

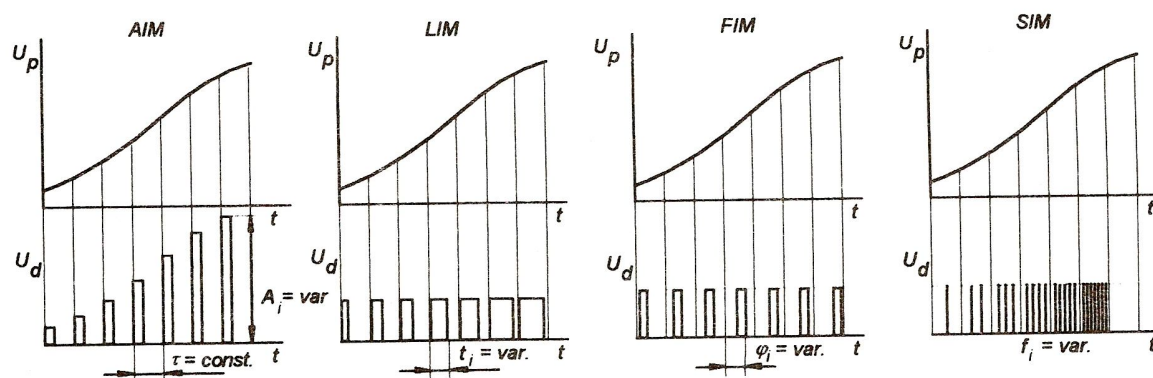
1. DIGITAALLELEKTROONIKA ALUSED

1.1. Diskreetsed ja arvsignaalid

1.1.1. Kvantimine

Kvantimine tähendab klassikaliselt füüsikateoorialt kvantteooriale siirdumise menetlust. Informaatikas on kvantimine signaalitöötlemise operatsioon, millega pidevale signaalile omistatakse kindlaks ajavahemikuks diskreetne väärtus. Kvantimine toimub nii signaali nivoo järgi kui ka ajas. Lisagem, et **signaal** on sõnumi (informatsiooni) füüsikaline kandja. Sõltuvalt füüsilisest olemusest liigitatakse signaale pneumo-, hüdro-, elektri-, valgus- jms signaalideks. Mikroprotsessortechnikas käsitletakse peamiselt elektrisignaale, kuid erijuhtudel ka optilisi ehk valgussignaale.

Suur osa looduslikest ja tehisprotsessidest on pidevatoimelised, s. t neid iseloomustavad pidevad olekusignaalid, mida saab mõõta või hinnata suvalisel ajahetkel. Pidevatoimelisi signaale nimetatakse neid töötlevate (analoog)seadmete järgi **analoogsignaalideks**. Mikroprotsessortechnika põhineb **diskreet- ehk katkelistel signaalidel**, millele omistatakse väärtus ainult kindlail ajahetkeil. Diskreetsignaalid jagunevad impulss- ja arvsignaalideks. **Impulss-signaalides** kodeeritakse informatsiooni impulsi parameetritega. Impulsi olulisemad parameetrid on amplituud (A_i) ehk kõrgus, kestus (t_i) ehk laius, sagedus (f_i) või periood (τ_i) ja faasinurk (φ_i) ehk nihe taktiimpulsi suhtes. Nende nelja parameetri alusel tuntakse signaalide nelja **impulssmodulatsiooni** liiki: 1) amplituud-impulssmodulatsiooni (**AIM**), 2) laius-impulssmodulatsiooni (**LIM**), 3) sagedus-impulssmodulatsiooni (**SIM**) ja 4) faasi-impulssmodulatsioon (**FIM**), mille olemusest annab ülevaate joonis 1. Märkigem, et nende terminite asemel võib kasutada ka pulsiamplituudi-, pulsilaiuse-, pulsisageduse- ja pulsifaasimodulatsiooni mõisteid.



Joonis 1.1. Impulssmodulatsiooni liigid

Kvantimisperiood ehk diskreetimisintervall τ valitakse sõltuvalt kvantitava signaali sageduslikest omadustest (spektrist) nii, et kvantimisega ei läheks kaduma signaaliga edastatav info. Selleks peab kvantimisperiood τ olema vähemalt kaks korda lühem kui signaali spektri suurima sagedusega harmoonilise komponendi periood

$$\tau = \frac{1}{2f_{\max}} \quad \text{või} \quad f_i = \frac{1}{\tau_i} = 2f_{\max}, \quad (1.1)$$

kus $f_{\max}$ on signaali spektri harmooniliste komponentide suurim sagedus.

Juhul kui impulsi parameetrid ei ole määratletud, sisaldab üks impulss ühe biti informatsiooni, s. t impulsi olemasolu võib lugeda signaaliks 1 ning selle puudumise signaaliks 0. Kahe impulsi opereerides saab edastada $2^2 = 4$ bitti infot jne. Mitmebitiseid impulssignaale saab kodeerida kahendarvude koodiga ning neid nimetatakse **arvsignaalideks**. Digitaaltehnikas kasutatakse kõige enam 8-, 10-, 12- või 16-bitiseid arvsignaale, mille infosisaldus on $2^8 = 256$, $2^{10} = 1024$, $2^{12} = 4096$ ja $2^{16} = 65536$ bitti.

1.1.2. Kodeerimine, dekodeerimine ja koodide liigid

Kodeerimine on informatsiooni esitusvormi muutmine sellekohase reeglistiku alusel. Numbritest koostatud kood nimetatakse **arvkoodideks**. Arvsignaale moodustatakse kodeerimisega. Eri arvusüsteemidele vastavad erinevad koodid. Arvusüsteemidest tuntakse kõige enam kümnendsüsteemi. Vähem on kasutusel nn rooma numbrite süsteem. Arvutustehnikas rakendatakse peamiselt kahendsüsteemi, kuid ka kaheksand- ja kuueteistkümnendsüsteemi. Kõiki arvusüsteeme võib jaotada positsioonilisteks süsteemideks ning mittepositsioonilisteks süsteemideks. Viimaste hulka kuulub näiteks rooma numbrite süsteem. **Positsiooniliseks süsteemiks** nimetatakse arvusüsteemi, kus ühel ja samal arvul on erinev väärtus sõltuvalt asukohast arvujadas. Neid süsteeme iseloomustab arvude esitamise selgus ning aritmeetiliste operatsioonide lihtsus. Positsiooniliste süsteemide hulka kuuluvad nii kümnend-, kahend-, kaheksand- kui ka kuueteistkümnendsüsteem (tabel 1). Positsioonilist süsteemi kirjeldatakse üldjuhul valemiga

$$X = a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_1 \cdot s + a_0 \cdot s^0 + a_{-1} \cdot s^{-1} + a_{-2} \cdot s^{-2} + \dots, \quad (1.2)$$

kus teguriteks a_i võivad olla suvalised süsteemis kasutatavad arvud.

Ka positsioonilisi süsteeme on erinevaid. Üldjuhul võib arvu igale numbrile anda suvalise kaalu, mida nimetatakse arvu **kohakaaluks**. Arvu mõistmiseks tuleb ette anda **koodi võti**, mis näitab arvu kõigi kohtade kaalu. Kümnendkoodi korral on kohakaaludeks arvud 10^n , kus n on koha järjenumber. 10 on koodi **põhiarv**, kuna selle järgi moodustatakse kõik kohakaalud. Kohti loendatakse alates paremalt, mille kohakaal kümnendkoodis on 10^0 . Kahendkoodi põhivarv on 2, kohakaaludeks aga arvud 2^n . Koodi võtmeks on harilikult 8421, s. o. 2^3 , 2^2 , 2^1 , 2^0 .

Kahendarvudel on järgmised omadused:

- kasutatakse kahte sümbolit 0 ja 1;
- põhiarvuks on 2;
- kohakaaludeks on arvud 2^n (1, 2, 4, 8, 16, 32, 64 jne), kus n on arvu kohanumber.

Kahendarvu väärtuse leidmiseks tuleb selle kohti tähistavad arvud korrutada kohakaaludega ning seejärel liita. Nii leitakse, et kahendarvule 11001101 vastab kümnendsüsteemis väärtus 205 (joonis 1.2). Nagu näha, on kahendarvul palju enam kohti kui vastaval kümnendarvul. Suurimale kaheksakohalisele kahendarvule vastab kolmekohaline kümnendarv 255 ning suurimale kuueteistkümnendkohalisele kahendarvule viiekohaline arv 65535. Sellest tingituna on inimesel kahendarvudega opereerida tülikas.

Sümbolid: 0, 1
Põhiarv: 2
Kohakaalud: 2^n , kus n on kohanumber

Näide:

1 1 0 0 1 1 0 1

1	x	1	=	1
0	x	2	=	0
1	x	4	=	4
1	x	8	=	8
0	x	16	=	0
0	x	32	=	0
1	x	64	=	64
1	x	128	=	128
				205

Joonis 1.2. Kahendarvu väärtuse leidmine

Sümbolid: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
Põhiarv: 16
Kohakaalud: 16^n , kus n on kohanumber

Näide:

C D

13	x	1	=	13
12	x	16	=	192
				205

Joonis 1.3. Kuueteistkümnendarvu väärtuse leidmine

Kuueteistkümnendarvu esitamiseks kasutatakse kümnendarvu sümboleid 0...9 ning ladina kirja suurtähti $A = 10$, $B = 11$, $C = 12$, $D = 13$, $E = 14$, $F = 15$, mida antud juhul tõlgendatakse kui numbreid (joonis 1.3). Kui võrrelda kahendarvu 1100 1101 vastava kuueteistkümnendarvuga CD , siis on näha, et üks kuueteistkümnendarvu koht vastab neljakohalisele kahendarvule. See tähendab, et neljakohalise kahendarvu saab esitada vaid ühe kuueteistkümnendarvu sümboliga 0...F ehk 0...15. Kuueteistkümnendarvude väärtus leitakse samuti kui kümnend- ja kahendarvude puhul kohaväärtuste liitmisega.

Levinumatest arvkodeidest ja arvusteemidest annab ülevaate tabel 1.1. Tehnikas kasutatakse laialt **kahend-kümnendkoodi**, mille mõistmiseks on vaja teada, et arvu kodeerimisel on kasutatud korraga kahend- ja kümnendkoodi positsioonilisi süsteeme, s. t mitmekohaline arv on kodeeritud kümnendkoodis, kuid iga selle number esitatakse kahendkoodis. Kahend-kümnendkoodide korral rakendatakse mitmesuguse võtmega koodi, lisaks tavalisele 8421 koodile näiteks ka 4221 või 2421 võtmega koodi.

Kahend-kümnendkoodiga 8421+3 saab lihtsustada kahend-kümnendarvude aritmeetika-tehteid. Koodi võti 8421+3 tähendab, et arvujada iga kümnendnumbri asemel kodeeritakse kahendkoodis 3 võrra suurem kümnendarv, näiteks arvu 5 asemel arv 8 või arvu 8 asemel 11. Saadakse neljakohalise kahendkoodi ülekanne järgmisele kohale vastavalt kümnendarvule (kümnendarvu ülekanne tekib arvu 9 järel, neljakohalise kahendarvu ülekanne aga $1111_2 = 15_{10}$ järel). Tsüklilistest ehk peegeldunud koodidest on levinud nn **Gray kood**, mida rakendatakse positsioonjuhtimisega ajamite asendiandurites. Koodi eeliseks võrreldes teiste kahendkoodidega on see, et anduri lähimatele naaberasenditele (naaberarvudele) vastavad koodid erinevad teineteisest minimaalselt, s. t ainult ühe järgu võrra. Seega muutub kood anduri modulatsiooniketta liikumisel sujuvalt ning asendisignaali töötlevate loogikaskeemide lülituste arv on minimaalne, mis suurendab seadmete töökindlust. **Ühikkoodi** kasutakse juhul, kui mingit füüsilist suurust iseloomustab impulsside arv, näiteks kui asendit määratakse impulssanduri impulsside loendamiseega.

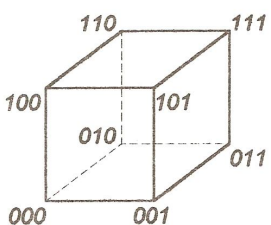
Tähtsad on **vigu avastavad** ning **vigu korrigeerivad koodid**, mida nimetatakse **veatõrjekoodideks**. Seoses informatsioonitehnika arenguga on oodata nende koodide üha laiemat rakendust. Veatõrjekoodide abil on võimalik vähendada seadmete keerukusest ning keskkonnahäiretest põhjustatud vigu ning järelikult suurendada seadmete töökindlust. Vigu avastavate ja korrigeerivate koodide puhul on lubatud ehk õigete koodide arv märksa väiksem võimalike koodide arvust. Seega on osa koodi keelatud ehk vigased. Lubatud ja keelatud koodide suhet iseloomustab koodi samm d . Tavalise kahendsüsteemi korral $d=1$, s. t kõik koodid on lubatud. Kui $d=2$, siis on lubatud ja keelatud koodide arv võrdne ning niisugune kood võimaldab avastada ühe vea. Kui avastatakse keelatud kood, teatatakse sellest juhtseadme või süsteemi operaatorile, kes teeb vastavad järeldused. Koodi sammu $d=3$ korral saab avastada korraga kahte viga. Kui eeldada, et korraga esineb vaid üks viga, siis saab niisuguse koodiga ka vigu korrigeerida, s. t keelatud koodid jagunevad omakorda rühmadesse, mille koodi korrigeeritakse kas üheks või teiseks lubatud koodiks. Koodi sammu suurenemisel avarduvad ka vigade avastamise ning korrigeerimise võimalused. Kui koodis on informatsiooni rohkem kui tema eristamiseks minimaalselt vaja, on tegu **liaskoodiga**. Seega sobivad liaskoodid veatõrjekoodideks. Tuntumad veatõrjekoodid on **Hammingi** ja **Reedi-Mulleri** koodid.

Tabel 1.1

Arvkoodid ja arvusüsteemid

Nr	Koodi nimi	Sümbolid a_i	Koodi valem või näide
1	Kahendkood	0, 1	$\dots a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$
2	Kaheksandkood	0, 1, 2, 3, 4, 5, 6, 7	$\dots a_3 \cdot 8^3 + a_2 \cdot 8^2 + a_1 \cdot 8^1 + a_0 \cdot 8^0$
3	Kümnendkood	0, 1, 2, 3, ..., 9	$\dots a_3 \cdot 10^3 + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0$
4	Kuueteistkümnendkood	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$\dots a_3 \cdot 16^3 + a_2 \cdot 16^2 + a_1 \cdot 16^1 + a_0 \cdot 16^0$

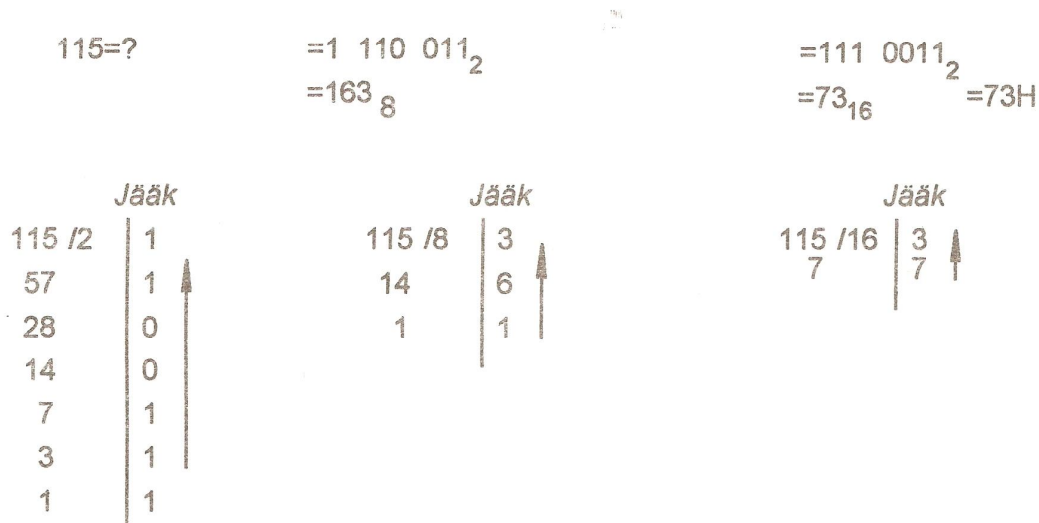
Tabeli 1.1 järg

5	Kahend-kümnendkood 8421	0, 1	Näide: 852 = 1000 0101 0010
6	Kahend-kümnendkood 4221	0, 1	Näide: 852 = 1110 0111 0010
7	Kahend-kümnendkood 4421	0, 1	Näide: 852 = 1100 0101 0010
8	Kahend-kümnendkood 2421	0, 1	Näide: 852 = 1110 0101 0010
9	Kahend-kümnendkood 8421+3	0, 1	Näide: 852 = 1011 1000 0101
10	Tsükliline ehk peegeldunud kümnendkood	0, 1, 2, 3, ..., 9	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, / 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, / 20, 21, 22, 23, ...
11	Tsükliline ehk peegeldunud kahendkood; Gray kood	0, 1	0 - 0000 1 - 0001 2 - 0011 3 - 0010 4 - 0110 5 - 0111 6 - 0101 7 - 0100 8 - 1100 9 - 1101 10 - 1111 11 - 1110 12 - 1010 13 - 1011 14 - 1001 15 - 1000 16 - 0000
12	Ühikkood	1	1, 11, 111, 1111, 11111
13	Vigu avastav kood $d = 2$	Lubatud 000 011 101 110 Keelatud 001 010 100 111	
14	Vigu avastav ja korrigeeriv kood $d = 3$	Lubatud 000 111 Keelatud 001 010 100 110 011 101	Koodid 001, 010 ja 100 parandatakse koodiks 000 Koodid 110, 011 ja 101 parandatakse koodiks 111

1.1.3. Kümnnendarvude teisendamine kahend-, kaheksand- ja kuueteistkümnnendarvudeks

Kümnnendarvu teisendamiseks kahendarvuks kasutatakse joonisel 1.4 näidatud skeemi. Arv 115 jagatakse tulbas arvuga 2 ning eraldatakse jagamise jääk 1. Jagamise tulemus 57 kirjutatakse esialgse arvu alla. Seejärel korratakse kirjeldatud tegevust seni, kuni jagamise tulemuseks saadakse arv 1. Jagamise jääkidest moodustub eraldi tulp, mis sisaldab arve 1 ja 0. Lugeses selles tulbas olevaid sümboleid alt üles leitakse lähtearvule 115 vastav kahendarv 1110011.

Nagu joonisel näidatud, saab sama toiminguga leida ka kaheksand- või kuueteistkümnnendarve.



Joonis 1.4. Kümnnendarvude teisendamine kahend-, kaheksand- ja kuueteistkümnnendarvudeks

Märksa lihtsam on kümnnendarve teisendada kaheksand- ja kuueteistkümnnendarvudeks kahendkoodi abil. Selleks tuleb kümnnendarv teisendada esmalt kahendarvuks ning jaotada selle kohad triaadideks või tetraadideks, olenevalt sellest kas soovitakse leida kaheksand- või kuueteistkümnnendkoodi. Edasi kodeeritakse kahendarvu iga triaadi eraldi vastava kaheksandarvu või tetraadi kuueteistkümnnendarvu sümboliga.

Arvude teisendamist ühest süsteemist teise näitab tabel 1.2, kuhu on koondatud mõned kümnnend-, kahend-, kuueteistkümnnend- ja kahend-kümnnendkoodis ehk BCD-koodis arvud.

Tabel 1.2

Arvude teisendamine

Kümnendarv	Kahendarv	Kuueteist-kümnendarv	BCD-kood
0	0	0	0000
1	01	1	0001
2	10	2	0010
3	11	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
8	1000	8	1000
9	1001	9	1001
10	1010	A	0001 0000
11	1011	B	0001 0001
12	1100	C	0001 0010
13	1101	D	0001 0011
14	1110	E	0001 0100
15	1111	F	0001 0101
16	1 0000	10	0001 0110
17	1 0001	11	0001 0111
18	1 0010	12	0001 1000
19	1 0011	13	0001 1001
20	1 0100	14	0010 0000
126	111 1110	7E	0001 0010 0110
127	111 1111	7F	0001 0010 0111
128	1000 0000	80	0001 0010 1000
510	1 1111 1110	1FE	0101 0001 0000
511	1 1111 1111	1FF	0101 0001 0001
512	10 0000 0000	200	0101 0001 0010

1.1.4. Informatsiooni hulk ja signaali viga

Arvsignaalis sisalduva informatsiooni hulga ja signaali vea vahel on olemas kindel sõltuvus. Arvsignaali bittide arv n ehk kahendarvu järkude arv määrab signaali diskreetsusastmete (diskreetide) arvu $N = 2^n$. Signaali kodeerimisveaks loetakse maksimaalselt ühe diskreeti väärtus. Seega on 10-bitise kahendsignaali viga $1/1024 \approx 0,1 \%$.

1.2. Loogikafunktsioonid ja loogikalülitused ning nende esitusviisid

1.2.1. Loogikatehted

Loogikalülituste projekteerimine, talitus ja selle analüüs põhineb loogikaalgebral (Boole'i algebra). Muutujatel saab siin olla ainult kaks väärtust 0 - väär ja 1 - tõene. Seepärast nimetatakse seda loogikat ka **binaarloomikaks**. Loogilisi muutujaid tähistatakse ladina tähestiku tähtedega.

Sõltumatuid muutujaid (sisendeid) nimetatakse argumentideks, neist sõltuvaid muutujaid aga funktsioonideks. Loogikafunktsiooni kõik argumendid on loogilised muutujad, millel on kaks väärtust 0 ja 1. Kõiki loogikafunktsioone väljendavad kolm põhitehet: loogiline korrutamine, loogiline liitmine ja loogiline eitus.

Loogiline korrutamine (NING). NING-funktsioon on võrdne ühega ainult juhul, kui kõik argumendid on võrdsed ühega. Tehte tähistamiseks kasutatakse nii harilikku korrutusmärke ($\cdot$) kui ka loogilise korrutamise eritähist - katust ($\wedge$). Loogilist korrutamist nimetatakse ka konjunktsiooniks.

Loogiline liitmine (VÕI). VÕI-funktsioon on üks siis, kui kas või üks argumentidest võrdub ühega. VÕI-tehte tähistamiseks kasutatakse kas pluss (+) märki või loogilise liitmise eritähist - V tähe kujulist märki ($\vee$). Loogilist liitmist nimetatakse ka disjunktsiooniks.

Loogiline eitus (EI). EI-funktsioonil on argumendi vastandväärtus. Kui argument on 1, siis funktsioon võrdub 0 ning vastupidi. EI-tehet tähistatakse kriipsuga sümboli peal, näiteks argumendi x eitus on $\bar{x}$. Loogilist eitust nimetatakse ka inversiooniks.

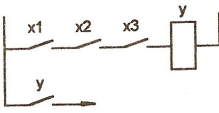
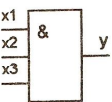
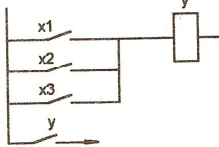
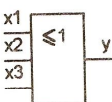
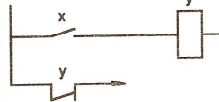
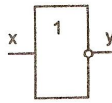
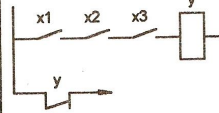
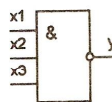
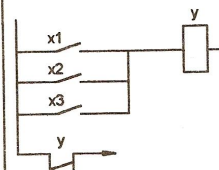
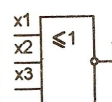
Loetletud kolm loogikatehet moodustavad loogiliselt täieliku süsteemi, mida rakendades saab realiseerida mis tahes loogikafunktsiooni. Kõiki kolme loogika põhifunktsiooni on loogikaalgebra reeglite alusel võimalik realiseerida ainult üht tüüpi loogikaelementide kas NING-EI või VÕI-EI abil. Järelikult võib NING-EI- ja VÕI-EI-elemente ning tehteid nendega nimetada universaalseteks loogikaelementideks ja -teheteks.

Lisaks põhifunktsioonidele leiavad kasutamist mitmed loogika tüüpfunktsioonid, nagu alternatiiv, ekvivalentsus, implikatsioon jt. Niisuguste funktsioonide ja elementide olemasolu lihtsustab loogikalülituste sünteesi. Loogikafunktsioonidest ja loogika tüüpelementidest annab ülevaate tabel 1.3.

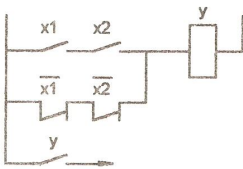
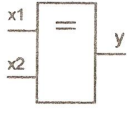
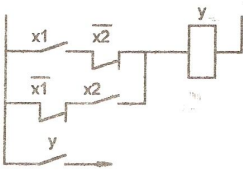
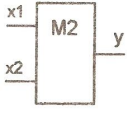
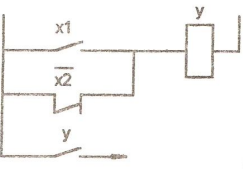
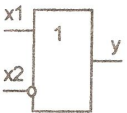
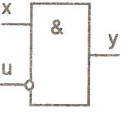
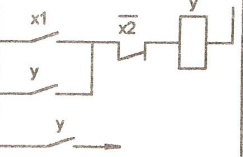
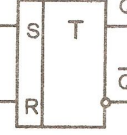
Keerukamaid loogikalülitusi, mis koosnevad paljudest loogikaelementidest ning on ette nähtud kindlate funktsioonide täitmiseks, nimetatakse funktsionaalseteks loogikalülitusteks, mille hulka võib lugeda ka tabelis 1.3 toodud mäluolelemendi.

Tabel 1.3

Loogikafunktsioonid ja -elemendid

Nr	Loogika-funktsiooni nimetus	Loogika-funktsiooni selgitus	Loogikafunktsiooni kontaktaseskeem	Loogikafunktsiooni matemaatiline esitus	Loogika-elementi tähis
1	2	3	4	5	6
1.	NING	Lüli väljundis on signaal 1, kui signaal 1 on tema 1. ning 2. ning 3. ning jne sisendis		$y = x1 \cdot x2 \cdot x3$ $y = x1 \wedge x2 \wedge x3$	
2.	VÕI	Lüli väljundis on signaal 1, kui signaal 1 on tema 1. või 2. või 3. või jne sisendis		$y = x1 + x2 + x3$ $y = x1 \vee x2 \vee x3$	
3.	EI	Lüli väljundis on signaal 1, kui tema sisendis on signaal 0 ja vastupidi		$y = \bar{x}$	
4.	NING-EI	Väljundis on signaal 0, kui kõigis sisendites on signaal 1		$\bar{y} = x1 \cdot x2 \cdot x3$ $\bar{y} = x1 \wedge x2 \wedge x3$ $y = \overline{x1 \cdot x2 \cdot x3}$	
5.	VÕI-EI	Väljundis on signaal 0, kui kas või ühes sisendis on signaal 1		$\bar{y} = x1 + x2 + x3$ $\bar{y} = x1 \vee x2 \vee x3$ $y = \overline{x1 + x2 + x3}$	

Tabeli 1.3 järg

1	2	3	4	5	6
6.	Ekvi- valent- sus	Väljundis on signaal 1 ainult siis, kui sisenditel on ühesugused väärtused		$y = x1 \cdot x2 + \overline{x1} \cdot \overline{x2}$	
7.	Välistav VÕI	Väljundis on signaal 1 ainult siis, kui sisendite olek on erinev		$y = \overline{x1} \cdot x2 + x1 \cdot \overline{x2}$ $y = x1 \oplus x2$	
8.	Impli- katsioon	Väljundis on signaal 0 ainult siis, kui $x2 = 1$ ja $x1 = 0$. Loogiline järelendus		$y = x1 + \overline{x2}$	
9.	Keeld	Väljund võrdub sisendiga x, kui signaal u on 0. Sinaali $u = 1$ korral on väl- jundis signaal 0 sõltumata sisendist x		$y = x \cdot \overline{u}$	
10.	Mälu	Sisendiga x1 lülitatakse väljund olekusse 1 ning see olek säilib, kuni seda ei muudeta sisendiga x2		$y = (x1 + y') \cdot x2$	

1.2.2. Loogikaseadused

Loogikaseadusteks nimetatakse tavaliselt binaarloogika algebra ehk Boole'i algebra seadusi. Algebraks nimetatakse üldjuhul elementide hulka, millega tehakse tehteid, kusjuures nende tehete aluseks on kindlad reeglid ehk aksioomid. Aksioomid määravad ära algebra põhitehete omadused ja seosed. Kuna nüüdismatemaatikas on palju algebra liike (universaalalgebra, hulgaalgebra, loogikaalgebra), siis kehtivad neis ka erinevad tehted ja aksioomid. Boole'i algebra elementideks on binaarloogika signaalid (argumendid) väärtustega 0 ja 1. Nende signaalidega saab sooritada kõiki loogikatehteid ning moodustada suvalisi loogikafunktsioone. Loogikatehete kohta kehtivad järgmised **binaarloogika aksioomid**:

1. Argumentide järjekorda võib tehtes muuta

$$a + b = b + a. \quad (1.3)$$

2. Sulgusid võib avada ehk funktsiooni võib teisendada loogiliste osakorrutiste summaks

$$a(b + c) = ab + ac. \quad (1.4)$$

3. Funktsiooni võib teisendada loogiliste osasummade korrutiseks

$$a + bc = (a + b)(a + c). \quad (1.5)$$

4. Argumendi ja tema eituse loogiline korrutis võrdub nulliga ega muuda loogilise summa väärtust

$$a + a \cdot \bar{a} = a. \quad (1.6)$$

5. Suvalise argumendi ja tema eituse loogiline summa võrdub alati ühega

$$a + \bar{a} = b + \bar{b} = 1. \quad (1.7)$$

6. Suvalise argumendi ja tema eituse loogiline korrutis võrdub alati nulliga

$$a \cdot \bar{a} = b \cdot \bar{b} = 0. \quad (1.8)$$

Loogikatehete ja aksioomide põhjal leitakse kahendarvude kohta kehtivad järgmised reeglid:

$$\begin{aligned} \bar{0} &= 1; \quad \bar{1} = 0; \\ 0 \cdot 0 &= 0; \quad 0 \cdot 1 = 0; \quad 1 \cdot 0 = 0; \quad 1 \cdot 1 = 1; \\ 0 + 0 &= 0; \quad 0 + 1 = 1; \quad 1 + 0 = 1; \quad 1 + 1 = 1. \end{aligned} \quad (1.9)$$