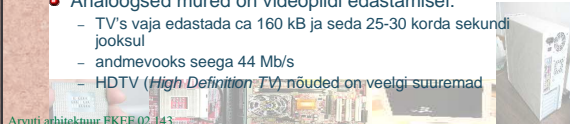
Arvuti arhitektuur FKEF.02.143



Seadme karakteristikud (3)

- Inimkõne jaoks on vajalik sämplimissagedus 8 kHz (sellega salvestatakse signaalid sagedusega kuni 4 kHz)
- Digitaalse heli standardiks on 44,1 kHz (meie videoloengud on tehtud sellise sämplimissagedusega)
- Iga sämpel sisaldab 4 baiti infot.
- Kokku saame $44,1\text{kHz} * 4\text{B} * 8\text{b/B} = 1,4\text{ Mb/s}$
- **Oluline on ajastus**, ajalised nihked ei ole aktsepteeritavad
- Samal ajal on üksikute samplite kadumine aktsepteeritav – kõrv seda ei taju!
- Analoogsed mured on videopildi edastamisel:
 - TV's vaja edastada ca 160 kB ja seda 25-30 korda sekundi jooksul
 - andmevooks seega 44 Mb/s
 - HDTV (*High Definition TV*) nõuded on veelgi suuremad

Arvuti arhitektuur FKEF.02.143



Plug-and-play

- Seadmeid peab olema võimalik lisada, ilma et tuleks arvuti välja lülitada või lausa kaan eemaldada
- Süsteem peab aru saama, kui uus seade lisati
 - tuvastab lisatud seadme automaatselt
 - leiab ise juhtprogrammi
 - loob suhtluskanali
- Süsteem peab aru saama, kui mõni seade eemaldati
 - ehk teisisõnu, süsteem peab omama ülevaadet hetkel kasutuses olevatest seadmetest

Arvuti arhitektuur FKFF.02.143



USB arhitektuur

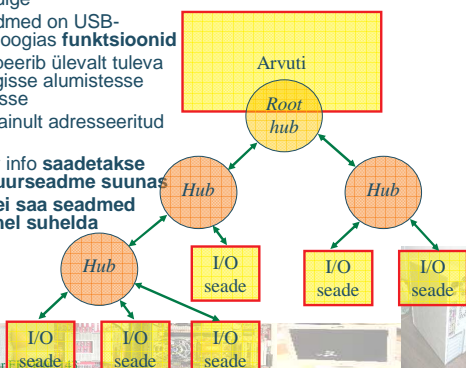
- Suur andmevoog eeldaks paljusid traate andmete paralleelseks edastamiseks
 - kallim hind
 - ebamugavam
 - pika vahemaa korral *data skew*
- Ülkirjeldataud murede tõttu kasutatakse jadaühendust
 - odav
 - paindlik
 - taksageduse üles ajamisega (*skew* ei esine jadaühenduses) saavutatakse väga suur andmeedastuskiirus (1,5 – 480 Mb/s)
- Suure arvu seadmete ühendamiseks kasutatakse puustruktuuri

Arvuti arhitektuur FKFF.02.143



- Hub'id on jaotuspunktideks
- Root-hub ühendab kogu puu arvuti külge
- I/O seadmed on USB-terminoloogias **funktsioonid**
- Hub kopeerib ülevalt tuleva info kõigisse alumistesse kanalitesse
- Vastab ainult adresseeritud seade
- All tulev info **saadetakse ainult juurseadme suunas**
- Seega ei saa seadmed omavahel suhelda

USB puustruktuur



Arvuti arhitektuur

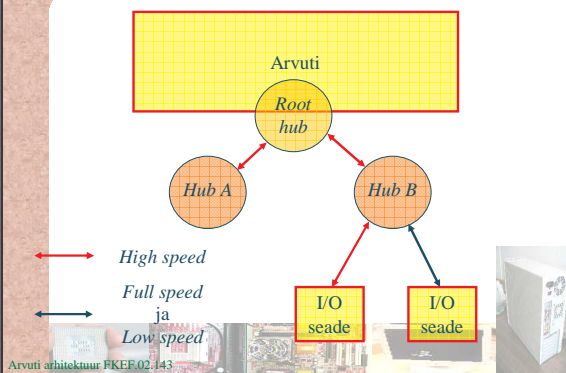


USB arhitektuur

- USB töötab päringusüsteemil (*polling*)
- Seade saadab infot ainult peale juhtarvutiil loa saamist
 - selline lähenemine vältib konflikte ja signaalide vastasmõju, kuna kaks seadet samaaegselt infot ei saada
 - seega saavad jaoturid (*hub*) olla lihtsad ja odavad seadmed
- Erandiks *High-speed* seadmetega suhtlus USB 2.0 protokollis
 - kui mõni aeglane seade ka suhelda tahab, saadetakse temaga ühenduses oleva jaoturini *high-speed* info, sealt edasi aeglaselt
 - Viimane võtab palju aega, seega saab lubada kiiretel seadmetel oma andmeid edasi vahetada
 - Oeldakse, et siin on sel ajal jagatud kiire ja aeglase andmevahetuse vahel
 - Jagatud moodis andmevahetusele eelneb ja järgneb spetsiaalne käsk

Arvuti arhitektuur FKKEE.02.143

Jagatud siin (*Split-bus*)



Arvuti arhitektuur FKKEE.02.143

Adresseerimine (1)

- Kui me ühendame USB seadme arvuti külge, siis on protsessori siiniga ühenduses ainult *root-hub*. See paistab ettepoole ühe seadmena.
- *Root-hub* saadab kõik pakettid edasi USB-puud pidi
- Igal seadmel ja jaoturil on oma 7-bitine address
- See address ei ole kuidagi seotud protsessori siinil kasutatavate adressidega
- Seadmeid saab jaoturi taha ühendada piiramatult

Arvuti arhitektuur FKKEE.02.143

Adresseerimine (2)

- Esmasel ühendamisel on seadme aadressiks 0
- Jaotur suudab detekteerida temaga ühendatud uue seadme ja lisab selle info enda staatusinfo hulka
- Arvuti küsitleb perioodiliselt kõiki jaotureid ja saab staatusinfost teada uue seadme ühendamisest
- Käivitub standardprotseduur:
 - loeb info seadme kohta seadme ROM mälust,
 - reset signaal seadmele ja soovitud konfiguratsioonimuudatused
 - unikaalse aadressi omistamine
- Nüüd on uus seade töökorras

Arvuti arhitektuur FKFE.02.143

Adresseerimine (3)

- Seadme väljalülitamisele järgneb sarnane protseduur:
 - vastav jaotur teatab sellest USB-tarkvarale
 - see omakorda uuendab oma infotabelid
 - Kui väljalülitatavaks seadmeks on jaotur ise, siis ühendatakse lahti kõik selle taga olevad seadmed
- Staatuse, juhtimise ja andmeregistrid seadmes on lõpp-punktideks (*endpoints*)
- Need tuvastatakse 4-bitise koodi abil
- Iga koodile vastab tegelikult kaks lõpp-punkti
 - üks sisendi
 - teine väljundi tarvis
- Lõpp-punktiga 0 on ühendus kogu aeg olemas
- Seda tarvitatakse seadme initsialiseerimisel
- Teisi lõpp-punkte kasutatakse vastavalt vajadusele

Arvuti arhitektuur FKFE.02.143

USB protokoll

- Info on koondatud pakettidesse
 - juhtpaketid
 - andmepaketid
- Iga paketi alguses on paketiidentifikaator (PID) 8 bitti (infot tegelikult 4 bitti)

PID ₀	PID ₁	PID ₂	PID ₃	PID ₀	PID ₁	PID ₂	PID ₃
------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------

- Inverteeritud väärtused on info edastamise õigsuse kontrolliks
- Nende 4 bitiga valitakse üks võimalikest 16 paketitüübist
- Kinnitus pakett (*acknowledge*) koosnebki ainult PID baidist

Arvuti arhitektuur FKFE.02.143

Juhtpakett

- Juhtpakett

8	7	4	5
PID	Address	Lõpp-punkt	CRC5

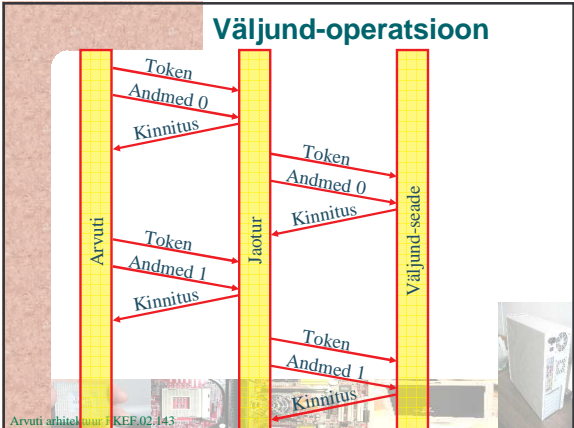
 - Address – seadme 7-bitine aadress
 - Lõpp-punkti kood (4 bitti) seadme sees
 - Cyclic redundancy check* (CRC) 5 bitti rehkendatakse aadressi ja lõpp-punkti alusel ja võimaldab veatuvastust
 - PID väärtus on üks kahest võimalikust, kas IN või OUT paketi jaoks

Andmepakett

- Andmepakett

8	0 – 8192	16
PID	Andmed	CRC16

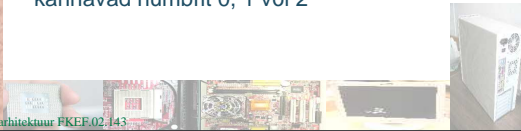
 - Andmed – kuni 8192 bitti
 - Cyclic redundancy check* (CRC) 16 bitti rehkendatakse andmete alusel ja võimaldab veatuvastust
 - PID identifitseerib paketti ja võib olla 0, 1 või 2
 - Nagu näete, pole andmepaketil enam aadressi ega lõpppunkti koodi küljes – see info saadakse juhtpaketest



USB protokoll (2)

- Järjestikused paketid *full-speed* ja *low-speed* režiimides kannavad numbrit 0 või 1
- Kui mõni pakett (juht-, andme- või kinnituspakett) läheb kaotsi, saadetakse kogu jada uuesti
- Paketi jrk-numbri kontroll võimaldab adressaadil ignoreerida duplikaat-pakette
- Järjestikused paketid *high-speed* režiimides kannavad numbrit 0, 1 või 2

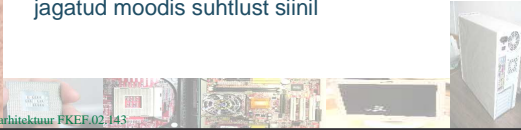
Arvuti arhitektuur FKFE02.143



Sisend-operatsioon

- Sisend-operatsioonid on analoogilised:
 - peremees (*host*) saadab juhtpaketi, küsides andmeid
 - seade vastab kas andmete saatmisega + ACK või
 - öeldes, et tal pole andmeid NAK (*negative acknowledge*)
- Jagatud siini moodis suhtlusele eelneb IN/OUT paketi spetsiaalne juhtpakett mis alustab jagatud moodis suhtlust siinil

Arvuti arhitektuur FKFE02.143

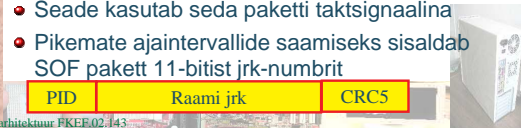


Isokroonne liiklus

- Isokroonseid andmeid saavad või lugevad seadmed vajavad ajainfot
- Selle andmiseks jagatakse suhtluspaketid võrdse pikkusega raamideks
- *Full-speed* režiimis on raamid 1 ms pikkused
- *High-speed* režiimis on raami pikkuseks 125 μ s
- Root-hub genereerib StartOfFrame paketi iga millisekundi järel
- Seade kasutab seda paketti taktsignaalina
- Pikemate ajaintervallide saamiseks sisaldab SOF pakett 11-bitist jrk-numbrit

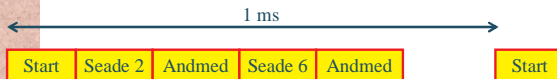
PID	Raami jrk	CRC5
-----	-----------	------

Arvuti arhitektuur FKFE02.143



Isokroonne liiklus (2)

- SOF paketi järel toimub andmevahetus järgemööda kõigi isokroonset andmevahetust vajavate seadmetega



- Kinnitust pole vaja praegu saata,
- samuti pole tarvis muretseda kaotsiläinud andmete pärast
- Peale isokroonsete seadmete teenindamist teenindatakse asünkroonsed seadmed

Arvuti arhitektuur FKEE02.143

Näide andmeedastusele kuluvast ajast

Olgu meil andmepaketis 10 baiti infot

- Andmevahetuse alustamiseks saadame 3-baidise juhtsignaali
- seejärel andmepaketi $1+10+2 = 13$ baiti
- lõpuks 3 baiti takti sünkroniseerimiseks ja jada lõpu tähistamiseks
- Kogu andmevoog seega $19 \cdot 8$ bitti, mis kiirusel 12 Mb/s teeb $12,7 \mu\text{s}$
- Seega on küllaldaselt aega ka teiste seadmetega suhtlemiseks!

Arvuti arhitektuur FKEE02.143

Kasutatud kirjandus

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

Arvuti arhitektuur FKEE02.143
