## L4 video algusest

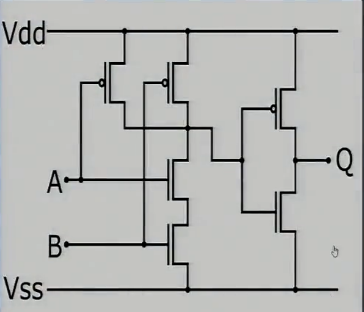
**Milline tehnoloogia on kasutusel tänapäeva arvutites?**

CMOS

**Kas esimesse arvutite põlvkonda kuulusid ENIAC, EDVAC; UNIVAC jt?**

Jah kuulusid küll.

**Mida on kujutatud joonisel?**



**CMOS AND.** Selle järgi saab aru, et lõpus on inverter. CMOS on vastupidine loogika.

**Mis summaatori ülekande meetod on kõige enam levinud?**

CLA (Järjestik ja paralleelülekande kombinatsioon)

**Milline summaatori ülekande meetod on kõige aeglasem?**

Järjestikülekanne on kõige aeglasem (ripple carry)  
Paralleelülekanne (paralleel carry) oleks kõige kiirem aga see oleks ka kõige kallim  
Järjestik ja paralleelülekande kombinatsioon (CLA) oleks kõige levinum.

**Järjesta kombinatsioonskeemid nende kasutusest lähtuvalt. 1 väheoluline, 5 vägaoluline**

5 multipleksor - oluline  
4 summaator  
3 dekooder  
2 võrdleja  
1 lahutaja – ebaoluline.

**Millist meetodit kasutatakse enim riistvaras lahutamise teostamiseks**

Summa = A + B täiendkood

Realiseerime tehte 7 – 6 = 1. Lahutamine on täiendkoodi liitmine.

Leia 6 **pöördkood**: 0110 → 1001  
Täiendkood = pöördkood + 1 = 1001 + 0001 = 1010 (See on 6 pöördkood).

7 – 6 = 0111 + 1010 = x0001 (esimene 1 kaob ära, sest 4 bitti).

## L6 algusest (L4 kordamine)

**Miks on trigeritel kaks väljundit?**

Sest oma ülesehituselt on neil vaja väljundi pöördväärtust nagunii tagasiside jaoks (complimentary output on ka alati olemas).

**Kuidas nimetatakse olukorda, kui taktsignaal läheb madalast (0st) kõrgeks (1ks)?**

Tegemist on esifront (rising edge). (pinge läheb üles).  
Millise asendi all triger oma väljundit muudab. Kas esinfrondiga või tagafrondiga takteeritav triger.

**Kuidas nimetatakse 1st 0i minekut?**

Tagafront (falling edge).

**Mitu korda on signaal ühe perioodi jooksul kõrge ja madal?**

Ühe perioodi jooksul mõlemat üks kord.

**Mis on latch ja flip flop erinevus? *Hea eksami küsimus.***

Eesti keeles on nende vasteks ainult triger (mõlemal).

Erinevus on selles, et flip-flopil on taktsignaal ehk ta on takteeritav. Muudab oma olekuid siis, kui taktsignaal tuleb sisse. Kas see on taktsignaal, mõne teise trigeri väljund, takteerimine käsitsi vms. sõltub olukorrast.

|  |  |
| --- | --- |
| Latch | Flipflo |
| Latch töötab ainult binaarse sisendiga, taktsignaali (clock) ei ole. | Erinevus on selles, et flip-flopil on taktsignaal ehk ta on takteeritav |
| Asünkroonne | Sünkroonne |
| Ei saa registrina kasutada | Saab registris kasutada |
|  |  |

**Kuipalju infot salvestab 1 triger?**

1 bit.

**Millise trigeri tõeväärtustabel on kõige lühem? SR, JK, D, MS, T?**

T-triger (toggl) on kõige lihtsam.

T   
0 Q – hoiab eelmist  
1 -Qt-1 – inverteerib oma väljundi

**Mis on trigeri väljund pärast käivitust? Kuidas seda juhtida?**

Me ei tea seda. Me tahame seda vastava trigeri preseti ja cleari abil.  
Kui tuleb reset signaal, siis selle jaoks on eraldi sisend ja surume kõik alguses nulliks.

**Mis mõtet omab registril infoallika sisendi valiku võimalus?**

Sisend W määrab, kas registrisse salvestatakse info sisendi a või sisendi b pealt.  
Selle pärast, et meie sisendid tulevad erinevatest kohtadest.

**SHL, SHR**

ALU teeb nihutamine vasakule (shift left) ja nihutamine paremale (shift right).

**Milliseid funktsioone saab täita niheregistri abil?**

Nihkeregistri abil saab korrutada (vasakule) ja jagada (paremale) arvusüsteemi baasiga. Nihutab nulli sisse, kui pole ringnihe.

**Kuidas nimetatakse nihkeregistrit, mis teostab neid mõlemaid?**

Reversiivne nihkeregister

**Kas asünkroonsel loenduril on järgmise trigeri sisendiks ainult eelmise trigeri väljund või kombinatsioon kõikidest eelmistest trigeritest?**

Ainult eelmise trigeri väljund

**Kas sünkroonsel loenduril on järgmise trigeri sisendiks ainult eelmise trigeri väljund või kombinatsioon kõikidest eelmistest trigeritest?**

Kui on sünkroonne, siis kõik teised annavad ka talle signaali.

## L6 kordamine – L8 alguses

**Millised on kaks peamiselt levinud arvuti arhitektuuri? Mis neid eristab?**

Von Neumann ja Harvard. Erinevuseks see, kuidas on mälu jaotatud andmete ja käskude jaoks.

* VN puhu mälu ühildatud,
* H puhul eraldi mälud ette nähtud kummagi jaoks.

**Kirjelda lihtsalt protsessori tööpõhimõtet. Joonis.**

Joonise leiab õpikust lk 143. **Protsessor on loogikaskeem**, mis interpreteerib ja täidab masinakäske ning koosneb vähemalt juhtseadmest ja aritmeetika-loogikaseadmest (ALU). CPU sooritab tehteid andmetega, mis on saadud mälust. Juhtseade peab käsu dekodeerima, seejärel tuleb käsk täita.

**Mis tähendab lühend ISA arvuti arhitektuuri juures?**

Instruction Set Architecture. Käsustiku arhitektuur. Kuidas on käsud, mida arvuti suudab täita, üles ehitatud. Mitu bitti on mingis käsus, kui palju on aadressi jaoks.

**Millised on loengus käsitletud käsustiku käskude liigid?**

Juhtkäsud  
Täidesaatvad käsud  
…

**Peab kindlasti teadma: Kuidas nimetatakse mehhanismi, mis hoiab protsessoris käskude järjekorda „meeles“?**

Tegu on käsuloenduriga.

**Mis eesmärki täitab käsuregister?**

Sisaldab kõiki käske, mida see protsessor täita oskab.

**Nimeta kaks peamist käsu adresseerimise viisi ja kirjelda neid.**

1. Vahetu (käsukood näitab ära operandi, konstandi jaoks. operandi maht või suurus on piiratud sellega, kui suur on käsu aadress) ja
2. otsene (käsukood näitab aadressi, kus asub operand).

**Kirjelda operatsioonautomaadi tööd. Joonis.**

**Kirjelda käsu täitmise tsüklit. Joonis.**

fetch-decode-execute tsükkel. Õpikus lk 135 joonis 3.10.

Käsukoodi laadimine (fetch) 🡪 Käsuloenduri modifitseerimine PC: PC+1 🡪 Käsukoodi dekodeerimine 🡪 Käsu täitmine

## L8 kordamine (L9 alguses)

**Vahetu ja otsese adresseerimisega käskude täitmine + joonis**

Vahetu adresseerimine on eelkõige kasulik, kui kasutame konstante või proges on konstante.  
Otsene adresseerimine on see,k ui käsule järgneb operandi aadress.

**Mis on juhtautomaat? Kirjeldus.**

Juhtautomaadi ülesa

**Automaatide erijuhud, L8 slaid 20. Joonised.**

Vastavalt joonisele peaks oskama määrata, millise automaadiga on tegu ja saada aru (sisendid, väljundid, hetke olek, olekute funktsioonid, väljundfunktsioonid.

**Mealy ja Moore automaadi kirjeldus, olekute määramine.**

Kuidas olekuid määratakse. Kas on seotud tegevusega või peab olema kahe tegevuse vahel olek.

## L9 kordamine (L10 alguses)

**RISC ja CISC arhitektuurid. Erinevused, võrdlus (joonis 3.35 põhjal).**

RISC – *seda on õpikus põhjalikumalt käsitletud*

CISC -

Kärbitud käsustikuga ja täisprotsessor.

**RISC arhitektuuri projekteerimise põhimõtted.**

**Konveier. Siirdekäsud, andmete sõltuvus, mälu poole pöördumine.**

Mis ta on, milleks ta on. Käsu võtt, kodeerimine, operandide võtt, operatsioon, operatsiooni teostamine ja operandide salvestamine.

*Konveieri puhu siirdekäsud, andmete sõltuvus ja mälu poole pöördumine on eraldi välja toodud. Missuguseid probleeme nad konveierile tekitavad.* Peamiselt viitega seotud või konveieri taaskäivitamisega.

**Hargnemiste ennustamine eesmärk + 3 lähenemist**

Kui on konveieriga protsessor ja meie koodis on hargnemised, siis on mõistlik vaadata kas koodis on hargnemisi. Lähenemised: eeldame, et kunagi pole hargnemist. Kõige nutikam on see, mis suudab kahe veaga jõuda õige kohani, kas hargemine on või pole.

**Vahemälu (cache) kirjeldus, eesmärk + joonised.**

Eesmärk on, et osa mälust (kuhu pöördumine on aeglane) saaks kopeerida kiiremasse mällu (koopia põhimälu ja protsessori vahel), et vähendada küsimiste arvu. Probleemiks põhimälu ja vahemälu sünkroniseerimine.

Vahemälu SRAM. Käivitamisel on vahemälu juhuslikus olekus (andmete juures kasutuses kehtivuse bitt).

**Vahemälu organiseerimise 3 viisi, kirjeldused, joonised.**

// Plokk ja sõna. Teatud mälu tüüpide puhul ka segment.

**Otsevastavusega vahemälu** (direct mapped cache) – kõige lihtsam. Mälu on jagatud segmentideks (nt 4 plokk).

**Assotsiatiivne vahemälu** (associative mapped cache) – segmendid puuduvad, ilma järjestuseta. Kiire ja paralleelne riistvaraline võrdlemine. Võrreldes otsevastavusega on riistavaraliselt kallim (transistore rohkem). Round-Robini – järjestikkune asendamine.

Otsekirjutamine (write through) – protsessorist tulev info kirjut vahemällu ja põhimällu. Aeglane.  
Write-back (tagasikirjutamine) – vahemälust põhimällu. Vaja lisa riistvara muudatuste arvepidamiseks.

**Kogumassotsiatiivne vahemälu** (set associative mapped cache)

*Nende vahel peab suutma vahet teha ja aru saada nende tööpõhimõttest.*

## L10 kordamine (L11 alguses)

**Infokandja nõuded (vs ideaalne mälu)**

Kiirus ideaalis protsessoriga sarnane.

* Peab olema võimalik eristada millises olekus infokandja on (Info lugemine)
* Peab olema võimalik viia infokandjat ühest olekust teise (Info kirjutamine)
* Infokandjal peab olema kaks olekut, mille vahel on energiabarjäär (püsivad olekut, mille vahel ei tekiks iseeneslikku ümberlülitust)

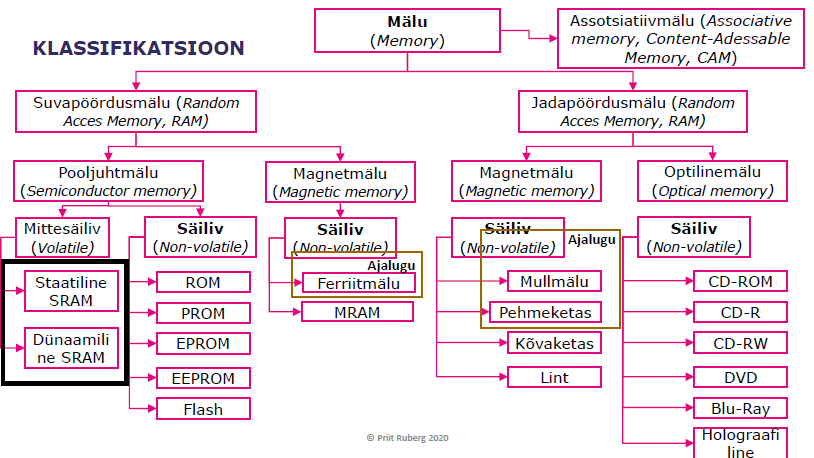
Üks infokandja ei rahulda kõiki tingimusi (suur infotihedus, suur kiirus lugemisel ja kirjutamisel, odav, säiliv (nonvolatile), lugemise ja kirjutamise võimalusega, väikese energiatarbega, suure töökindlusega, eemaldatav/mitteeemaldatav).

**Mälu hierarhia mudel, selgitus + joonis (püramiid)**

L0 (kõige kallim, väikseim mahutavus, kiireim, protsessorile lähemal (isegi protsessori sees olev registermälu)) kuni   
Ln (kõige odavam, suurem mahutavus, aeglaseim, protsessorist kaugemal (nt CD))

**Mälude klassifikatsioon**

Selline suur skeem. Suva ja jada pöördus mälud. Rohkem rääkisime mittesäilivast



**Suvapöördumälud + alamliigid**

**Staatiline SRAM - infot ei pea pidevalt uuendama (võrreldes dünaamilise RAMiga)**

* Mäluelemendid on realiseeritud trigeritena ( latch või flip flop), üks pesik 4-6 transistori.
* Piisavalt kiire, et pidada protsessoriga „sammu“
* Kasutatakse registermälu ja vahemälu realiseerimiseks

**Dünaamiline SRAM** – peab infot uuendama (nt iga sekundi tagant), kasutab kondensaatoreid ja transistore

* odavam, 1 pesik = 1 transistor
* infot peab regulaarselt uuendama (info kaob kondensaatori tühjaks laadimise tõttu)
* Tulenevalt aadressiliinide multipleksimisest aeglasem kui SRAM

**Jadapöördusmälud + alamliigid**

**Assotsiatiivmälu (tänane loeng!)**

*Associative memory, Content Adessable Memory, CAM*

**SRAM ja DRAM tööpõhimõte, iseloomustus, MOSFET**

**SRAM** – kallim, üks pesik 4-6 transistori 1 biti hoidmiseks (2 p-Mos (info sisse ja väljalaadimiseks), 4 n-MOS transistori). Staatiline infot ei pea pidevalt uuendama (võrreldes dünaamilise RAMiga). Kasutatakse registermälu ja vahemälu realiseerimiseks.

**DRAM** – **tüüpiliselt arvuti põhimälu.** odavam, transistoride mõttes (üks pesik = 1 transistor ühe biti hoidmiseks). Info salvestatakse laenguma väljatransistoris ([MOS]FET). Dünaamiline RAM kasutab kondensaatoreid ja transistore. Infot tuleb uuendada ajavahemiku tagant (~1 sek tagant, kulutab energiat). Kui infot ei uuendata, siis info hajub kondensaatori tühjaks laadimise tõttu. Tüüpiliselt arvuti põhimälu (RAM), aeglasem kui SRAM.

**Säilivad pooljuhtmälud (ROM, EPROM jne.), tööpõhimõte, FGMOS (floating gate metal oxide semiconductor)**

**MROM** - maskiga (stencil) tootja poolt programmeeritav, realiseeritakse nii riistvara kui tarkvara, kasutaja ei saa sisu muuta, pole paindlik. Suurte seeriate tootmisel, **odavaim ja töökindlaim.**

**PROM** – programmeeritav püsimälu. Kasutaja programmeeritav ühe korra, info salvestamine destruktiivne protsess. Saab programmeerida üks kord. Nt RFID tagi info sisse progemiseks.

**EPROM** – sellel omakorda erinevad klassid (EPROM, EEPROM ja Flash EPROM). Ümberkirjutatavad püsimälud. Tegu on ujuva paisuga väljatransistoridega mäludega. Põhimõte baseerub MOSFETidel.

**Magnetmälu: MRAM (Magnetoresistive random access memory (MRAM))** – pole tootma hakatud aga pmst oleks universaalne mälu. Taluvad kõrget radiatsiooni. On võimelised töötama väga kõrgel temperatuuril. On „surkimis“ kindlad. Sama kiirelt kui lugeda ja kirjutada saab. Nonvolatile, nondestructive. Tegemist on pmst DRAMi või SRAMiga, mis ei kaotaks andmeid, kui toide ära võtta.

**Magnetmälu, kõvaketas, lugemine/kirjutamine, kodeeringud**

**Magnetmälu:** Esimesene kõvaketas 1956. Säilivad, odavad, suur infotihedus. Ei sobi põhimäluks, sest reageerimisaeg suur (mikrosekundites). Baseerub magnetiliste doomenite ümbermagneetimisel. Kirjutamiseks/lugemiseks kasutatakse vastavad „pead“. Info lugemiseks erinevad kodeeringud (NRZ, RZ, NRZI jne).

**Kõvaketas.** Pöörlevad kettad. Mittemagneetiline alus (klaas, alumiinium). Kaetud magnetmaterjali kihiga. Kirjutamiseks/lugemiseks kasutatakse vastavad „pead“ kummagi ketta poole jaoks. Pead liiguvad ketta kohal (ei puuduta). Olulisemad andmed pöördumise aeg ja ülekande kiirus.

**Suurused:** 1.8, 2.5 ja 3.5 tolli.  
**Pöörlemiskiirused. RPMid** 5400, 7200 ja 10,000

**Optiline mälu, EFM kodeering**

Optilised andmekandjad CD, DVD, Blu-ray. Info salvestamiseks süvendid ja põhipind.

EFM – eight to fourteen modulation – Optilises mälus kasutatav kodeering (liinikood). Kaks üleminekut ei saa olla kõrvuti. 16384 koodi on piisav, et moodustada 256 sellist koodi kus on binaaris kahe ühe vahel vähemalt kaks nulli! Ei saa olla sellist koodi, kus on kaks 2 ühte kõrvuti või et nende vahel on ainult 1 null.

**CD läheb hukka**, sest kaitsekihi paksus põhipinna ja markeeringu vahel ei ole tihti piisav ja põhjustab peegelpinnal oksüdatsiooni. CD puhul lainepikkus 780 nm. DVD ja Blue-ray puhul väiksemad, tänu millele saab rohkem infot mahutada.

## L11 kordamine (L12 alguses) Mälu organiseerimine, mälu vaheldamine, erinevate adresseerimisviisidega mälud, Sõltumatute ketaste massiiv, Pooljuhtketas

**Mälu organiseerimine, mitu plokki, aadressi järgud**

* Andmeliinid sõna järgulisus mälus
* Aadressiliin võimalik mälu pesade arv
* Mälu koosneb mitmest plokist (mitte ühest suurest)

Näiteks: Aadressiga küsitakse kummast mälublokist missugusel konkreetsel aadressil olevat informatsiooni protsessor vajab (või tahab kirjutada sellesse mälupesasse).

**Vaheldatud ja vaheldamata mälu, mälu konveier (low order mälu puhul)**

**Memory interleaving** - Eesmärk tõsta mälu läbilaskevõimet (suurendada kiirust). Mälu jagatakse väiksemateks, eraldiseisvateks pankadeks ehk kasutatakse mitut mälumoodulit ühe suure asemel. Mälust lugemine ja kirjutamine toimub mälumoodulite vahel paralleelselt

**Vaheldamata mälu** - Järjestikuse aadressiga pesad samas mäluplokis. Viide tekib siis, kui tahame samast plokist mitmest pesast infot. Tavaline/tüüpiline mälu organiseerimine.

**Vaheldatud mälu** - Järjestikuste aadressidega sõnad eri mäluplokis. Pöördumine mitme sõna poole on võimalik. Eri mäluplokkides on sõltumatu adresseerimine ja lugemise/kirjutamise riistvara. Lisa riistvara teeb lahenduse kalliks. Võimaldab mälu poole pöörduda kordades kiiremini. **Konvereri meetodil** (3 mäluploki korral)alates 3 taktist, kui esimene sõna on loetud väljastab konveier ühe sõna takti kohta.

*Mäluaadresside järjest lugemine on kirjanduse järgi üpris tõenäoline.*

**Pinumälu ja kuhimälu, andmete hoidmine, salvestamine ja laadimine**

*Erinevad adresseerimise viisiga mälud.*

**Pinumälu (stack)** - Osa põhimälust. Saab organiseerida kas FILO või FIFO (tavaline järjekord poes). Saab võtta järjest. Stack pointer näitab ära viimase objekti hetkel pinumälus, väärtus hoitakse protsessori registris (eraldi lipp selle jaoks). Kasutatakse alamprogrammi pöördumise jaoks. Pinusse Pinusse salvestatakse parameetrid ja tagasipöördumise aadress. Kui stack saab täis, siis tekib ületäitumine ja programm jookseb kokku.

**Heap’i** (kuhi) suurus ei ole fikseeritud. Programmi töö ajal vabalt kasutatav, dünaamiliselt hallatav. Saame kasutada nii kaua kuni süsteemi mälu otsa saab. Suur ja vaba mälu ruum. Vabastatakse free käsuga või programm lõpetab.

**Kahe pordiga mälu**

Kasutus: https://www.youtube.com/watch?v= pqJz9JsVWWk

Samaaegne lugemine ja kirjutamine. On kirjutamise ja lugemise liides. Ei ole nii, et kõik käib sama kanalit kaudu.

**Assotsiatiivne mälu**

Otsimiseks kasutatakse andmeid või sisu, mitte ei otsita aadressi järgi

Otsitakse aadressi, kus sobivad andmed asuvad

Väga kiire mälu, otsing ühe taktiga (ühe taktiga saame teada, kus aadressil meid huvitav mälupesa on). Kasutatakse prioriteerenkooderit.

**Kirjutamine:** Info kirjutatakse vabale positsioonile, aadressi ei jäeta meelde.  
**Lugemine:** Otsitakse andmeid või andme osa, mis sisaldab otsitavat sõna.

Vastupidine SRAMile ja DRAMile otsingu mõttele.

**Sõltumatute ketaste massiiv (RAID, redundant array of inexpensive/independent disks)**

Nii riistavaralised kui tarkvaralised realisatsioonid. Riistvaraline RAID tavaliselt lülitatakse BIOSis sisse ja enne BIOSi aktiveerimist riistvaralise RAIDi kontrolleri informatsioon tuleb süsteemi käivitamisel ette.

Süsteemi töökindluse tagamiseks. Kiirust tõstab paralleelne pöördumine.

Standard konfiguratsioonid

Standard RAIDi ketaste konfiguratsioone on 7 (RAID 0 kuni RAID 6)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Omadus** | **Raid 0** | **1 (Mirror)** | **2** | **3** | **4** | **5** | **6 (5+1)** | **0+1** | **1+0** |
| Liiasus | **Puudub** | Olemas | Olemas | Olemas | Olemas | Olemas | Olemas | **Olemas** | **Olemas** |
| Kiirus | väga hea | sõltub |  | sõltub | Parem kui 2 ja 3 | sõltub |  | **Väga hea** | **Väga hea** |
| Veakindlus | 0 | N-1 | 1 | 1 ketas | 1 ketas | 1 ketas | 2 ketast | Üks pool | Üks pool |
| Maht | 1 | 1/N | 1-1/N\*log2(N-1) | 1-1/n, 66% saab kätte | 1-1/n | 1-1/n | 1-1/n | N/2 | N/2 |
| Min kettaid | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 |
|  | Super-arvuti | Andmebaas nt, Priit kasutab seda | pole kasutuses | Tava-kasutuses pole | pole | Kasutusel. | Kasutusel, kulukas | Väga populaarne | andmebaasid, emaili ja veebiserver |

Kui tavaliselt nõue, et samad kettad (suurus, seeria jms), siis SHR eeliseks see, et pole tähtis, mis ketas – suudab kõik ära kasutada.

Hübriid tasemed

SHR (SYNOLOGY HYBRID RAID) – paned massiivi ükskõik, mis kettaid (erinevate suurustega). Protokoll suudab kogu mahu ära kasutada.

Ebastandardsed on patenditud ja tasulised RAIDid.

**Pooljuhtketas (SSD)**

AHCI ja NVMe (Samsung).  
Põhineb Flash mälul (CMOS). Oluliselt kiirem kui HDD + muud eelised.  
Puudusteks HDD odavam ja viga hüüab tulles.

**ACHI** (kõvaketta liides arvutiga) on liides, mis oli pudelikaelaks SSD ja SATA jaoks tuli sellest liidesest. M.2 tehnoloogial ketas NVMe’ga, mis kasutab PCIe siini on kiirem. Kiirus sõltub emaplaadist ja kettast.

Kui saadetakse kettale käske, siis on järjekord…**ACHI** 1 järjekord 32 käsku. **NVME** 64k järjekorda ja igaühes 64k käsku. (4 miljardit käsku saab olla korraga ootel). **Oluline oleks, et emaplaadil oleks M.2 pesa (mitte M.2 kujuga SATA pesa nt).**

## L12 kordamine (L13 alguses) – Virtuaalmälu, adresseerimise viisid

**Virtuaalmälu kaks eesmärki (mäluruumi laiendamine, ühildamine)**

* Mälu „suurendamine“
* Mälu ühtlustamine – vürtuaalmälu aitab ühtlustada vastendamise abil.

Tulemus, et mälu on suur, kiire ja homogeenne. Virtuaalmälu ühendab põhimälu ja **osa ketasmälust** üheks aadressiruumiks. Kasutame, sest Programmid vajavad rohkem mälu kui varem ning kui põhimälu võimaldab. Sageli kasutame ka mitut programmi korraga, mis igaüks vajavad mälu.

Varasemalt ühele programmile võis anda 4GB (32bit mäluruumi puhul). Hetkel rohkem (ja rohkem ka vaja).

VM kasutamine pärsib mälupöördumise kiirust. DDR4 on 34,000 MB/s, kettaga võrreldes põhimälu 10x kiirem.  
Eeliseks on programmeerimisel väiksem keerukus ja programmeerija.

**Virtuaalmälu organiseerimine**

Virtuaalsest mäluaadressist füüsiliseks mäluaadressiks transleerimine toimub **riistvara** (MMU, memory management unit) ja op. süsteemi koostöös.

MMU vaatab, missugune programm põhimälus on väiksemas kasutuses tõstetakse ketasmällu ja pannakse viide sinna. Ja programm, mida rohkem vaja tõstetakse põhimällu.

1:1 virtuaalse mälu vastendamine füüsilisele mälule (mahuga 4GB = 2 32 bitti) tähendab, et igale virtuaalsele aadressile vastab füüsiline aadress – nii teha ei saa. **Programm nõuab 1GB mälu** -> virtuaalmälu info hoidmiseks on vaja 1GB mälu, et teha vastendamine. Kaotaksime suure osa lihtsalt selleks, et hoida virtuaalmälu tabelit põhimälus. Kasutatakse:

* Lehekülgedeks jagamine ( Pageing; Paged virtual memory
* Segmenteerimine ( Segmentation; Segmented virtual memory
* Segmenteerimine lehekülgedeks jagamisega ( Segmentation with pageing)

**Käsu adresseerimise viisid**

**Vahetu** adresseerimine ( immediate addressing) – operand (konstant) on käsuga kaasas. Käsu pikkus määrab operandi maks. väärtuse.

**Otsene** adresseerimine ( direct addressing, absolute addressing) – käsuga kaasas kaasas operandi aadress. Operandi väärtus võib muutuda aga aadress muutuda ei saa. Annab paindlikkust võrreldes vahetu adresseerimisega. Väga kiire.

Otsene register adresseerimine ( register direct addressing) – pikk aadress näitab põhimälus olevale aadressile, **lühike aadress**iga näiteb registermälus olevale aadressile. Väga kiire.

**Kaudne** adresseerimine ( indirect addressing) – aeglasem kui otsene adresseerimine, 2 mälu poole pöördumist (käsk näitab aadressi, mis sisaldab omakorda aadressi, kus on operand). Operand on mälu aadress, mis sisaldab mälu aadressi.

Kaudne register adresseerimine ( register indirect addressing) – läbi registri käimine on kiire. Operand on registermälus olev mälu aadress. 1 mälupöördus, kiire.

**Pinu** adresseerimine ( stack addressing) – oluline on stack-pointer ehk pinumälu osuti, mis hoiab väärtust selle kohta, kus on järgmine vaba mälupesa.

Autodekrementne -

autoinkrementne -

**Baseerimisega** adresseerimine ( based addressing) - …

**Indekseerimisega** adresseerimine indexed addressing -

**Baseerimisega ja indekseerimisega** adresseerimine based indexed addressing -

**Suhteline adresseerimine** ( program counter relative addressing) -

**Operandide arv käsus**

0-3 operandiga käsud on olemas.

Võib olla 2 käsku ühes mälusõnad (CISC arhitektuuris), võib olla selliseid käske, mis hõlmavad mitu mälusõna pikkust.

RISCis peaks olema ühe mälusõna pikkused, kiiresti täidetavad.

**Kolme operandiga käsud - Programmeerija vaatest ideaalne on kolme aadressiga arvuti** (sest enamasti kolm operandi (tegur, tegur, tulemus)).

**Kahe operandiga käsud** on enim kasutst leidvad. Ühe aadressi kohale salvestatakse ka tulemus.

**Ühe operandiga käsud** - Käsuga antakse ühe operandi aadress. Teine operand paikneb registermälus (tavaliselt akumulaator, AC). Tulemus salvestatakse ka registrisse. Akumulaatori „pudelikael“ – akumulaatori poole pöördumist esineb väga palju.

**Null operandiga käsud** - Käsukoodi juurde aadress ei kuulu. Operand võetakse pinumälu „pealt“. Väga haruldane kasutus.

## L13 kordamine (L14 alguses)

**Siinid üldiselt, põhjasild, lõunasild**

Siinid on trükiplaadil olevad rajad, andmeedastussüsteem, mis ühendab arvuti komponente (ja ka arvuteid). Tänapäeval pannakse neid sinna 3 tasandis.

1 rada = 1 bit, 64 raja = 64 bitti.

**Tänapäeval arvutites:**

* Andmesiin (data bus)
* Aadressisiin (address bus)
* Juhtsiin (control bus) – juhtautomaadi vahendusel mälust lugemine ja mällu kirjutamine, IO seadmetest lugemine ja kirjutamine

**Põhjasild** – CPU osa, ühendab protsessorit, mälu ja PCI E siini. Ühendatud lõunasillaga.

**Lõunasild** – Portide ja seadmete jaoks, ühendab endas perifeeriaseadmeid, liideseid (USB, PCI jms)

Süsteemisiin - https://www.youtube.com/watch?v=QzWW CBugZo

Andmesiin, aadressisiin ja juhtsiin - https://www.youtube.com/watch?v=TgYAj7mlRT8

**Siinipuhver**

Seade mis võimaldab siine kasutada rohkem kui kahe seadmel. 3 olekuga puhver, kus 0 ja 1 üks olekule on ka high-Z (ujuv olek, Elektriliselt lahtiühendatud, määramata olek). Võimaldab teha siinil kahesuunalist andmevahetust (lugemist ja kirjutamist).

https://www.youtube.com/watch?v=faAjse109Q8

**Liidese portide adresseerimine**

Mäluna adresseeritavad S/V seadmed ( memory mapped I/O) - S/V seadmed osa mälust, ühendatud mälusiinile, pole vaja eraldi käske seadmete poole pöördumiseks. Mingi osa mäluruumist on eraldatud S/V seadmetele. **Eelised** lihtne riistvara, vaja ühte siini ja üht dekoodrit, eraldi käske pole vaja, saab kasutada kõiki adresseerimise viise. **Puudused** on mälu augud (VM aitab) ja raske eristada mälu pöördumist S/V seadme operatsioonist.

Eraldi adresseeritavad pordid ( separate I/O) - S/V seadmete jaoks eraldi aadressiruum. Protsessoris on S/V seadmetelt lugemiseks/kirjutamseks eraldi käsud. Mälu pöördus ja S/V pöördus on eraldatud (eraldi aadressiruum). **Eelised** püõhimälus pole auke, Kogumäluruum on mäluseadmetele kasutatav, Lihtne eristada mälu ja S/V seadmete käske. **Puudused:** vaja lisariistvara (teist dekoodrit), keerukam käsustik. **Rohkem levinud lahendus.**

S/V seadmetena adresseeritav mälu ( I/O mapped memory) - Osa S/V seadmete aadressiruumist on kasutusel mälu adresseerimiseks. Sellise mälu pöördumiseks tuleb kasutada S/V seadme poole pöördumise käske ( In, Out). On kasutusel videomälude juures.

**Andmevahetus sisend väljundseadmetega** - Tavaliselt seadme ja liidese kontrollerid lahus. Probleemiks, kui siini ja seadme kiiruse erinevus. Abiks puhverdatud andmeahetus IO seadmega (nt printer).

* **Pollimine** ehk programe ( Polling, Programmed I/O) – protsessor kontrollib kogu aeg IO staatusregistrit. Kõige lihtsam, kõige ajakulukam, ebaefektiivne CPU kasutus.
* **Katkestustega andmevahetus** ( Interrupt driven I/O) - Süsteemi töö organiseerimiseks, katkestusi käivitatatakse programselt, erindina (exceptions) ja riistvaraliselt (IO seade genereerib katkestuse andes märku, et soovib andmevahetust). Näiteks: kasutaja nupuvajutus.
* Otsemällupöördus (Direct Memory Access) - Eraldi seade siini hõivamiseks DMA kontroller, DMA eesmärk on vähendada protsessori osakaalu andmeedastuses. IO seadmed seadmed saavad pöörduda läbi DMA kontrolleri otse mälupoole, ilma protsessori sekkumiseta DMA protsess.

**Siinide arbitreerimine**

Siinil külge on tavaliselt ühendatud mitu seadet (protsessor, DMA kontroller, mälukontroller). Kellel on õigus siinile infot saata, sellega tegeleb siini arbiiter ehk siini info vahendamise seade. Tavaliselt juhib protsessor siini protokolli täitmist.

**Staatiline arbitreerimine** vs dünaamiline arbitreerimine

**Tsentraliseeritud arbitreerimine** vs detsentraliseeritud arbitreerimine

**Mõned arbitreerimise meetodid**

*Erinevused selles, kes saab prioriteedi, töökindlus (kas mõne seadme rike mõjutab kasutust).*

* Daisy Chain (pärgühendus)
* Polling (pollimine)
* Independent Request (eraldatud liinidega ühendus)

**Paralleel ja järjestikliidesed**

*Liideles protsessori pool (täidab siini protokolli) ja IO seadme pool (järjestikune või paralleelne infovahetus).*

* **Järjestikandmeedastus** – üks andmeliin + nullnivoo. Kui info tuleb sisse, siis peab teadma, mis järjekorras bitte tuleb (kas alguses LSB (least significant bit, madalam järk) ja lõpus MSB (most sign.) või vastupidi).
  + **Asünkroonne järjestikandmeedastus** – Andmete saatja ja vastuvõtja taktsignaal võib olla erinev, Paketid võivad tulla suvalise ajavahega
  + **Sünkroonne –** USB, FireWire.Saatja ja vastuvõtja taktgeneraatorid on sünkroniseeritud,kallim ja kiirem kui asünkroonne edastus (täiendav riistvara).

**Veakindlad koodid**

Andmeedastuses tekib viga (kas juhe on halb vms).

**Vigu avastavad koodid** – andmetele lisatakse lisabitte, mis aitavad vigu avastada. Paarsuskontroll (paarsusbitt, sobib ainult ühe biti vea korral), kontrollsumma (raiskab rohkem andmemahtu võrreldes paarsuskontrolliga) ja tsükkelkoodkontroll (CRC). Vigase ploki esinemisel saadetakse andmed uuesti.

**Vigu parandavad koodid** (nt Hamming Error Correction Code, Paarsus blokk Block parity ECC).

* paarsus blokk - Andme plokile lisatakse paarsus tulp ning paarsus rida (sõna), Võimaldab tuvastada rohkem kui ühte viga maatriksit kasutades.
* Hamming – eraldi kood, kindla sammuga, leiab vea ja parandab.

## L14 kordamine (L15 alguses)

Aine on 6 EAP, seega peaks tegema iga nädal 10h tööd 😊

**Kuvarid** – neid on väga palju.

* **CRT (cathode ray tube)** - kaadri laotamine, tööpõhimõte, osad  
    
  Katood kuumutatakse, elektronide emissioon tekib. Kallutusmähised aitavad elektronide voogu juhtida. Skaneeritakse ridade kaupa üleval valakult alla paremale. Elektronide heledust saab juhtida. Fosforiga kaetud ekraan hakkab helendama, kui elektroniga pihta saab.
* **LCD (liquid crystal display, vedelkristall)** tööpõhimõte, erinevad valgustused  
    
  On olemas kristallid, mis säilitavad kristallilisi omadusi temperatuurivahemikus aga omavad ka muid omadusi. Pingega juhitakse vedelkristalli polarisatsiooni. Ekraanid moduleerivad taustavalgust (juhivad läbilaskmist) – taustast tuleb valgust ja kristall juhib seda, kas tuleb läbi või mitte. Peamised on peegeldunud (nt kalkulaator või kell) ja tagumise valgusega LCD kuvarid (monitor, telekas).
* Passiiv ja aktiivmaatriksiga LCD kuvarid, kasutatavad tehnoloogiad
* **OLED kuvarid, passiiv ja aktiivmaatriks**Kui tavaline LCD tahab taustavalgust, siis OLED emiteerib ise valgust. Orgaanilised materjalid eraldavad valgust kui neile voolu juhtida! Õhem, kiirem reaktsiooniaeg, parem musta värvi kontrastsus (sest pole kogu aeg põlevat taustavalgust), väiksem energiatarve.  
  AMOLED – aktiivmaatriksiga (kasutab TFT maatriksit)  
  PMOLED - masiivmaatriksiga
* Plasmaekraanid – kaks klaasplaati, mille vahel on gaasiga täidetud kambrid. Plaadide sees on elektroodid jne.

**Puutetundlikud ekraanid**

**ITO** – indium tin oxide, indium tina oksiid. Takistuslik materjal.

**Takistuslik** – peamine esimene, läbipaistev takistusliku materjaliga kaetud kiht. 4 juhtmega - kumbagi kihti kontrollitakse eraldi, pealmine (painduv) ITO kiht võib mõraneda. 5 juhtmega x,y juhtmed alumiselt kihilt, pealmine kiht annab pinge väärtuse. Ekraani võib vajutada millega iganes.

**Mahtuvuslik ja liigid**

**Pindmahtuvuslik** – peamine teine (70% turust). Vajab juhtivat keha. Ekraani pind on kaetud üheltpool ITOga. Ekraani pinnal on ühlane elektriväli (vahelduvvoolu). Juhtiv keha viiakse tekkinud välja. Voolu muutuse suurus sõltub mõõtepunkti kaugusest andurist.

**Projekteeritudmahtuvuslik** - ekraani pinnal on juhtivatest ribadest võrk (üksteisest isoleeritud ITO ribad). Näpu vajutusega võtab sõrm osa laengust endale. Laengu liikumise fikseerib kontroller. Kasutus: nutitelefonide ekraanid.

**Infrapunapuuteekraan** - Ekraani ees on infrapuna kiire võrgustik. Ekraan ei vaja lisakihte. Infrapuna signaali võib häirida mustus/tolm. ATM näiteks.

**Täielik sisepeegeldumine** - IR kiir levib ekraani ees oleva kaitseklaasi **sees**. Sama nagu inrapuna puuteekraan v.a. kiir levib pinna sees. Puudutus muudab kiirte liikumise nurka, mis registreeritakse anduritega, Mitu puudutust hästi ei saa korraga.

**Tagant hajutatud** - Ekraani ette tekib ühtlane infrapuna foon (võrdluseks see, kus klaasis oli kiire levik). Võimalik tuvastada mitmeid puudutusi, kui valgust jätkub.

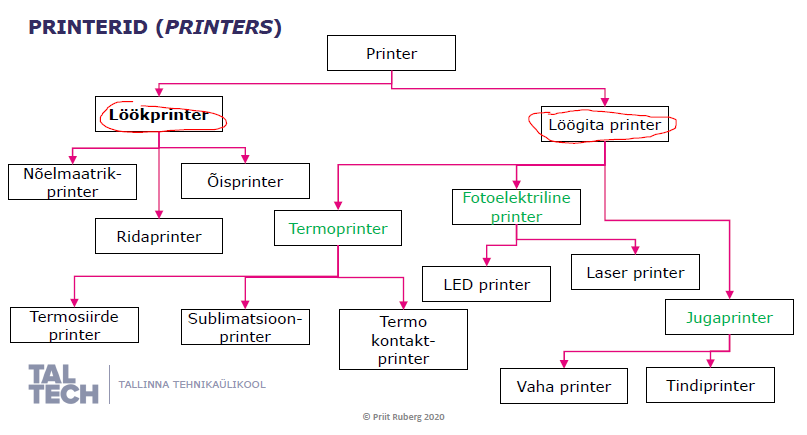
**Hajutatud pinnavalgusega** - Ekraani servades on infrapuna dioodid. Ekraani kattena kasutatakse spetsiaalset pleksiklaasi, mille osakesed jaotavad IR valguse ühtlaselt. Puutepinna peale pannakse valguse hajutaja diffuser. Kaamera registreerib puutekohas varju, mille järgi registreerib kaamera puute koordinaadid.

**Akustiline laine** - Ekraani servades on piesoandurid. Pieso andurid muudavad mehaanilise võngete energia elektrienergiaks. Võrreldakse eelnevalt salvestatud lainepildiga. Ekraan ei vaja täiendavaid kihte. Puute tajumine mehaaniline (vajuta millega tahad).

Pindakustilised lained - Lained tekitatakse piesogeneraatorite abil. Kahes nurgas generaatorid, vastas nurkades vastuvõtjad, külgedel peegeldajad. Signaali genereeritake üle pinna ja kindlaks tegemise võimalused on piiratud.

Jõutundlik - Puutepind ( panel glass ) on kinnitatud piesoanduritele (piezoelectric film). Füüsiline vajutus tekitab andurites surve ja elektrisignaali. Erinevad surved registreeritud ja annab võimaluse määrata survepunkti koordinaadi. Keskonna ja vandaalikindel, hea pildikvaliteet. Ei saa eristada mitut puudet, puudete vahel peab olema paus.

**Printerid löögita ja löögiga + kõik liigid**



**Laserprinter** – fotoelektriline löögita printer

**Tindiprinter** – löögita jugaprinter

**Teised seadmed**

Klaviatuur – olekud 1 ja 0. Mehaanilise klaviatuuri põhjal contact bouce. Kontroller kontrollib pidevalt nupuvajutusi. Nupud on ühendatud maatriksina.

**Hiir erinevad tehnoloogiad** – mehaaniline, optiline ja   
mehaanilisoptiline – kuul jätkuvalt alles aga kasutati led’e pöörlemissuuna tuvastamiseks.  
**optiline** – led või laser. Pinna peegeldusest tuvastatakse kaameraga hiire liikumine.

**Skänner** – laengusidestusseade.

**Analoogliides ADC ja DAC** – analooginfo teisendus (heli, kõne, muusika, vajutus).

**Analoog digitaalmuundur** ( analog digital converter, ADC) – enam levinud. Digitaliseerimisel saame ilusast analoogsignaali lainest sakilise (kui üksikud punktid sagedusega registreerime). Standard ADC on 10bit ka lahjematel kontrolleritel. 0-5V vahemikus 1024. Üpris täpne.

Samplimise sagedus on min lainepikkus / 2. ADC resolutsioon ADC 4 bit = 16. 0-5V vahemikus 5V/16 = 0.3125V (see on resolutsioon)?

Digitaal analoogmuundur ( digital analog converter, DAC) – vähe on kontrollereid, kuhu see on sisse ehitatud.

**Puhvertoiteallikad** (UPSid, katkematu toiteallikas) ja nende liigitamine ( Offline, Online ja Line Interactive)

**Offline** – Sisend pole lahti sidestatud arvutist/sisendist. Arvuti on liiniga kogu aeg seotud. Liinikõikumised lähevad arvutisse ka. Kui võrgu pinge kukub alla piirnormi lülitatakse toiteahel läbi aku. Odav hind. Kaitseb elektrikatkestuse korral.

**Online** – Kahekordne muundamine (vahelduvvoolust alalisvooluks ja tagasi). Mürasid jääb vähemaks. Kallis lahendus, sest vaja Kvaliteetse vahelduvpinge genereerimine.

Line Interactive - kompromiss kahe eelneva lahenduse vahel. Astmeline pinge väärtuste korrigeerimine. Võrgu häireid ei filtreerita nii hästi kui online UPSis.