

**1.Loogikaelemendid : NAND, NOR, OR, AND, XOR, jne. // NOT** - loog.eitus e. inversioon.  $y = \neg x$ .  
 $x \ y \ 0 \ 1 \ 1 \ 0$ . **AND** - loog.korrutamine e. konjunksioon.  $y = x_1 x_2$ .  $x_1 x_2 \ y \ 00 \ 01 \ 01 \ 10 \ 11 \ 1$ . **NAND** - konjunksiooni eitus e. Shefteri funkts.  $y = \neg(x_1 x_2)$ .  $x_1 x_2 \ y \ 00 \ 11 \ 10 \ 10 \ 11 \ 1 \ 0$ . **OR** - loog.liitmine e. disjunksioon.  $y = x_1 + x_2$ .  $x_1 x_2 \ y \ 00 \ 01 \ 11 \ 10 \ 11 \ 1$ . **NOR** - disjunksiooni eitus e. Pierce'i funkts.  $y = \neg(x_1 + x_2)$ .  $x_1 x_2 \ y \ 00 \ 11 \ 01 \ 01 \ 10 \ 01 \ 11 \ 0$ . **XOR** - antivalentsfunkts.  $y = \neg x_1 x_2 + x_1 \neg x_2 = x_1 (+) x_2$ .  $x_1 x_2 \ y \ 00 \ 01 \ 11 \ 10 \ 11 \ 0$ . **XNOR** - ekvivalentsfunkts.  $y = \neg x_1 \neg x_2 + x_1 x_2$ .  $x_1 x_2 \ y \ 00 \ 10 \ 10 \ 01 \ 11 \ 1$ .

**2.Mikroskeemide valmistamise tehnoloogiad : TTL, STTL, ECL, MOS, jne. // DTL** (Diod Transistor Logic) 3-osa: 1.kombinaator, mis realiseerib loogikaf-i, 2.taastaja mis taastab õiged nivood, 3.puhver väljundi hargnemisteguri tõstmiseks. 1. on diodidest, 2. ja 3. transid. **TTL** (Transistor-Transistor Logic) sama, mis DTL aga 1.osa on samuti transidest. Bipolaarne tehnoloogia. **STTL** (Schollky TTL e. Low TTL) - kasut Šoti diodi. **ECL** (Emitter Coupled Logic) - bipolaartransidel põhinev, kiiretoimeline. **MOS** (Metal Oxyde Silicon) - unipolaarne tehnoloogia. **NMOS** (n-channel MOS). **CMOS** (Complementary MNOS). Kasut. arvutiskeemides. Probleem selles, et on aeglasem kui bipolaarsed, kuid võimaldavad suurema pakkimis-tiheduse, energiatarve väiksem.

**3.Trigerid : RS, JK, MS, D, T. // Triger** - mäluga loogikaelement. T on võimeline säilitama ühe biti informatsiooni ja seda väljundist edasiseks kasut. võimalik saada niikaua, kui toide on sisse lülitatud. T jagunevad asünkroonseteks ja sünkroonseteks e. takteeritavateks. **RS-triger**. (Reset-Set, ühetaktiline). Asünkroonsel RS-trigeril on kaks sisendit S ja R ning kaks väljundit Q ja  $\neg Q$ . Sünkroonsel trigeril on lisaks C (clock) sisend. RS-trigeri võib koostada NOR- või NAND-elementidest. Keelatud kombinatsioon R=1, S=1. **JK-triger** (kahetaktiline). Sisendsignaali kombinatsioonide 0 0, 1 0 ja 0 1 korral toimib see triger analoogselt NOR-elementidest koosneva RS-trigeriga, kuid olulise iseärasusena ei ole tal keelatud kombinatsioonid. J=1, K=1 korral muudab JK-triger oma väljundoleku vastupidiseks. **MS-triger** (Master-Slave) kahetaktiline triger. Tegevus käib kas läbi Slave'i või Master'i. **D-trigeri** (D-delay, ühetaktiline) e. nihketrigeri võib saada sünkroonset RS-trigerist, kui selle ühte sisendisse anda D-signaali ja teise sisendisse sama signaali NAND-lüli abil inverteeritult. D-triger viivitab tema sisendisse saabuvat infot ühe sünkroniseerimistakti võrra. **T-triger** (kahetaktiline loendustriger). T-trigeri võib saada JK-trigerist, kui signaali anda C-sisendisse ning kokkuühendatud J- ja K-sisendile rakendada 1-signaali vastav alalispinge. Niisugune triger lülitub ümber iga kord, kui sisendisse saabub järjekordne impulss.

**4.Loendurid : kahend, kümnend, suvalise mooduliga, sünkroonne, asünkroonne, jne. //**

**Kahendloendur**. Kui ühendada mitu T-trigerit järjestikku, kujuneb impulsisagedus lülituse väljundist seda väiksemaks, mida suurem on trigerite arv. Sedasama T-trigerite jadühendust saab kasutada sisendisse saabuvate impulsside loendamiseks kahendsüsteemis. Suurim loendavate impulsside arv on ühe võrra väiksem loendustegurist  $K=2^n$ . St. et neljast trigerist koosneva lülitusega saab loendada 15 impulssi. Kahendloenduritest sagedamini kasut. **kümnendloendureid**. Need moodustatakse dekaadidest. Dekaad on trigerite rühm, mille loendustegur on 10. Dekaadloenduris on kümnendaru iga numbrikoha jaoks üks dekaad. **Sünkroonne** - ümberlülitumine toimub samaaegselt v. paralleelselt.

**5.Registrid : nihkega ja ilma, reverssiivsed, jne. // Register** on trigeritel põhinev lülitus kahendarvude registreerimiseks. Info võib säilida registris kui tahes kaua tingimusel, et toide on sees. Vajadusel saab registris sisalduvat infot kustutada. Iga biti salvestamiseks on vajalik omaette triger. Bitte on registrit moodustavatesse trigeritesse võimalik sisestada ja neist väljastada rööbiti (korruga) või järjestikku (üksteise järel). Rööbitise sisendi ja väljundiga registreid nim. ka **mäluregistreiks**. Registreid, millesse info sisestamine ja väljastamine toimub järjestikku, nim. **nihkeregistriks**. See register nihutab iga taktimpulsi toimel sisestatava kahendarvu kohti ühe võrra paremale või vasakule. **Reverssiivne register** - kahesuunaline nihe.

**6.Summaatorid:pool- ja täissummaatorid, paraleel- ja järjestikülekanega, kiire ülekanne, lahutajad. // A+B=S. Täissummaator** - võimaldab liita arvestades ka ülekanneid. **Järjestikülekanne summaator** - aeglane. **Paralleelülekanega summaator** - kasvab järguliselt j ka riistvara kulu. **Kiire ülekanne** - järjestik-paralleelne meetod. Iga järgu saamiseks arvutatakse eraldi Generation ja Propagation. **Lahutajad**  $A-B=V$ . 1. otseteel (kõigi variantide analüüs), 2. Matemaatiliselt.

**7.Dekooder** // on lülitus, mis tunneb ära registri või loenduri väljundist dekodeeritud sisendisse saabuva kahendarvu ja annab signaali sellele vastavasse väljundisse. Dekoodril on nii mitu sisendit, kui mitu kohta n on kahendarvudel. Väljundite arv võib ulatuda võimalike koodikombinatsioonide arvuni  $2^n$ . Dekodeeritud põhiülesanne on muundada kahendkoodis arv kümnendsüsteemi arvuk.

**8.Multipleksori** // ülesanne on vastavalt juhtkoodile ühendada üks mitmest sisendist ainsa väljundiga.

**9.Koodimuundur**. // (Code converter) kombinatsiooniskeem, mis muudab ühe koodi teiseks.

Indikaatorite numbrid koostatakse 7 segmentidest. Seega ei piisa kümnendnumbrit nähtavaks tegemiseks enam ühest signaalist, vaid mitmesugustes kombinatsioonides tuleb sisse lülitada 2...7 segmenti. Seega on vaja kahendkood muundada teistsuguseks koodiks. Koodimuundurid põhinevad

samuti loogikamentidel (NAND, AND) nagu dekodeerid, kuid on neist palju keerukamad.

Koodimuundureid vadeldakse sageli ka kui dekodeerite ühte alaliiki.

**10. Komparaator (võrdlusskeem).** /// näitab operandide suuruse suhet. Sisendiks: A - a1a0, B - b1b0.

Väljundid: \*L=1, kui A<B => G=0, \*G=1, kui A>B => L=0, \*Kui L=0 ja G=0 => A=B.

**11. Mälud : muutmälud, püsिमälud, staatilised, dünaamilised, suva- ja jadapöördusega,**

**pinumälu, jne.** /// **Muutmälu** (RAM - Random Access Memory) suvapöördusega mälu st. võib pöörduda ükskõik millise aadressi osa poole sama ajaga. Saab kirjutada, lugeda, kustutada. **Püsिमälu** (ROM - Read Only Memory) - ainult lugemiseks. Ühekordselt program. püsिमälu PROM. Ümber progetavad püsिमälud EPROM (kustutatakse ultravioletkiirgusega), EAPROM (kustutus toimub koos uue infi salvestamisega), EEPROM (kustutus toim elektriliselt). **Staatiline mälu** (SRAM) - trigerite peal, ümberkirjutust pole vaja. **Dünaamiline mälu** (DRAM) - vajalik tsüklik info uuendamine, info kandjaks kondensaator. **Jadapöördusmälu** - eri mälulementide poole pöördumine nõuab eri aja.

**Suvapöördumine** - pöördumise aeg sama. **Pinumälu** (LIFO - Last In, First Out).

**12. Käsu täitmine protsessoris (von Neumani tsükkel).** /// 1. käskukoodi lugemine käsuloenduri (PC) järgi (FETCH), 2. käsuloenduri modifitseerimine PC+1, 3. käskukoodi desifreerimine, 4. kasutäitmine 5. resultaati. **Käskukood** - a) milline käsk kuulub täitmisele, b) kus asuvad operandid, c) kuhu salvestada resultaati.

**13. Ühe, kahe, kolme ja 1,5 aadressiga arvutid.** ///

**1 aadressiga arvuti:** 1. käskukood, 2. operandi/resultaadi pikk aadress

**1,5 aadressiga arvuti:** 1. käskukood; 2. oper/result pikk aadress, 3. oper/result pikk aadress.

**2 aadressiga arvuti:** 1. käskukood; 2. esim oper pikk aadress, 3. teise oper pikk aadr/result pikk aadress.

**3 aadressiga arvuti:** 1. käskukood; 2. esim operandi pikk aadress; 3. teise oper pikk aadress; 4. resultaadi pikk aadress.

**14. Adresseerimise viisid : otsene, vahetu, jne.** /// 1) vahetu adr. - operand kohe käsus, 2) otsene adr. - käsu juurde kuulub aadress, 3) kaudne adr. - aadressi aadress, 4) autoinkrementne adr. - loetakse operand välja ja aadress säilitatakse modifitseeritult. Pinumälu, pinuosuti. 5) autodekrementne - lahutamise (sama, mis eelmisel), 6) segmenteerimine - vanem osa aadr.-st hoitakse lehekülje nr. alles ja modifitseeritakse, 7) adr. indekseerimisega - käsu juurde kuulub pikk baasaadress ja liidetakse juurde nihe, mis annab uue aadressi, 8) Adr. baseerimisega - käskukoodi juurde kuulub nihe, mis võib olla lühem kui pikk aadress, 9) Adr. baseerimise ja indekseerimisega - liidetakse kõik (baas kui ka indeks) kokku. 10) suhteline adr. - PC-käsuloendur + nihe ja saadakse uus aadress

**15. Juht- ja operatsioonautomaadi osa käsu täitmisel.** /// JA sisaldab mikroprogrammi ja väljastab signaale. OA sisaldab ALU-d ja registrit ning on vahetu mikrooperatsioonide teostaja.

**16. Protsessori struktuur : käsuloendur, käsuregister, käsudekooder, jne.** /// Protsessoris toimub andmete teisendamine. **Käsuloendur** näitab järgmise täidetava käsu aadressi. **Käsuregistris** hoitakse käske. **Käsudekooder e. desifraator** teeb kindlaks käsu. Juhtautomaat e. mikroprogramm. Mingi väljund, mis ütleb, mis käsuga on tegu. **ALU** - aritmeetika-loogika seade - so. kombinatsiooniskeem, kus on mingisugune hulk operatsioone. Registre blokk. Akumulaator - spetsiaalne register, kus salvestatakse tulemus.

**17. Juhtautomaat: abstraktne, Mealy, Moore.** /// **Abstraktne automaat** on põhimingimust kast millel on {Z}-sisendit {A}-olekut ja {W}-väljundit,  $\lambda$  - väljunditef.,  $\delta$ -üleminekuf. Kui on määratud {A,Z,W, $\lambda$ , $\delta$ } siis öeldakse, et abstr. automaat on määratud. Kui on määratud ka algolek  $a_1$  siis tegem initsiaalse automaadiga. **Mealy mudel:**  $AxZ \rightarrow W$ ,  $W(t) = \lambda(A(t), Z(t))$ . **Moore mudel:**  $W(t) = \lambda(A(t))$  (sisend tähtsust ei oma)

**18. Operatsioonautomaat : registermälu, ALU.** /// **Operatsioonautomaat(OA)** koosneb mingist registermälest ja aritm.-loog. Seadmest (ALU). OA määrangud: 1) sisend-sõnad, 2) väljundisõnad, 3) sisesõnad (nii sisend kui väljundisõnad jm), 4) mikro-operatsioonid  $\{\phi_m\}$ , 5) loogika tingimused  $\{X_i\}$ . OA-d iseloomustab: tootlikkus (ühe takti jooksul teost mikro oper. arv) sõltub palju algoritmist; kiirus; hind; universaalsus (võimalus täita erinevaid mikroprogesid); regulaarsus (mida regul seda universaalsem)

**19. Programmeeritavad loogikamaatriksid : PAL, PLA.** /// **PAL** (Program Array Logic) - koosneb AND- ja OR- tüüpi maatriksist. Kitsendus: OR-maatriks on ära fikseeritud. **PLA** - koosneb AND- ja OR-tüüpi maatriksist. Mõlemad on programmeeritavad.

**20. Püsिमäluga juhtautomaat : vertikaalne, horisontaalne ja vertikaal-horisontaalne mikroprogramm (MP), mikrokäskude adresseerimine.** /// **Vert. MP** - on kodeeritud MP. **Horisont. MP** - juhtsignaal on kodeerimata. **Vert.-horisont. MP** - osaliselt kodeeritud. **Mikrokäskude adresseerimine:** \* sundadresseerimine, \* loomulik adresseerimine (järgmine juhtautomaat määratakse tavalise loenduri abil (parall. Loendamine)

**21. Silpprotsessor : operatsioonautomaat ja käsujärjesti (sekventser).** /// (**BS - Bit-Sliced**) Sõna lius ja käsusüsteem vabalt valitav. Bipolaarne tehnoloogia. Kiire TTL. Erineb monoliitprotsessorist,

kus: \*sõna laius ja käsusüsteem fikseeritud, \* unipolaarne tehnoloogia, \* MOS. **Op.automaat** - koosneb registermälust ja ALU-st. **Käsujärjesti** arvutab järgmise mikrokäsu aadressi.

**22.Siinid : siinidraiverid, andmesiin, aadressisiin ja juhtsiin.** /// **Siin** - üks komplekt juhtmeid.

**Andmesiinid** - toimub andmevahetus mõlemas suunas. **Aadressiin** - määrab ära maksimaalne adresseeritav mälu. **Juhtsiin** - komplekt juhtsignaale mõlemas suunas. **Siinidraiver** - element, mis eraldab siinist mingi seadme.

**23.Mikroprotsessori üldstruktuur (monoliitprots.): akumulaator, registermälu, ALU, siinipuhvrid, pinumälu osuti (SP),jne.** ///lisaks käsudekooder (INC/DEC), üldregistrite blokk

**24.Andmevahetus mikroprotsessorsüsteemis (mikroarvutis) : siini tsükkel, olekusõna, siinikontroller, pöördumine mälu ja välisseadmete poole, andmevahetus katkestustega ja ilma, prioriteetid, otsepöördusrežiim (DMA).** /// **Siinitsükkel** v. masinatsükkel - iga siini poole pöördumine. I/O READ - lugemine, I/O WRITE - kirjutamine. **Olekusõna** - kood, mis siini kaudu väljastatakse. **Siinikontroller** - register, kus säilitatakse infot siinitsükli kohta. **Pöördumine mälu ja välisseadmete poole** - siinitsükli kaudu. **Andmevahetus katkestusega** - antakse aktiivsus I/O-seadmetele. **Ilma katkestuseta andmevahetus** - kõik väljundseadmed on passiivsed ja protsessor määrab ära pöördumise ja lahendab prioriteetiprobleemi. **Otsepöördusrežiim e. DMA** - korraldab ise andmevahetuse. Haarab juhtsiinid enda alla. Andmevahetus läbi DMA kontrolleri.

**25.Programm - käsk - mikroprogramm - mikrokäsk.** /// **Programm** - jada käske. **Käsk** - ühele käsule vastab mikroprogramm. **Mikroprogramm** - käsukood määrab mikroprogrammi. **Mikrokäsk** - elementaartegevus, mis täidetakse operatsioonautomaadis.

**26.Mikroprotsessorsüsteemi (mikroarvuti) komponendid : sisend-väljund kontroller (parallel ja järjestik), katkestuste kontroller, DMA kontroller, taimer.** /// **Sisend-väljund kontroller:**

\*paralleelne - sobitab kokku siiniprotokollid ja sisend-väljund seadme. \*järjestik - info edastatakse järjestikkoodis. **Katkestuste kontroller** - annab katkestuste nõudmise protsessorile ja lahendab prioriteetide probleemi. **DMA kontroller** - andmevahetus ilma protsessori osavõtuta. **Taimer** - võimalik genereerida erinevaid impulssjadasid.