3. Kombinatsioonloogikalülitused

3.1. Kommutaatorid

Kommutaatorid jagunevad multipleksoriteks ja demultipleksoriteks. Nende tööpõhimõtte paremaks mõistmiseks on joonisel 3.1 näidatud kommutaatorite kontaktaseskeemid. **Multipleksoril** on mitu sisendit ja üks väljund. Sisendid jagunevad infosisenditeks ja juhtsisenditeks, kusjuures infosisendite arv määrab ära juhtsisendite arvu ning vastupidi. Vastavalt juhtsignaalile kommuteeritakse multipleksori väljundisse signaal ühest infosisendist. Kommuteeritavate infosisendite arv võrdub 2^n, kus n on juhtsisendite arv. Järelikult saab kahe juhtsisendiga ehk kahebitise koodiga kommuteerida 4 sisendit, kolme juhtsisendiga 8 sisendit jne.

Nelja ja kaheksa sisendiga multipleksori tööd kirjeldavad loogikavõrrandid:

$$Y_{1-4} = x_0\bar{u}_1\bar{u}_0 + x_1\bar{u}_1 u_0 + x_2 u_1\bar{u}_0 + x_3 u_1 u_0, \tag{3.1}$$

$$\begin{aligned} Y_{1-8} = {} & x_0\bar{u}_2\bar{u}_1\bar{u}_0 + x_1\bar{u}_2\bar{u}_1 u_0 + x_2\bar{u}_2 u_1\bar{u}_0 + x_3\bar{u}_2 u_1 u_0 + \cdots \\ & \cdots + x_4 u_2\bar{u}_1\bar{u}_0 + x_5 u_2\bar{u}_1 u_0 + x_6 u_2 u_1\bar{u}_0 + x_7 u_2 u_1 u_0. \end{aligned} \tag{3.2}$$

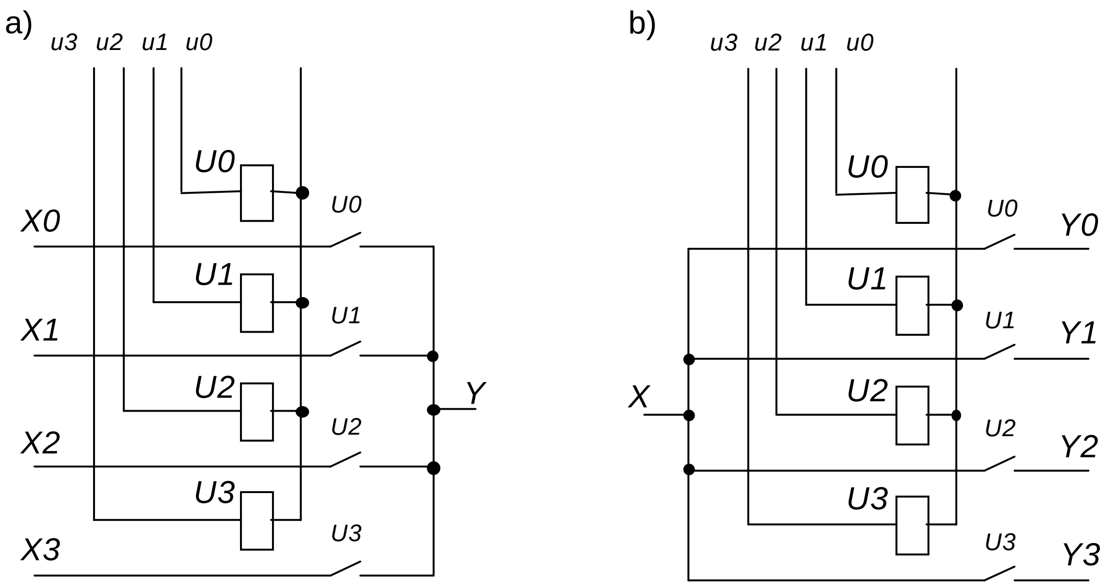
Demultipleksoril on üks infosisend ja mitu väljundit. Juhtsisendite arv sõltub väljundite arvust ja vastupidi. Vastavalt juhtsignaalile kommuteeritakse infosisendi signaal ühte väljundisse. Väljundite arv on 2^n, kus n on juhtsisendite arv. Nelja väljundiga demultipleksori tööd kirjeldavad järmised loogikavõrrandid:

$$Y_0 = x\bar{u}_1\bar{u}_0, \quad Y_1 = x\bar{u}_1 u_0, \quad Y_2 = x u_1\bar{u}_0, \quad Y_3 = x u_1 u_0. \tag{3.3}$$

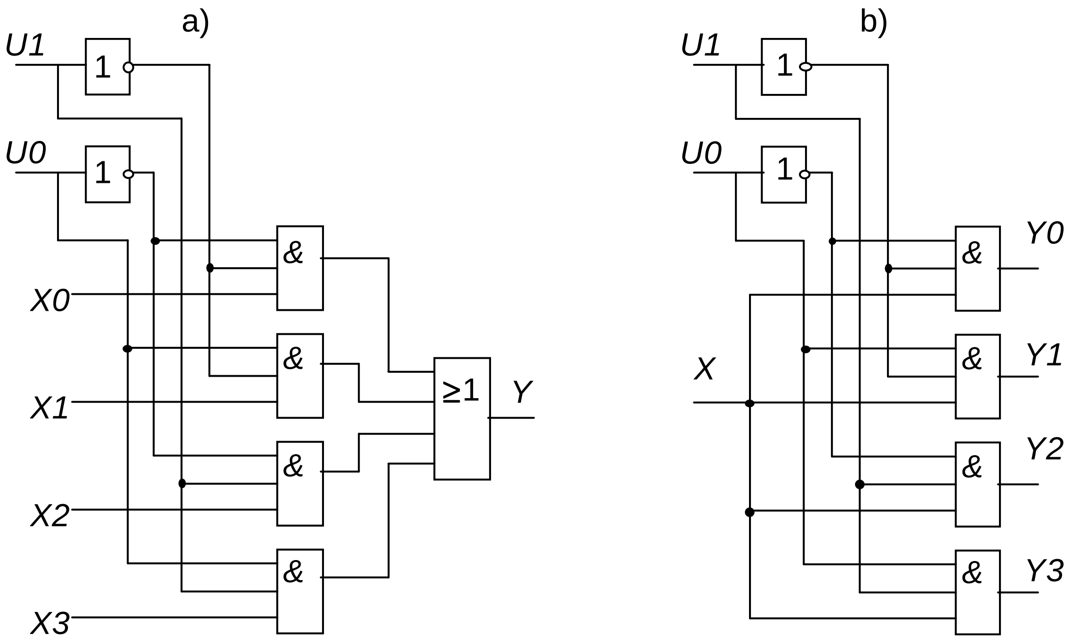
Kaheksa väljundiga demultipleksori tööd kirjeldavad võrrandid:

$$\begin{aligned} Y_0 &= x\bar{u}_2\bar{u}_1\bar{u}_0, & Y_4 &= x u_2\bar{u}_1\bar{u}_0, \\ Y_1 &= x\bar{u}_2\bar{u}_1 u_0, & Y_5 &= x u_2\bar{u}_1 u_0, \\ Y_2 &= x\bar{u}_2 u_1\bar{u}_0, & Y_6 &= x u_2 u_1\bar{u}_0, \\ Y_3 &= x\bar{u}_2 u_1 u_0, & Y_7 &= x u_2 u_1 u_0. \end{aligned} \tag{3.4}$$

Kommutaatorite loogikaskeemid on joonisel 3.1. Enam kasutatakse multipleksorit ja vähem demultipleksorit. Multipleksori väljundfunktsioon (võrrandid 3.1 ja 3.2) on esitatud loogikafunktsiooni täielikul disjunktiivsel normaalkujul. Järelikult saab piisava arvu sisenditega multipleksori abil realiseerida suvalisi loogikafunktsioone. Kolme muutuja loogikafunktsiooni realiseerimiseks tuleb kasutada multipleksorit $K_{1\text{-}4}$, nelja muutuja funktsiooni jaoks $K_{1\text{-}8}$ ja viie muutuja funktsiooni korral $K_{1\text{-}16}$.



Joonis 3.1. Kommutaatorite kontaktskeemid: a) multipleksor, b) demultipleksor



Joonis 3.2. Kommutaatorite loogikaskeemid: a) multipleksor, b) demultipleksor

3.2. Summaatorid

Summaatoriks nimetatakse arvuti loogikalülitust, mis on ette nähtud arvkoodide aritmeetiliseks summeerimiseks. Mitmejärgulise kahendarvu summaator koosneb mitmest ühejärgulisest summaatorist. Arvu summeerimisel tuleb lisaks kahe summeeritava arvu vastavatele järkudele liita nendega ka nooremate järkude summeerimisel tekkinud ülekanne. Seega on ühejärgulisel summaatoril kolm sisendit ning kaks väljundit. Summaatori loogikatabeli ning loogikafunktsiooni saab tuletada tavapärasest arvude tulba liitmise skeemist:

$$
\begin{array}{cccccc}
 & C_4 & C_3 & C_2 & C_1 & \\
 & & a_3 & a_2 & a_1 & a_0 \\
+ & & & & & \\
 & & b_3 & b_2 & b_1 & b_0 \\
\hline
 & S_4 & S_3 & S_2 & S_1 & S_0 \\
 & & C_4 & C_3 & C_2 & C_1
\end{array}
$$

Vastavalt liitmise skeemile ning loogikale leitakse kõigile sisendite kombinatsioonidele väljundite väärtused ning esitatakse need tabelina. Seda tabelit 3.1 nimetatakse summaatori loogikatabeliks.

Tabel 3.1

Summaatori loogikatabel

a_i	b_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Loogikatabeli põhjal kirjutatakse väljundite jaoks loogikafunktsioonid:

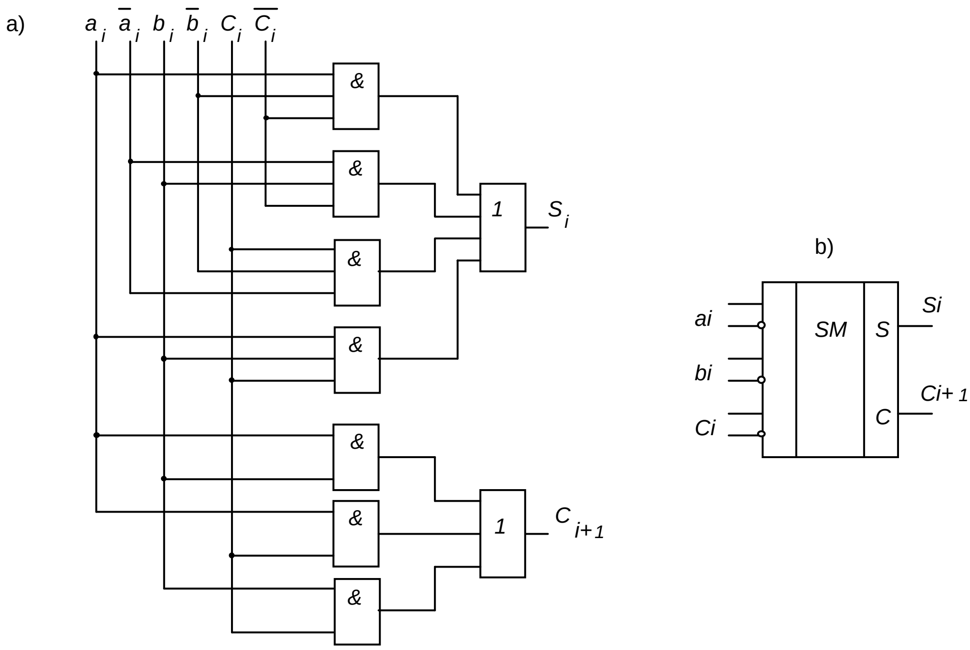
$$S_i = a_i \cdot \overline{b} \cdot_i \overline{C_i} \vee \overline{a_i} \cdot b_i \cdot \overline{C_i} \vee \overline{a_i} \cdot \overline{b_i} \cdot C_i \vee a \cdot b_i \cdot C_i; \quad (3.5)$$

$$C_{i+1} = a_i \cdot b_i \cdot \overline{C_i} \vee a_i \cdot \overline{b_i} \cdot C_i \vee \overline{a_i} \cdot b_i \cdot C_i \vee a_i \cdot b_i \cdot C_i . \quad (3.6)$$

Viimase avaldise saab lihtsustada kujule

$$C_{i+1} = a_i \cdot b_i \vee a_i \cdot C_i \vee b_i \cdot C_i \,. \tag{3.7}$$

Vastavalt võrranditele (3.5) ja (3.7) koostatakse ühejärgulise (ühebitise) kahendsummaatori loogikaskeem (joonis 3.3). Samas on näidatud ka summaatori tingmärk.

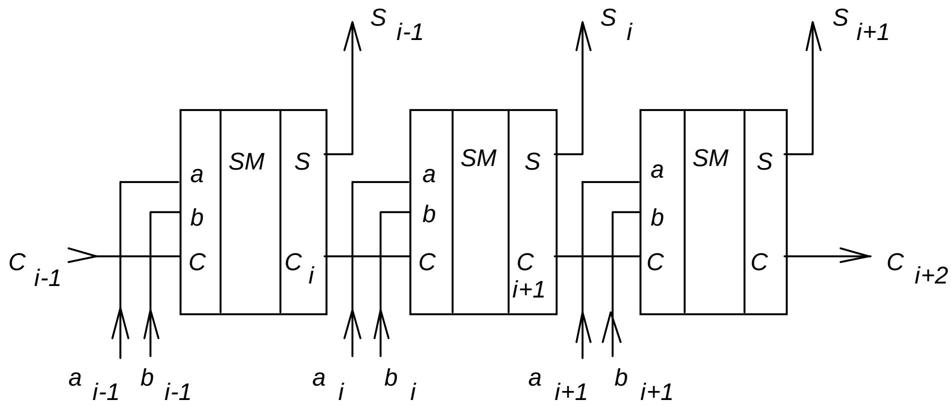


Joonis 3.3. Ühebitine täissummaator: a) loogikaskeem, b) tingmärk

Mitmejärgulised kahendsummaatorid jagunevad:

1) jadaülekandega,
2) rööpülekandega ja
3) rühmaülekandega summaatoriteks.

Jadaülekandega summaatoris moodustatakse väljundsignaal arvukohtade järjestikku summeerimisega, alates kõige nooremast (parempoolsest) kuni kõige vanema ehk vasakpoolsemani välja. Seega moodustatakse arvu summa ja ülekandesignaalid kõige nooremas kohas ning alles pärast seda summeeritakse arvude järgmised kohad. Arvukoha summeerimiseks ja ülekande moodustamiseks kulub teatud aeg, mida ülekande seisukohalt võib vaadelda hilistumisena. Kuna ülekanne toimub järjestikku, siis aeglustab see summaatori tööd. Suure kohtade arvu korral on koguhilistumine võrdne hilistumiste summaga üksikutes kohtades.



Joonis 3.4. Jadaülekandega kahendsummaatori loogikaskeem

Rööpülekandega summaatorid töötavad palju kiiremini kui jadaülekandega summaatorid. Mitmekohalise kahendarvu summeerimisel moodustatakse ülekanne korraga kõigi kohtade jaoks. Seetõttu ei kulu ülekandeks lisaaega ning summaator töötab kiiremini kui jadaülekande korral. Rööpülekandega summaatori tööpõhimõte on järgmine.

Ülekanne $i+1$ järku on avaldatav võrrandiga

$$C_{i+1} = a_i \cdot b_i \vee C_i (a_i \vee b_i). \tag{3.8}$$

Ülekanne i järku on omakorda avaldatav võrrandiga

$$C_i = a_{i-1} \cdot b_{i-1} \vee C_{i-1}(a_{i-1} \vee b_{i-1}). \tag{3.9}$$

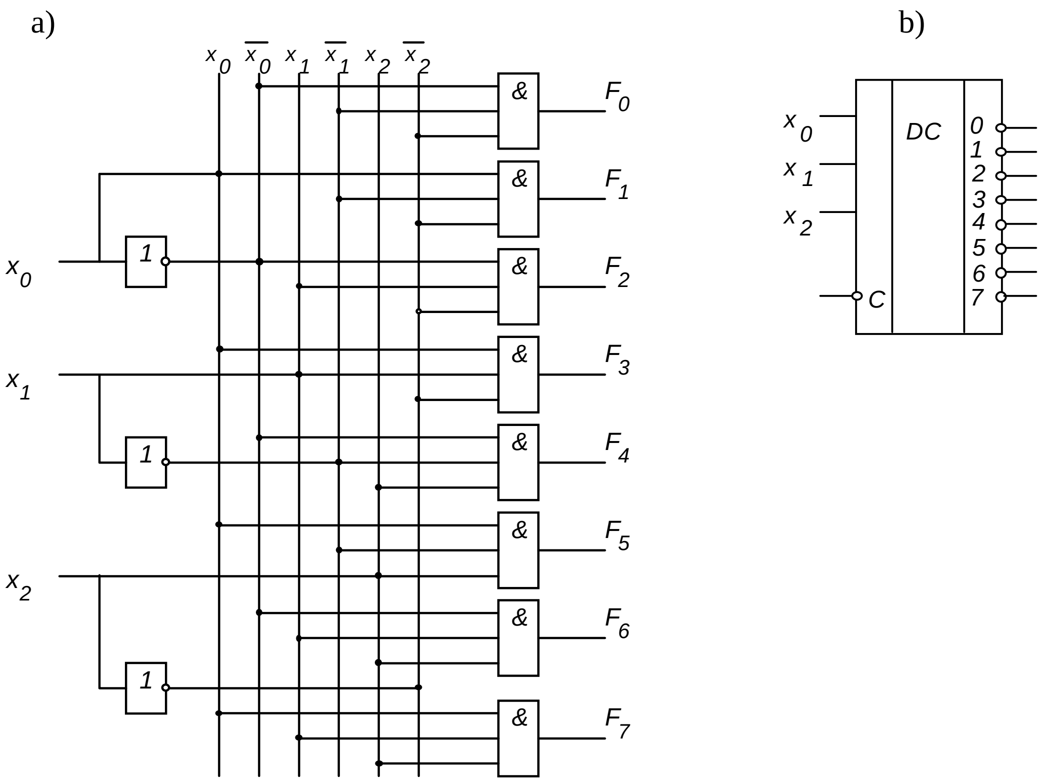
Nii jätkates saab kirjutada ülekande avaldised summaatori kõigi kohtade jaoks kuni noorema kohani välja. Kui asendada seejärel ülekanded, alates kõige nooremast, vastavate avaldistega, siis

$$\begin{aligned} C_{i+1} &= a_i \cdot b_i \vee a_{i-1} \cdot b_{i-1}(a_i \vee b_i) \vee a_{i-2} \cdot b_{i-2}(a_i \vee b_i)(a_{i-1} \vee b_{i-1}) \vee \\ &\vee \cdots \vee a_1 \cdot b_1 (a_i \vee b_i)(a_{i-1} \vee b_{i-1}) \cdots (a_2 \vee b_2) \vee C_0 (a_i \vee b_i)(a_{i-1} \vee b_{i-1}) \cdots (a_2 \cdot b_2)(a_1 \vee b_1). \end{aligned} \tag{3.10}$$

Valemi (1.34) järgi võib konstrueerida skeemi, mis moodustab ülekanded summaatori kõigi kohtade jaoks korraga. Suure kohtade arvu puhul muutub skeem aga sedavõrd keeruliseks, et rööpülekandega summaatori ehitamine osutub ebaotstarbekaks. Seepärast rakendatakse rööpülekande põhimõtet kombineeritult koos jadaülekandega. Vastavaid summaatoreid nimetatakse **rühmaülekandega summaatoriteks**.

3.3. Koodrid ja dekoodrid

Dekooder on lülitus, mis on ette nähtud etteantud sisendkoodi muundamiseks soovitud väljundkoodiks. Ta tunneb ära sisestatava kahendarvu ja annab siganaali vastavasse väljundisse (joonis 3.5). Dekoodri ülesanneteks on muundada kahendkoodis arv niisuguseks koodiks, millega saab aktiveerida nõutava mälupesa, juhtida number- või tähtindikaatorit (joonis 3.6 ja 3.7), tunda ära mitmesuguseid kodeeritud signaale jne. Kuna dekoodri väljundisse ühendatavad seadmed on erinevad, siis kasutatakse nende juhtimiseks ka erinevaid dekoodreid. Näiteks on indikaatoritest levinumad 7-segmendilised vedelkristall- ja valgusdioodindikaatorid ning 10-numbrilised huumlahendusindikaatorid. Seitsme-segmendilise indikaatori dekoodril on reeglina 4 sisendit ning 7 väljundit, kümnenumbrilisel aga 4 sisendit ja 10 väljundit. Üldjuhul on dekoodril nii mitu sisendit *n*, kui mitu kohta on sisendisse antaval kahendarvul. Maksimaalne väljundite arv võrdub kombinatsioonide arvuga 2^n. Dekoodreid koostatakse peamiselt NING-elementidest.



Joonis 3.5. Kolmebitine kahendsignaali dekooder a) loogikaskeem, b) tingmärk

Suure sisendite arvu korral kasutatakse dekodeerimiseks nn kaskaadlülitust, kus esimese astme dekooder aktiveerib ühe teise astme dekoodri ning see omakorda ühe väljundi. Kahendkoodi dekoodri tööd kirjeldavad harilikult järgmised võrrandid:

$$
\begin{aligned}
F_0 &= \bar{x}_1\bar{x}_2\ldots\bar{x}_{n-1}\bar{x}_n,\\
F_1 &= \bar{x}_1\bar{x}_2\ldots\bar{x}_{n-1}x_n,\\
F_2 &= \bar{x}_1\bar{x}_2\ldots x_{n-1}\bar{x}_n,
\end{aligned}
\qquad (3.11)
$$

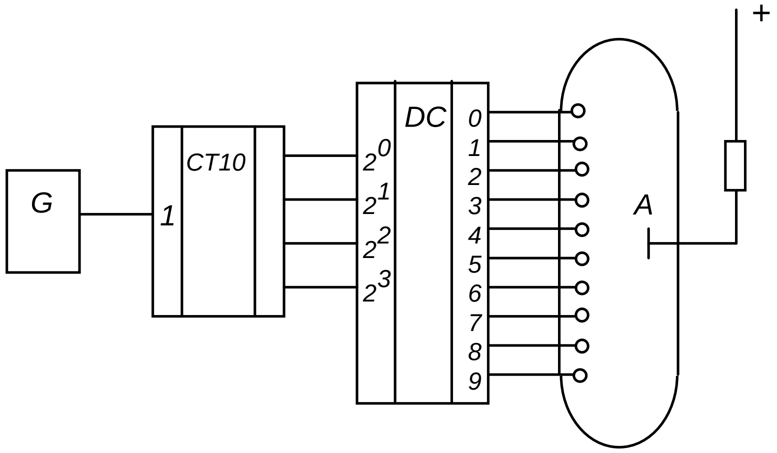
$\ldots$

$$F_{2^n-1} = x_1 x_2 \dots x_{n-1} x_n.$$

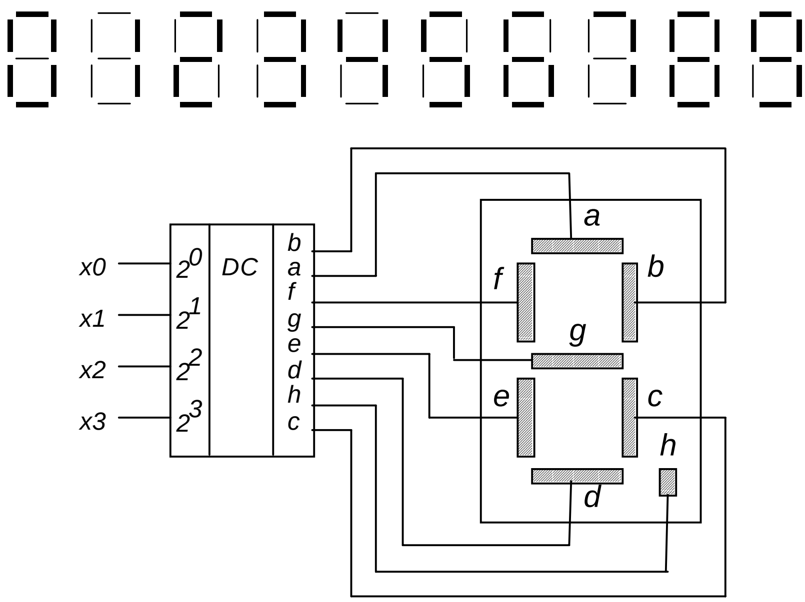
Seitsmesegmendilise indikaatori dekooder peab sisendisse antud kahendkoodi kohaselt lülitama sisse indikaatori segmendid nii, et hakkaks helendama arvule vastav kümnendnumber. Dekoodril on neli sisendit ja seitse väljundit. Kaheksandat, komasegmenti, dekoodriga ei juhita. Kuna segment *a* ei helendu numbrite 1 ja 4 korral, siis võib kirjutada, et

$$\bar{a} = \bar{x}_3\bar{x}_2\bar{x}_1 x_0 + \bar{x}_3 x_2 \bar{x}_1 \bar{x}_0 \tag{3.12}$$

Analoogilised avaldised saab kirjutada ka kõigi ülejäänud segmentide kohta.

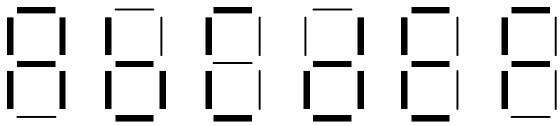


Joonis 3.6. Dekoodri kasutamine kümnenumbrilise huumlahendusindikaatori juhtimiseks



Joonis 3.7. Dekoodri kasutamine seitsmesegmendilise indikaatori juhtimiseks

Kuueteistkümnendarvude tähistamiseks kasutatavaid ladina tähestiku tähti esitatakse 7-segmendilise indikaatori abil alljärgnevalt.



Tabel 3.2

Seitsmesegmendilise indikaatori dekooderi loogikatabel

	Sisendid				Väljundid						
	2^3	2^2	2^1	2^0	*a*	*b*	*c*	*d*	*e*	*f*	*g*
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0							
B	1	0	1	1							
C	1	1	0	0							
D	1	1	0	1							
E	1	1	1	0							
F	1	1	1	1							

Ülesanne iseseisvaks lahendamiseks

Täitke 7-segmendilise indikaatori dekooderi loogikatabeli tühjad ruudud vastavalt joonisel esitatud tähemärkidele.

3.4. Aritmeetika-loogikaplokk

Aritmeetika-loogikaplokk (*ALU - arithmetic logic unit*) on ette nähtud aritmeetika- ja loogikateheteks kahendarvudega (joonis 3.8) [1]. Kõik aritmeetikatehted sooritatakse arvude või nende täiendkoodide summeerimisega ja nihutamisega. Peamised loogikatehted on NING, VÕI, EI ja Mod2, mille täitmiseks on *ALU*-s vastavad loogikalülitused.

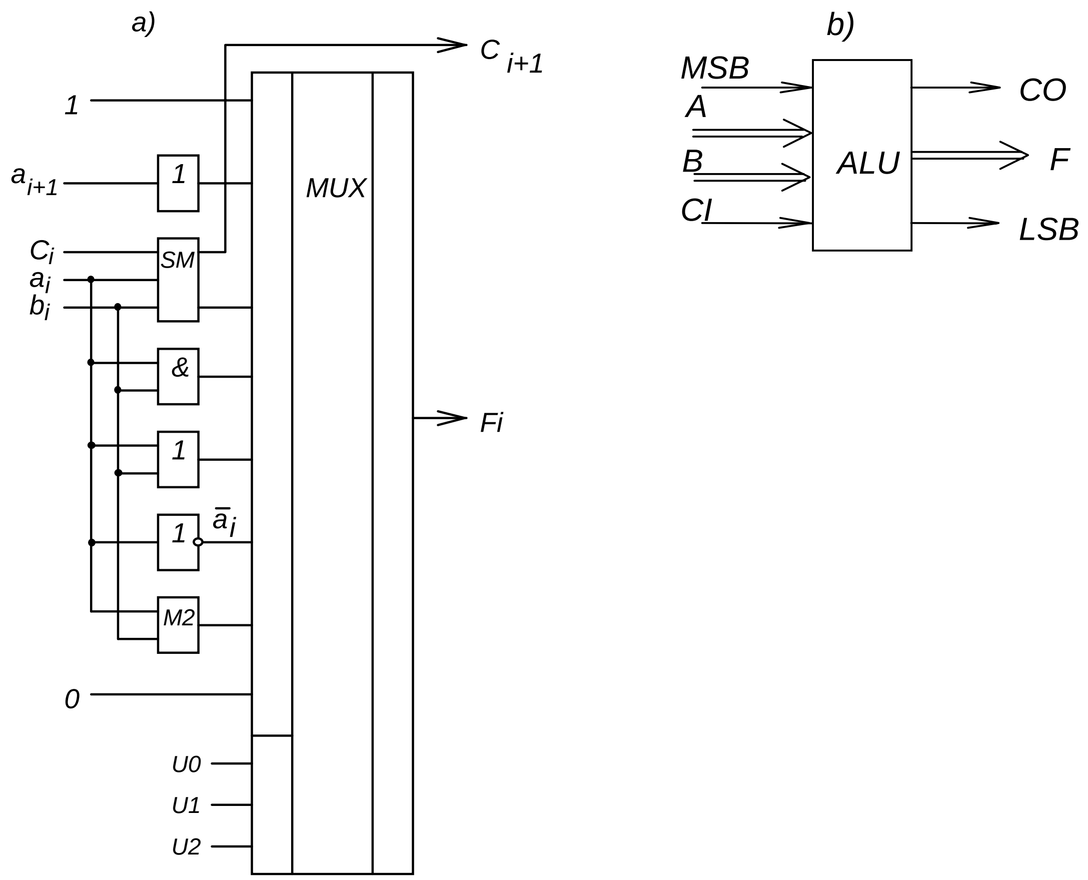
Erinevate tehete selekteerimiseks on aritmeetika-loogikaplokil kommutaator *MUX*. Mitmebitiste operandide $A = a_n, a_{n-1} \ldots a_1, a_0$, ja $B = b_n, b_{n-1} \ldots b_1, b_0$ ning bittide *MSB* (*most significant bit*) ja *CI* (*carry in*) summeerimisel kombinatsioonsummaatoriga saadakse kahendsumma $S = s_n, s_{n-1} \ldots s_1, s_0$ ning ülekandebitid *CO* (*carry out*) ja *LSB* (*least *significant bit*). Ülekandebitt *CI* tähistab ülekannet kõrgemast bitist madalamasse ja *CO* vastupidi madalamast bitist kõrgemasse. Mitmebitise *ALU* madalaima ja kõrgeima biti sisendmuutujad on vastavalt *LSB* ehk madalaim bitt (viimane oluline bitt) ja *MSB* ehk kõrgeim bitt (kõige tähtsam bitt). Loogikatehetel ülekannet ei esine.

Multipleksor valib etteantud juhtkoodi $u_2\ u_1\ u_0$ järgi ühe funktsionaalsetest sisenditest ja suunab selle tulemi väljundisse F_i. Näiteks koodi 101 puhul $F_i = S_i$ (kahendliitmine ülekandega C_{i+1}), koodi 011 puhul $F_i = a_i$Ùb_i jne. Koodi 000 puhul $F_i = 0$ ja koodi 111 puhul $F_i = 1$.

Aritmeetikatehete operandide ja tulemite salvestamiseks kasutatakse registreid. Kahendsõnad suunatakse registritest *ALU* sisenditesse ja *ALU* väljundist registritesse multipleksorite ja demultipleksorite abil.

Otstarbekas on registreerida ka tehte tulemi teisi tunnuseid, nagu ületäitumine, nulltulem, negatiivne tulem jms. Selleks kasutatakse mikroprotsessoris olekuregistrit.

Teheteks mitmebitiste kahendarvudega kasutatakse ka vastava bittide arvuga *ALU*-sid. Mitmebitise *ALU* saab koostada ühebitistest *ALU*-dest. *ALU* loogikaskeem ja lihtsustatud tähis on joonisel 3.8.



Joonis 3.8. Aritmeetika-loogikaplokk: a) loogikaskeem, b) tingmärk,