

Tõnu Lehtla, Argo Rosin

***LOOGIKA JA
PROGRAMMEERIMINE***

Tallinn
2001

This study material has been compiled in the framework and by financial support of the Leonardo da Vinci pilot project "International Curricula of Mechatronics and Training Materials for Initial Vocational Training", EE/99/1/87301/PI.1.1.A./FPI

Sisukord

I OSA DIGITAALTEHNIKA JA LOOGIKA ALUSED

	Lk	
1	Diskreetsed ja arvsignaalid	6
1.1	Kvantimine	6
1.2	Kodeerimine, dekodeerimine ja koodide liigid	8
1.3	Kümnendarvude teisendamine	13
1.4	Informatsiooni hulk ja signaali viga	14
2	Loogikafunktsioonid ja loogikalülitused	15
2.1	Loogikatehted	15
2.2	Loogikalülitused	15
2.3	Loogikaseadused	18
2.4	Loogikalülituste süntees ja minimeerimine	22
3	Kombinatsioonloogikalülitused	25
3.1	Kommutaatorid	25
3.2	Summaatorid	27
3.3	Koodrid ja dekodeerid	30
3.4	Aritmeetika-loogikaplokk	32
4.	Järjendloogikalülitused	34
4.1	Mittestabiilsed ja monostabiilsed loogikalülitused.	34
4.2	Trigerid	35
4.3	Registrid	39
4.4	Loendurid	41
5	Homogeensed struktuurid	44
5.1	Loogikamaatriksid	44
5.2	Mälud	47
6.	Diskreetsed automaadid	51
6.1	Diskreetsete automaatide olemus	51
6.2	Algoritmi riistvaraline realiseerimine	54
6.3	Algoritmi tarkvaraline realiseerimine	62
6.4	Riist- ja tarkvara hierarhia	64

II OSA LOOGIKAKONTROLLERID JA NENDE PROGRAMMEERIMISE ALUSED

	Lk	
7	Programmeeritavate kontrolleriite riistvara	70
7.1.	Kontrollerite ehitus	70
7.2.	Tööpõhimõte	72
8	Programmeerimise alused	75
8.1.	Kasutajaprogrammide struktuur	75
8.2.	Käsurea struktuur	77
8.3.	Andmetüübid ja adresseerimisviisid	79
9	Programmeeritavate kontrolleriite tarkvara ja käsustik	83
9.1.	Programmeerimiskeeled	84
9.2.	Kompaktkontrollerite tarkvara	85
9.3.	Moodulkontrollerite tarkvara	92
9.4.	Keele Step-7 käsustik	94
10	Protsessijuhtimisprogrammi koostamine	109
10.1.	Programmeerimiseks ettevalmistamine	109
10.2.	Programmeerimine	111
10.3.	Programmi testimine ja protsessi diagnostika	112
11.	Juhtimisprogrammide näiteid	114
11.1.	Ventilatsiooni signalisatsioon	114
11.2.	Tehase värava juhtimine	119
11.3.	Hädaseiskamis- ja peakäivituslüüti	123
11.4.	Valgusfooride juhtimine	125
11.5.	Vaheladu	127
11.6.	Sulatusahju ukse juhtimine	129
11.7.	Torupainutusseadme juhtimine	131

1. OSA

DIGITAALTEHNIKA JA LOOGIKA ALUSED

Digitaaltehnika on tihedalt seotud **matemaatilise loogikaga**. Nüüdisarvutite imetlusväärset võimalused põhinevad asjaolul, et meid ümbritseva maailma ka kõige keerukamad nähtused ja protsessid on taandatavad lihtsate loogikatehete korduvale sooritamisele. Näiteks saab arvutusmatemaatika meetodite abil astendamise-, juurimis- ja trigonomeetriafunktsioone arvutada nelja aritmeetikatehete (liitmise, lahutamise, korrutamise ja jagamise) korduva rakendamisega. Kahendarvude ja binaarsignaaside puhul on need neli tehet aga asendatavad vaid ühe korduva tehete, s. o. liitmis- ehk summeerimistehete ja sooritatavad vastava loogikaseadme - summaatoriga. Summaatori talitus, nagu ka kõigi muude digitaalseadmete talitus, põhineb omakorda vaid kolmel elementaarloogika tehete: **loogilistel NING, VÕI ja EI tehete!** Sellest, kuidas see kõik võimalik on, annab ülevaate käesoleva kursuse 1. osa "Digitaaltehnik

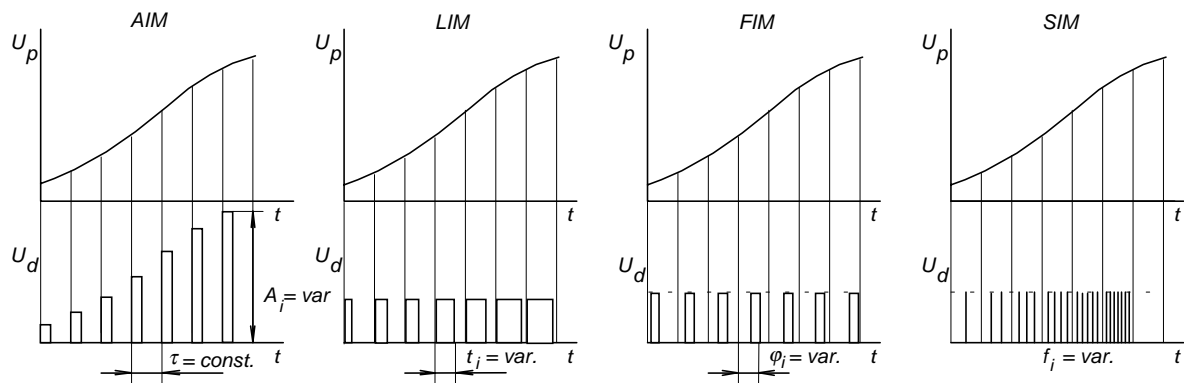
1. Diskreetsed ja arvsignaalid

1.1. Kvantimine

Kvantimine tähendab klassikaliselt füüsikateoorialt kvantteooriale siirdumise menetlust. Informaatikas on **kvantimine** signaalitöötlemise operatsioon, millega pidevsignaale omistatakse kindlaks ajavahemikuks diskreetne väärtus. Seetõttu nimetatakse antud protseduuri ka **diskreetimiseks** (*ingl. sample, sampling*). Diskreetimine toimub nii signaali nivoo järgi kui ka ajas. Lisagem, et **signaal** on sõnumi (informatsiooni) füüsikaline kandja. Sõltuvalt füüsikalisest olemusest liigitatakse signaale pneumo-, hüdro-, elektri-, valgus- jms signaalideks. Mikroprotsessortehnikas käsitletakse peamiselt elektrisignaale, kuid erijuhtudel ka optilisi- ehk valgussignaale.

Suur osa looduslikest ja tehispetsessidest on pidevatoimelised, s. t neid iseloomustavad pidevad olekusignaalid, mida saab mõõta või hinnata suvalisel ajahetkel ja millel on lõpmatu arv olekuid. Pidevatoimelisi signaale nimetatakse neid töötlevate (analoog)seadmete järgi **analoogsignaalideks** (*analogue signal*). Mikroprotsessortehnik

põhineb **diskreet- ehk katkelistel signaalidel** (*discrete signal*), millele omistatakse väärtus ainult kindlail ajahetkeil. Diskreetsignaalidel on lõplik arv olekuid. Need jagunevad omakorda impulss- ja arvsignaalideks. **Impulss-signaalides** (*pulse signal*) 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 taktimpulsi suhtes. Nende nelja parameetri alusel tuntakse signaalide nelja **impulssmodulatsiooni** (*pulse modulation*) liiki: 1) impulsi **amplituudmodulatsiooni** (*pulse amplitude modulation, PAM*), 2) impulsi **laiusmodulatsiooni** (*pulse width modulation, PWM*), 3) impulsi **sagedusmodulatsiooni** (*pulse frequency modulation, PFM*) ja 4) impulsi **faasimodulatsiooni** (*pulse phase modulation, PPM*), 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

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

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

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

Mitme impulsi edastatavaid impulssignaale saab kahedkoodiga kodeerida. Vastavat toiminut nimetatakse impulsside **koodmodulatsiooniks** (*pulse code modulation, PCM*). Kui kodeerimiseks kasutatakse arvkoodi, nt. kahendkoodi, nimetatakse vastavat signaali **arv- ehk digitaalsignaali** (*digital signal, numerical signal*), kahendkoodi puhul ka **binaarsignaali** (*binary signal*). Digitaalsignaali edastamise ja muundamisega tegelevat tehnikaharu nimetatakse **digitaaltehnikaks**, vastavat elektroonikaharu aga **digitaalelektronikaks**. Digitaaltehnikas kasutatakse kõige enam 8-, 10-, 12- või 16-kohalisi arvsignaale.

Kahe impulsi opereerides saab edastada $2^2 = 4$ erinevat signaalikombinatsiooni, kolme impulsi $2^3 = 8$ signaalikombinatsiooni jne. 8-, 10-, 12- või 16-kohaliste arvsignaali kombinatsioonide arv on vastavalt $2^8 = 256$, $2^{10} = 1024$, $2^{12} = 4096$ ja $2^{16} = 65536$.

1.2. Kodeerimine, dekodeerimine ja koodide liigid

Bitt (*bit*) on väikseim informatsiooni ühik, mis tähendab valiku "üks kahest" tulemust, nt. mündi viskamisel kulli või kirja. Kahendarvude puhul tähistab bitt kahendarvu kohta, millel on kaks võimalikku olekut - 0 ja 1.

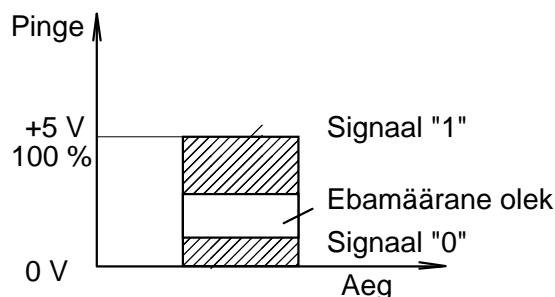
Mitmekohalise kahendarvu võib ühendada **kahendsilbiks** või **-sõnaks**.

Silpi, mis moodustub kaheksakohalisest kahendarvust, nimetatakse **baidiks** (*byte*). Sel juhul võib öelda, et silbi pikkuseks 8 bitti ehk üks bait. **Sõna** (saks. *wort*, ingl. *word*) pikkus võib olla üks või mitu (2, 3, 4, 6, 8) baiti. Üheбайдise infosõnaga juhtseadmes edastatakse korraga nt. 8 sisendsignaali või 8 väljundsignaali. Programmeeritav kontrolleri töötleb enamuses 8 bitist moodustatud silpe ehk baite. Kahendsilbi igas järgus võivad olla arvud 0 ja 1.

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.

- **Signaal 0** tähistab olekut, kus pinget pole (s. o. pinge on null, 0 V).
- **Signaal 1** tähistab olekut, kus pinge on olemas (nt. 5 V, 24 V või 220 V)

Reaalsete impulsside parameetrite väärtused erinevad nende ideaalväärtustest. Impulssi arväärtus loetakse võrdseks 1-ga kui tema pinge on suurem teatud läviväärtusest ning võrdseks 0-ga kui tema pinge on väiksem vastavast läviväärtusest (vt. joonis).



Joonis 1.2. Signaali nivood

Kodeerimine on informatsiooni esitusvormi muutmine sellekohase reeglistiku alusel. Numbritest koostatud koodi nimetatakse **arvkoodideks** (*digital code*). Arvsignaale moodustatakse kodeerimisega. Eri arvusteemidele vastavad erinevad koodid. Arvusteemidest tuntakse kõige enam kümnendsüsteemi ja sellele vastavat kümnendkoodi (*decimal code*). Vähem on kasutusel nn rooma numbrite süsteem. Arvutustehnikas rakendatakse peamiselt kahendsüsteemi (*binary code*), kuid ka kaheksand- (*octal number system*) ja kuueteistkümnendsüsteemi (*hexadecimal system*). Kõiki arvusteeme võib jaotada mittepositsioonilisteks ning positsioonilisteks süsteemideks. **Mittepositsiooniliste** arvusteemide hulka kuulub näiteks rooma numbrite süsteem.

Positsiooniliseks süsteemiks (*positional system*) nimetatakse arvusüsteemi, kus ühele ja samale märgile omistatakse erinev väärtus sõltuvalt tema asukohast märgijadas. 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 (märgid).

Ka positsioonilisi süsteeme on erinevaid. Üldjuhul võib arvu igale (numbri)märgile 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 põhiarv on 2, kohakaaludeks aga arvud 2^n . Koodi võtmeks on harilikult 8421, s. o $2^3, 2^2, 2^1, 2^0$.

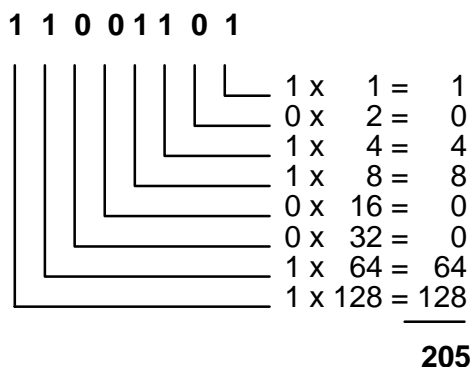
Kahendarvudel on järgmised omadused:

- 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

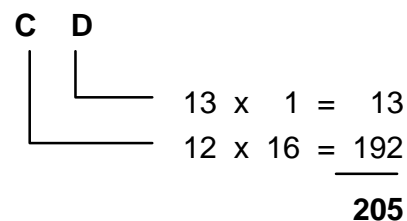
Näide:



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:



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 arvusüsteemidest annab ülevaate tabel 1.1. Tehnikas kasutatakse laialt **kahend-kümnendkoodi** (BCD , *binary coded decimal*), 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 koode, lisaks tavalisele 8421 koodile näiteks ka 4221 või 2421 võtmega koode.

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 loendamisega.

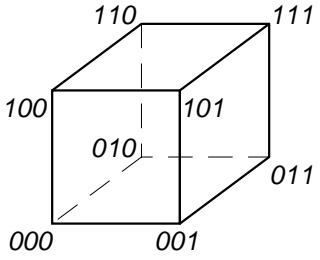
Tähtsad on **vigu avastavad** ning **vigu korrigeerivad koodid**, mida nimetatakse **veatõrjekoodideks**. Seoses informatsioonitehnika arenguga on oodata nende koodide üha laiemat rakendust. Veatõrjekoodide abil on võimalik vähendada seadmete keerukusest ning keskkonnanähäiretest põhjustatud vigu ning järelikult suurendada seadmete töökindlust. Vigu avastavate ja korrigeerivate koodide puhul on lubatud ehk õigete koodide arv märksa väiksem võimalike koodide arvust. Seega on osa koode 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 koode 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	Kuuteistkümnendkood	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$\dots a_3 \cdot 16^3 + a_2 \cdot 16^2 + a_1 \cdot 16^1 + a_0 \cdot 16^0$
5	Kahendkümnendkood 8421	0, 1	Näide: 852 = 1000 0101 0010
6	Kahendkümnendkood 4221	0, 1	Näide: 852 = 1110 0111 0010
7	Kahendkümnendkood 4421	0, 1	Näide: 852 = 1100 0101 0010
8	Kahendkümnendkood 2421	0, 1	Näide: 852 = 1110 0101 0010
9	Kahendkümnendkood 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 6 - 0101 11 - 1110 1 - 0001 7 - 0100 12 - 1010 2 - 0011 8 - 1100 13 - 1011 3 - 0010 9 - 1101 14 - 1001 4 - 0110 10 - 1111 15 - 1000 5 - 0111 16 - 0000
12	Ühikkood	1	1, 11, 111, 1111, 11111

Tabeli 1.1 järg

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

1.3. Kümnenarvude teisendamine kahend-, kaheksand- ja kuuteistkümnenarvudeks

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

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

$$\begin{aligned}
 115 &=? & =1\ 110\ 011_2 & & =111\ 0011_2 & & =73_{16} & & =73H \\
 & & =163_8 & & & & & &
 \end{aligned}$$

	Jääk		Jääk		Jääk
115 / 2	1		115 / 8	3	
57	1	↑	14	6	↑
28	0		1	1	
14	0				
7	1				
3	1				
1	1				

Joonis 1.4. Kümnenarvude teisendamine kahend-, kaheksand- ja kuuteistkümnenarvudeks

Reaalarvude teisendamisel tuleb arvestada, et kahendarvu kohakaalud pärast koma on vastavalt

1/2	1/4	1/8	1/16	1/32	1/64	jne.
ehk						
0,5	0,25	0,125	0,0625	0,03125	0,015625	jne.

Märksa lihtsam on kümnenarve teisendada kaheksand- ja kuuteistkümnenarvudeks kahendkoodi abil. Selleks tuleb kümnenarv teisendada esmalt kahendarvuks ning jaotada selle kohad triaadideks või tetraadideks, olenevalt sellest kas soovitakse leida kaheksand- või kuuteistkümnenarkoodi. Edasi kodeeritakse kahendarvu iga triaadi eraldi vastava kaheksandarvu või tetraadi kuuteistkümnenarvu sümboliga.

Arvude teisendamist ühest süsteemist teise näitab tabel 1.2, kuhu on koondatud mõned kümnen-, kahend-, kuuteistkümnen- ja kahend-kümnenarkoodis ehk *BCD*-koodis arvud.

Tabel 1.2

Arvude teisendamine

Kümnendar v	Kahendarv	Kaheksandar v	Kuueteist- kümnendarv	BCD-kood
0	0	0	0	0000
1	01	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0100
15	1111	17	F	0001 0101
16	1 0000	20	10	0001 0110
17	1 0001	21	11	0001 0111
18	1 0010	22	12	0001 1000
19	1 0011	23	13	0001 1001
20	1 0100	24	14	0010 0000
126	111 1110	176	7E	0001 0010 0110
127	111 1111	177	7F	0001 0010 0111
128	1000 0000	200	80	0001 0010 1000
510	1 1111 1110	776	1FE	0101 0001 0000
511	1 1111 1111	777	1FF	0101 0001 0001
512	10 0000 0000	1000	200	0101 0001 0010

1.4. Informatsiooni hulk ja signaali viga

Informatsiooni hulka mõõdetakse bittides (*bit*) ja baitides (*byte*). Bitt vastab kahendarvu kohale. Kuna 1 bait = 8 bitti, siis 8 kohaline kahendarv edastab 1 baidi informatsiooni. Arvsignaalis sisalduva informatsiooni hulga ja signaali vea vahel on olemas kindel sõltuvus. Arvsignaali bittide arv n ehk kahendarvu kohtade arv määrab signaali diskreetsusastmete (diskreetide) arvu $N = 2^n$. Signaali kodeerimisveaks loetakse maksimaalselt ühe diskreeti väärtust. Seega on 10-bitise kahendsignaali viga $1/2^{10} = 1/1024 \cong 0,1 \%$.