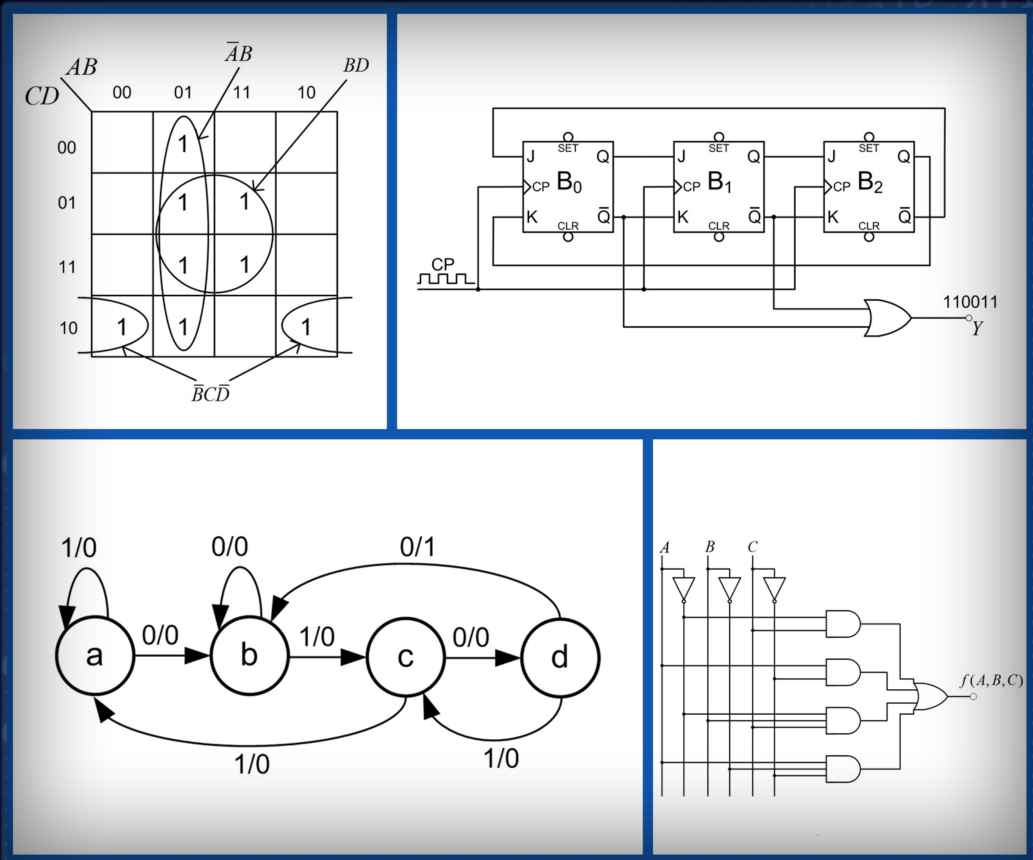
VISOKA TEHNIČKA ŠKOLA U BJELOVARU

Zoran Vrhovski ▪ Ivan Šumiga



DIGITALNA TEHNIKA

Zbirka riješenih zadataka

Bjelovar, 2015.

VISOKA TEHNIČKA ŠKOLA U BJELOVARU

DIGITALNA TEHNIKA

Zbirka riješenih zadataka

Prvo izdanje

Bjelovar, 2015.

Zoran Vrhovski, mag. ing. el. techn. inf.
Mr. sc. Ivan Šumiga

DIGITALNA TEHNIKA
Zbirka riješenih zadataka

Nakladnik:
Visoka tehnička škola u Bjelovaru

Za nakladnika:
prof. dr. sc. Ante Čikić

Recenzenti:
dr. sc. Igor Petrović
dr. sc. Mihael Kukec

Lektorica:
Ivana Jurković, prof.

Priprema za tisak:
Zoran Vrhovski, mag. ing. el. techn. inf.

Oblikovanje naslovnice:
Josip Horvat

Tisak:
Croatiagraf d.o.o. Markovac Križevački, www.croatiagraf.hr
Veljača, 2015.

Naklada:
120 primjeraka

CIP zapis dostupan u računalnome katalogu Nacionalne i sveučilišne knjižnice u Zagrebu pod brojem 000897735.
ISBN 978-953-7676-20-9

Zoran Vrhovski, mag. ing. el. techn. inf.
Mr. sc. Ivan Šumiga

DIGITALNA TEHNIKA

Zbirka riješenih zadataka

Prvo izdanje

Bjelovar, 2015.

Predgovor

Tko želi nešto naučiti, naći će način;
tko ne želi, naći će izliku.
Pablo Picasso

Ova zbirka zadataka iz Digitalne tehnike namijenjena je prvenstveno studentima Stručnog studija mehatronike na Visokoj tehničkoj školi u Bjelovaru koji slušaju kolegij *Digitalna tehnika*. Zbirka zadataka može se koristiti i na svim drugim veleučilištima i sveučilištima koja se u sklopu svog programa bave digitalnim sklopovima.

Osnovni je cilj ove zbirke zadataka približiti digitalnu tehniku studentima i ostalima koji će čitati i proučavati ovaj materijal. Zbirka zadataka sadrži 149 stranica sa 63 riješena primjera i 63 zadataka za vježbu. Svako poglavlje u uvodnom dijelu sadrži teorijski dio koji je potrebno proučiti za što bolje razumijevanje riješenih vježbi i zadataka za vježbu. Studentima se preporučuje da najprije prouče riješene primjere, a nakon toga da rješavaju zadatke za vježbu.

Autori su zbirku zadataka napisali zajedničkim snagama s jednakim doprinosom u svakom poglavlju.

Zahvaljujemo recenzentima, dr. sc. Igoru Petroviću i dr. sc. Mihaelu Kukecu na korisnim savjetima i sugestijama te lektorici Ivani Jurković, prof. na strpljenju u čitanju ove knjige i usklađivanju teksta s hrvatskim standardnim jezikom.

Posebno zahvaljujemo Visokoj tehničkoj školi u Bjelovaru na financijskoj potpori bez koje izdavanje ove zbirke zadataka ne bi bilo moguće.

Zoran Vrhovski, mag. ing. el. techn. inf.
Mr. sc. Ivan Šumiga

Sadržaj

Poglavlje 1

Uvod

Digitalni sustavi različitog stupnja složenosti, zahvaljujući razvoju mikroelektronike i računala, danas su prisutni u svim područjima ljudske djelatnosti. Studenti Visoke tehničke škole u Bjelovaru s osnovama rada takvih sustava susreću se na kolegiju Digitalna tehnika. Ova zbirka kroz riješene primjere studentima nastoji približiti najvažnije probleme kako bi lakše svladali temeljna znanja iz digitalne tehnike.

U ovom uvodnom poglavlju proći ćemo ukratko kroz sva poglavlja ove zbirke zadataka kako bismo zainteresirali čitatelje, a to su na prvom mjestu studenti Stručnog studija mehatronike Visoke tehničke škole u Bjelovaru.

U drugom poglavlju opisani su težinski brojevni sustavi s naglaskom na binarne, oktalne, dekadske i heksadekadske sustave. Binarni se sustav uz heksadekadski najviše koristi kod projektiranja digitalnih sustava. Nadalje, opisane su operacije s binarnim brojevima, primjeri kodiranja, zaštite podataka te otkrivanje i ispravljanje pogrešaka.

Osnove digitalne logike, pravila Booleove algebre i zapis logičkih funkcija u kanonskom obliku opisani su u trećem poglavlju. Nadalje, u ovom poglavlju prikazane su realizacije logičkih funkcija pomoću osnovnih logičkih sklopova I, ILI i NE te pomoću univerzalnih logičkih sklopova NI i NILI. Realizacija logičkih funkcija u kanonskom obliku pomoću osnovnih logičkih sklopova zahtjeva velik broj sklopova pa je logičke funkcije potrebno minimizirati. Minimizacija logičkih funkcija prikazana je u četvrtom poglavlju. Opisana je algebarska metoda minimizacije i minimizacija pomoću K-tablica.

U petom poglavlju opisani su SR, JK, T i D bistabili. Za navedene bistabile prikazane su tablice stanja i tablice uzbude. Bistabili su temeljni elementi za pamćenje binarnih podataka. Svojstvo pamćenja koristi se kod projektiranja sekvencijskih sklopova što je opisano u zadnjem, šestom poglavlju. Od sekvencijskih sklopova u šestom su poglavlju opisani binarno sinkrono i asinkrono brojilo, prstenasto brojilo, Johnsonovo brojilo, generator niza i detektor niza.

Poglavlje 2

Brojevni sustavi

2.1 Pretvorbe između brojevnih sustava

Brojevni sustavi mogu biti pozicijski i nepozicijski. Najpoznatiji nepozicijski brojevni sustav koji se može naći u primjeni (npr. oznaka mjeseca u godini) je rimski brojevni sustav. Danas su u praktičnoj primjeni uglavnom pozicijski brojevni sustavi kod kojih vrijednost znamenke ovisi o njenoj poziciji u broju. Općenito se cijeli broj s n znamenaka može zapisati u obliku polinoma kao:

$$N = a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + a_{n-3}B^{n-3} + ... + a_2B^2 + a_1B^1 + a_0B^0 \tag{2.1}$$

pri čemu su:

- N - cijeli broj,
- a_{n-1}, a_{n-2}, a_{n-3},... , a_2, a_1, a_0 - znamenke broja N,
- B - baza brojevnog sustava.

Najpoznatiji i najčešće korišten brojevni sustav je dekadski sustav. Prema izrazu (2.1) cijeli dekadski broj 19263 može se napisati kao $1 \cdot 10^4 + 9 \cdot 10^3 + 2 \cdot 10^2 + 6 \cdot 10^1 + 3 \cdot 10^0$.

Brojevni sustavi koji imaju značajniju primjenu su:

- binarni ($B = 2$, znamenke: 0, 1),
- oktalni ($B = 8$, znamenke: 0, 1, 2, 3, 4, 5, 6, 7),
- dekadski ($B = 10$, znamenke: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9),
- heksadekadski ($B = 16$, znamenke: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

Realni se brojevi mogu prikazati u brojevnom sustavu s proizvoljnom bazom prema izrazu:

$$R = a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + ... + a_1B^1 + a_0B^0 + a_{-1}B^{-1} + ... + a_{-m+1}B^{-m+1} + a_{-m}B^{-m} \tag{2.2}$$

gdje su:

- R - realni broj,
- a_{n-1}, ..., a_1, a_0, a_{-1}, ..., a_{-m} - znamenke broja R,

- m - broj decimalnih mjesta,
- B - baza brojevnog sustava.

Iza znamenke a_0 stavlja se decimalni zarez ili decimalna točka. Broj R u brojevnom sustavu s bazom B zapisivat ćemo kao R_B. Na primjer, broj $1A3F,B1_{16}$ pripada heksadekadskom sustavu s bazom $B = 16$. U dekadskom sustavu se baza $B = 10$ podrazumijeva te se u tom slučaju ona najčešće izostavlja u zapisu broja.

☞ Primjer 2.1.1

U obliku polinoma prikažite sljedeće cijele brojeve:

a) 101010_2,

b) 102020_3,

c) 264_8,

d) 1234_{10},

e) $F4B_{16}$.

Navedene brojeve prikažite u dekadskom sustavu.

✍ **Rješenje:** Pretvorbu brojeva napravit ćemo pomoću relacije (2.1).

$$\begin{aligned}
\text{a) } 101010_2 &= 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \\
&= 32 + 0 + 8 + 0 + 2 + 0 \\
&= 42_{10}
\end{aligned}$$

$$\begin{aligned}
\text{b) } 102020_3 &= 1 \cdot 3^5 + 0 \cdot 3^4 + 2 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 0 \cdot 3^0 \\
&= 243 + 0 + 54 + 0 + 6 + 0 \\
&= 303_{10}
\end{aligned}$$

$$\begin{aligned}
\text{c) } 264_8 &= 2 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0 \\
&= 128 + 48 + 4 \\
&= 180_{10}
\end{aligned}$$

$$\begin{aligned}
\text{d) } 1234_{10} &= 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 \\
&= 1000 + 200 + 30 + 4 \\
&= 1234_{10}
\end{aligned}$$

$$\begin{aligned}
\text{e) } \mathrm{F4B}_{16} &= \mathrm{F} \cdot 16^2 + 4 \cdot 16^1 + \mathrm{B} \cdot 16^0 \\
&= 15 \cdot 16^2 + 4 \cdot 16^1 + 11 \cdot 16^0 \\
&= 3840 + 64 + 11 \\
&= 3915_{10}
\end{aligned}$$

☞ Primjer 2.1.2

U obliku polinoma prikažite sljedeće realne brojeve:

a) $111001,1101_2$,

b) $112001,11_3$,

c) $615{,}4_8$,

d) $123{,}987_{10}$,

e) $F{,}5A_{16}$.

Navedene brojeve prikažite u dekadskom sustavu.

✍ **Rješenje:** Pretvorbu brojeva napravit ćemo pomoću relacije (2.2).

$$\begin{aligned}
\text{a) } 111001{,}101_2 &= 1\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3}\\
&= 32 + 16 + 8 + 0 + 0 + 1 + 0{,}5 + 0 + 0{,}125\\
&= 57{,}625_{10}\\
\text{b) } 112001{,}11_3 &= 1\cdot 3^5 + 1\cdot 3^4 + 2\cdot 3^3 + 0\cdot 3^2 + 0\cdot 3^1 + 1\cdot 3^0 + 1\cdot 3^{-1} + 1\cdot 3^{-2}\\
&= 243 + 81 + 54 + 0 + 0 + 1 + 0{,}\dot{3} + 0{,}\dot{1}\\
&= 379{,}\dot{4}_{10}\\
\text{c) } 615{,}4_8 &= 6\cdot 8^2 + 1\cdot 8^1 + 5\cdot 8^0 + 4\cdot 8^{-1}\\
&= 384 + 8 + 5 + 0{,}5\\
&= 397{,}5_{10}\\
\text{e) } 123{,}987_{10} &= 1\cdot 10^2 + 2\cdot 10^1 + 3\cdot 10^0 + 9\cdot 10^{-1} + 8\cdot 10^{-2} + 7\cdot 10^{-3}\\
&= 100 + 20 + 3 + 0{,}9 + 0{,}08 + 0{,}007\\
&= 123{,}987_{10}\\
\text{f) } F{,}5A_{16} &= F\cdot 16^0 + 5\cdot 16^{-1} + A\cdot 16^{-2}\\
&= 15\cdot 16^0 + 5\cdot 16^{-1} + 10\cdot 16^{-2}\\
&= 15 + 0{,}3125 + 0{,}0390625\\
&= 15{,}3515625_{10}
\end{aligned}$$

☞ Primjer 2.1.3

Pretvorite cijeli broj 123_{10} u binarni, ternarni, oktalni i heksadekadski sustav.

✍ **Rješenje:**

Pretvorba se provodi sukcesivnim dijeljenjem bazom brojevnog sustava pri čemu ostatak dijeljenja predstavlja znamenku broja, a ostatak se dalje dijeli bazom brojevnog sustava.

a) Binarni brojevni sustav ima bazu $B = 2$. Pretvorba broja 123_{10} u binarni brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rrll}
123 : 2 = & 61 & \text{i ostatak} & 1\\
61 : 2 = & 30 & \text{i ostatak} & 1\\
30 : 2 = & 15 & \text{i ostatak} & 0\\
15 : 2 = & 7 & \text{i ostatak} & 1 \uparrow\\
7 : 2 = & 3 & \text{i ostatak} & 1\\
3 : 2 = & 1 & \text{i ostatak} & 1\\
1 : 2 = & 0 & \text{i ostatak} & 1
\end{array}$$

Ostatak pri dijeljenju s bazom $B = 2$ su znamenke binarnog broja koje se čitaju od dna prema vrhu pa vrijedi da je $123_{10} = 1111011_2$.

b) Ternarni brojevni sustav ima bazu $B = 3$. Pretvorba broja 123_{10} u ternarni brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rrll} 123:3= & 41 & \text{i ostatak} & 0 \\ 41:3= & 13 & \text{i ostatak} & 2 \\ 13:3= & 4 & \text{i ostatak} & 1 \ \uparrow \\ 4:3= & 1 & \text{i ostatak} & 1 \\ 1:3= & 0 & \text{i ostatak} & 1 \end{array}$$

Ostatak pri dijeljenju s bazom $B = 3$ su znamenke ternarnog broja koje se čitaju od dna prema vrhu pa vrijedi da je $123_{10} = 11120_3$.

c) Oktalni brojevni sustav ima bazu $B = 8$. Pretvorba broja 123_{10} u oktalni brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rrll} 123:8= & 15 & \text{i ostatak} & 3 \\ 15:8= & 1 & \text{i ostatak} & 7 \ \uparrow \\ 1:8= & 0 & \text{i ostatak} & 1 \end{array}$$

Ostatak pri dijeljenju s bazom $B = 8$ su znamenke oktalnog broja koje se čitaju od dna prema vrhu pa vrijedi da je $123_{10} = 173_8$.

d) Heksadekadski brojevni sustav ima bazu $B = 16$. Pretvorba broja 123_{10} u heksadekadski brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rrll} 123:16= & 7 & \text{i ostatak} & 11(\text{B}) \\ 7:16= & 0 & \text{i ostatak} & 7 \end{array} \uparrow$$

Ostatak pri dijeljenju s bazom $B = 16$ su znamenke heksadekadskog broja koje se čitaju od dna prema vrhu pa vrijedi da je $123_{10} = 7\text{B}_{16}$.

☞ Primjer 2.1.4

Pretvorite cijeli broj 235_8 u ternarni sustav.

✍ **Rješenje:** Cijeli broj 235_8 može se pretvoriti u ternarni sustav sukcesivnim dijeljenjem s bazom $B = 3$ u oktalnom sustavu. Ovaj bi postupak doveo do konfuzije u ljudskoj glavi jer su ljudi navikli množiti, dijeliti, zbrajati i oduzimati u dekadskom sustavu. Prema tome, cijeli broj 235_8 najprije ćemo pretvoriti u dekadski, a zatim u ternarni sustav. Pretvorbu cijelog broja 235_8 u dekadski sustav napravit ćemo pomoću relacije (2.1).

$$\begin{aligned} 235_8 &= 2 \cdot 8^2 + 3 \cdot 8^1 + 5 \cdot 8^0 \\ &= 128 + 24 + 5 \\ &= 157 \end{aligned}$$

Ternarni brojevni sustav ima bazu $B = 3$. Pretvorba broja 157_{10} u ternarni brojevni sustav provodi se na sljedeći način:

$$\begin{aligned} 157:3 &= 52 \quad \text{i ostatak} \quad 1 \\ 52:3 &= 17 \quad \text{i ostatak} \quad 1 \\ 17:3 &= 5 \quad \text{i ostatak} \quad 2 \ \uparrow \\ 5:3 &= 1 \quad \text{i ostatak} \quad 2 \\ 1:3 &= 0 \quad \text{i ostatak} \quad 1 \end{aligned}$$

Ostatak pri dijeljenju s bazom $B = 3$ su znamenke ternarnog broja koje se čitaju od dna prema vrhu pa vrijedi da je $157_{10} = 12211_3$. Budući da je cilj zadatka bio pretvorba oktalnog broja u ternarni broj, vrijedi da je $235_8 = 12211_3$.

☞ Primjer 2.1.5

Pretvorite sljedeće realne brojeve iz dekadskog sustava u zadani sustav:
a) $0{,}75_{10}$ u binarni sustav,
b) $15{,}65_{10}$ u binarni sustav,
c) $0{,}125_{10}$ u oktalni sustav,
d) $61{,}24_{10}$ u oktalni sustav,
e) $0{,}0625_{10}$ u heksadekadski sustav,
f) $61{,}1_{10}$ u heksadekadski sustav.

✍ **Rješenje:** Kod pretvorbe realnih brojeva iz dekadskog u neki drugi sustava, posebno se pretvara cjelobrojni dio, a posebno decimalni dio. Pretvorba decimalnog djela provodi se sukcesivnim množenjem s bazom sustava u koji se realni broj pretvara. Cjelobrojni dio u svakom koraku množenja predstavlja znamenku broja u željenoj bazi, dok se decimalni dio dalje množi s bazom. Postupak završava kada decimalni dio iznosi nula. Konačan rezultat je povezivanje cijelog dijela i decimalnog dijela pomoću decimalnog zareza. Na primjer, ako cijeli dio iznosi 11_2, a decimalni dio 1001_2, tada realan binarni broj iznosi $11{,}1001_2$.

a) Realan broj $0{,}75_{10}$ potrebno je pretvoriti u binarni sustav. Cijeli dio realnog broja $0{,}75_{10}$ je 0. Broj 0 nije potrebno pretvarati u binarni sustav jer je 0 u svim sustavima jednaka. Vrijedi da je $0_{10} = 0_2$. Decimalni dio realnog broja $0{,}75_{10}$ iznosi 0,75. Pretvorba decimalnog dijela broja $0{,}75_{10}$ u binarni brojevni sustav provodi se na sljedeći način:

$$\begin{aligned} 0{,}75 \cdot 2 &= 1{,}5 \quad a_{-1} = 1 \quad \text{i decimalni dio} \quad 0{,}5 \\ 0{,}5 \cdot 2 &= 1{,}0 \quad a_{-2} = 1 \quad \text{i decimalni dio} \quad 0{,}0 \ \downarrow \end{aligned}$$

Cjelobrojni dijelovi dobiveni množenjem s bazom $B = 2$ su znamenke decimalnog dijela binarnog broja koje se čitaju od vrha prema dnu pa vrijedi da je $0{,}75_{10} = 0{,}11_2$.

b) Realan broj $15{,}65_{10}$ potrebno je pretvoriti u binarni sustav. Cijeli dio realnog broja $15{,}65_{10}$ je 15. Binarni brojevni sustav ima bazu $B = 2$. Pretvorba broja 15_{10} u binarni brojevni sustav provodi se na sljedeći način:

$$\begin{aligned} 15:2 &= 7 \quad \text{i ostatak} \quad 1 \\ 7:2 &= 3 \quad \text{i ostatak} \quad 1 \\ 3:2 &= 1 \quad \text{i ostatak} \quad 1 \ \uparrow \\ 1:2 &= 0 \quad \text{i ostatak} \quad 1 \end{aligned}$$

Ostatak pri dijeljenju s bazom $B = 2$ su znamenke binarnog broja koje se čitaju od dna prema vrhu pa vrijedi da je $15_{10} = 1111_2$.

Decimalni dio realnog broja $15{,}65_{10}$ iznosi 0,65. Pretvorba decimalnog dijela broja $15{,}65_{10}$ u binarni brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rrlll}
0{,}65 \cdot 2 = & 1{,}3 & a_{-1} = 1 & \text{i decimalni dio} & 0{,}3 \\
0{,}3 \cdot 2 = & 0{,}6 & a_{-2} = 0 & \text{i decimalni dio} & 0{,}6 \\
0{,}6 \cdot 2 = & 1{,}2 & a_{-3} = 1 & \text{i decimalni dio} & 0{,}2 \\
0{,}2 \cdot 2 = & 0{,}4 & a_{-4} = 0 & \text{i decimalni dio} & 0{,}4 \\
0{,}4 \cdot 2 = & 0{,}8 & a_{-5} = 0 & \text{i decimalni dio} & 0{,}8 \downarrow \\
0{,}8 \cdot 2 = & 1{,}6 & a_{-6} = 1 & \text{i decimalni dio} & 0{,}6 \\
 & & \vdots & &
\end{array}$$

Postupak ćemo zaprovoditi na šestom decimalnom mjestu. Cjelobrojni dijelovi dobiveni množenjem s bazom $B = 2$ su znamenke decimalnog dijela binarnog broja koje se čitaju od vrha prema dnu pa vrijedi da je $0{,}65_{10} = 0{,}101001..._2$. Prema tome, vrijedi da je $15{,}65_{10} = 1111{,}101001..._2$.

c) Realan broj $0{,}125_{10}$ potrebno je pretvoriti u oktalni sustav. Cijeli dio realnog broja $0{,}125_{10}$ je 0. Vrijedi da je $0_{10} = 0_8$. Decimalni dio realnog broja $0{,}125_{10}$ iznosi 0,125. Pretvorba decimalnog dijela broja $0{,}125_{10}$ u oktalni brojevni sustav provodi se na sljedeći način:

$$0{,}125 \cdot 8 = \quad 1{,}0 \quad a_{-1} = 1 \quad \text{i decimalni dio} \quad 0{,}0 \downarrow$$

Cjelobrojni dijelovi dobiveni množenjem s bazom $B = 8$ su znamenke decimalnog dijela oktalnog broja koje se čitaju od vrha prema dnu pa vrijedi da je $0{,}125_{10} = 0{,}1_8$.

d) Realan broj $61{,}24_{10}$ potrebno je pretvoriti u oktalni sustav. Cijeli dio realnog broja $61{,}24_{10}$ je 61. Oktalni brojevni sustav ima bazu $B = 8$. Pretvorba broja 61_{10} u oktalni brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rrll}
61 : 8 = & 7 & \text{i ostatak} & 5 \\
7 : 8 = & 0 & \text{i ostatak} & 7
\end{array} \uparrow$$

Ostatak pri dijeljenju s bazom $B = 8$ su znamenke oktalnog broja koje se čitaju od dna prema vrhu pa vrijedi da je $61_{10} = 75_8$.

Decimalni dio realnog broja $61{,}24_{10}$ iznosi 0,24. Pretvorba decimalnog dijela broja $61{,}24_{10}$ u oktalni brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rrlll}
0{,}24 \cdot 8 = & 1{,}92 & a_{-1} = 1 & \text{i decimalni dio} & 0{,}92 \\
0{,}92 \cdot 8 = & 7{,}36 & a_{-2} = 7 & \text{i decimalni dio} & 0{,}36 \\
0{,}36 \cdot 8 = & 2{,}88 & a_{-3} = 2 & \text{i decimalni dio} & 0{,}88 \\
0{,}88 \cdot 2 = & 7{,}04 & a_{-4} = 7 & \text{i decimalni dio} & 0{,}04 \downarrow \\
 & & \vdots & &
\end{array}$$

Postupak ćemo zaprovoditi na četvrtom decimalnom mjestu. Cjelobrojni dijelovi dobiveni množenjem s bazom $B = 8$ su znamenke decimalnog dijela oktalnog broja koje se čitaju od vrha prema dnu pa vrijedi da je $0{,}24_{10} = 0{,}1727..._8$. Prema tome, vrijedi da je $61{,}24_{10} = 75{,}1727..._8$.

e) Realan broj $0{,}0625_{10}$ potrebno je pretvoriti u heksadekadski sustav. Cijeli dio realnog broja $0{,}0625_{10}$ je 0. Vrijedi da je $0_{10} = 0_{16}$. Decimalni dio realnog broja $0{,}0625_{10}$ iznosi 0,0625.

Pretvorba decimalnog dijela broja $0{,}0625_{10}$ u heksadekadski brojevni sustav provodi se na sljedeći način:

$$0{,}0625 \cdot 16 = \quad 1{,}0 \quad a_{-1} = 1 \quad \text{i decimalni dio} \quad 0{,}0 \ \downarrow$$

Cjelobrojni dijelovi dobiveni množenjem s bazom $B = 16$ su znamenke decimalnog dijela oktalnog broja koje se čitaju od vrha prema dnu pa vrijedi da je $0{,}0625_{10} = 0{,}1_{16}$.

f) Realan broj $61{,}1_{10}$ potrebno je pretvoriti u heksadekadski sustav. Cijeli dio realnog broja $61{,}1_{10}$ je 61. Heksadekadski brojevni sustav ima bazu $B = 16$. Pretvorba broja 61_{10} u heksadekadski brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rlll} 61 : 16 = & 3 & \text{i ostatak} & 13(\mathrm{D}) \\ 3 : 16 = & 0 & \text{i ostatak} & 3 \end{array} \uparrow$$

Ostatak pri dijeljenju s bazom $B = 8$ su znamenke heksadekadskog broja koje se čitaju od dna prema vrhu pa vrijedi da je $61_{10} = 3\mathrm{D}_{16}$.

Decimalni dio realnog broja $61{,}1_{10}$ iznosi 0,1. Pretvorba decimalnog dijela broja $61{,}1_{10}$ u heksadekadski brojevni sustav provodi se na sljedeći način:

$$\begin{array}{rllll} 0{,}1 \cdot 16 = & 1{,}6 & a_{-1} = 1 & \text{i decimalni dio} & 0{,}6 \\ 0{,}6 \cdot 8 = & 9{,}6 & a_{-2} = 9 & \text{i decimalni dio} & 0{,}6 \\ 0{,}6 \cdot 8 = & 9{,}6 & a_{-3} = 9 & \text{i decimalni dio} & 0{,}6 \\ & & \vdots & & \end{array} \downarrow$$

Množenjem smo dobili periodično ponavljanje znamenke 9. Cjelobrojni dijelovi dobiveni množenjem s bazom $B = 16$ su znamenke decimalnog dijela heksadecimalnog broja koje se čitaju od vrha prema dnu pa vrijedi da je $0{,}1_{10} = 0{,}1999..._{16}$. Prema tome, vrijedi da je $61{,}1_{10} = 3\mathrm{D{,}1\dot{9}}_{16}$.

☞ Primjer 2.1.6

Direktnom metodom pretvorite sljedeće brojeve iz oktalnog u binarni sustav:
a) 365_8,
b) $0{,}35_8$,
c) $71{,}26_8$.

✍ Rješenje:

Brojevi iz sustava čija je baza potencija broja 2 ($B = 2^n, n = 2, 3, 4, ...$) mogu se pretvoriti u binarni sustav direktnom metodom. Svaka se znamenka broja iz sustava s bazom $B = 2^n$ može direktno pretvoriti u binarne brojeve s n binarnih znamenaka. Na primjer, znamenka oktalnog sustava ($B = 8 = 2^3$) može se prikazati pomoću tri binarne znamenke. Pretvorba oktalnih znamenki u binarne provodi se pomoću tablice 2.1.

a) $365_8 = \underbrace{011}_{3}\underbrace{110}_{6}\underbrace{101}_{5}{}_2 = 11110101_2$

b) $0{,}35_8 = \underbrace{000}_{0},\underbrace{011}_{3}\underbrace{101}_{5}{}_2 = 0{,}011101_2$

c) $71{,}26_8 = \underbrace{111}_{7}\underbrace{001}_{1},\underbrace{010}_{2}\underbrace{110}_{6}{}_2 = 111001{,}01011_2$

Nakon pretvorbe oktalnih znamenki u binarne pomoću tablice 2.1, poželjno je izbrisati vodeće i prateće nule.

Tablica 2.1: Pretvorba oktalnih znamenki u binarne

Znamenka oktalnog sustava	**Ekvivalent u binarnom sustavu**
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

☞ Primjer 2.1.7

Direktnom metodom pretvorite sljedeće brojeve iz heksadekadskog u binarni sustav:
a) FC_{16},
b) $0{,}1AB_{16}$,
c) $6B{,}C4_{16}$.

✍ Rješenje:

Pretvorba heksadekadskih znamenki u binarne provodi se pomoću tablice 2.2 prema istom načelu po kojem se oktalne znamenke pretvaraju u binarne.

Tablica 2.2: Pretvorba heksadekadskih znamenki u binarne

Znamenka heksadekadskog sustava	**Ekvivalent u binarnom sustavu**
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

a) $FC_{16} = \underbrace{1111}_{F}\underbrace{1100}_{C}{}_2 = 11111100_2$

b) $0{,}1\mathrm{AB}_{16} = \underbrace{0000}_{0},\underbrace{0001}_{1}\underbrace{1010}_{\mathrm{A}}\underbrace{1011}_{\mathrm{B}}{}_2 = 0{,}000110101011_2$

c) $6\mathrm{B{,}C4}_{16} = \underbrace{0110}_{6}\underbrace{1011}_{\mathrm{B}},\underbrace{1100}_{\mathrm{C}}\underbrace{0100}_{4}{}_2 = 1101011{,}110001_2$

Nakon pretvorbe heksadekadskih znamenki u binarne pomoću tablice 2.2, poželjno je izbrisati vodeće i prateće nule.

☞ Primjer 2.1.8

Direktnom metodom pretvorite sljedeće brojeve iz binarnog u oktalni sustav:
a) 101011011_2,
b) $11101{,}001101_2$.

✍ **Rješenje:** Kod pretvorbe brojeva iz binarnog sustava u oktalni primjenjuje se obrnuti postupak od pretvorbe iz oktalnog sustava u binarni. Pri tome treba paziti da se ispred broja doda potreban broj nula kako bi broj binarnih znamenki bio višekratnik broja 3. U slučaju decimalnog broja na kraj je potrebno dodati određeni broj nula tako da broj decimalnih znamenki bude višekratnik broja 3. Binarne znamenke grupiraju se u skupinu od 3 binarne znamenke lijevo i desno od decimalnog zareza. Pretvorba binarnih znamenki u oktalne provodi se pomoću tablice 2.1.

a) $101011011_2 = \underbrace{101}_{5}\underbrace{011}_{3}\underbrace{011}_{3}{}_2 = 533_8$

b) $11101{,}00111_2 = \underbrace{011}_{3}\underbrace{101}_{5},\underbrace{001}_{1}\underbrace{110}_{6}{}_2 = 35{,}16_8$

☞ Primjer 2.1.9

Direktnom metodom pretvorite sljedeće brojeve iz binarnog u heksadekadski sustav:
a) 101011011_2,
b) $11101{,}001101_2$.

✍ **Rješenje:** Kod pretvorbe brojeva iz binarnog sustava u heksadekadski primjenjuje se obrnuti postupak od pretvorbe iz heksadekadskog sustava u binarni. Pri tome treba paziti da se ispred broja doda potreban broj nula kako bi broj binarnih znamenki bio višekratnik broja 4. U slučaju decimalnog broja na kraj je potrebno dodati određeni broj nula tako da broj decimalnih znamenki bude višekratnik broja 4. Binarne znamenke grupiraju se u skupinu od 4 binarne znamenke lijevo i desno od decimalnog zareza. Pretvorba binarnih znamenki u heksadekadske provodi se pomoću tablice 2.2.

a) $101011011_2 = \underbrace{0001}_{1}\underbrace{0101}_{5}\underbrace{1011}_{\mathrm{B}}{}_2 = 15\mathrm{B}_{16}$

b) $11101{,}00111 = \underbrace{0001}_{1}\underbrace{1101}_{\mathrm{D}},\underbrace{0011}_{3}\underbrace{1000}_{8}{}_2 = 1\mathrm{D}{,}38_{16}$

2.1.1 Zadaci za vježbu

✎ Zadatak 2.1.1

U obliku polinoma prikažite sljedeće cijele brojeve:

a) 111011_2,

b) 201220_3,

c) 7426_8,

d) 874_{10},

e) $ABC2_{16}$.

Navedene brojeve prikažite u dekadskom sustavu.

✎ Zadatak 2.1.2

U obliku polinoma prikažite sljedeće realne brojeve:

a) $111001{,}1101_2$,

b) $112001{,}11_3$,

c) $615{,}4_8$,

d) $123{,}987_{10}$,

e) $F{,}5A_{16}$.

Navedene brojeve prikažite u dekadskom sustavu.

✎ Zadatak 2.1.3

Pretvorite cijeli broj 637_{10} u binarni, ternarni, oktalni i heksadekadski sustav.

✎ Zadatak 2.1.4

Pretvorite cijeli broj $3BF_{16}$ u ternarni sustav.

✎ Zadatak 2.1.5

Pretvorite sljedeće realne brojeve iz dekadskog sustava u zadani sustav:
a) $0{,}125_{10}$ u binarni sustav,
b) $23{,}23_{10}$ u binarni sustav,
c) $0{,}5_{10}$ u oktalni sustav,
d) $387{,}12_{10}$ u oktalni sustav,
e) $0{,}125_{10}$ u heksadekadski sustav,
f) $127{,}33_{10}$ u heksadekadski sustav.

✎ Zadatak 2.1.6

Direktnom metodom pretvorite sljedeće brojeve iz oktalnog u binarni sustav:
a) 672_8,
b) $0{,}173_8$,
c) $32{,}41_8$.

✎ Zadatak 2.1.7

Direktnom metodom pretvorite sljedeće brojeve iz heksadekadskog u binarni sustav:
a) $\mathrm{B5}_{16}$,
b) $0{,}\mathrm{FE3}_{16}$,
c) $\mathrm{D1{,}A7}_{16}$.

✎ Zadatak 2.1.8

Direktnom metodom pretvorite sljedeće brojeve iz binarnog u oktalni sustav:
a) 1100101101_2,
b) $101110010{,}01011011_2$.

✎ Zadatak 2.1.9

Direktnom metodom pretvorite sljedeće brojeve iz binarnog u heksadekadski sustav:
a) 1100101101_2,
b) $101110010{,}01011011_2$.

2.2 Zbrajanje i oduzimanje prirodnih binarnih brojeva

Pravila za zbrajanje dviju binarnih znamenki redom su:

$$\begin{aligned}
&0+0=0\\
&0+1=1\\
&1+0=1\\
&1+1=0 \quad \text{i prijenos 1 na više brojno mjesto (eng. } \textit{carry}\text{)}
\end{aligned} \tag{2.3}$$

Pravila za oduzimanje dviju binarnih znamenki redom su:

$$\begin{aligned}
&0-0=0\\
&0-1=1 \quad \text{i posudba 1 s višeg brojnog mjesta (eng. } \textit{borrow}\text{)}\\
&1-0=1\\
&1-1=0
\end{aligned} \tag{2.4}$$

☞ Primjer 2.2.1

Zbrojite sljedeće binarne brojeve:
a) 1101+1011,
b) 1111+1111,
c) 10001+11000,
d) 10001+111.

✍ **Rješenje:** Zbrajanje je potrebno izvršiti prema pravilu (2.3).

a)

$$\begin{array}{cccccc} & & 1 & 1 & 0 & 1 \\ + & & 1 & 0 & 1 & 1 \\ \hline & 1 & 1 & 0 & 0 & 0 \end{array}$$

b)

$$\begin{array}{cccccc} & & 1 & 1 & 1 & 1 \\ + & & 1 & 1 & 1 & 1 \\ \hline & 1 & 1 & 1 & 1 & 0 \end{array}$$

c)

$$\begin{array}{ccccccc} & & 1 & 0 & 0 & 0 & 1 \\ + & & 1 & 1 & 0 & 0 & 0 \\ \hline & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

d)

$$\begin{array}{cccccc} & 1 & 0 & 0 & 0 & 1 \\ + & & & 1 & 1 & 1 \\ \hline & 1 & 1 & 0 & 0 & 0 \end{array}$$

☞ Primjer 2.2.2

Oduzmite sljedeće binarne brojeve:
a) 1111-1001,
b) 101011-100101,
c) 100011-1001,
d) 101110-100101.

✍ **Rješenje:** Oduzimanje je potrebno izvršiti prema pravilu (2.4).

a)

$$\begin{array}{ccccc} & 1 & 1 & 1 & 1 \\ - & 1 & 0 & 0 & 1 \\ \hline & 0 & 1 & 1 & 0 \end{array}$$

b)

$$\begin{array}{ccccccc} & 1 & 0 & 1 & 0 & 1 & 1 \\ - & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

c)

$$
\begin{array}{r@{\quad}cccccc}
 & 1 & 0 & 0 & 0 & 1 & 1 \\
- & & & 1 & 0 & 0 & 1 \\
\hline
 & 0 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

d)

$$
\begin{array}{r@{\quad}cccccc}
 & 1 & 0 & 1 & 1 & 1 & 0 \\
- & 1 & 0 & 0 & 1 & 0 & 1 \\
\hline
 & 0 & 0 & 1 & 0 & 0 & 1
\end{array}
$$

2.2.1 Zadaci za vježbu

✎ Zadatak 2.2.1

Zbrojite sljedeće binarne brojeve:
a) 1111+1011,
b) 1001+1011,
c) 11001+11011,
d) 10101+1011.

✎ Zadatak 2.2.2

Oduzmite sljedeće binarne brojeve:
a) 1101-1101,
b) 111111-110111,
c) 101011-10101,
d) 101010-101.

2.3 Prikaz brojeva u modulu

Računala i općenito digitalni sklopovi ne mogu prikazivati beskonačno velike brojeve. Razlog tome je što za prikaz svake binarne znamenke binarnog broja mora postojati sklop koji pohranjuje tu znamenku. Prema tome, binarni se brojevi prikazuju s konačnim brojem bitova.

Mikrokontroleri su upravljačka mikroračunala koja posjeduju velik broj registara za pohranu binarnih znamenaka. Pretpostavimo da su ti registri širine 8 bitova. Najmanji binarni broj koji se može pohraniti u taj registar je $00000000_2 = 0_{10}$, a najveći je $11111111_2 = 255_{10}$. Pokušajmo sada prema pravilima (2.3) pribrojiti broj 1 najvećem broju u 8-bitnom registru:

$$
\begin{array}{c|cccccccc}
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
+ & & & & & & & & 1 \\
\hline
\boxed{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Kao rezultat dobili smo 9-bitni broj $100000000_2 = 256_{10}$. Rezultat zbrajanja 100000000_2 sprema se u 8-bitni registar pri čemu će se u registar spremiti samo donjih 8 bitova rezultata zbrajanja. Sadržaj 8-bitnog registra bit će $00000000_2 = 0_{10}$ umjesto $100000000_2 = 256_{10}$. Kažemo da je nastao preljev registra (eng. *overflow*).

Pokušajmo sada prema pravilima (2.4) oduzeti broj 1 najmanjem broju u 8-bitnom registru:

$$\begin{array}{cccccccccc} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ - & & & & & & & & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Kao rezultat dobili smo 8-bitni broj $11111111_2 = 255_{10}$. Rezultat zbrajanja 11111111_2 sprema se u 8-bitni registar. Sadržaj 8-bitnog registra bit će $11111111_2 = 255_{10}$ umjesto -1_{10}. Kažemo da je nastao podljev registra (eng. *underflow*).

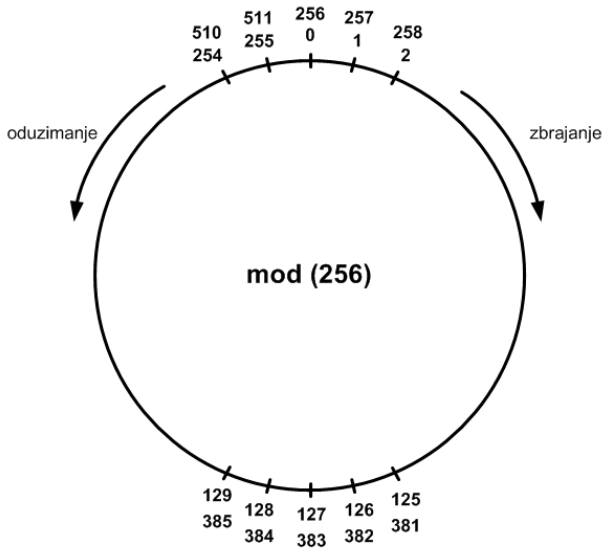
Općenito se u registar dimenzije n bitova mogu pohraniti cijeli brojevi iz intervala $[0, 2^n - 1]$. Broj y koji je veći i jednak broju 2^n u registar će se zapisati kao modul broja 2^n:

$$x = y \mod 2^n \tag{2.5}$$

gdje je:

- x - broj zapisan u registru,
- y - stvarni broj.

Modul broja y cjelobrojni je ostatak pri dijeljenju broja y s brojem 2^n. Ako u 8-bitni registar želimo pohraniti broj 256, tada će u registru biti pohranjen broj $x = 256 \mod 2^8 = 256 \mod 256 = 0$. Na slici 2.1 prikazani su brojevi u modulu 256.



Slika 2.1: Prikaz brojeva u modulu 256

Kod zbrajanja dva broja y i z te spremanja rezultata zbrajanja u registar dimenzije n bitova u registru će biti pohranjen broj:

$$x = (y + z) \mod 2^n. \tag{2.6}$$

Općenito za modul bilo kojeg broja vrijedi:

$$x = y \mod m \tag{2.7}$$

gdje je:

- y - broj,
- m - modul,
- x - modul broja y, odnosno cjelobrojni ostatak pri dijeljenju broja y s brojem m.

☞ Primjer 2.3.1

Izračunajte:
a) $123 \mod 35$,
b) $15 \mod 5$,
c) $17 \mod 16$,
d) $123 \mod 150$.

✍ Rješenje:

Koristeći relaciju (2.7) slijedi:

a) $123 \mod 35 = 18$ jer je $123/35 = 3$ i ostatak 18.

b) $15 \mod 5 = 0$ jer je $15/5 = 3$ i ostatak 0.

c) $17 \mod 16 = 1$ jer je $17/16 = 1$ i ostatak 1.

d) $123 \mod 150 = 123$ jer je $123/150 = 0$ i ostatak 123.

☞ Primjer 2.3.2

Mikrokontroler rezultate zbrajanja sprema u 8-bitni registar. Izračunajte sadržaj registra nakon zbrajanja:
a) 123 + 245,
b) 35 + 220,
c) 100 + 160.

✍ Rješenje:

U registar dimenzije 8 bitova mogu se pohraniti brojevi iz intervala $[0,\ 2^8 - 1] = [0,\ 255]$. Prema relaciji (2.6) vrijedi:

a) Sadržaj registra bit će $x = (123+245) \mod 256 = 368 \mod 256 = 112$. Dakle, u registru će biti zapisan neispravan rezultat jer je ispravan rezultat 368. Zbroj 123 + 245 = 368 izvan je intervala [0, 255].

b) a) Sadržaj registra bit će $x = (35 + 220) \mod 256 = 255 \mod 256 = 255$. Dakle, u registru će biti zapisan ispravan rezultat. Zbroj 35 + 220 = 255 unutar je intervala [0, 255].

c) Sadržaj registra bit će $x = (100 + 160) \mod 256 = 260 \mod 256 = 4$. Dakle, u registru će biti zapisan neispravan rezultat jer je ispravan rezultat 260. Zbroj 100 + 160 = 260 izvan je intervala [0, 255].

2.3.1 Zadaci za vježbu

✎ Zadatak 2.3.1

Izračunajte:

a) 35 mod 45,
b) 15 mod 7,
c) 23 mod 20,
d) 123 mod 50.

✎ **Zadatak 2.3.2**

Mikrokontroler rezultate zbrajanja sprema u 8-bitni registar. Izračunajte sadržaj registra nakon zbrajanja:
a) 110 + 110,
b) 225 + 220,
c) 243 + 13.

2.4 Komplementi brojeva

Definirat ćemo dva komplementa:

$$\overline{N_B} = B^n - N \tag{2.8}$$

$$\overline{N_{B-1}} = B^n - N - 1 \tag{2.9}$$

gdje je:

- $\overline{N_B}$ - B-komplement broja N,
- $\overline{N_{B-1}}$ - $(B-1)$-komplement broja N,
- B - baza brojevnog sustava,
- n - dimenzija registra za prikaz broja.

U digitalnim sklopovima važna je baza 2 pa ćemo stoga najveću pozornost posvetiti 2-komplementu broja i 1-komplementu broja.

1-komplement binarnog broja dobije se tako da se komplementiraju svi bitovi (0 prelazi u 1, a 1 prelazi u 0 ($0 \rightarrow 1$, i $1 \rightarrow 0$)). Na primjer, 1-komplement broja 1001_2 je 0110_2. Skraćeno pišemo $\overline{1001}_2 = 0110$. 2-komplement binarnog broja dobije se tako da se 1-komplementu pribroji broj 1. Na primjer, 2-komplement broja 1001_2 je $\overline{1001}_2 + 1 = 0110_2 + 0001_2 = 0111_2$.

Kod određivanja komplementa treba voditi računa o širini registra za prikaz podataka. Na primjer, 1-komplement broja 101 u 8-bitnom registru je $\overline{101} = \overline{00000101} = 11111010$. Dakle, prije komplementiranja potrebno je broj proširiti s vodećim nulama do širine 8 bitova.

☞ **Primjer 2.4.1**

Izračunajte B-komplement za sljedeće brojeve:
a) 47_{10} ako se broj prikazuje s dvije znamenke,
b) 47_{10} ako se broj prikazuje s tri znamenke,
c) 999_{10} ako se broj prikazuje s tri znamenke.

✍ **Rješenje:** Svi brojevi zapisani su u dekadskom sustavu gdje je baza $B = 10$ pa, prema tome, tražimo 10-komplement zadanih brojeva.

a) Broj se prikazuje s dvije znamenke ($n = 2$) pa prema relaciji (2.8) vrijedi:

$$\overline{47}_{10} = 10^2 - 47 = 100 - 47 = 53.$$

b) Broj se prikazuje s dvije znamenke ($n = 3$) pa prema relaciji (2.8) vrijedi:

$$\overline{47}_{10} = 10^3 - 47 = 1000 - 47 = 953.$$

c) Broj se prikazuje s dvije znamenke ($n = 3$) pa prema relaciji (2.8) vrijedi:

$$\overline{999}_{10} = 10^3 - 999 = 1000 - 999 = 1.$$

☞ Primjer 2.4.2

Izračunajte 1-komplement i 2-komplement za sljedeće brojeve:
a) 1011_2 ako se broj prikazuje u registru širine 4 bita,
a) 1011_2 ako se broj prikazuje u registru širine 8 bitova,
c) 10011001_2 ako se broj prikazuje u registru širine 8 bitova.

✍ **Rješenje:** Svi brojevi zapisani su u binarnom sustavu gdje je baza $B = 2$. Za binarne brojeve 1-komplement i 2-komplement možemo izračunati prema relacijama (2.8) i (2.9). Jednostavnije je 1-komplement binarnog broja izračunati tako da se komplementiraju svi bitovi, a 2-komplement binarnog broja izračunati tako da se 1-komplementu pribroji broj 1.

a) 1-komplement broja 1011_2 u registru širine 4 bita je:

$$\overline{1011}_2 = 0100_2.$$

2-komplement broja 1011_2 u registru širine 4 bita je:

$$\overline{1011}_2 + 1 = 0100_2 + 1 = 0101_2.$$

b) 1-komplement broja 1011_2 u registru širine 8 bitova je:

$$\overline{1011}_2 = \overline{00001011}_2 = 11110100_2.$$

2-komplement broja 1011_2 u registru širine 8 bitova je:

$$\overline{1011}_2 + 1 = \overline{00001011}_2 + 1 = 11110100_2 + 1 = 11110101_2.$$

b) 1-komplement broja 10011001_2 u registru širine 8 bitova je:

$$\overline{10011001}_2 = 01100110_2.$$

2-komplement broja 10011001_2 u registru širine 8 bitova je:

$$\overline{10011001}_2 + 1 = 01100110_2 + 1 = 01100111_2.$$

2.4.1 Zadaci za vježbu

✎ Zadatak 2.4.1

Izračunajte B-komplement za sljedeće brojeve:
a) 63_{10} ako se broj prikazuje s dvije znamenke,
b) 63_{10} ako se broj prikazuje s tri znamenke,
c) 500_{10} ako se broj prikazuje s tri znamenke.

✎ Zadatak 2.4.2

Izračunajte 1-komplement i 2-komplement za sljedeće brojeve:
a) 1011_2 ako se broj prikazuje u registru širine 4 bita,
a) 1011_2 ako se broj prikazuje u registru širine 8 bitova,
c) 10011001_2 ako se broj prikazuje u registru širine 8 bitova.

2.5 Brojevi s predznakom i oduzimanje pomoću komplemenata

U tablici 2.3 dan je prikaz 4-bitnih brojeva s predznakom.

Tablica 2.3: Prikaz 4-bitnih brojeva s predznakom

Dekadski broj	Prikaz s predznakom i veličinom	Prikaz s predznakom i 2-komplementom	Prikaz s predznakom i 1-komplementom
-8	-	1000	-
-7	1111	1001	1000
-6	1110	1010	1001
-5	1101	1011	1010
-4	1100	1100	1011
-3	1011	1101	1100
-2	1010	1110	1101
-1	1001	1111	1110
0	1000 0000	0000	1111 0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111

Binarni brojevi s predznakom u registru mogu biti zapisani na tri načina (tablica 2.3):

1. s predznakom i veličinom - najznačajniji bit u registru predstavlja predznak, a ostali bitovi veličinu broja (postoje dva prikaza nule). Na primjer, broj 6_{10} iznosi 110_2 u binarnom zapisu. U zapisu s predznakom i veličinom u 4-bitnom registru broj 6_{10} je 0110_2, a broj -6_{10} je 1110_2.

2. s predznakom i 2-komplementom - pozitivni brojevi prikazuju se s predznakom i veličinom, a negativni s predznakom i 2- komplementom (jedan prikaz nule). Na primjer, broj 4_{10} iznosi 100_2 u binarnom zapisu. U zapisu s predznakom i 2-komplementom u 4-bitnom registru broj 4_{10} je 0100_2, a broj -4_{10} je 2-komplement broja 4_{10}, odnosno $\overline{0100}_2 + 1 = 1011 + 1 = 1100$.

3. s predznakom i 1-komplementom - pozitivni brojevi prikazuju se s predznakom i veličinom, a negativni s predznakom i 1- komplementom (dva prikaza nule). Na primjer, broj 3_{10} iznosi 11_2 u binarnom zapisu. U zapisu s predznakom i 1-komplementom u 4-bitnom registru broj 3_{10} je 0011_2, a broj -3_{10} je 1-komplement broja 3_{10}, odnosno $\overline{0011}_2 = 1100$.

Operacije zbrajanja i oduzimanja u sustavu 2-komplementa su jednostavne i danas se najčešće koriste u računalima (mikroračunala, mikroprocesori, mikrokontroleri, procesori). Komplement broja omogućuje da se oduzimanje dvaju brojeva svodi na zbrajanje. Pravila za oduzimanje brojeva zapisanih u 2-komplementu prikazana su pomoću sljedeće relacije:

$$D = M - S = \begin{cases} \underline{M + \overline{S_B}} & \text{ako je} \quad M > S \\ -(M + \overline{S_B})_B & \text{ako je} \quad M < S \end{cases} \tag{2.10}$$

gdje je:

- D - razlika (eng. *difference*),
- M - umanjenik (eng. *minuend*),
- S - umanjitelj (eng. *subtrahend*).

Najveći broj koji se u 2-komplementu može upisati u registar je $B^{n-1} - 1$ gdje je n širina registra. Najmanji broj koji se u 2-komplementu može upisati u registar je $-B^{n-1}$ gdje je n širina registra. Ako je binarni broj zapisan s predznakom i 2-komplementom u registru širine n bitova, tada se on u dekadski sustav može pretvoriti pomoću sljedeće relacije:

$$Z = a_{n-1}(-2^{n-1}) + a_{n-2}2^{n-2} + a_{n-3}B^{n-3} + ... + a_2 2^2 + a_1 2^1 + a_0 2^0 \tag{2.11}$$

Najznačajnija znamenka binarnog broja zapisanog s predznakom i 2-komplementom ima težinu -2^{n-1}. Na primjer, broj 10000110_2 u zapisu s predznakom i 2-komplementom iznosi:

$$\overset{-2^7}{1}\ \overset{2^6}{0}\ \overset{2^5}{0}\ \overset{2^4}{0}\ \overset{2^3}{0}\ \overset{2^2}{1}\ \overset{2^1}{1}\ \overset{2^0}{0} = -128 + 0 + 0 + 0 + 0 + 4 + 2 + 0 = -122.$$

☞ Primjer 2.5.1

Izračunajte sumu binarnih brojeva u zapisu 2-komplementa. Brojevi se pohranjuju u 8-bitnim registrima. Je li suma točna?
a) $00100001_2 + 01000010_2$,
b) $10100001_2 + 01000010_2$,
c) $01100001_2 + 01000010_2$.

✍ **Rješenje:**

a)

$$\begin{array}{rcccccccccl} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1_2 & & (33_{10}) \\ + & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0_2 & & (66_{10}) \\ \hline & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1_2 & & (99_{10}) \end{array}$$

Rezultat je točan.

b)

$$\begin{array}{rcccccccccl} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1_2 & & (-95_{10}) \\ + & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0_2 & & (66_{10}) \\ \hline & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1_2 & & (-29_{10}) \end{array}$$

Rezultat je točan.

c)

$$\begin{array}{rcccccccccl} & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1_2 & & (97_{10}) \\ + & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0_2 & & (66_{10}) \\ \hline & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1_2 & & (-93_{10}) \end{array}$$

Rezultat je netočan jer zbroj dvaju pozitivnih brojeva ne može biti negativan. Razlog tome je preljev registra jer je $97 + 66 = 163$ što je veće od 127.

☞ Primjer 2.5.2

Izračunajte razliku binarnih brojeva u zapisu 2-komplementa. Brojevi se pohranjuju u 8-bitnim registrima. Je li suma točna?
a) 00100001_2-01000010_2,
b) 10100001_2-01000010_2,
c) 01100001_2 -01000010_2.

✍ **Rješenje:**

Pravila za oduzimanje brojeva u zapisu 2-komplementa prikazana su pomoću relacije (2.10).

a) U ovom primjeru vrijedi da je $M = 00100001_2 = 33_{10}$, a $S = 01000010_2 = 66_{10}$. Prema relaciji (2.10) vrijedi da je $M < S$ pa ćemo razliku brojeva izračunati prema relaciji $D = -(\overline{M + \overline{S}_B})_B$. Dakle, najprije je potrebno pronaći 2-komplement umanjitelja S (1-komplement + 1):

$$\overline{S}_2 = \overline{01000010}_2 + 1 = 10111101_2 + 1 = 10111110_2.$$

Zbroj $M + \overline{S}_2$ je:

$$\begin{array}{rcccccccccl} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1_2 & & (33_{10}) \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0_2 & & (-66_{10}) \\ \hline & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1_2 & & (-33_{10}) \end{array}$$

Provjerimo sada (-) 2-komplement zbroja $M + \overline{S}_2$:

$$-(\overline{M + \overline{S}_2})_2 = -(\overline{11011111}_2 + 1) = -(00100000_2 + 1) = -00100001_2 = -33_{10}.$$

Rezultat je točan. Jednakost $11011111_2 = -33_{10}$ možete provjeriti relacijom (2.11).

b) U ovom primjeru vrijedi da je $M = 10100001_2 = -95_{10}$, a $S = 01000010_2 = 66_{10}$. Prema relaciji (2.10) vrijedi da je $M < S$ pa ćemo razliku brojeva izračunati prema relaciji $D = -\overline{(M + \overline{S_B})_B}$. Dakle, najprije je potrebno pronaći 2-komplement umanjitelja S (1-komplement + 1):

$$\overline{S}_2 = \overline{01000010}_2 + 1 = 10111101_2 + 1 = 10111110_2.$$

Zbroj $M + \overline{S}_2$ je:

$$\begin{array}{cccccccccc} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1_2 & (-95_{10}) \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0_2 & (-66_{10}) \\ \hline & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1_2 & (\ 95_{10}) \end{array}$$

Rezultat je netočan jer smo zbrajanjem dvaju negativnih brojeva dobili pozitivan broj 01011111_2 (najznačajniji bit je 0).

c) U ovom primjeru vrijedi da je $M = 01100001_2 = 97_{10}$, a $S = 01000010_2 = 66_{10}$. Prema relaciji (2.10) vrijedi da je $M > S$ pa ćemo razliku brojeva izračunati prema relaciji $D = M + \overline{S_B}$. Dakle, potrebno je pronaći 2-komplement umanjitelja S (1-komplement + 1):

$$\overline{S}_2 = \overline{01000010}_2 + 1 = 10111101_2 + 1 = 10111110_2.$$

Zbroj $M + \overline{S}_2$ je:

$$\begin{array}{cccccccccc} & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1_2 & (97_{10}) \\ + & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0_2 & (-66_{10}) \\ \hline & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1_2 & (\ 31_{10}) \end{array}$$

Rezultat je točan.

2.5.1 Zadaci za vježbu

✎ Zadatak 2.5.1

Izračunajte sumu binarnih brojeva u zapisu 2-komplementa. Brojevi se pohranjuju u 8-bitnim registrima. Je li suma točna?
a) 00101011_2+01011010_2,
b) 10101101_2+01011010_2,
c) 01101101_2 +01011010_2.

✎ Zadatak 2.5.2

Izračunajte razliku binarnih brojeva u zapisu 2-komplementa. Brojevi se pohranjuju u 8-bitnim registrima. Je li suma točna?
a) 00101011_2-01011010_2,
b) 10101101_2-01011010_2,
c) 01101101_2 -01011010_2.

2.6 Binarni kodovi, zaštita podataka

Najjednostavniji način zaštite je višestruko ponavljanje svakog bita podatka. Na primjer, pretpostavimo da komunikacijskim kanalom želimo poslati podatak 101_2. Višestrukim ponavljanjem svakog bita 5 puta taj bi podatak komunikacijskim kanalom bio poslan kao 11111 00000 11111. Nedostatak ove vrste zaštite je velika količina bitova koju je potrebno poslati za određeni podatak.

Jedan od čestih načina otkrivanja grešaka je dodavanje paritetnog bita kodnoj riječi. Kodnoj riječi dodaje se paritetni bit tako da ukupan broj jedinica (1) u riječi bude paran (parni paritet) ili neparan (neparni paritet). Na primjer, podatak 100110 ima neparni broj jedinica. Za parni paritet paritetni bit je jedinica pa je kodirani podatak (1)100110. Za neparni paritet paritetni bit je nula pa je kodirani podatak (0)100110. Na ovaj se način mogu otkriti jednostruke greške. Općenito, paritetom se mogu otkriti greške pri promjeni neparnog broja bitova. Otkrivene greške ne mogu se ispraviti.

Binarni kodovi koje ćemo koristiti su redom:

- kod 8421 (BCD kod),
- kod 2421 (Aikenov kod),
- kod *excess* - 3 (Stibitzov kod),
- bikvinarni kod (kod 504321),
- Grayev kod,
- ASCII kod.

Tablice navedenih kodova bit će prikazane u primjerima.

☞ Primjer 2.6.1

Podatak 011 prenosi se tako da se svaki bit podatka prenosi 7 puta. Kodirani podatak u tom je slučaju 0000000 1111111 1111111. Analizirajte ovaj način zaštite s obzirom na mogućnost otkrivanja i ispravljanja pogrešaka.

✍ Rješenje:

Kod prijenosa podataka može se dogoditi greška. Ako se promijeni jedan bit, govori se o jednostrukoj grešci, ako se promijene dva bita govori se o dvostrukoj grešci itd. Vjerojatnost pojave greške najveća je za jednostruke greške i smanjuje se povećanjem broja pogrešaka. Ako za podatak 011 promatramo kod prvog bita (0000000), tada se mogu pojaviti sljedeće greške:

- jednostruka (samo je jedan bit u kodu pogrešan):
  - 1000000, 0100000, 0010000, 0001000, 0000100, 0000010, 000001,
- dvostruka (dva bita u kodu su pogrešna):
  - 1100000, 0110000, 0011000, 0101000, 1000100, 1000010, 1000001, itd.,
- trostruka (tri bita u kodu su pogrešna):
  - 1110000, 0111000, 0011100, 0101010, 1001100, 1010010, 1100001, itd.,

- četverostruka (četiri bita u kodu su pogrešna):
  - 1111000, 0111100, 0011101, 0111010, 1011100, 1010110, 1110001, itd.,
- peterostruka (pet bitova u kodu su pogrešna):
  - 1111100, 0111110, 0111101, 0111110, 1011110, itd.,
- šesterostruka (šest bitova u kodu su pogrešna):
  - 1111110, 0111111, 1011111, 1101111, 1110111, 1111011, 1111101.

Definirajmo distancu d kao udaljenost između dva susjedna ispravna koda. U slučaju kodova 0000000 i 1111111 vrijedi da je distanca $d = 7$ jer je potrebno promijeniti 7 bitova da bi kod 0000000 prešao u kod 1111111 i obratno.

Distanca d može se odrediti i za pogrešne podatke:

- jednostruka pogreška,
  - za kod 0000000 distanca je $d = 1$ jer se jedan bit mora promijeniti da bi nastala jednostruka pogreška koda 0000000,
  - za kod 1111111 distanca je $d = 6$ jer se šest bitova moraju promijeniti da bi nastala jednostruka pogreška koda 0000000,
  - budući da je veća vjerojatnost jednostruke pogreške koda 0000000 od šesterostruke pogreške koda 1111111, možemo zaključiti da je ispravan kod 0000000.
- dvostruka pogreška,
  - za kod 0000000 distanca je $d = 2$ jer se dva bita moraju promijeniti da bi nastala dvostruka pogreška koda 0000000,
  - za kod 1111111 distanca je $d = 5$ jer se pet bitova moraju promijeniti da bi nastala dvostruka pogreška koda 0000000,
  - budući da je veća vjerojatnost dvostruke pogreške koda 0000000 od peterostruke pogreške koda 1111111, možemo zaključiti da je ispravan kod 0000000.
- trostruka pogreška,
  - za kod 0000000 distanca je $d = 3$ jer se tri bita moraju promijeniti da bi nastala trostruka pogreška koda 0000000,
  - za kod 1111111 distanca je $d = 4$ jer se četiri bita moraju promijeniti da bi nastala trostruka pogreška koda 0000000,
  - budući da je veća vjerojatnost trostruke pogreške koda 0000000 od četverostruke pogreške koda 1111111, možemo zaključiti da je ispravan kod 0000000.
- četverostruka pogreška,
  - za kod 0000000 distanca je $d = 4$ jer se četiri bita moraju promijeniti da bi nastala četverostruka pogreška koda 0000000,
  - za kod 1111111 distanca je $d = 3$ jer se tri bita moraju promijeniti da bi nastala četverostruka pogreška koda 0000000,
  - budući da je manja vjerojatnost četverostruke pogreške koda 0000000 od trostruke pogreške koda 1111111, možemo zaključiti da je ispravan kod 1111111.

Prema navedenoj analizi možemo zaključiti da je moguće ispraviti g pogrešaka:

$$g = (d-1)/2. \tag{2.12}$$

Ako se želi ispraviti g pogreški, za distancu mora vrijediti $d \geq 2g - 1$.

☞ Primjer 2.6.2

Napišite sljedeće brojeve u kodu 8421:
a) 2_{10},
b) 98_{10},
c) 356_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✍ **Rješenje:** U kodu 8421 svaka se dekadska znamenka pretvara u 4-bitni binarni broj prema tablici 2.4. Ovaj kod naziva se još i BCD kod.

Tablica 2.4: Kod 8421 (BCD kod (eng. *Binary-coded decimal*))

Dekadski broj	**Kod 8421**
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
	1010
	1011
	1100
	1101
	1110
	1111

Težine znamenaka u ovom kodu su 8, 4, 2 i 1. U kodu 8421 koriste se prvih deset kombinacija 4-bitnog binarnog broja, a ostalih šest je neiskorišteno.

a) Prema tablici 2.4 vrijedi da je $2_{10} = 0010_{\mathrm{BCD}}$. Podatak 0010 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)0010 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)0010 - broj jedinica je neparan u zaštićenoj riječi.

b) Prema tablici 2.4 vrijedi da je $98_{10} = 1001\ 1000_{\mathrm{BCD}}$. Podatak 1001 1000 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)1001 1000 - broj jedinica je paran u zaštićenoj riječi,

- neparnim paritetom: (0)1001 1000 - broj jedinica je neparan u zaštićenoj riječi.

c) Prema tablici 2.4 vrijedi da je $356_{10} = 0011\ 0101\ 0110_{BCD}$. Podatak 0011 0101 0110 ima paran broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (0)0011 0101 0110 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (1)0011 0101 0110 - broj jedinica je neparan u zaštićenoj riječi.

☞ Primjer 2.6.3

Napišite sljedeće brojeve u kodu 2421:
a) 2_{10},
b) 98_{10},
c) 356_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✍ **Rješenje:** U kodu 2421 svaka se dekadska znamenka pretvara u 4-bitni binarni broj prema tablici 2.5. Ovaj kod naziva se još i Aikenov kod.

Tablica 2.5: Kod 2421 (Aikenov kod)

Dekadski broj	**Kod 2421**
0	0000
1	0001
2	0010
3	0011
4	0100
	0101
	0110
	0111
	1000
	1001
	1010
5	1011
6	1100
7	1101
8	1110
9	1111

Težine znamenaka u ovom kodu su 2, 4, 2 i 1. U kodu 2421 koriste se prvih pet i zadnjih pet kombinacija 4-bitnog binarnog broja, a ostalih šest je neiskorišteno.

a) Prema tablici 2.5 vrijedi da je $2_{10} = 0010_{2421}$. Podatak 0010 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)0010 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)0010 - broj jedinica je neparan u zaštićenoj riječi.

b) Prema tablici 2.5 vrijedi da je $98_{10} = 1111\ 1110_{2421}$. Podatak 1111 1110 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)1111 1110 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)1111 1110 - broj jedinica je neparan u zaštićenoj riječi.

c) Prema tablici 2.5 vrijedi da je $356_{10} = 0011\ 1011\ 1100_{2421}$. Podatak 0011 1011 1100 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)0011 1011 1100 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)0011 1011 1100 - broj jedinica je neparan u zaštićenoj riječi.

☞ Primjer 2.6.4

Napišite sljedeće brojeve u kodu *excess* - 3:
a) 2_{10},
b) 98_{10},
c) 356_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✍ **Rješenje:** U kodu *excess* - 3 svaka se dekadska znamenka pretvara u 4-bitni binarni broj prema tablici 2.6. Ovaj kod naziva se još i Stibitzov kod.

Tablica 2.6: Kod *excess* - 3 (Stibitzov kod)

Dekadski broj	**Binarni broj**
	0000
	0001
	0010
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100
	1101
	1110
	1111

Kod *excess* - 3 je kod s prekoračenjem od 3. U kodu *excess* - 3 ne koriste se prve tri i zadnje tri kombinacije 4-bitnog binarnog broja, a ostalih deset se koriste. Karakteristika koda *excess* - 3 je da ne sadrži kombinaciju 0000, što olakšava detekciju prekida kod prijenosa podataka (svaka 4-bitna kombinacija sadrži barem jednu jedinicu).

a) Prema tablici 2.6 vrijedi da je $2_{10} = 0101_{\text{excess-3}}$. Podatak 0101 ima paran broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (0)0101 - broj jedinica je paran u zaštićenoj riječi,

- neparnim paritetom: (1)0101 - broj jedinica je neparan u zaštićenoj riječi.

b) Prema tablici 2.6 vrijedi da je $98_{10} = 1100\ 1011_{\text{excess-3}}$. Podatak 1100 1011 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)1100 1011 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)1100 1011 - broj jedinica je neparan u zaštićenoj riječi.

c) Prema tablici 2.6 vrijedi da je $356_{10} = 0110\ 1000\ 1001_{\text{excess-3}}$. Podatak 0110 1000 1001 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)0110 1000 1001 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)0110 1000 1001 - broj jedinica je neparan u zaštićenoj riječi.

☞ Primjer 2.6.5

Napišite sljedeće brojeve u bikvinarnom kodu (kod 5043210):
a) 2_{10},
b) 98_{10},
c) 356_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✍ **Rješenje:** U kodu 5043210 svaka se dekadska znamenka pretvara u 4-bitni binarni broj prema tablici 2.7. Ovaj kod naziva se još i bikvinarni kod.

Tablica 2.7: Kod 5043210 (bikvinarni kod)

Dekadski broj	**Kod 5043210**
0	0100001
1	0100010
2	0100100
3	0101000
4	0110000
5	1000001
6	1000010
7	1000100
8	1001000
9	1010000

Kod 5043210 je 7-bitni kod s težinama 5, 0, 4, 3, 2, 1 i 0. Kodne riječi uvijek imaju dvije jedinice i pet nula pa se svaka greška koja promijeni broj jedinica lako detektira.

a) Prema tablici 2.7 vrijedi da je $2_{10} = 0100100_{5043210}$. Podatak 0100100 ima paran broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (0)0100100 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (1)0100100 - broj jedinica je neparan u zaštićenoj riječi.

b) Prema tablici 2.7 vrijedi da je $98_{10} = 1010000\ 1001000_{5043210}$. Podatak 1010000 1001000 ima paran broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (0)1010000 1001000 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (1)1010000 1001000 - broj jedinica je neparan u zaštićenoj riječi.

c) Prema tablici 2.7 vrijedi da je $356_{10} = 0101000\ 1000001\ 1000010_{5043210}$. Podatak 0101000 1000001 1000010 ima paran broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (0)0101000 1000001 1000010 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (1)0101000 1000001 1000010 - broj jedinica je neparan u zaštićenoj riječi.

Primijetite da kod 5043210 u kodnim riječima uvijek ima paran broj jedinica.

☞ Primjer 2.6.6

Napišite sljedeće brojeve u Grayevom kodu:
a) 2_{10},
b) 98_{10},
c) 356_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✍ **Rješenje:** U Grayevom kodu svaka se dekadska znamenka pretvara u 4-bitni binarni broj prema tablici 2.8. Ovaj kod naziva se još i reflektirani binarni kod.

Tablica 2.8: Grayev kod

Dekadski broj	**Grayev kod**
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
	1111
	1110
	1010
	1011
	1001
	1000

Grayev kod je 4-bitni netežinski kod s minimalnom promjenom. Susjedne kodne riječi razlikuju se za samo jedan bit. Konstruira se zrcaljenjem u svakom bitovnom mjestu.

a) Prema tablici 2.8 vrijedi da je $2_{10} = 0011_{\text{Gray}}$. Podatak 0011 ima paran broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (0)0011 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (1)0011 - broj jedinica je neparan u zaštićenoj riječi.

b) Prema tablici 2.8 vrijedi da je $98_{10} = 1101\ 1100_{\text{Gray}}$. Podatak 1101 1100 ima neparan broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (1)1101 1100 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)1101 1100 - broj jedinica je neparan u zaštićenoj riječi.

c) Prema tablici 2.8 vrijedi da je $356_{10} = 0010\ 0111\ 0101_{\text{Gray}}$. Podatak 0010 0111 0101 ima paran broj jedinica. Slijedi zaštita podatka:

- parnim paritetom: (0)0010 0111 0101 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (1)0010 0111 0101 - broj jedinica je neparan u zaštićenoj riječi.

☞ Primjer 2.6.7

Napišite sljedeće znakove u ASCII kodu:
a) 2,
b) 98,
c) BJ.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✍ Rješenje:

U ASCII kodu svaki se znak pretvara u 7-bitni binarni broj prema tablici 2.9. Mikrokontroleri informacije s vanjskim svijetom često razmjenjuju pomoću znakova ASCII koda. Podaci se najčešće prenose asinkrono, znak po znak, pa je u ovom slučaju svaki znak potrebno zaštititi paritetnim bitom.

a) Prema tablici 2.9 vrijedi da je $2 = 0110010_{\text{ASCII}}$. Znak kodiran s binarnim kodom 0110010 ima neparan broj jedinica. Slijedi zaštita znaka:

- parnim paritetom: (1)0110010 - broj jedinica je paran u zaštićenoj riječi,
- neparnim paritetom: (0)0110010 - broj jedinica je neparan u zaštićenoj riječi.

b) Prema tablici 2.9 vrijedi da je $98 = 0111001\ 0111000_{\text{ASCII}}$. Znak kodiran s binarnim kodom 0111001 ima paran broj jedinica, a znak kodiran s binarnim kodom 0111000 ima neparan broj jedinica. Slijedi zaštita svakog znaka posebno:

- parnim paritetom: (0)0111001 (1)0111000 - broj jedinica je paran u svakoj zaštićenoj riječi,
- neparnim paritetom: (1)0111001 (0)0111000 - broj jedinica je neparan u svakoj zaštićenoj riječi.

c) Prema tablici 2.9 vrijedi da je $\text{BJ} = 1000010\ 1001010_{\text{ASCII}}$. Znak kodiran s binarnim kodom 1000010 ima paran broj jedinica, a znak kodiran s binarnim kodom 1001010 ima neparan broj jedinica. Slijedi zaštita svakog znaka posebno:

- parnim paritetom: (0)1000010 (1)1001010 - broj jedinica je paran u svakoj zaštićenoj riječi,
- neparnim paritetom: (1)1000010 (0)1001010 - broj jedinica je neparan u svakoj zaštićenoj riječi.

Tablica 2.9: ASCII kod

Dekadski kod	Binarni kod	Simbol	Dekadski kod	Binarni kod	Simbol	Dekadski kod	Binarni kod	Simbol
0	0000000	NUL	43	0101011	+	86	1010110	V
1	0000001	SOH	44	0101100	,	87	1010111	W
2	0000010	STX	45	0101101	-	88	1011000	X
3	0000011	ETX	46	0101110	.	89	1011001	Y
4	0000100	EOT	47	0101111	/	90	1011010	Z
5	0000101	ENQ	48	0110000	0	91	1011011	[
6	0000110	ACK	49	0110001	1	92	1011100	\
7	0000111	BEL	50	0110010	2	93	1011101	]
8	0001000	BS	51	0110011	3	94	1011110	^
9	0001001	HT	52	0110100	4	95	1011111	_
10	0001010	LF	53	0110101	5	96	1100000	‘
11	0001011	VT	54	0110110	6	97	1100001	a
12	0001100	FF	55	0110111	7	98	1100010	b
13	0001101	CR	56	0111000	8	99	1100011	c
14	0001110	SO	57	0111001	9	100	1100100	d
15	0001111	SI	58	0111010	:	101	1100101	e
16	0010000	DLE	59	0111011	;	102	1100110	f
17	0010001	DC1	60	0111100	<	103	1100111	g
18	0010010	DC2	61	0111101	=	104	1101000	h
19	0010011	DC3	62	0111110	>	105	1101001	i
20	0010100	DC4	63	0111111	?	106	1101010	j
21	0010101	NAK	64	1000000	@	107	1101011	k
22	0010110	SYN	65	1000001	A	108	1101100	l
23	0010111	ETB	66	1000010	B	109	1101101	m
24	0011000	CAN	67	1000011	C	110	1101110	n
25	0011001	EM	68	1000100	D	111	1101111	o
26	0011010	SUB	69	1000101	E	112	1110000	p
27	0011011	ESC	70	1000110	F	113	1110001	q
28	0011100	FS	71	1000111	G	114	1110010	r
29	0011101	GS	72	1001000	H	115	1110011	s
30	0011110	RS	73	1001001	I	116	1110100	t
31	0011111	US	74	1001010	J	117	1110101	u
32	0100000		75	1001011	K	118	1110110	v
33	0100001	!	76	1001100	L	119	1110111	w
34	0100010	"	77	1001101	M	120	1111000	x
35	0100011	#	78	1001110	N	121	1111001	y
36	0100100	$	79	1001111	O	122	1111010	z
37	0100101	%	80	1010000	P	123	1111011	{
38	0100110	&	81	1010001	Q	124	1111100	|
39	0100111	’	82	1010010	R	125	1111101	}
40	0101000	(	83	1010011	S	126	1111110	~
41	0101001	)	84	1010100	T	127	1111111	DEL
42	0101010	*	85	1010101	U			

2.6.1 Zadaci za vježbu

✎ Zadatak 2.6.1

Podatak 1101 prenosi se tako da se svaki bit podatka prenosi 5 puta. Kodirani podatak u tom je slučaju 11111 11111 00000 11111. Analizirajte ovaj način zaštite s obzirom na mogućnost otkrivanja i ispravljanja pogrešaka.

✎ Zadatak 2.6.2

Napišite sljedeće brojeve u kodu 8421:
a)8_{10},
b) 35_{10},
c) 147_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✎ Zadatak 2.6.3

Napišite sljedeće brojeve u kodu 2421:
a) 8_{10},
b) 35_{10},
c) 147_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✎ Zadatak 2.6.4

Napišite sljedeće brojeve u kodu *excess* - 3:
a) 8_{10},
b) 35_{10},
c) 147_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✎ Zadatak 2.6.5

Napišite sljedeće brojeve u bikvinarnom kodu (kod 5043210):
a) 8_{10},
b) 35_{10},
c) 147_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✎ Zadatak 2.6.6

Napišite sljedeće brojeve u Grayevom kodu:

a) 8_{10},
b) 35_{10},
c) 147_{10}.
Zaštitite kodirane riječi parnim i neparnim paritetom.

✎ Zadatak 2.6.7

Napišite sljedeće znakove u ASCII kodu:
a) 5,
b) A3,
c) VTSBJ.
Zaštitite kodirane riječi parnim i neparnim paritetom.

2.7 Hammingovi kodovi za otkrivanje i ispravljanje pogrešaka

Na temelju principa višestrukog ispitivanja pariteta, R. W. Hamming je definirao općeniti način konstruiranja kodova za ispravljanje grešaka. Po njemu su nazvani Hammingovi kodovi.

Podatak koji se šalje kodira se na sljedeći način:

- Paritetni bitovi ubacuju se na pozicije koje su potencije broja 2 (2^n, $n = 0, 1, 2, 3, \dots$), odnosno na pozicije 1, 2, 4, 8..., ovisno o broju informacijskih bitova.
- Ako podatak ima 1 informacijski bit (I_1), kod sadrži 1 informacijski bit + 2 paritetna bita što je ukupno 3 bita ($P_1P_2I_1$).
- Ako podatak ima 2 informacijska bita (I_1I_2), kod sadrži 2 informacijska bita + 3 paritetna bita što je ukupno 5 bitova ($P_1P_2I_1P_3I_2$).
- Ako podatak ima 3 informacijska bita ($I_1I_2I_3$), kod sadrži 3 informacijska bita + 3 paritetna bita što je ukupno 6 bitova ($P_1P_2I_1P_3I_2I_3$).
- ... .
- Ako podatak ima 8 informacijskih bitova ($I_1I_2I_3I_4I_5I_6I_7I_8$), kod sadrži 8 informacijskih bitova + 4 paritetna bita što je ukupno 12 bitova ($P_1P_2I_1P_3I_2I_3I_4P_4I_5I_6I_7I_8$).

Broj paritetnih bitova u odnosu na informacijske bitove prikazan je u tablici 2.10.

Tablica 2.10: Broj paritetnih bitova u odnosu na informacijske bitove Hammingova koda

Informacijski bitovi	**Paritetni bitovi**
1	2
2-4	3
5-11	4
12-26	5
27-57	6
58-120	7
121-247	8

Prema tablici 2.10 možemo zaključiti da je primjena ovog koda efikasnija što je veći broj informacijskih bitova:

- za zaštitu 8 bitova potrebna su dodatna 4 paritetna bita ili povećanje ukupnog broja bitova za 50%,
- za zaštitu 128 bitova potrebno je dodatnih 8 paritetnih bitova ili povećanje ukupnog broja bitova za 6,25%.

U nastavku će biti prikazano konstruiranje Hammingovog koda. Konstruiranje Hammingovog koda bit će opisano na 8-bitnom podatku s informacijskim bitovima $I_1I_2I_3I_4I_5I_6I_7I_8$. Iz tablice 2.10 vidimo da su za kodiranje 8 informacijskih bitova potrebna 4 paritetna bita. Hammingov kod imat će ukupno 12 bitova. Kod određivanja paritetnih bitova koristit ćemo parni paritet.

Najprije je potrebno pozicionirati paritetne bitove na pozicije koje su potencije broja 2 (1 , 2, 4, 8) kako je prikazano u tablici 2.11.

Tablica 2.11: Kreiranje tablice za Hammingov kod - pozicioniranje paritetnih bitova

Pozicija bitova	1	2	3	4	5	6	7	8	9	10	11	12
Hammingov kod	P_1	P_2		P_3				P_4				

Nakon pozicioniranja paritetnih bitova u Hammingov kod, potrebno je pozicionirati informacijske bitove redom na slobodne pozicije počevši od pozicije broj 3 (tablica 2.12).

Tablica 2.12: Kreiranje tablice za Hammingov kod - pozicioniranje informacijskih bitova

Pozicija bitova	1	2	3	4	5	6	7	8	9	10	11	12
Hammingov kod	P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8

Bitovi se grupiraju u grupe za određivanje paritetnih bitova. Paritetni bit grupe određuje se tako da paritet grupe bude paran ili neparan. U ovom primjeru koristit ćemo parni paritet.

Za određivanje paritetnog bita prve grupe P_1 koristi se svaki drugi bit u tablici počevši od paritetnog bita P_1 (tablica 2.13).

Tablica 2.13: Kreiranje tablice za Hammingov kod - određivanje paritetnog bita P_1

Pozicija bitova		1	2	3	4	5	6	7	8	9	10	11	12
Hammingova kodirana riječ		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8
Grupa bitova za paritetne bitove	P_1	P_1		I_1		I_2		I_4		I_5		I_7	
	P_2												
	P_3												
	P_4												

Paritetni bit P_1 određuje se tako da bitovi prve grupe čine parni paritet. Formalno se ovaj uvjet može zapisati pomoću EX ILI operacije $\oplus$[1]:

$$P_1 \oplus I_1 \oplus I_2 \oplus I_4 \oplus I_5 \oplus I_7 = 0. \qquad (2.13)$$

[1] EX-ILI operator će na izlazu dati nulu kada je na ulazu paran broj jedinica.

Prema tome, paritetni bit P_1 može se izračunati na sljedeći način:

$$P_1 = I_1 \oplus I_2 \oplus I_4 \oplus I_5 \oplus I_7. \tag{2.14}$$

Za određivanje paritetnog bita druge grupe P_2 koristi se svaka druga skupina po dva bita u tablici počevši od paritetnog bita P_2 (tablica 2.14).

Tablica 2.14: Kreiranje tablice za Hammingov kod - određivanje paritetnog bita P_1

Pozicija bitova		1	2	3	4	5	6	7	8	9	10	11	12
Hammingova kodirana riječ		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8
Grupa bitova za paritetne bitove	P_1	P_1		I_1		I_2		I_4		I_5		I_7	
	P_2		P_2	I_1			I_3	I_4			I_6	I_7	
	P_3												
	P_4												

Paritetni bit P_2 određuje se tako da bitovi druge grupe čine parni paritet. Formalno se ovaj uvjet može zapisati pomoću EX ILI operacije $\oplus$:

$$P_2 \oplus I_1 \oplus I_3 \oplus I_4 \oplus I_6 \oplus I_7 = 0. \tag{2.15}$$

Prema tome, paritetni bit P_2 može se izračunati na sljedeći način:

$$P_2 = I_1 \oplus I_3 \oplus I_4 \oplus I_6 \oplus I_7. \tag{2.16}$$

Za određivanje paritetnog bita treće grupe P_3 koristi se svaka druga skupina po četiri bita u tablici počevši od paritetnog bita P_3 (tablica 2.15).

Tablica 2.15: Kreiranje tablice za Hammingov kod - određivanje paritetnog bita P_3

Pozicija bitova		1	2	3	4	5	6	7	8	9	10	11	12
Hammingova kodirana riječ		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8
Grupa bitova za paritetne bitove	P_1	P_1		I_1		I_2		I_4		I_5		I_7	
	P_2		P_2	I_1			I_3	I_4			I_6	I_7	
	P_3				P_3	I_2	I_3	I_4					I_8
	P_4												

Paritetni bit P_3 određuje se tako da bitovi treće grupe čine parni paritet. Formalno se ovaj uvjet može zapisati pomoću EX ILI operacije $\oplus$:

$$P_3 \oplus I_2 \oplus I_3 \oplus I_4 \oplus I_8 = 0. \tag{2.17}$$

Prema tome, paritetni bit P_3 može se izračunati na sljedeći način:

$$P_3 = I_2 \oplus I_3 \oplus I_4 \oplus I_8. \tag{2.18}$$

Za određivanje paritetnog bita četvrte grupe P_4 koristi se svaka druga skupina po osam bitova u tablici počevši od paritetnog bita P_4 (tablica 2.16).

Tablica 2.16: Kreiranje tablice za Hammingov kod - određivanje paritetnog bita P_4

Pozicija bitova		1	2	3	4	5	6	7	8	9	10	11	12
Hammingova kodirana riječ		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8
Grupa bitova za paritetne bitove	P_1	P_1		I_1		I_2		I_4		I_5		I_7	
	P_2		P_2	I_1			I_3	I_4			I_6	I_7	
	P_3				P_3	I_2	I_3	I_4					I_8
	P_4								P_4	I_5	I_6	I_7	I_8

Paritetni bit P_4 određuje se tako da bitovi četvrte grupe čine parni paritet. Formalno se ovaj uvjet može zapisati pomoću EX ILI operacije $\oplus$:

$$P_4 \oplus I_5 \oplus I_6 \oplus I_7 \oplus I_8 = 0. \tag{2.19}$$

Prema tome, paritetni bit P_4 može se izračunati na sljedeći način:

$$P_4 = I_5 \oplus I_6 \oplus I_7 \oplus I_8. \tag{2.20}$$

Općenito, za određivanje paritetnog bita n-te grupe P_n koristi se svaka druga skupina po 2^{n-1} bita u tablici počevši od paritetnog bita P_n. Paritetni bit P_n određuje se tako da bitovi n-te grupe čine parni paritet.

☞ Primjer 2.7.1

Hammingovim kodom zaštitite 6-bitnu riječ 101110. Kod određivanja paritetnih bitova koristite parni paritet. Pokažite kako se na prijemnoj strani može otkriti i ispraviti greška ako je pri prijenosu promijenjen informacijski bit I_4.

✍ Rješenje:

Iz tablice 2.10 vidimo da su za kodiranje 6 informacijskih bitova potrebna 4 paritetna bita. Hammingov kod imat će ukupno 10 bitova. Kreiranje Hammingovog koda za riječ 101110 prikazano je u tablici 2.17.

Tablica 2.17: Kreiranje tablice za Hammingovo kodiranje riječi 101110 - određivanje paritetnih bitova

Pozicija bitova		1	2	3	4	5	6	7	8	9	10
Hammingov kod		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6
		?	?	1	?	0	1	1	?	1	0
Grupa bitova za paritetne bitove	P_1	P_1		1		0		1		1	
	P_2		P_2	1			1	1			0
	P_3				P_3	0	1	1			
	P_4								P_4	1	0

Koristit ćemo parni paritet kod određivanja paritetnih bitova. U prvoj grupi bitova koje štiti paritetni bit P_1 neparan je broj jedinica te je stoga $P_1 = 1$. U drugoj grupi bitova koje štiti paritetni bit P_2 neparan je broj jedinica te je stoga $P_2 = 1$. U trećoj grupi bitova koje štiti paritetni bit P_3 paran je broj jedinica te je stoga $P_3 = 0$. U četvrtoj grupi bitova koje štiti paritetni bit P_4 neparan je broj jedinica te je stoga $P_4 = 1$.

Isti rezultat dobili bismo korištenjem relacija (2.14), (2.16), (2.18) i (2.20):

$$
\begin{aligned}
P_1 &= I_1 \oplus I_2 \oplus I_4 \oplus I_5 = 1 \oplus 0 \oplus 1 \oplus 1 = 1\\
P_2 &= I_1 \oplus I_3 \oplus I_4 \oplus I_6 = 1 \oplus 1 \oplus 1 \oplus 0 = 1\\
P_3 &= I_2 \oplus I_3 \oplus I_4 = 0 \oplus 1 \oplus 1 = 0\\
P_4 &= I_5 \oplus I_6 = 1 \oplus 0 = 1
\end{aligned}
$$

Potpuna tablica Hammingovog kodiranja riječi 101110 prikazana je tablicom 2.18.

Tablica 2.18: Kreiranje tablice za Hammingov kodiranje riječi 101110 - potpuna tablica

<table>
<tr><td colspan="2">Pozicija bitova</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td colspan="2" rowspan="2">Hammingov kod</td><td>P_1</td><td>P_2</td><td>I_1</td><td>P_3</td><td>I_2</td><td>I_3</td><td>I_4</td><td>P_4</td><td>I_5</td><td>I_6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td rowspan="4">Grupa bitova za paritetne bitove</td><td>P_1</td><td>1</td><td></td><td>1</td><td></td><td>0</td><td></td><td>1</td><td></td><td>1</td><td></td></tr>
<tr><td>P_2</td><td></td><td>1</td><td>1</td><td></td><td></td><td>1</td><td>1</td><td></td><td></td><td>0</td></tr>
<tr><td>P_3</td><td></td><td></td><td></td><td>0</td><td>0</td><td>1</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>P_4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>1</td><td>0</td></tr>
</table>

Prema gore opisanom postupku, 6-bitni podatak 101110 zaštićen Hammingovim kodom može se napisati kao 1110011110. Predajnik šalje 10-bitni kod. Prijemnik iz tog koda treba izdvojiti 6 informacijskih bitova od I_1 do I_6. Na temelju ispitivanja pariteta svake grupe bitova koju štite paritetni bitovi P_1 do P_4 možemo otkriti jednostruku grešku u prenesenoj informaciji. Neka je pri prijenosu informacija nastala pogreška na informacijskom bitu I_4:

$$
\begin{array}{cccccccccc}
P_1 & P_2 & I_1 & P_3 & I_2 & I_3 & I_4 & P_4 & I_5 & I_6\\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0\\
 & & & & & & \downarrow & & & \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0
\end{array}
$$

Na prijemnoj se strani radi korekcija jednostruke greške. Za svaku grupu bitova koju štite paritetni bitovi P_1 do P_4 ispituje se paritet koji se naziva sindrom. Sindrom C_x za svaku grupu bitova formira se po istom obrascu kao i paritetni bit:

$$
\begin{aligned}
C_1 &= P_1 \oplus I_1 \oplus I_2 \oplus I_4 \oplus I_5\\
C_2 &= P_2 \oplus I_1 \oplus I_3 \oplus I_4 \oplus I_6\\
C_3 &= P_3 \oplus I_2 \oplus I_3 \oplus I_4\\
C_4 &= P_4 \oplus I_5 \oplus I_6
\end{aligned}
\tag{2.21}
$$

Ako je podatak ispravno prenesen, na prijemnoj strani vrijedi C_1=C_2=C_3=C_4=0. Ako se promijeni bit I_4 tada prema relaciji (2.21) vrijedi:

$$
\begin{aligned}
C_1 &= P_1 \oplus I_1 \oplus I_2 \oplus I_4 \oplus I_5 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1\\
C_2 &= P_2 \oplus I_1 \oplus I_3 \oplus I_4 \oplus I_6 = 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1\\
C_3 &= P_3 \oplus I_2 \oplus I_3 \oplus I_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1\\
C_4 &= P_4 \oplus I_5 \oplus I_6 = 1 \oplus 1 \oplus 0 = 0
\end{aligned}
$$

Do istog rezultata moguće je doći i tablično (tablica 2.19).

Tablica 2.19: Određivanje sindroma Hammingova koda 1110010110

Pozicija bitova		1	2	3	4	5	6	7	8	9	10
Hammingov kod		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6
		1	1	1	0	0	1	0	1	1	0
Sindrom	$C_1 = 1$	1		1		0		0		1	
	$C_2 = 1$		1	1			1	0			0
	$C_3 = 1$				0	0	1	0			
	$C_4 = 0$								0	1	0

U prvoj grupi bitova koje štiti paritetni bit P_1 neparan je broj jedinica te je stoga sindrom $C_1 = 1$. U drugoj grupi bitova koje štiti paritetni bit P_2 neparan je broj jedinica te je stoga sindrom $C_2 = 1$. U trećoj grupi bitova koje štiti paritetni bit P_3 paran je broj jedinica te je stoga sindrom $C_3 = 1$. U četvrtoj grupi bitova koje štiti paritetni bit P_4 neparan je broj jedinica te je stoga sindrom $C_4 = 0$.

Nakon što se izračunaju svi sindromi, kreira se riječ $C_4C_3C_2C_1 = 0111_2 = 7_{10}$. Iz tablice je tu riječ moguće pročitati tako da se sindromi zapišu od dna prema vrhu. Sindrom daje poziciju bita na kojoj se pojavila greška. Iz tablice 2.19 vidimo da je na poziciji broj 7 informacijski bit I_4 na kojem se pojavila greška. Sindrom na opisani način omogućuje otkrivanje jednostruke greške.

☞ Primjer 2.7.2

Hammingovim kodom zaštitite 8-bitnu riječ 11110000. Kod određivanja paritetnih bitova koristite neparni paritet. Pokažite kako se na prijemnoj strani može otkriti i ispraviti greška ako je pri prijenosu promijenjen informacijski bit I_2.

✍ Rješenje:

Iz tablice 2.10 vidimo da su za kodiranje 8 informacijskih bitova potrebna 4 paritetna bita. Hammingov kod će imati ukupno 12 bitova. Kreiranje Hammingovog koda za riječ 11110000 prikazano je u tablici 2.20.

Tablica 2.20: Kreiranje tablice za Hammingovo kodiranje riječi 11110000 - određivanje paritetnih bitova

Pozicija bitova		1	2	3	4	5	6	7	8	9	10	11	12
Hammingov kod		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8
		?	?	1	?	1	1	1	?	0	0	0	0
Grupa bitova za paritetne bitove	P_1	P_1		1		1		1		0		0	
	P_2		P_2	1			1	1			0	0	
	P_3				P_3	1	1	1					0
	P_4								P_4	0	0	0	0

Koristit ćemo neparni paritet kod određivanja paritetnih bitova. U prvoj grupi bitova koje štiti paritetni bit P_1 neparan je broj jedinica te je stoga $P_1 = 0$. U drugoj grupi bitova koje štiti paritetni bit P_2 neparan je broj jedinica te je stoga $P_2 = 0$. U trećoj grupi bitova koje štiti paritetni bit P_3 neparan je broj jedinica te je stoga $P_3 = 0$. U četvrtoj grupi bitova koje štiti paritetni bit P_4 paran je broj jedinica te je stoga $P_4 = 1$.

Isti rezultat dobili bismo korištenjem negiranih relacija (2.14), (2.16), (2.18) i (2.20):

$$\begin{aligned}
P_1 &= \overline{I_1 \oplus I_2 \oplus I_4 \oplus I_5 \oplus I_7} = \overline{1 \oplus 1 \oplus 1 \oplus 0 \oplus 0} = 0\\
P_2 &= \overline{I_1 \oplus I_3 \oplus I_4 \oplus I_6 \oplus I_7} = \overline{1 \oplus 1 \oplus 1 \oplus 0 \oplus 0} = 0\\
P_3 &= \overline{I_2 \oplus I_3 \oplus I_4 \oplus I_8} = \overline{1 \oplus 1 \oplus 1 \oplus 0} = 0\\
P_4 &= \overline{I_5 \oplus I_6 \oplus I_7 \oplus I_8} = \overline{0 \oplus 0 \oplus 0 \oplus 0} = 1
\end{aligned}$$

Tablica 2.21: Kreiranje tablice za Hammingovo kodiranje riječi 11110000 - potpuna tablica

Pozicija bitova		1	2	3	4	5	6	7	8	9	10	11	12
Hammingov		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8
kod		0	0	1	0	1	1	1	1	0	0	0	0
Grupa bitova za paritetne bitove	P_1	0		1		1		1		0		0	
	P_2		0	1			1	1			0	0	
	P_3				0	1	1	1					0
	P_4								1	0	0	0	0

Potpuna tablica Hammingovog kodiranja riječi 11110000 prikazana je tablicom 2.21. Prema gore opisanom postupku 8-bitni podatak 11110000 zaštićen Hammingovim kodom može se napisati kao 00101111000. Predajnik šalje 12-bitni kod. Prijemnik iz tog koda treba izdvojiti 8 informacijskih bitova od I_1 do I_8. Na temelju ispitivanja pariteta svake grupe bitova koju štite paritetni bitovi P_1 do P_4 možemo otkriti jednostruku grešku u prenesenoj informaciji. Neka je pri prijenosu informacija nastala pogreška na informacijskom bitu I_2:

$$\begin{array}{cccccccccccc}
P_1 & P_2 & I_1 & P_3 & I_2 & I_3 & I_4 & P_4 & I_5 & I_6 & I_7 & I_8\\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0\\
 & & & & \downarrow & & & & & & & \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0
\end{array}$$

Na prijemnoj se strani radi korekcija jednostruke greške. Za svaku grupu bitova koju štite paritetni bitovi P_1 P_4 ispituje se paritet, odnosno određuje se sindrom C_x za svaku grupu bitova:

$$\begin{aligned}
C_1 &= \overline{P_1 \oplus I_1 \oplus I_2 \oplus I_4 \oplus I_5 \oplus I_7}\\
C_2 &= \overline{P_2 \oplus I_1 \oplus I_3 \oplus I_4 \oplus I_6 \oplus I_7}\\
C_3 &= \overline{P_3 \oplus I_2 \oplus I_3 \oplus I_4 \oplus I_8}\\
C_4 &= \overline{P_4 \oplus I_5 \oplus I_6 \oplus I_7 \oplus I_8}
\end{aligned} \tag{2.22}$$

Ako je podatak ispravno prenesen, na prijemnoj strani vrijedi C_1=C_2=C_3=C_4=0. Ako se promijeni bit I_2 tada prema relaciji (2.22) vrijedi:

$$\begin{aligned}
C_1 &= \overline{P_1 \oplus I_1 \oplus I_2 \oplus I_4 \oplus I_5 \oplus I_7} = \overline{0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0} = 1\\
C_2 &= \overline{P_2 \oplus I_1 \oplus I_3 \oplus I_4 \oplus I_6 \oplus I_7} = \overline{0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0} = 0\\
C_3 &= \overline{P_3 \oplus I_2 \oplus I_3 \oplus I_4 \oplus I_8} = \overline{0 \oplus 0 \oplus 1 \oplus 1 \oplus 0} = 1\\
C_4 &= \overline{P_4 \oplus I_5 \oplus I_6 \oplus I_7 \oplus I_8} = \overline{1 \oplus 0 \oplus 0 \oplus 0 \oplus 0} = 0
\end{aligned}$$

Do istog rezultata moguće je doći i tablično (tablica 2.22).

Tablica 2.22: Određivanje sindroma Hammingova koda 00101111000

Pozicija bitova		1	2	3	4	5	6	7	8	9	10	11	12
Hammingov kod		P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8
		0	0	1	0	0	1	1	1	0	0	0	0
Sindrom	$C_1 = 1$	0		1		0		1		0		0	
	$C_2 = 0$		0	1			1	1			0	0	
	$C_3 = 1$				0	0	1	1					0
	$C_4 = 0$								1	0	0	0	0

U prvoj grupi bitova koje štiti paritetni bit P_1 paran je broj jedinica te je stoga sindrom $C_1 = 1$. U drugoj grupi bitova koje štiti paritetni bit P_2 neparan je broj jedinica te je stoga sindrom $C_2 = 0$. U trećoj grupi bitova koje štiti paritetni bit P_3 paran je broj jedinica te je stoga sindrom $C_3 = 1$. U četvrtoj grupi bitova koje štiti paritetni bit P_4 neparan je broj jedinica te je stoga sindrom $C_4 = 0$.

Nakon što se izračunaju svi sindromi kreira se riječ $C_4C_3C_2C_1 = 0101_2 = 5_{10}$. Iz tablice je tu riječ moguće pročitati tako da se sindromi zapišu od dna prema vrhu. Sindrom daje poziciju bita na kojoj se pojavila greška. Iz tablice 2.22 vidimo da je na poziciji broj 5 informacijski bit I_2 na kojem se pojavila greška. Sindrom na opisani način omogućuje otkrivanje jednostruke greške.

2.7.1 Zadaci za vježbu

✎ Zadatak 2.7.1

Hammingovim kodom zaštitite 6-bitnu riječ 110011. Kod određivanja paritetnih bitova koristite parni paritet. Pokažite kako se na prijemnoj strani može otkriti i ispraviti greška ako je pri prijenosu promijenjen informacijski bit I_5.

✎ Zadatak 2.7.2

Hammingovim kodom zaštitite 8-bitnu riječ 00001111. Kod određivanja paritetnih bitova koristite neparni paritet. Pokažite kako se na prijemnoj strani može otkriti i ispraviti greška ako je pri prijenosu promijenjen informacijski bit I_3.

Poglavlje 3

Digitalna logika

3.1 Booleova algebra

Donošenje odluka na temelju logičkog zaključivanja u govornom jeziku uobičajeno koristi ključne riječi 'i', 'ili', 'ne', 'ako' i slične. Znanost koja se bavi procesima logičkog zaključivanja, odnosno formama i zakonima mišljenja poznata je još iz drevne Grčke, a zove se logika. Engleski matematičar George Boole prvi je proveo matematičku analizu logike sudova. Algebra za analizu logike, odnosno logička algebra zove se Booleova algebra. Značajniji interes za razvoj ove grane matematike porastao je nakon njene uspješne primjene u analizi relejnih sklopova u telefonskim centralama i računskim strojevima.

Da bi se Booleova algebra mogla uspješno primijeniti, potrebno je definirati sljedeće:

- skup od dva ili više različitih članova (elemenata) $S = \{a, b, c, d,...\}$,
  - za dvočlanu Booleovu algebru skup S ima dva člana: $S = \{0, 1\}$,
- operatore $+$ i $\times$ koji primijenjeni na članove skupa S proizvode novog člana iz skupa S,
- skup aksioma.

Za praktičnu primjenu Booleove algebre važno je poznavati zakone Booleove algebre koji se sastoje od aksioma i teorema. U nastavku će biti spomenuti najvažniji.

1. aksiom o neutralnom elementu
$$\begin{aligned} A + 0 = A \\ A \cdot 1 = A \end{aligned} \tag{3.1}$$

2. aksiom o komplementu
$$\begin{aligned} A + \overline{A} = 1 \\ A \cdot \overline{A} = 0 \end{aligned} \tag{3.2}$$

3. zakon komutacije
$$\begin{aligned} A + B = B + A \\ A \cdot B = B \cdot A \end{aligned} \tag{3.3}$$

4. zakon distribucije

$$\begin{aligned} A \cdot (B + C) = A \cdot B + A \cdot C \\ A + B \cdot C = (A + B) \cdot (A + C) \end{aligned} \tag{3.4}$$

Iz aksioma se izvode neki od sljedećih teorema.

1. zakon dominacije

$$\begin{aligned} A + 1 = 1 \\ A \cdot 0 = 0 \end{aligned} \tag{3.5}$$

2. zakon idempotencije

$$\begin{aligned} A + A = A \\ A \cdot A = A \end{aligned} \tag{3.6}$$

3. zakon involucije

$$\overline{\overline{A}} = A \tag{3.7}$$

4. zakon apsorcije

$$\begin{aligned} A + A \cdot B = A \\ A \cdot (A + B) = A \end{aligned} \tag{3.8}$$

5. zakon asocijacije

$$\begin{aligned} A + (B + C) = (A + B) + C \\ A \cdot (B \cdot C) = (A \cdot B) \cdot C \end{aligned} \tag{3.9}$$

6. De Morganov zakon

$$\begin{aligned} \overline{A + B} = \overline{A} \cdot \overline{B} \\ \overline{A \cdot B} = \overline{A} + \overline{B} \end{aligned} \tag{3.10}$$

7. generalizirani De Morganov zakon

$$\begin{aligned} \overline{A + B + C + ...} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot ... \\ \overline{A \cdot B \cdot C \cdot ...} = \overline{A} + \overline{B} + \overline{C} + ... \end{aligned} \tag{3.11}$$

8. zakon simplifikacije

$$\begin{aligned} A \cdot B + A \cdot \overline{B} = A \\ (A + B) \cdot (A + \overline{B}) = A \end{aligned} \tag{3.12}$$

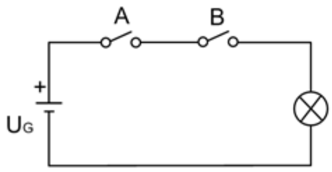
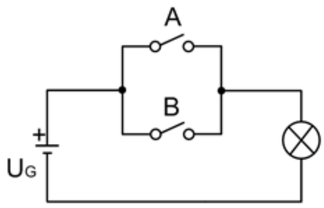
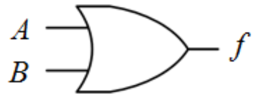
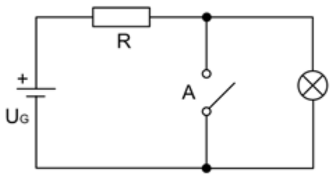
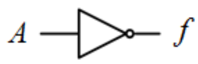
Logička funkcija je logička varijabla koja poprima logičke vrijednosti 0 ili 1, ovisno o vrijednostima nezavisnih logičkih varijabli povezanih logičkim operatorima u logičkom izrazu. Osnovne logičke funkcije, njihove tablice kombinacija, simboli, Booleovi izrazi i realizacija pomoću sklopki prikazani su na slici 3.1.

Osnovne logičke funkcije su (slika 3.1):

- I - logička funkcija koja na svom izlazu poprima logičku vrijednost 1 samo ako su svi njeni ulazi u logičkoj vrijednosti 1. I funkcija dviju logičkih varijabli definira se kao $f = A \cdot B$. Izraz $f = A \cdot B$ može se skraćeno pisati kao $f = AB$.

- ILI - logička funkcija koja na svom izlazu poprima logičku vrijednost 0 samo ako su svi njeni ulazi u logičkoj vrijednosti 0. ILI funkcija dviju logičkih varijabli definira se kao $f = A + B$.

- NE - logička funkcija koja na svom izlazu poprima negaciju (komplement) logičke vrijednosti na svom ulazu. NE funkcija jedne logičke varijable definira se kao $f = \overline{A}$.

<table>
<tr><th>Operator</th><th>Sklop sa sklopkama</th><th>Tablica kombinacija</th><th>Simbol</th><th>Booleov izraz</th></tr>
<tr><td>I</td><td>
</td><td><table><tr><th>A</th><th>B</th><th>f</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>
</td><td>$f = A \cdot B$</td></tr>
<tr><td>ILI</td><td>
</td><td><table><tr><th>A</th><th>B</th><th>f</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>
</td><td>$f = A + B$</td></tr>
<tr><td>NE</td><td>
</td><td><table><tr><th>A</th><th>f</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td><td>
</td><td>$f = \overline{A}$</td></tr>
</table>

Slika 3.1: Osnovne logičke funkcije

☞ Primjer 3.1.1

Napišite tablicu kombinacija za logičku funkciju $f(A,B,C,D) = AD + B\overline{C} + \overline{B}CD$.

✍ Rješenje:

Logička funkcija četiri varijable (A, B, C, D) ima $2^4 = 16$ mogućih kombinacija. Općenito, logička funkcija n varijabli ima 2^n mogućih kombinacija. Tablica kombinacija popunjava se najprije sa svim mogućim kombinacijama ulaznih varijabli (A, B, C, D). Prva kombinacija je 0000, druga kombinacija je 0001 itd. Kombinacije su 4-bitni binarni brojevi dekadskih vrijednosti redom od 0 (0000_2) do 15 (1111_2). Logička funkcija $f(A,B,C,D)$ sastoji se od tri produkta (I operator) koji su povezani operatorom sume (ILI operator). Tablica kombinacija za logičku funkciju $f(A,B,C,D) = AD + B\overline{C} + \overline{B}CD$ prikazana je tablicom 3.1. Za svaki produkt logičke funkcije $f(A,B,C,D)$ napravljen je stupac istinitosti. Prvi produkt logičke funkcije $f(A,B,C,D)$ je AD. Produkt AD imat će logičku vrijednost 1 samo kada su logičke varijable A i D u logičkoj vrijednosti 1. U tablici 3.1 vidimo da je to slučaj kod kombinacija 10001, 1011, 1101 i 1111. Za ostale vrijednosti logičkih varijabli A i D produkt AD imat će vrijednost 0. Drugi produkt logičke funkcije $f(A,B,C,D)$ je $B\overline{C}$. Produkt $B\overline{C}$ imat će logičku vrijednost 1 samo kada je logička varijabla B u logičkoj vrijednosti 1 i C u logičkoj vrijednosti 0. U tablici 3.1 vidimo da je to slučaj kod kombinacija 0100, 0101, 1100 i 1101. Za ostale vrijednosti logičkih varijabli B i C produkt $B\overline{C}$ imat će vrijednost 0. Po istom obrascu možemo popuniti stupac produkta $\overline{B}CD$. Vrijednost logičke funkcije $f(A,B,C,D)$ bit će 1 kada je barem jedan produkt u stanju logičke vrijednosti 1.

Tablica 3.1: Tablica kombinacija za logičku funkciju $f(A,B,C,D) = AD + B\overline{C} + \overline{B}CD$

A	B	C	D	AD	$B\overline{C}$	$\overline{B}CD$	f
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	0	0	1	0	1
1	1	0	1	1	1	0	1
1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	1

☞ Primjer 3.1.2

Zadane su logičke funkcije:

$$f_1(A,B,C) = \overline{A}B\overline{C} + AB\overline{C} + ABC$$
$$f_2(A,B,C) = AB + B\overline{C}.$$

Pokažite da su logičke funkcije $f_1(A,B,C)$ i $f_2(A,B,C)$ jednake pomoću:
a) tablice kombinacija,
b) zakona Booleove algebre.

✍ Rješenje:

Dvije jednake logičke funkcije mogu biti zapisane na različite načine. Jednakost dviju logičkih funkcija moguće je jednoznačno utvrditi pomoću tablice kombinacija ili pomoću zakona Booleove algebre.

a) Tablica kombinacija za logičke funkcije $f_1(A,B,C) = \overline{A}B\overline{C} + AB\overline{C} + ABC$ i $f_2(A,B,C) = AB + B\overline{C}$ prikazana je tablicom 3.2.

Tablica 3.2: Tablica kombinacija za logičke funkcije $f_1(A,B,C) = \overline{A}B\overline{C} + AB\overline{C} + ABC$ i $f_2(A,B,C) = AB + B\overline{C}$

A	B	C	$\overline{A}B\overline{C}$	$AB\overline{C}$	ABC	f_1	AB	$B\overline{C}$	f_2
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0
1	1	0	0	1	0	1	1	1	1
1	1	1	0	0	1	1	1	0	1

Tablica 3.2 ima 8 mogućih kombinacija jer su logičke funkcije $f_1(A{,}B{,}C)$ i $f_2(A{,}B{,}C)$ funkcije tri varijable ($2^3 = 8$). Popunjavanje tablice kombinacija radi se po istom obrascu kao u prethodnom primjeru. Kako bi logičke funkcije $f_1(A{,}B{,}C)$ i $f_2(A{,}B{,}C)$ bile jednake, stupci u tablici kombinacija moraju im biti jednaki. Prema tablici 3.2 logičke funkcije $f_1(A{,}B{,}C)$ i $f_2(A{,}B{,}C)$ jednake su jer su im stupci u tablici kombinacija jednaki.

b) Jednakost dviju logičkih funkcija moguće je pokazati i zakonima Booleove algebre. Pokušajmo logičku funkciju $f_1(A{,}B{,}C)$ svesti na funkciju $f_2(A{,}B{,}C)$. Koristeći zakon idempotencije (3.6), a zatim zakon distribucije (3.4) slijedi:

$$
\begin{aligned}
f_1(A{,}B{,}C) &= \overline{A}B\overline{C} + AB\overline{C} + ABC\\
f_1(A{,}B{,}C) &= \overline{A}B\overline{C} + \underbrace{AB\overline{C} + AB\overline{C}}_{(3.6)} + ABC\\
f_1(A{,}B{,}C) &= \underbrace{(\overline{A} + A)B\overline{C}}_{(3.4)} + \underbrace{AB(\overline{C} + C)}_{(3.4)}.
\end{aligned}
$$

Prema aksiomu o komplementu (3.2) vrijedi da je $\overline{A} + A = \overline{C} + C = 1$ pa se prethodni izraz može pojednostaviti:

$$
\begin{aligned}
f_1(A{,}B{,}C) &= \underbrace{(\overline{A} + A)}_{1}B\overline{C} + AB\underbrace{(\overline{C} + C)}_{1}\\
f_1(A{,}B{,}C) &= B\overline{C} + AB = f_2(A{,}B{,}C).
\end{aligned}
$$

Pokazali smo da su logičke funkcije $f_1(A{,}B{,}C)$ i $f_2(A{,}B{,}C)$ jednake.

3.1.1 Zadaci za vježbu

✎ Zadatak 3.1.1

Napišite tablicu kombinacija za logičku funkciju $f(A{,}B{,}C{,}D) = BC + A\overline{D} + \overline{A}B\overline{D}$.

✎ Zadatak 3.1.2

Zadane su logičke funkcije:

$$
\begin{aligned}
f_1(A{,}B{,}C) &= \overline{A}\overline{B} + A\overline{B} + \overline{A}B\overline{C}\\
f_2(A{,}B{,}C) &= \overline{B} + \overline{A}\overline{C}.
\end{aligned}
$$

Pokažite da su logičke funkcije $f_1(A{,}B{,}C)$ i $f_2(A{,}B{,}C)$ jednake pomoću:
a) tablice kombinacija,
b) zakona Booleove algebre.

3.2 Kanonski oblik logičkih funkcija

Logička funkcija zadana tablicom kombinacija može se prikazati na dva načina:

1. polazi se od uvjeta da vrijednost logičke funkcije bude logička vrijednost 1,
2. polazi se od uvjeta da vrijednost logičke funkcije bude logička vrijednost 0.

Na ta se dva načina logička funkcija može prikazati u dva kanonska oblika.

Tablica 3.3: Tablica kombinacija za proizvoljnu logičku funkciju dvije varijable (mintermi)

A	B	Vrijednost funkcije	Mintermi
0	0	$v_0 = 0$	$m_0 = \overline{A}\overline{B}$
0	1	$v_1 = 0$	$m_1 = \overline{A}B$
1	0	$v_2 = 0$	$m_2 = A\overline{B}$
1	1	$v_3 = 1$	$m_3 = AB$

Vrijednost funkcije v_i ($i = 0{,}1{,}2{,}3$) definirana je za svaku kombinaciju logičkih varijabli A i B (v_i može poprimiti logičku vrijednost 0 ili 1). Minterm je produktni član koji sadrži sve nezavisne varijable logičke funkcije tako da je vrijednost produkta jednaka 1 (ako je vrijednost logičke varijable 0, tada se varijabla invertira). Logička funkcija definirana tablicom 3.3 može se zapisati kao:

$$f_m = v_0 \cdot m_0 + v_1 \cdot m_1 + v_2 \cdot m_2 + v_3 \cdot m_3$$
$$f_m = 0 \cdot (\overline{A}\overline{B}) + 0 \cdot (\overline{A}B) + 0 \cdot (A\overline{B}) + 1 \cdot (AB) = AB.$$

Općenito za logičku funkciju n varijabli vrijedi:

$$f = \sum_{i=0}^{2^n-1} v_i m_i. \tag{3.13}$$

Zapis logičke funkcije (3.13) prvi je kanonski zapis i zove se standardni disjunktivni oblik jer su mintermi povezani ILI funkcijom, odnosno disjunkcijom. Logička se funkcija prema relaciji (3.13) zapisuje u obliku sume minterma m_i.

Tablica 3.4: Tablica kombinacija za proizvoljnu logičku funkciju dvije varijable (makstermi)

A	B	Vrijednost funkcije	Makstermi
0	0	$v_0 = 0$	$M_0 = (A + B)$
0	1	$v_1 = 0$	$M_1 = (A + \overline{B})$
1	0	$v_2 = 0$	$M_2 = (\overline{A} + B)$
1	1	$v_3 = 1$	$M_3 = (\overline{A} + \overline{B})$

Vrijednost funkcije v_i ($i = 0{,}1{,}2{,}3$) definirana je za svaku kombinaciju logičkih varijabli A i B (v_i može poprimiti logičku vrijednost 0 ili 1). Maksterm je suma član koji sadrži sve nezavisne varijable logičke funkcije tako da je vrijednost sume jednaka 0 (ako je vrijednost logičke varijable 1, tada se varijabla invertira). Logička funkcija definirana tablicom 3.4 može se zapisati kao:

$$f_M = (v_0 + M_0) \cdot (v_1 + M_1) \cdot (v_2 + M_2) \cdot (v_3 + M_3)$$
$$f_M = (0 + (A + B)) \cdot (0 + (A + \overline{B})) \cdot (0 + (\overline{A} + B)) \cdot (1 + (\overline{A} + \overline{B}))$$
$$f_M = (A + B) \cdot (A + \overline{B}) \cdot (\overline{A} + B).$$

Općenito za logičku funkciju n varijabli vrijedi:

$$f = \prod_{i=0}^{2^n-1} (v_i + M_i). \tag{3.14}$$

Zapis logičke funkcije (3.14) drugi je kanonski zapis i zove se standardni konjuktivni oblik jer su makstermi povezani I funkcijom, odnosno konjukcijom. Logička se funkcija prema relaciji (3.14) zapisuje u obliku produkta maksterma M_i. U oba kanonska oblika logička funkcija mora biti jednaka. Pomoću sume minterma dobili smo funkciju $f_m = AB$, a pomoću produkta maksterma dobili smo funkciju $f_M = (A+B)\cdot(A+\overline{B})\cdot(\overline{A}+B)$.

Primjenom teorema i aksioma Booleove algebre pokazat ćemo da vrijedi $f_M = f_m$:

$$f_M = (A+B)\cdot(A+\overline{B})(\overline{A}+B) = (\underbrace{AA}_{A} + A\overline{B} + AB + \underbrace{B\overline{B}}_{0})(\overline{A}+B)$$
$$f_M = (A + A\overline{B} + AB)(\overline{A}+B) = \underbrace{A\overline{A}}_{0} + \underbrace{A\overline{B}\overline{A}}_{0} + \underbrace{AB\overline{A}}_{0} + AB + \underbrace{A\overline{B}B}_{0} + \underbrace{ABB}_{AB}$$
$$f_M = AB + AB = AB = f_m.$$

Logička funkcija prikazana u kanonskom obliku sume minterma i produkta maksterma može se pisati skraćeno. Na primjer, logička funkcija tri varijable definirana sumom minterma:

$$f(A,B,C) = m_0 + m_1 + m_5 + m_6 + m_7$$

skraćeno se može zapisati na sljedeći način:

$$f(A,B,C) = \sum(0,1,5,6,7).$$

Logička funkcija tri varijable definirana produktom maksterma:

$$f(A,B,C) = M_2 \cdot M_3 \cdot M_4$$

skraćeno se može zapisati na sljedeći način:

$$f(A,B,C) = \prod(2,3,4).$$

☞ Primjer 3.2.1

Zadana je logička funkcija $f(A,B,C)$ u obliku tablice kombinacija (tablica 3.5).

Tablica 3.5: Tablica kombinacija za logičku funkciju $f(A,B,C)$

A	B	C	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a) Zapišite logičku funkciju $f(A,B,C)$ u obliku sume minterma.
b) Zapišite logičku funkciju $f(A,B,C)$ u obliku produkta maksterma.

c) Primjenom zakona logičke algebre pokažite da su obje logičke funkcije jednake.

✍ **Rješenje:** Za prvo ćemo razmatranje tablicu kombinacija 3.5 proširiti s mintermima i makstermima (tablica 3.6).

Tablica 3.6: Proširena tablica kombinacija za logičke funkcije $f(A,B,C)$

A	B	C	f	Mintermi	Makstermi
0	0	0	1	m_0	M_0
0	0	1	1	m_1	M_1
0	1	0	1	m_2	M_2
0	1	1	0	m_3	M_3
1	0	0	0	m_4	M_4
1	0	1	0	m_5	M_5
1	1	0	1	m_6	M_6
1	1	1	1	m_7	M_7

a) Logičku funkciju $f(A,B,C)$ definiraju mintermi za koje funkcija ima vrijednost 1. Prema tome, vrijedi sljedeća relacija:

$$f(A,B,C) = \sum(0,1,2,6,7) = m_0 + m_1 + m_2 + m_6 + m_7.$$

Za minterm m_0 logičke varijable imaju vrijednosti $A = 0$, $B = 0$ i $C = 0$. Za te vrijednosti varijabli minterm m_0 mora imati vrijednost 1 ($m_0 = 1$). Budući da je minterm umnožak logičkih varijabli, za minterm m_0 vrijedi sljedeće: $m_0 = \overline{A}\overline{B}\overline{C} = \bar{0} \cdot \bar{0} \cdot \bar{0} = 1 \cdot 1 \cdot 1 = 1$. Za minterm m_1 logičke varijable imaju vrijednosti $A = 0$, $B = 0$ i $C = 1$ pa je $m_1 = \overline{A}\overline{B}C = \bar{0} \cdot \bar{0} \cdot 1 = 1 \cdot 1 \cdot 1 = 1$. Na isti način dobit ćemo i ostale minterme.

Logička funkcija $f(A,B,C)$ definirana sumom minterma je:

$$f(A,B,C) = \sum(0,1,2,6,7) = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} + ABC.$$

Općenito, kod kreiranja minterma logička varijabla bit će invertirana ako je vrijednost varijable za taj minterm 0, a neće biti invertirana ako je vrijednost varijable za taj minterm 1. Nakon ovog razmatranja tablicu kombinacija nećemo više proširivati mintermima, već ćemo sumu minterma ispisati direktno iz tablice kombinacija.

b) Logičku funkciju $f(A,B,C)$ definiraju makstermi za koje funkcija ima vrijednost 0. Prema tome, vrijedi sljedeća relacija:

$$f(A,B,C) = \prod(3,4,5) = M_3 \cdot M_4 \cdot M_5.$$

Za maksterm M_3 logičke varijable imaju vrijednosti $A = 0$, $B = 1$ i $C = 1$. Za te vrijednosti varijabli maksterm M_3 mora imati vrijednost 0 ($M_3 = 0$). Budući da je maksterm suma logičkih varijabli, za maksterm M_3 vrijedi sljedeće: $M_3 = A + \overline{B} + \overline{C} = 0 + \bar{1} + \bar{1} = 0 + 0 + 0 = 0$. Za maksterm M_4 logičke varijable imaju vrijednosti $A = 1$, $B = 0$ i $C = 0$ pa je $M_4 = \overline{A} + B + C = \bar{1} + 0 + 0 = 0 + 0 + 0 = 0$. Na isti način dobit ćemo i ostale maksterme.

Logička funkcija $f(A,B,C)$ definirana produktom maksterma je:

$$f(A,B,C) = \prod(3,4,5) = (A + \overline{B} + \overline{C}) \cdot (\overline{A} + B + C) \cdot (\overline{A} + B + \overline{C}).$$

Općenito, kod kreiranja maksterma logička varijabla bit će invertirana ako je vrijednost varijable za taj maksterm 1, a neće biti invertirana ako je vrijednost varijable za taj maksterm

0. Nakon ovog razmatranja tablicu kombinacija nećemo više proširivati makstermima, već ćemo produkt maksterma ispisati direktno iz tablice kombinacija.

c) Primjenom teorema i aksioma Booleove algebre pokazat ćemo da vrijedi $f(A,B,C) = \sum(0,1,2,6,7) = \prod(3,4,5)$. Jednakost ćemo pokazati svođenjem produkta maksterma na sumu minterma:

$$
\begin{aligned}
& f(A,B,C) = \prod(3,4,5) = (A+\overline{B}+\overline{C}) \cdot (\overline{A}+B+C) \cdot (\overline{A}+B+\overline{C}) = \\
& = (\underbrace{A\overline{A}}_{0} + AB + AC + \overline{A}\,\overline{B} + \underbrace{\overline{B}B}_{0} + \overline{B}C + \overline{A}\,\overline{C} + B\overline{C} + \underbrace{\overline{C}C}_{0}) \cdot (\overline{A}+B+\overline{C}) \\
& = (AB + AC + \overline{A}\,\overline{B} + \overline{B}C + \overline{A}\,\overline{C} + B\overline{C}) \cdot (\overline{A}+B+\overline{C}) = \\
& = \underbrace{\overline{A}AB}_{0} + \underbrace{\overline{A}AC}_{0} + \overline{A}\,\overline{A}\,\overline{B} + \overline{A}\,\overline{B}C + \overline{A}\,\overline{A}\,\overline{C} + \overline{A}B\overline{C} + ABB + ABC + \underbrace{\overline{A}\,\overline{B}B}_{0} + \underbrace{\overline{B}BC}_{0} + \overline{A}B\overline{C} + \\
& + BB\overline{C} + AB\overline{C} + \underbrace{A\overline{C}C}_{0} + \overline{A}\,\overline{B}\,\overline{C} + \underbrace{\overline{B}\,\overline{C}C}_{0} + \overline{A}\,\overline{C}\,\overline{C} + B\overline{C}\,\overline{C} = \\
& = \overline{A}\,\overline{B} + \overline{A}\,\overline{B}C + \overline{A}\,\overline{C} + \overline{A}B\overline{C} + AB + ABC + \overline{A}B\overline{C} + B\overline{C} + AB\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{C} + B\overline{C} = \\
& = \overline{A}\,\overline{B}\underbrace{(1+C)}_{1} + \overline{A}\underbrace{(1+B)}_{1}\overline{C} + AB\underbrace{(1+C)}_{1} + \underbrace{(\overline{A}+1)}_{1}B\overline{C} + \underbrace{(A+1)}_{1}B\overline{C} + \overline{A}\underbrace{(\overline{B}+1)}_{1}\overline{C} = \\
& = \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + AB + B\overline{C} + B\overline{C} + \overline{A}\,\overline{C} = \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + AB + B\overline{C} = \\
& = \overline{A}\,\overline{B}(\overline{C}+C) + \overline{A}(\overline{B}+B)\overline{C} + AB(\overline{C}+C) + (\overline{A}+A)B\overline{C} = \\
& = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + AB\overline{C} + ABC + \overline{A}B\overline{C} + AB\overline{C} = \\
& = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} + ABC = \\
& = \sum(0,1,2,6,7).
\end{aligned}
$$

☞ Primjer 3.2.2

Zadana je logička funkcija $f(A,B,C)$:

$$f(A,B,C) = \sum(0,3,6,7).$$

a) Zapišite logičku funkciju $f(A,B,C)$ u obliku sume minterma.
b) Zapišite logičku funkciju $f(A,B,C)$ u obliku produkta maksterma.
c) Napišite tablicu kombinacija za logičku funkciju $f(A,B,C)$.

✍ Rješenje:

a) Logička funkcija $f(A,B,C)$ u obliku sume minterma ima sljedeći oblik:

$$f(A,B,C) = \sum(0,3,6,7) = m_0 + m_3 + m_6 + m_7 = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}BC + AB\overline{C} + ABC.$$

b) Logičku funkciju $f(A,B,C)$ definiraju mintermi za koje funkcija ima vrijednost 1, odnosno makstermi za koje funkcija ima vrijednost 0. U skraćenom zapisu funkcije u obliku sume minterma nalaze se indeksi minterma za koje logička funkcija ima vrijednost 1. Prema tome, skraćeni zapis funkcije u obliku produkta maksterma imat će sve indekse koji nisu sadržani u skraćenom zapisu funkcije pomoću sume minterma:

$$f(A,B,C) = \sum(0,3,6,7) = \prod(1,2,4,5).$$

Logička funkcija $f(A,B,C)$ u obliku produkta maksterma ima sljedeći oblik:

$$f(A,B,C) = \prod(1,2,4,5) = M_1 \cdot M_2 \cdot M_4 \cdot M_5 = (A+B+\overline{C}) \cdot (A+\overline{B}+C) \cdot (\overline{A}+B+C) \cdot (\overline{A}+B+\overline{C}).$$

c) Tablica kombinacija za logičku funkciju $f(A,B,C) = \sum(0,3,6,7) = \prod(1,2,4,5)$ prikazana je tablicom 3.7. U tablici 3.7 dodan je stupac s indeksima minterma, odnosno maksterma kako bi lakše bilo popuniti tablicu. Logička funkcija $f(A,B,C)$ ima vrijednost 1 za indekse 0, 3, 6 i 7, a vrijednost 0 za indekse 1, 2, 4 i 5. Tablica 3.7 može se popuniti direktno ili na temelju sume minterma ili na temelju produkta maksterma.

Tablica 3.7: Tablica kombinacija za logičku funkciju $f(A,B,C) = \sum(0,3,6,7) = \prod(1,2,4,5)$

i	A	B	C	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

☞ Primjer 3.2.3

Zadana je logička funkcija $f(A,B,C,D)$:

$$f(A,B,C,D) = \prod(3,4,5,9,10,11,15).$$

a) Zapišite logičku funkciju $f(A,B,C,D)$ u obliku produkta maksterma.
b) Zapišite logičku funkciju $f(A,B,C,D)$ u obliku sume minterma.
c) Napišite tablicu kombinacija za logičku funkciju $f(A,B,C,D)$.

✍ Rješenje:

a) Logička funkcija $f(A,B,C,D)$ u obliku produkta maksterma ima sljedeći oblik:

$$\begin{aligned} &f(A,B,C,D) = \prod(3,4,5,9,10,11,15) = M_3 \cdot M_4 \cdot M_5 \cdot M_9 \cdot M_{10} \cdot M_{11} \cdot M_{15} = \\ &= (A + B + \overline{C} + \overline{D}) \cdot (A + \overline{B} + C + D) \cdot (A + \overline{B} + C + \overline{D}) \cdot (\overline{A} + B + C + \overline{D}) \cdot \\ &\cdot (\overline{A} + B + \overline{C} + D) \cdot (\overline{A} + B + \overline{C} + \overline{D}) \cdot (\overline{A} + \overline{B} + \overline{C} + \overline{D}). \end{aligned}$$

b) U skraćenom zapisu funkcije u obliku produkta maksterma nalaze se indeksi maksterma za koje logička funkcija ima vrijednost 0. Prema tome, skraćeni zapis funkcije pomoću sume minterma imat će sve indekse koji nisu sadržani u skraćenom zapisu funkcije pomoću produkta maksterma:

$$f(A,B,C,D) = \prod(3,4,5,9,10,11,15) = \sum(0,1,2,6,7,8,12,13,14).$$

Logička funkcija $f(A,B,C,D)$ u obliku sume minterma ima sljedeći oblik:

$$\begin{aligned} &f(A,B,C,D) = \sum(0,1,2,6,7,8,12,13,14) = m_0 + m_1 + m_2 + m_6 + m_7 + m_8 + m_{12} + m_{13} + m_{14} = \\ &= \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}C\overline{D} + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D}. \end{aligned}$$

c) Tablica kombinacija za logičku funkciju $f(A,B,C,D) = \prod(3,4,5,9,10,11,15) = \sum(0,1,2,6,7,8,12,13,14)$ prikazana je tablicom 3.8. U tablici 3.8 dodan je stupac s indeksima minterma, odnosno maksterma kako bi lakše bilo popuniti tablicu. Logička funkcija $f(A,B,C,D)$ ima vrijednost 1 za indekse 0, 1, 2, 6, 7, 8, 12, 13 i 14, a vrijednost 0 za indekse 3, 4, 5, 9, 10, 11 i 15. Tablica 3.8 može se popuniti direktno ili na temelju sume minterma ili na temelju produkta maksterma.

Tablica 3.8: Tablica kombinacija za logičku funkciju
$f(A,B,C,D) = \prod(3,4,5,9,10,11,15) = \sum(0,1,2,6,7,8,12,13,14)$

i	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

☞ Primjer 3.2.4

Zadana je logička funkcija $f(A,B,C)$:

$$f(A,B,C) = B + AC.$$

a) Zapišite logičku funkciju $f(A,B,C)$ u obliku sume minterma.
b) Zapišite logičku funkciju $f(A,B,C)$ u obliku produkta maksterma.
c) Napišite tablicu kombinacija za logičku funkciju $f(A,B,C)$.

✍ **Rješenje:** a) Logička funkcija $f(A,B,C) = B + AC$ nije napisana u standardnom kanonskom obliku. Potrebno ju je proširiti na kanonski oblik koristeći zakone Booleove algebre (aksiom o komplementu $\overline{A} + A = 1$ i zakon distribucije $A \cdot (B + C) = AB + AC$):

$$\begin{aligned}
&f(A,B,C) = B + AC = 1 \cdot B \cdot 1 + A \cdot 1 \cdot C = (\overline{A} + A) \cdot B \cdot (\overline{C} + C) + A \cdot (\overline{B} + B) \cdot C = \\
&= \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C} + ABC + A\overline{B}C + ABC = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = \\
&= \sum(2,3,5,6,7).
\end{aligned}$$

Dobiveni izraz predstavlja zapis logičke funkcije $f(A,B,C)$ u obliku sume minterma.

b) Logička funkcija $f(A,B,C) = B + AC$ može se proširiti na kanonski oblik koristeći zakone Booleove algebre (aksiom o komplementu $\overline{A}A = 0$ i zakon distribucije $A{+}BC = (A{+}B){\cdot}(A{+}C)$):

$$\begin{aligned}
&f(A,B,C) = B + AC = (B + A)(B + C) = (A + B + 0)(0 + B + C) = \\
&= (A + B + \overline{C}C)(\overline{A}A + B + C) = \big((A + B + \overline{C}) \cdot (A + B + C)\big)\big((\overline{A} + B + C) \cdot (A + B + C)\big) = \\
&= (A + B + C) \cdot (A + B + \overline{C}) \cdot (\overline{A} + B + C) = \prod(0,1,4)
\end{aligned}$$

Dobiveni izraz predstavlja zapis logičke funkcije $f(A,B,C)$ u obliku produkta maksterma.

c) Tablica kombinacija za logičku funkciju $f(A,B,C) = \sum(2,3,5,6,7) = \prod(0,1,4)$ prikazana je tablicom 3.9. U tablici 3.9 dodan je stupac s indeksima minterma, odnosno maksterma kako bi lakše bilo popuniti tablicu. Logička funkcija $f(A,B,C)$ ima vrijednost 1 za indekse 2, 3, 5, 6 i 7, a vrijednost 0 za indekse 0, 1, i 4. Tablica 3.9 može se popuniti direktno ili na temelju sume minterma ili na temelju produkta maksterma.

Tablica 3.9: Tablica kombinacija za logičku funkciju $f(A,B,C) = \sum(2,3,5,6,7) = \prod(0,1,4)$

i	A	B	C	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

3.2.1 Zadaci za vježbu

✎ Zadatak 3.2.1

Zadana je logička funkcija $f(A,B,C)$ u obliku tablice kombinacija (tablica 3.10).

Tablica 3.10: Tablica kombinacija za logičke funkcije $f(A,B,C)$

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

a) Zapišite logičku funkciju $f(A,B,C)$ u obliku sume minterma.
b) Zapišite logičku funkciju $f(A,B,C)$ u obliku produkta maksterma.
c) Primjenom zakona logičke algebre pokažite da su obje logičke funkcije jednake.

✎ Zadatak 3.2.2

Zadana je logička funkcija $f(A,B,C)$:

$$f(A,B,C) = \sum(1,2,4,5).$$

a) Zapišite logičku funkciju $f(A,B,C)$ u obliku sume minterma.
b) Zapišite logičku funkciju $f(A,B,C)$ u obliku produkta maksterma.
c) Napišite tablicu kombinacija za logičku funkciju $f(A,B,C)$.

✎ Zadatak 3.2.3

Zadana je logička funkcija $f(A,B,C,D)$:

$$f(A,B,C,D) = \prod(0,1,2,6,7,8,12,13,14).$$

a) Zapišite logičku funkciju $f(A,B,C,D)$ u obliku produkta maksterma.
b) Zapišite logičku funkciju $f(A,B,C,D)$ u obliku sume minterma.
c) Napišite tablicu kombinacija za logičku funkciju $f(A,B,C,D)$.

✎ Zadatak 3.2.4

Zadana je logička funkcija $f(A,B,C)$:

$$f(A,B,C) = \overline{A}B + AC.$$

a) Zapišite logičku funkciju $f(A,B,C)$ u obliku sume minterma.
b) Zapišite logičku funkciju $f(A,B,C)$ u obliku produkta maksterma.
c) Napišite tablicu kombinacija za logičku funkciju $f(A,B,C)$.

3.3 Komplementarne i dualne funkcije

Logička funkcija čija je vrijednost komplementarna funkciji f zove se komplementarna funkcija. Za logičku funkciju zadanu sumom minterma:

$$f = \sum_{i=0}^{2^n-1} v_i m_i \tag{3.15}$$

komplementarna funkcija $\overline{f}$ ima oblik:

$$\overline{f} = \sum_{i=0}^{2^n-1} \overline{v_i} m_i. \tag{3.16}$$

Za logičku funkciju zadanu produktom maksterma:

$$f = \prod_{i=0}^{2^n-1} (v_i + M_i) \tag{3.17}$$

komplementarna funkcija $\overline{f}$ ima oblik:

$$\overline{f} = \prod_{i=0}^{2^n-1} (\overline{v_i} + M_i). \tag{3.18}$$

U izrazima (3.15) do (3.18) n je broj logičkih varijabli funkcija f i $\overline{f}$. Ako u logičkoj funkciji f zamijenimo operatore $\cdot$ i $+$ te vrijednosti konstanti 0 i 1 (ako ih ima), dobit ćemo dualnu funkciju f_D. Na primjer, za funkciju $f(A,B,C) = \overline{A}B + B\overline{C}$ dualna je funkcija $f_D(A,B,C) = (\overline{A}+B)\cdot(B+\overline{C})$. Izraz za dualnu funkciju koristi se za jednostavnije dobivanje komplementarne funkcije:

$$\overline{f}(A,B,C,...) = f_D(\overline{A},\overline{B},\overline{C},...). \tag{3.19}$$

☞ Primjer 3.3.1

Zadana je logička funkcija $f(A,B,C)$:

$$f(A,B,C) = \overline{A}\,\overline{B}C + AB\overline{C} + ABC$$

a) Nađite komplementarnu funkciju prema izrazu (3.16).
b) Nađite komplementarnu funkciju komplementiranjem zadane funkcije.
c) Nađite komplementarnu funkciju pomoću dualne funkcije.

✍ **Rješenje:** a) Raspišimo funkciju $f(A,B,C)$ prema izrazu (3.15):

$$\begin{aligned} f(A,B,C) &= \overline{A}\,\overline{B}C + AB\overline{C} + ABC = m_1 + m_6 + m_7 = \\ &= 0\cdot m_0 + 1\cdot m_1 + 0\cdot m_2 + 0\cdot m_3 + 0\cdot m_4 + 0\cdot m_5 + 1\cdot m_6 + 1\cdot m_7 \end{aligned}$$

Komplementarnu funkciju $\overline{f}(A,B,C)$ dobit ćemo pomoću izraza (3.16):

$$\begin{aligned} \overline{f}(A,B,C) &= \overline{0}\cdot m_0 + \overline{1}\cdot m_1 + \overline{0}\cdot m_2 + \overline{0}\cdot m_3 + \overline{0}\cdot m_4 + \overline{0}\cdot m_5 + \overline{1}\cdot m_6 + \overline{1}\cdot m_7 = \\ &= 1\cdot m_0 + 0\cdot m_1 + 1\cdot m_2 + 1\cdot m_3 + 1\cdot m_4 + 1\cdot m_5 + 0\cdot m_6 + 0\cdot m_7 = \\ &= m_0 + m_2 + m_3 + m_4 + m_5 = \sum(0,2,3,4,5). \end{aligned}$$

b) Ako komplementiramo zadanu logičku funkciju $f(A,B,C)$, dobit ćemo komplementarnu funkciju $\overline{f}(A,B,C)$:

$$\overline{f}(A,B,C) = \overline{f(A,B,C)} = \overline{\overline{A}\,\overline{B}C + AB\overline{C} + ABC}.$$

Prema generaliziranom De Morganovom teoremu (3.11) vrijedi:

$$\begin{aligned} \overline{f}(A,B,C) &= \overline{f(A,B,C)} = \overline{\overline{A}\,\overline{B}C + AB\overline{C} + ABC} = \overline{\overline{A}\,\overline{B}C}\cdot\overline{AB\overline{C}}\cdot\overline{ABC} = \\ &= (\overline{\overline{A}} + \overline{\overline{B}} + \overline{C})\cdot(\overline{A} + \overline{B} + \overline{\overline{C}})\cdot(\overline{A} + \overline{B} + \overline{C}) = (A + B + \overline{C})\cdot(\overline{A} + \overline{B} + C)\cdot(\overline{A} + \overline{B} + \overline{C}) = \\ &= M_1\cdot M_6\cdot M_7 = \prod(1,6,7) = \sum(0,2,3,4,5). \end{aligned}$$

Rezultat koji smo dobili jednak je rezultatu dobivenom u a) dijelu primjera, no zapisan u obliku produkta maksterma.

c) Dualnu funkciju $f_D(A,B,C)$ dobit ćemo tako da zamijenimo operatore $\cdot$ i $+$ u logičkoj funkciji $f(A,B,C)$:

$$f_D(A,B,C) = (\overline{A} + \overline{B} + C)\cdot(A + B + \overline{C})\cdot(A + B + C).$$

Ako u dualnoj funkciji $f_D(A,B,C)$ komplementiramo varijable, dobit ćemo komplementarnu funkciju $\overline{f}(A,B,C)$:

$$\begin{aligned} \overline{f}(A,B,C) &= f_D(\overline{A},\overline{B},\overline{C}) = (\overline{\overline{A}} + \overline{\overline{B}} + \overline{C})\cdot(\overline{A} + \overline{B} + \overline{\overline{C}})\cdot(\overline{A} + \overline{B} + \overline{C}) = \\ &= (A + B + \overline{C})\cdot(\overline{A} + \overline{B} + C)\cdot(\overline{A} + \overline{B} + \overline{C}) = M_1\cdot M_6\cdot M_7 = \prod(1,6,7) = \sum(0,2,3,4,5). \end{aligned}$$

Rezultat koji smo dobili jednak je rezultatu dobivenom u b) dijelu primjera.

3.3.1 Zadaci za vježbu

✎ Zadatak 3.3.1

Zadana je logička funkcija $f(A,B,C)$:

$$f(A,B,C) = \overline{A}B\overline{C} + A\overline{B}C + A\overline{B}C + ABC$$

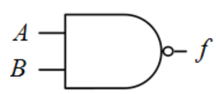
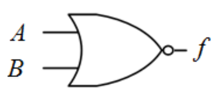
a) Nađite komplementarnu funkciju prema izrazu (3.16).
b) Nađite komplementarnu funkciju komplementiranjem zadane funkcije.
c) Nađite komplementarnu funkciju pomoću dualne funkcije.

3.4 Osnovne logičke funkcije i univerzalne logičke funkcije

Svaka logička funkcija može se prikazati u kanonskom obliku korištenjem tri logička operatora: logičkog zbrajanja, logičkog množenja i komplementiranja. Primitivne ili osnovne funkcije one su funkcije pomoću kojih se mogu prikazati sve ostale funkcije. To ne moraju biti samo spomenute tri funkcije, već bilo koja grupa funkcija pomoću kojih se mogu prikazati sve ostale. Primjeri skupova osnovnih funkcija su:

- I, ILI, NE,
- I, NE,
- ILI, NE,
- NI,
- NILI.

Logičke funkcije NI i NILI univerzalne su funkcije jer se pomoću njih mogu prikazati sve ostale logičke funkcije. Univerzalne logičke funkcije, njihove tablice kombinacija, simboli i Booleovi izrazi prikazani su na slici 3.2.

<table>
<tr><th>Operator</th><th>Tablica kombinacija</th><th>Simbol</th><th>Booleov izraz</th></tr>
<tr><td>NI</td><td><table><tr><th>A</th><th>B</th><th>f</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td></td><td>$f = \overline{A \cdot B}$</td></tr>
<tr><td>NILI</td><td><table><tr><th>A</th><th>B</th><th>f</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td></td><td>$f = \overline{A + B}$</td></tr>
</table>

Slika 3.2: Univerzalne logičke funkcije NI i NILI

Osnovne logičke funkcije i univerzalne logičke funkcije ostvaruju se istoimenim sklopovima (npr. NI logička funkcija ostvaruje se NI sklopom).

☞ Primjer 3.4.1

Izvedite sljedeće osnovne logičke funkcije pomoću NI logičkih sklopova:
a) NE - $f = \overline{A}$,
b) I - $f = AB$,
c) ILI - $f = A + B$.

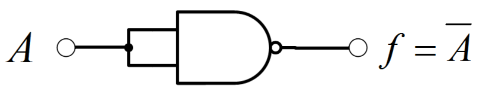
✍ Rješenje:

Za sve izvode u ovom primjeru koristit će se De Morganov zakon (3.10) i osnovni zakoni Booleove algebre.

a) NE sklop ima jedan ulaz, dok NI sklop ima dva ili više ulaza. Za NE logičku funkciju vrijedi:

$$f = \overline{A} = \overline{AA}.$$

NE logičku funkciju moguće je ostvariti pomoću jednog NI sklopa čiji su ulazi kratko spojeni (slika 3.3).

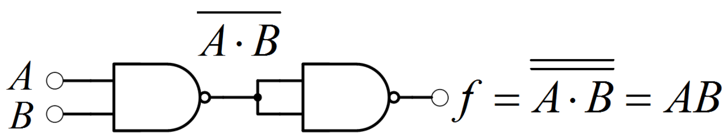


Slika 3.3: NE logička funkcija izvedena NI sklopom

b) Za I logičku funkciju vrijedi:

$$f = AB = \overline{\overline{AB}}.$$

I logičku funkciju moguće je ostvariti pomoću dva NI sklopa (slika 3.4).

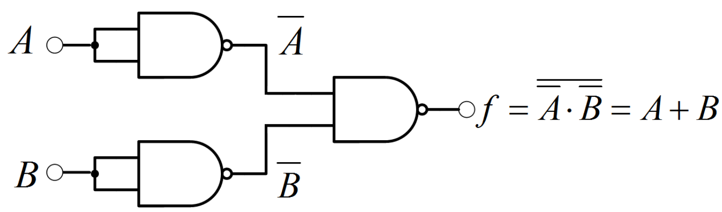


Slika 3.4: I logička funkcija izvedena NI sklopovima

c) Za ILI logičku funkciju vrijedi:

$$f = A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}}.$$

ILI logičku funkciju moguće je ostvariti pomoću tri NI sklopa (slika 3.5).



Slika 3.5: ILI logička funkcija izvedena NI sklopovima

☞ Primjer 3.4.2

Izvedite sljedeće osnovne logičke funkcije pomoću NILI logičkih sklopova:
a) NE - $f = \overline{A}$,
b) I - $f = AB$,
c) ILI - $f = A + B$.

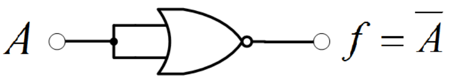
✍ Rješenje:

Za sve izvode u ovom primjeru koristit će se De Morganov zakon (3.10) i osnovni zakoni Booleove algebre.

a) NE sklop ima jedan ulaz, dok NILI sklop ima dva ili više ulaza. Za NE logičku funkciju vrijedi:

$$f = \overline{A} = \overline{A + A}.$$

NE logičku funkciju moguće je ostvariti pomoću jednog NILI sklopa čiji su ulazi kratko spojeni (slika 3.6).

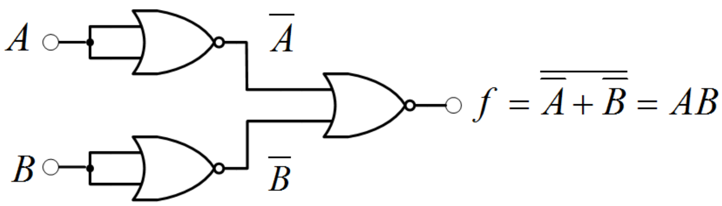


Slika 3.6: NE logička funkcija izvedena NILI sklopom

b) Za I logičku funkciju vrijedi:

$$f = AB = \overline{\overline{AB}} = \overline{\overline{A} + \overline{B}}.$$

I logičku funkciju moguće je ostvariti pomoću tri NILI sklopa (slika 3.7).

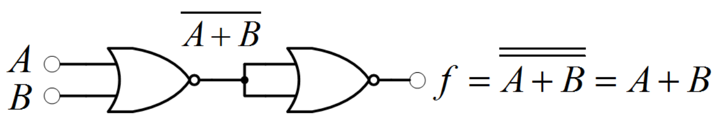


Slika 3.7: I logička funkcija izvedena NILI sklopovima

c) Za ILI logičku funkciju vrijedi:

$$f = A + B = \overline{\overline{A + B}}.$$

ILI logičku funkciju moguće je ostvariti pomoću dva NILI sklopa (slika 3.8).



Slika 3.8: ILI logička funkcija izvedena NILI sklopovima

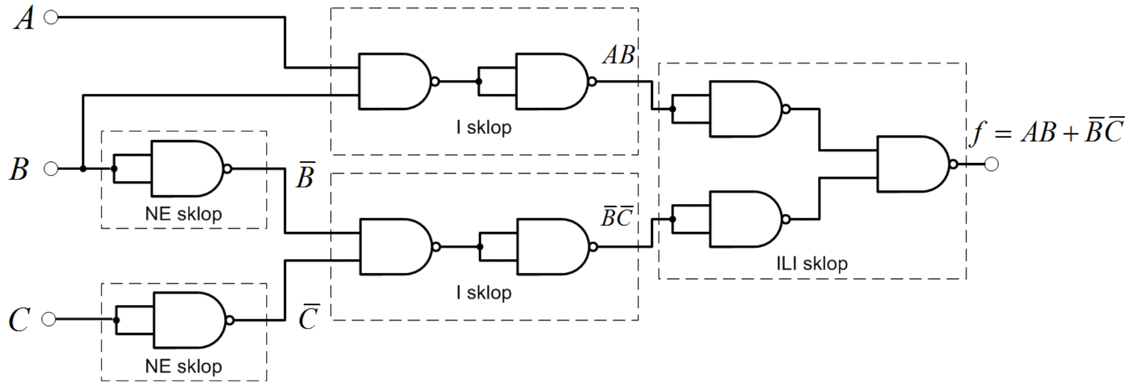
☞ Primjer 3.4.3

Zadana je logička funkcija:

$$f(A,B,C) = AB + \overline{B}\overline{C}.$$

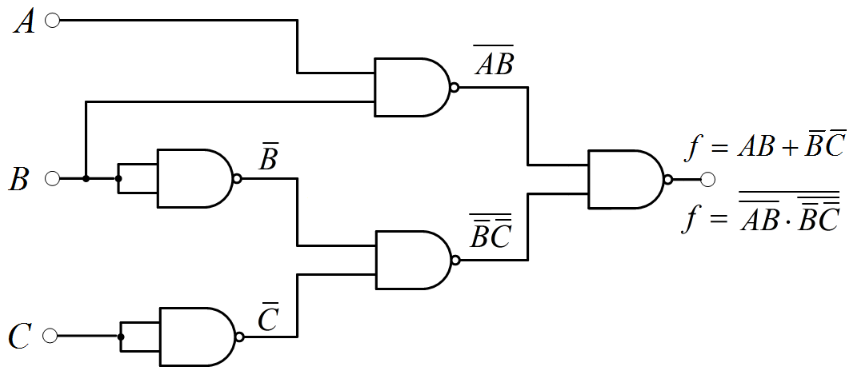
a) Realizirajte funkciju $f(A,B,C)$ pomoću NI sklopova.
b) Realizirajte funkciju $f(A,B,C)$ pomoću NILI sklopova.

✍ **Rješenje:** a) Logička funkcija $f(A,B,C) = AB + \overline{B}\overline{C}$ sklopovski se može realizirati pomoću dva NE sklopa, dva I sklopa i jednog ILI sklopa. Navedene sklopove potrebno je zamijeniti sklopovima sa slika 3.3, 3.4 i 3.5. Logička funkcija $f(A,B,C) = AB + \overline{B}\overline{C}$ realizirana NI sklopovima prikazana je na slici 3.9.



Slika 3.9: Logička funkcija $f(A,B,C) = AB + \overline{B}\overline{C}$ realizirana NI sklopovima

Na slici 3.9 postoje serijski povezani NI sklopovi u funkciji NE sklopa. Prema teoremu (3.7) ($\overline{\overline{A}} = A$), dva serijski povezana NI sklopa u funkciji NE sklopa mogu se izbaciti pa se dobije sklop sa slike 3.10.



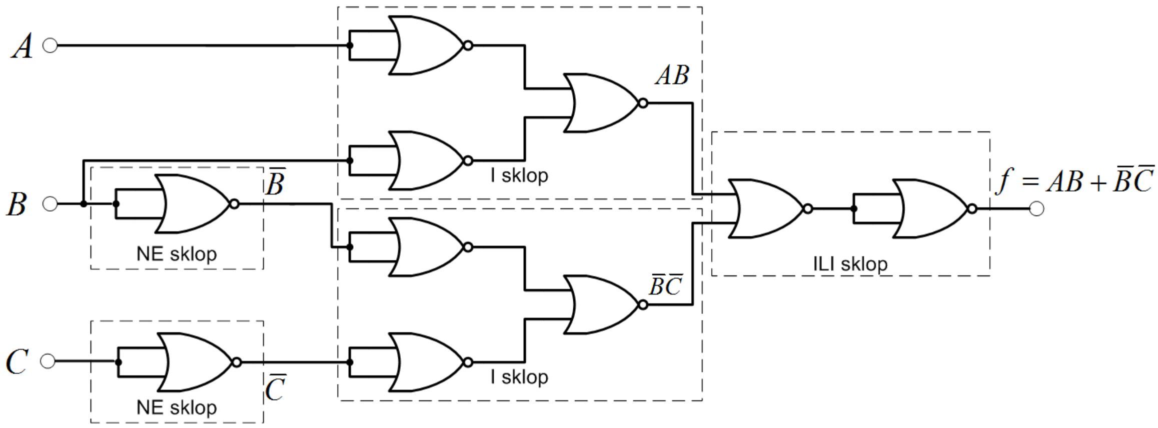
Slika 3.10: Logička funkcija $f(A,B,C) = AB + \overline{B}\overline{C}$ realizirana NI sklopovima - reducirani sklop

Primjenom De Morganovog teorema na logičku funkciju $f(A,B,C)$ dobit ćemo:

$$f(A,B,C) = \overline{\overline{f(A,B,C)}} = \overline{\overline{AB + \overline{B}\overline{C}}} = \overline{\overline{AB} \cdot \overline{\overline{B}\overline{C}}}.$$

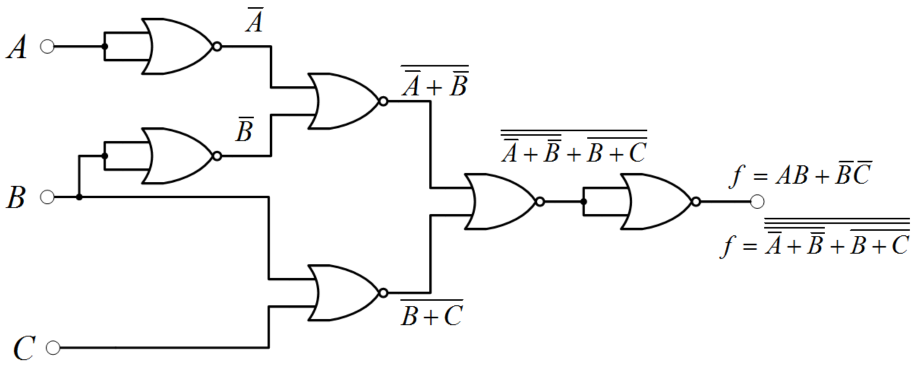
Dobiveni izraz za logičku funkciju $f(A,B,C)$ može se realizirati pomoću 5 NI sklopova, a realizacija je ista kao na slici 3.10.

b) Logička funkcija $f(A,B,C) = AB + \overline{B}\overline{C}$ sklopovski se može realizirati pomoću dva NE sklopa, dva I sklopa i jednog ILI sklopa. Navedene sklopove potrebno je zamijeniti sklopovima sa slika 3.6, 3.7 i 3.8. Logička funkcija $f(A,B,C) = AB + \overline{B}\,\overline{C}$ realizirana NILI sklopovima prikazana je na slici 3.11.



Slika 3.11: Logička funkcija $f(A,B,C) = AB + \overline{B}\,\overline{C}$ realizirana NILI sklopovima

Na slici 3.11 postoje serijski povezani NILI sklopovi u funkciji NE sklopa. Prema teoremu (3.7) ($\overline{\overline{A}} = A$), dva serijski povezana NILI sklopa u funkciji NE sklopa mogu se izbaciti pa se dobije sklop sa slike 3.12.



Slika 3.12: Logička funkcija $f(A,B,C) = AB + \overline{B}\,\overline{C}$ realizirana NILI sklopovima - reducirani sklop

Primjenom De Morganovog teorema na logičku funkciju $f(A,B,C)$ dobit ćemo:

$$f(A,B,C) = \overline{\overline{f(A,B,C)}} = \overline{\overline{AB + \overline{B}\,\overline{C}}} = \overline{\overline{AB} \cdot \overline{\overline{B}\,\overline{C}}} = \overline{\overline{\overline{A} + \overline{B}} + \overline{B + C}}.$$

Dobiveni izraz za logičku funkciju $f(A,B,C)$ može se realizirati pomoću 5 NILI sklopova, a realizacija je ista kao na slici 3.12.

☞ Primjer 3.4.4

Zadana je logička funkcija (EX ILI (isključivo ILI)):

$$f(A,B) = A \oplus B.$$

a) Realizirajte funkciju $f(A,B)$ pomoću osnovnih logičkih sklopova.
b) Realizirajte funkciju $f(A,B)$ pomoću NI sklopova.

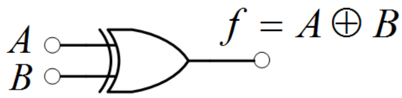
✍ **Rješenje:**

Tablica kombinacija logičke funkcije EX ILI prikazana je tablicom 3.11. Logička funkcija EX ILI ima logičku vrijednost 1 kada su logičke varijable A i B različite.

Tablica 3.11: Tablica kombinacija logičke funkcije EX ILI

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

Simbol logičkog sklopa EX ILI prikazan je na slici 3.13.

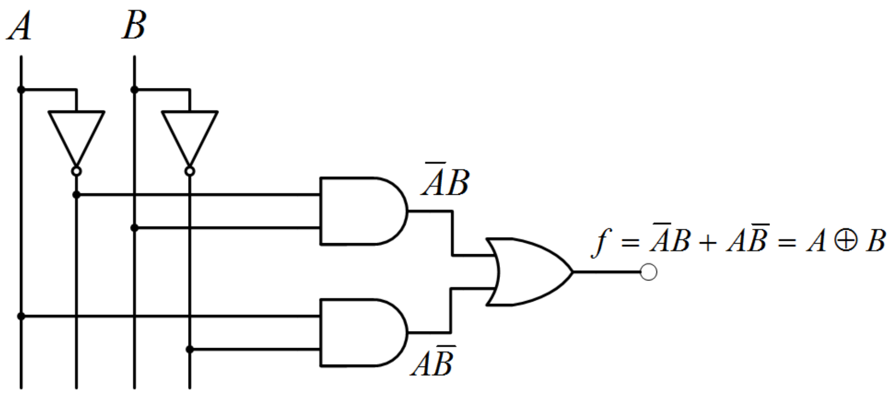


Slika 3.13: Simbol logičkog sklopa EX ILI

a) Da bi logičku funkciju $f(A,B) = A \oplus B$ realizirali osnovnim logičkim sklopovima, potrebno ju je zapisati u obliku sume minterma (ili produkta maksterma). Prema tablici 3.11, logička se funkcija $f(A,B)$ može opisati sumom minterma na sljedeći način:

$$f(A,B) = A \oplus B = \overline{A}B + A\overline{B}.$$

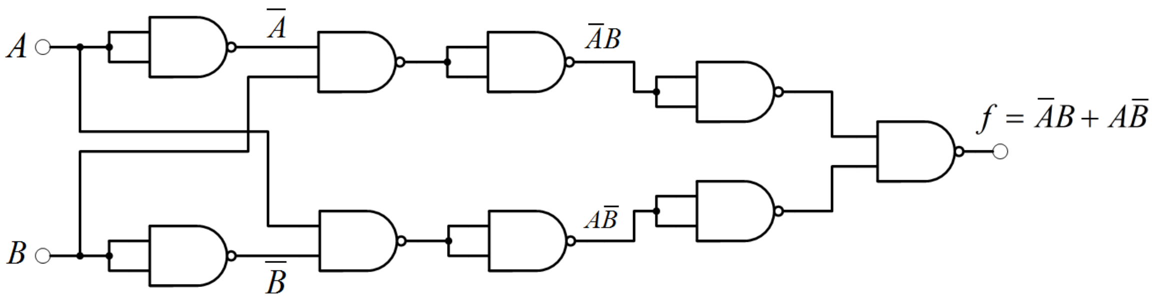
Logička funkcija $f(A,B) = A \oplus B$ realizirana osnovnim logičkim sklopovima prikazana je na slici 3.14.



Slika 3.14: Logička funkcija $f(A,B) = A \oplus B$ realizirana osnovnim logičkim sklopovima

b) Logička funkcija $f(A,B) = A \oplus B = \overline{A}B + A\overline{B}$ sklopovski se može realizirati pomoću dva NE sklopa, dva I sklopa i jednog ILI sklopa. Navedene sklopove potrebno je zamijeniti sklopovima sa slika 3.3, 3.4 i 3.5. Logička funkcija $f(A,B) = A \oplus B$ realizirana NI sklopovima prikazana je na slici 3.15.

Na slici 3.15 postoje serijski povezani NI sklopovi u funkciji NE sklopa. Prema teoremu (3.7) ($\overline{\overline{A}} = A$), dva serijski povezana NI sklopa u funkciji NE sklopa mogu se izbaciti pa se dobije sklop sa slike 3.16.

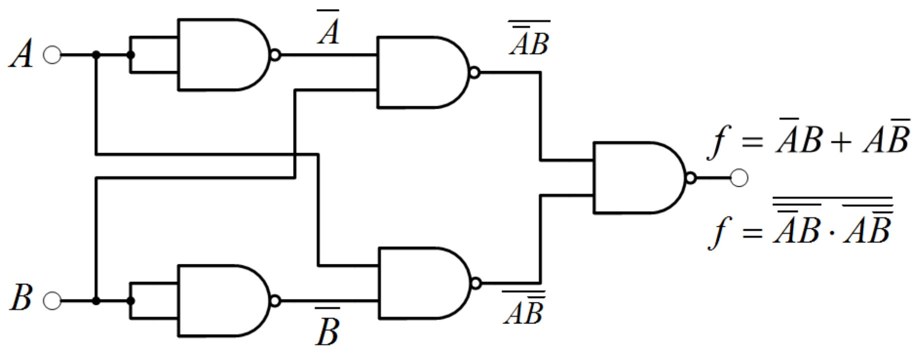


Slika 3.15: Logička funkcija $f(A,B) = A \oplus B$ realizirana NI sklopovima

Primjenom De Morganovog teorema na logičku funkciju $f(A,B) = A \oplus B$ dobit ćemo:

$$f(A,B) = \overline{\overline{f(A,B)}} = \overline{\overline{A \oplus B}} = \overline{\overline{\overline{A}B + A\overline{B}}} = \overline{\overline{\overline{A}B} \cdot \overline{A\overline{B}}}.$$

Dobiveni izraz za logičku funkciju $f(A,B)$ može se realizirati pomoću 5 NI sklopova, a realizacija je ista kao na slici 3.16.



Slika 3.16: Logička funkcija $f(A,B) = A \oplus B$ realizirana NI sklopovima - reducirani sklop

☞ Primjer 3.4.5

Zadana je logička funkcija:

$$f(A,B,C) = \big(A(A + \overline{B})\big) \oplus C$$

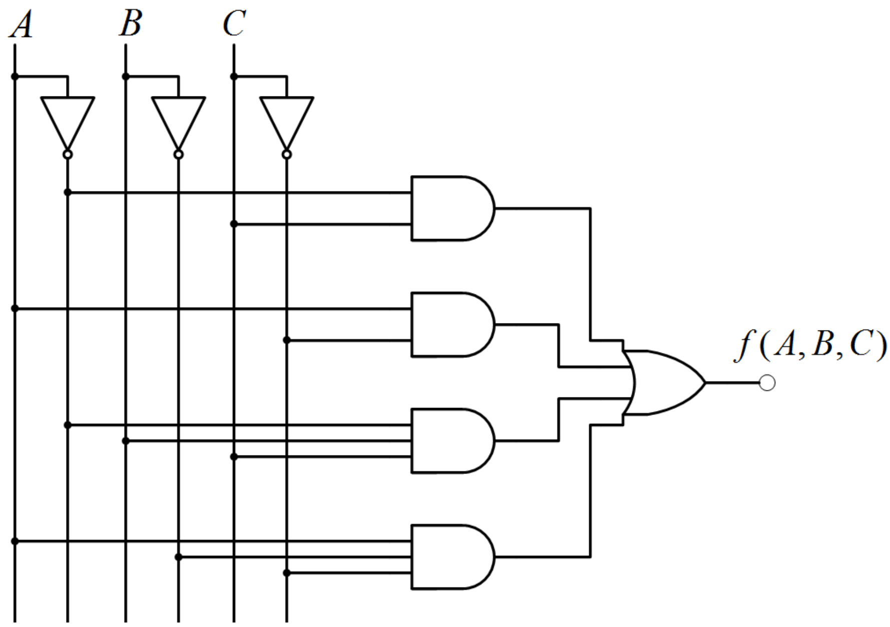
a) Realizirajte funkciju $f(A,B,C)$ pomoću osnovnih logičkih sklopova.
b) Realizirajte funkciju $f(A,B,C)$ pomoću NI sklopova.

✍ Rješenje:

a) Da bi zadanu logičku funkciju $f(A,B,C)$ realizirali osnovnim logičkim sklopovima, potrebno ju je pomoću zakona Booleove algebre svesti na samo osnovne logičke funkcije:

$$\begin{aligned} f(A,B,C) &= \big(A(A + \overline{B})\big) \oplus C = \big(A + A\overline{B}\big) \oplus C = \overline{A + A\overline{B}} \cdot C + \big(A + A\overline{B}\big)\,\overline{C} = \\ &= \overline{A} \cdot \overline{A\overline{B}} \cdot C + A\overline{C} + A\overline{B}\,\overline{C} = \overline{A} \cdot \big(\overline{A} + B\big) \cdot C + A\overline{C} + A\overline{B}\,\overline{C} = \overline{A}C + A\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C}. \end{aligned}$$

Logička funkcija $f(A,B,C)$ realizirana osnovnim logičkim sklopovima prikazana je na slici 3.17.

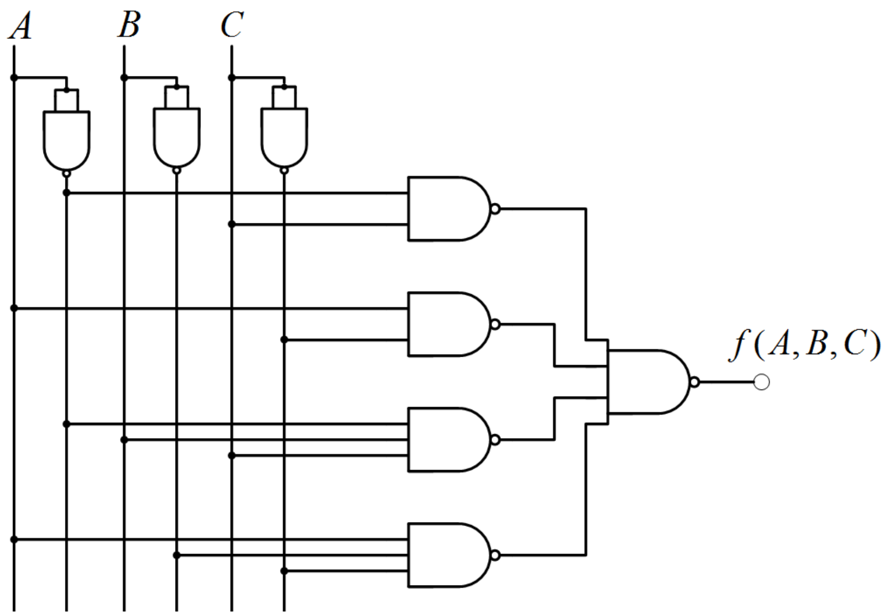


Slika 3.17: Logička funkcija $f(A,B,C) = \big(A(A+\overline{B})\big) \oplus C$ realizirana osnovnim logičkim sklopovima

b) Primjenom De Morganovog teorema na logičku funkciju $f(A,B,C) = \big(A(A+\overline{B})\big) \oplus C = \overline{A}C + A\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C}$ dobit ćemo:

$$f(A,B,C) = \overline{\overline{f(A,B,C)}} = \overline{\overline{\overline{A}C + A\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C}}} = \overline{\overline{\overline{A}C} \cdot \overline{A\overline{C}} \cdot \overline{\overline{A}BC} \cdot \overline{A\overline{B}\,\overline{C}}}.$$

Dobiveni izraz za logičku funkciju $f(A,B,C)$ može se realizirati NI sklopovima, a realizacija je prikazana na slici 3.18.



Slika 3.18: Logička funkcija $f(A,B,C) = \big(A(A+\overline{B})\big) \oplus C$ realizirana NI sklopovima

☞ Primjer 3.4.6

Logička funkcija četiri varijable $f(A,B,C,D)$ na izlazu mora dati vrijednost 1 ako je na ulazu paran broj nula.
a) Za logičku funkciju $f(A,B,C,D)$ napišite tablicu kombinacija.
b) Logičku funkciju $f(A,B,C,D)$ zapišite u obliku sume minterma i izvedite je osnovnim logičkim sklopovima.
c) Logičku funkciju $f(A,B,C,D)$ zapisanu u obliku sume minterma prilagodite i izvedite NI sklopovima.

✍ Rješenje:

a) Tablica kombinacija za logičku funkciju $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je na ulazu paran broj nula prikazana je tablicom 3.12.

Tablica 3.12: Tablica kombinacija za logičku funkciju $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je na ulazu paran broj nula

i	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

b) Logička funkcija $f(A,B,C,D)$ prikazana tablicom kombinacija 3.12 u obliku sume minterma je:

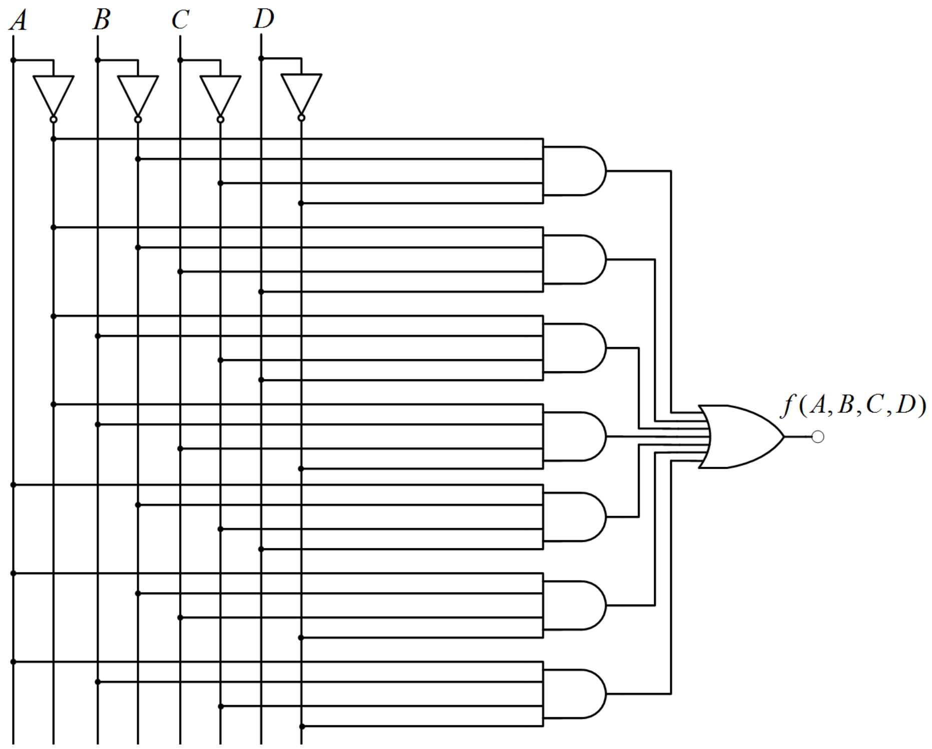
$$f(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + AB\overline{C}\,\overline{D}$$

Logička funkcija $f(A,B,C,D)$ realizirana osnovnim logičkim sklopovima prikazana je na slici 3.21.

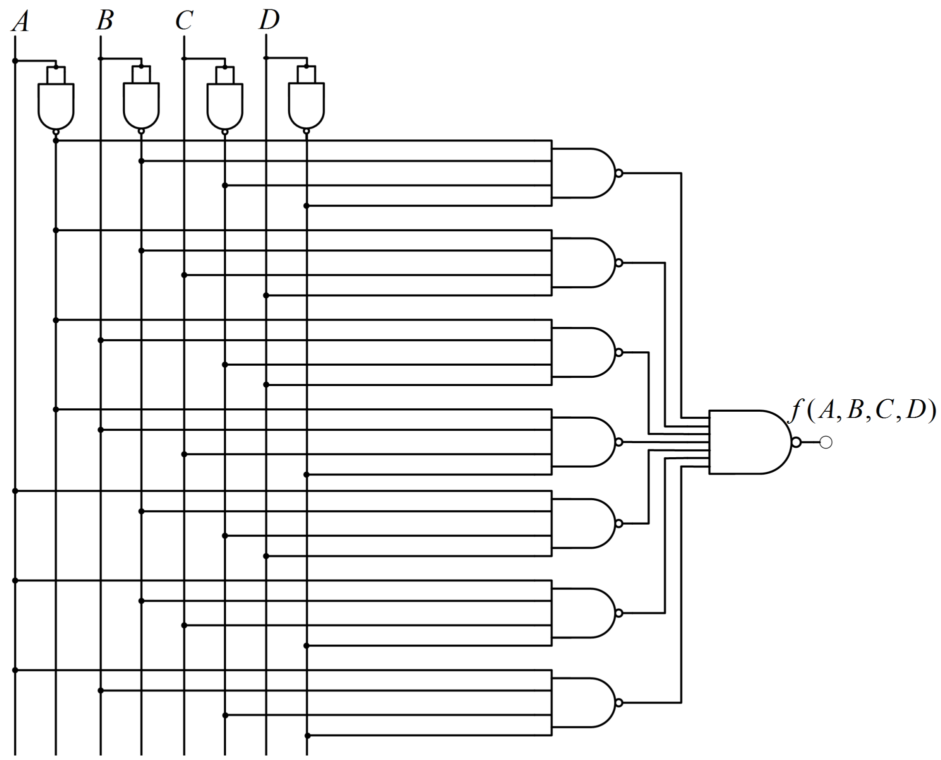
c) Primjenom De Morganovog teorema na logičku funkciju $f(A,B,C,D)$ iz tablice kombinacija 3.12 dobit ćemo:

$$\begin{aligned}
f(A,B,C,D) &= \overline{\overline{f(A,B,C,D)}} = \\
&= \overline{\overline{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + AB\overline{C}\,\overline{D}}} = \\
&= \overline{\overline{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}} \cdot \overline{\overline{A}\,\overline{B}CD} \cdot \overline{\overline{A}B\overline{C}D} \cdot \overline{\overline{A}BC\overline{D}} \cdot \overline{A\overline{B}\,\overline{C}D} \cdot \overline{A\overline{B}C\overline{D}} \cdot \overline{AB\overline{C}\,\overline{D}}}
\end{aligned}$$

Dobiveni izraz za logičku funkciju $f(A,B,C,D)$ može se realizirati NI sklopovima, a realizacija je prikazana na slici 3.22.



Slika 3.19: Logička funkcija $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je na ulazu paran broj nula realizirana osnovnim logičkim sklopovima



Slika 3.20: Logička funkcija $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je na ulazu paran broj nula realizirana NI sklopovima

☞ Primjer 3.4.7

Logička funkcija četiri varijable $f(A,B,C,D)$ na izlazu mora dati vrijednost 1 ako je binarna kombinacija na ulazu djeljiva s 5 bez ostatka. Pri tome je logička varijabla A najznačajnija znamenka 4-bitnog binarnog broja.
a) Za logičku funkciju $f(A,B,C,D)$ napišite tablicu kombinacija.
b) Logičku funkciju $f(A,B,C,D)$ zapišite u obliku sume minterma i izvedite je osnovnim logičkim sklopovima.
c) Logičku funkciju $f(A,B,C,D)$ zapisanu u obliku sume minterma prilagodite i izvedite NI sklopovima.

✍ Rješenje:

a) Tablica kombinacija za logičku funkciju $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je binarna kombinacija na ulazu djeljiva s 5 bez ostatka prikazana je tablicom 3.13.

Tablica 3.13: Tablica kombinacija za logičku funkciju $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je binarna kombinacija na ulazu djeljiva s 5 bez ostatka

i	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

b) Logička funkcija $f(A,B,C,D)$ prikazana tablicom kombinacija 3.13 u obliku sume minterma je:

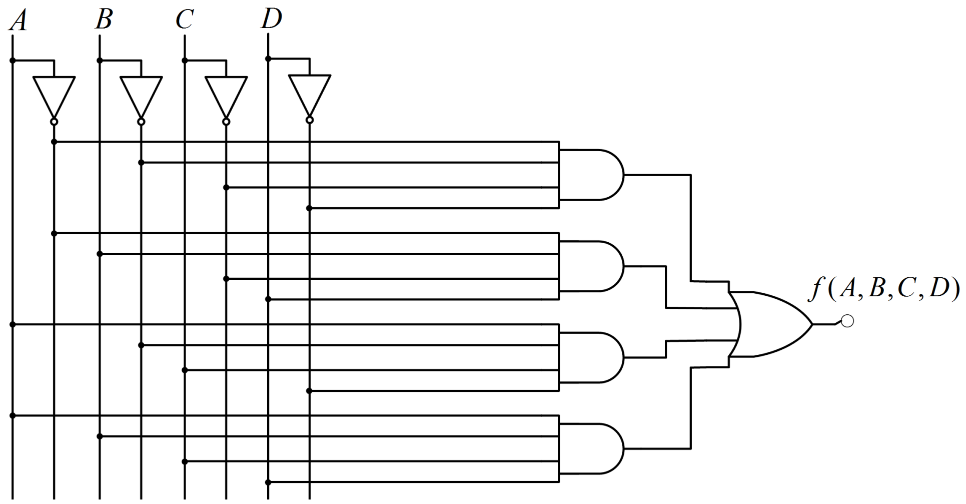
$$f(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + ABCD$$

Logička funkcija $f(A,B,C,D)$ realizirana osnovnim logičkim sklopovima prikazana je na slici 3.21.

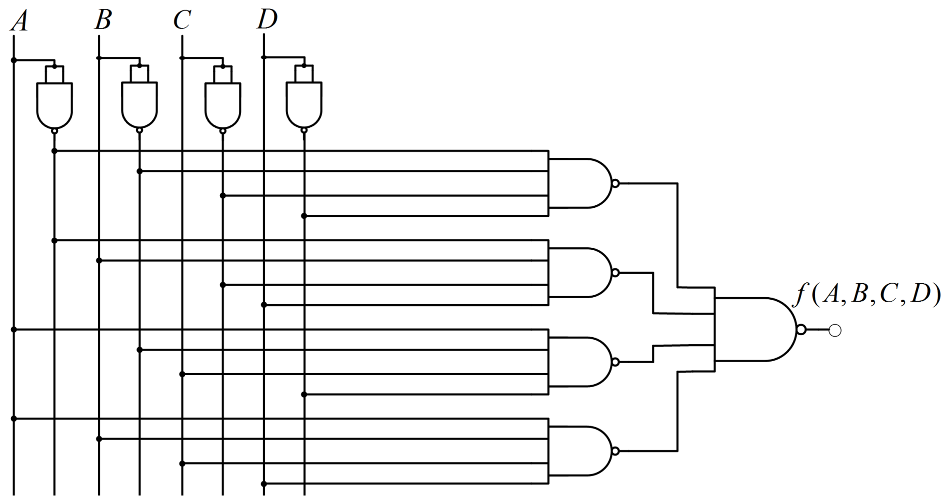
c) Primjenom De Morganovog teorema na logičku funkciju $f(A,B,C,D)$ iz tablice kombinacija 3.13 dobit ćemo:

$$f(A,B,C,D) = \overline{\overline{f(A,B,C,D)}} = \overline{\overline{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + ABCD}} =$$
$$= \overline{\overline{\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}} \cdot \overline{\overline{A}B\overline{C}D} \cdot \overline{A\overline{B}C\overline{D}} \cdot \overline{ABCD}}$$

Dobiveni izraz za logičku funkciju $f(A,B,C,D)$ može se realizirati NI sklopovima, a realizacija je prikazana na slici 3.22.



Slika 3.21: Logička funkcija $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je binarna kombinacija na ulazu djeljiva s 5 bez ostatka realizirana osnovnim logičkim sklopovima



Slika 3.22: Logička funkcija $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako je binarna kombinacija na ulazu djeljiva s 5 bez ostatka realizirana NI sklopovima

3.4.1 Zadaci za vježbu

✎ Zadatak 3.4.1

Izvedite logičku funkciju NILI pomoću NI logičkih sklopova.

✎ Zadatak 3.4.2

Izvedite logičku funkciju NI pomoću NILI logičkih sklopova.

✎ Zadatak 3.4.3

Zadana je logička funkcija:

$$f(A,B,C) = \overline{A}B + A\overline{C}.$$

a) Realizirajte funkciju $f(A,B,C)$ pomoću NI sklopova.
b) Realizirajte funkciju $f(A,B,C)$ pomoću NILI sklopova.

✎ Zadatak 3.4.4

Zadana je logička funkcija (EX NILI (isključivo NILI)):

$$f(A,B) = \overline{A \oplus B}.$$

a) Realizirajte funkciju $f(A,B)$ pomoću osnovnih logičkih sklopova.
b) Realizirajte funkciju $f(A,B)$ pomoću NI sklopova.
Napomena: logička funkcija EX NILI komplementarna je funkciji EX ILI čija je tablica kombinacija prikazana tablicom 3.11.

✎ Zadatak 3.4.5

Zadana je logička funkcija:

$$f(A,B,C) = \big(AB(B + \overline{B})\big) \oplus C$$

a) Realizirajte funkciju $f(A,B,C)$ pomoću osnovnih logičkih sklopova.
b) Realizirajte funkciju $f(A,B,C)$ pomoću NI sklopova.

✎ Zadatak 3.4.6

Logička funkcija četiri varijable $f(A,B,C,D)$ na izlazu mora dati vrijednost 1 ako je na ulazu neparan broj jedinica.
a) Za logičku funkciju $f(A,B,C,D)$ napišite tablicu kombinacija.
b) Logičku funkciju $f(A,B,C,D)$ zapišite u obliku sume minterma i izvedite je osnovnim logičkim sklopovima.
c) Logičku funkciju $f(A,B,C,D)$ zapisanu u obliku sume minterma prilagodite i izvedite NI sklopovima.

✎ Zadatak 3.4.7

Logička funkcija četiri varijable $f(A,B,C,D)$ na izlazu mora dati vrijednost 1 ako je binarna kombinacija na ulazu djeljiva s 3 bez ostatka. Pri tome je logička varijabla A najznačajnija znamenka 4-bitnog binarnog broja.
a) Za logičku funkciju $f(A,B,C,D)$ napišite tablicu kombinacija.
b) Logičku funkciju $f(A,B,C,D)$ zapišite u obliku sume minterma i izvedite je osnovnim logičkim sklopovima.
c) Logičku funkciju $f(A,B,C,D)$ zapisanu u obliku sume minterma prilagodite i izvedite NI sklopovima.

Poglavlje 4

Minimizacija logičkih funkcija

Logička funkcija algebarski može biti prikazana na više načina. Cilj minimizacije je dobivanje minimalnog oblika logičkih funkcija. Kod projektiranja to znači sklop s najmanjim brojem elemenata. Takav sklop je manji, pouzdaniji i jeftiniji te s manjom potrošnjom energije i većom brzinom rada. Ne postoji jedinstveno rješenje i jedan način minimizacije. Metode koje se najčešće koriste za minimizaciju logičkih funkcija su:

- algebarska metoda minimizacije,
- minimizacija pomoću Karnaughove tablice (u nastavku ćemo ih zvati K-tablice),
- Quine-McCluskeyjeva metoda.

U ovom poglavlju opisat ćemo algebarsku metodu minimizacije i minimizaciju K-tablicom.

4.1 Algebarska metoda minimizacije

Na funkciju izraženu u algebarskom obliku mogu se uzastopno primijeniti zakoni Booleove algebre da bi se dobio jednostavniji oblik. Kako ne postoji definirani postupak koji bi doveo do minimalnog izraza (uspješnost ovisi o znanju, intuiciji i iskustvu), metoda je prikladna za jednostavnije logičke funkcije.

☞ Primjer 4.1.1

Algebarskom metodom minimizirajte logičku funkciju:

$$f(A,B,C) = \overline{B}C + C + B(\overline{B} + B + \overline{C}C) + AB.$$

✍ **Rješenje:** Ako na logičku funkciju $f(A,B,C)$ primijenimo zakone Booleove algebre dobit ćemo sljedeći minimalni oblik funkcije:

$$f(A,B,C) = \overline{B}C + C + B(\underbrace{\overline{B} + B}_{1} + \underbrace{\overline{C}C}_{0}) + AB = (\underbrace{\overline{B} + 1}_{1})C + B(\underbrace{1 + 0}_{1}) + AB =$$
$$= C + B + AB = C + (\underbrace{1 + A}_{1})B = B + C$$

☞ Primjer 4.1.2

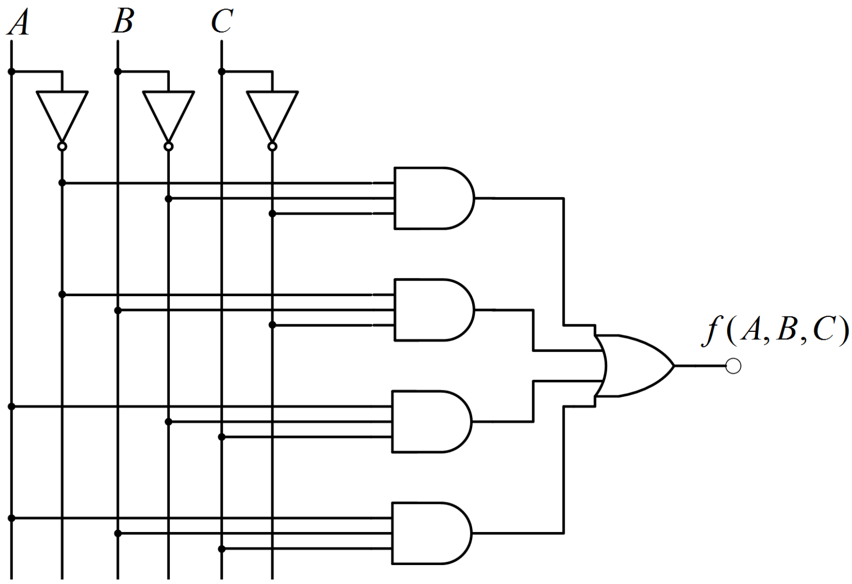
Zadana je logička funkcija:

$$f(A,B,C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + ABC.$$

a) Realizirajte zadanu logičku funkciju $f(A,B,C)$ osnovnim logičkim sklopovima.
b) Minimizirajte zadanu logičku funkciju $f(A,B,C)$ algebarskom metodom, a zatim minimalni oblik funkcije realizirajte osnovnim logičkim sklopovima.

✍ Rješenje:

a) Logička funkcija $f(A,B,C)$ realizirana osnovnim logičkim sklopovima prikazana je na slici 4.1.

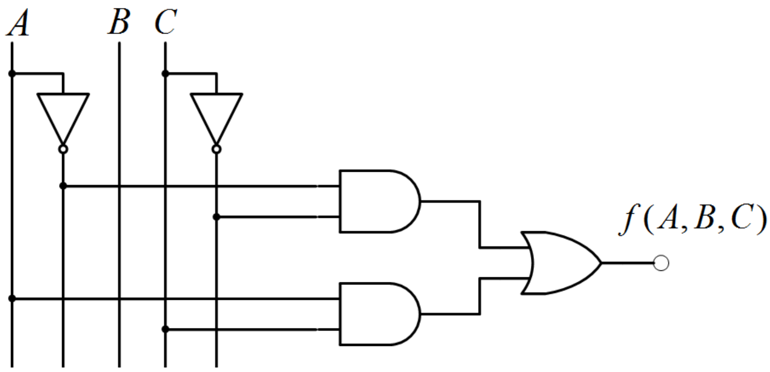


Slika 4.1: Logička funkcija $f(A,B,C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + ABC$ realizirana osnovnim logičkim sklopovima

b) Ako na logičku funkciju $f(A,B,C)$ primijenimo zakone Booleove algebre dobit ćemo sljedeći minimalni oblik funkcije:

$$f(A,B,C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + ABC = \overline{A}(\underbrace{\overline{B}+B}_{1})\overline{C} + A(\underbrace{\overline{B}+B}_{1})C = \overline{A}\overline{C} + AC$$

Minimizirana logička funkcija $f(A,B,C)$ realizirana osnovnim logičkim sklopovima prikazana je na slici 4.2. Minimizacijom logičke funkcije $f(A,B,C)$ smanjili smo broj potrebnih sklopova za njenu realizaciju što se može vidjeti usporedbom slika 4.1 i 4.2.



Slika 4.2: Minimizirana logička funkcija $f(A,B,C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + ABC = \overline{A}\overline{C} + AC$ realizirana osnovnim logičkim sklopovima

4.1.1 Zadaci za vježbu

✎ Zadatak 4.1.1

Algebarskom metodom minimizirajte logičku funkciju:

$$f(A,B,C) = \overline{A}B + B + ABC((\overline{B} + B) \cdot \overline{C}C) + AB.$$

✎ Zadatak 4.1.2

Zadana je logička funkcija:

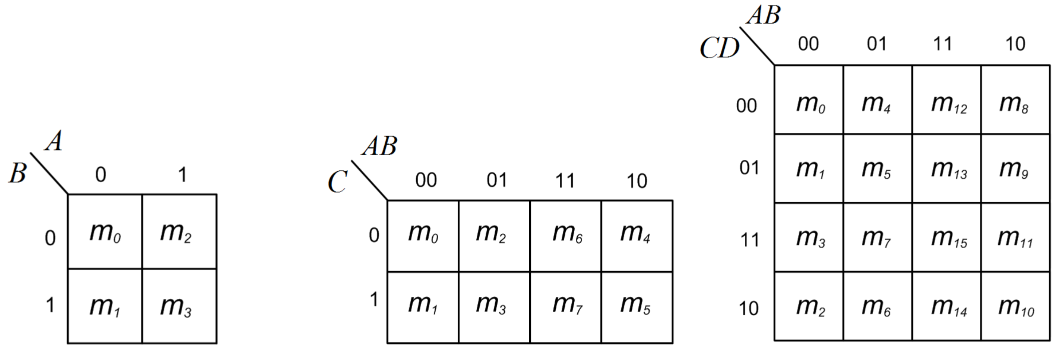
$$f(A,B,C) = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C.$$

a) Realizirajte zadanu logičku funkciju $f(A,B,C)$ osnovnim logičkim sklopovima.
b) Minimizirajte zadanu logičku funkciju $f(A,B,C)$ algebarskom metodom, a zatim minimalni oblik funkcije realizirajte osnovnim logičkim sklopovima.

4.2 Minimizacija pomoću Karnaughove tablice (K-tablice)

K-tablica grafički je prikaz funkcije izvedena iz tablice kombinacija. Svako polje tablice pripada jednoj kombinaciji ulaznih varijabli. Svakom polju pripada jedan minterm (ili maksterm) i vrijednost funkcije za tu kombinaciju. U nastavku ćemo promatrati K-tablice s mintermima.

Ulazne kombinacije logičkih varijabli raspoređene su u polja K-tablice (ćelije K-tablice) tako da je razmak ili distanca između dvije susjedne kombinacije 1. Također, susjedne su kombinacije s distancom 1 u svakom stupcu ili retku prva i zadnja ćelija. Raspored minterma po ćelijama K-tablice za funkcije 2, 3 i 4 logičke varijable prikazan je na slici 4.3.



(a) K-tablica za 2 logičke varijable

(b) K-tablica za 3 logičke varijable

(c) K-tablica za 4 logičke varijable

Slika 4.3: K-tablice za 2, 3 i 4 logičke varijable

☞ Primjer 4.2.1

Zadane logičke funkcije minimizirajte K-tablicom:
a) $f(A,B) = \overline{A}B + AB$,
b) $f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC$,
c) $f(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + ABC\overline{D}$.

✍ Rješenje:

U prvom primjeru minimiziranja logičkih funkcija K-tablicom detaljno ćemo objasniti postupak minimizacije.

a) Za logičku funkciju $f(A,B) = \overline{A}B + AB$ vrijedi:

$$f(A,B) = \overline{A}B + AB = m_1 + m_3.$$

U K-tablicu sa slike 4.3a upišu se vrijednosti 1 na mjestu minterma koji definiraju logičku funkciju $f(A,B)$. Popunjena K-tablica prikazana je na slici 4.4a. U popunjenoj K-tablici potrebno je pronaći najmanji broj najvećih mogućih pravokutnih područja koja su prekrivena vrijednostima 1. Pravokutna područja moraju imati dimenzije koje su potencije broja 2 (2^i, $i = 0, 1, 2, 3, ...$). Prema tome, pravokutno područje 3×1 nije kandidat za daljnju minimizaciju, dok pravokutno područje 2×1 jest kandidat za daljnju minimizaciju. Pravokutno područje predstavlja grupu susjednih vrijednosti 1. Pronađeno pravokutno područje potrebno je zaokružiti (slika 4.4b).

(a) Upisivanje vrijednosti 1 na mjesto minterma

(b) Grupiranje minterma

Slika 4.4: K-tablica za logičku funkciju $f(A,B) = \overline{A}B + AB$

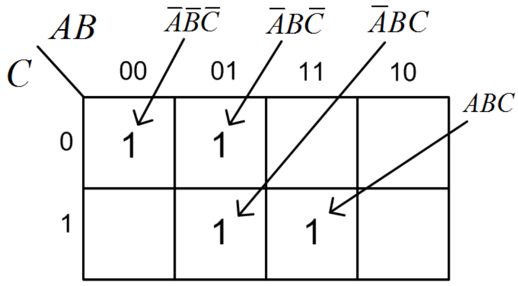
U zaokruženom pravokutnom području potrebno je uočiti koja se logička varijabla mijenja, a koja je konstantna na tom području. U području sa slike 4.4b mijenja se varijabla A jer zaokružena grupa sadrži logičku varijablu $\overline{A}$ i A. Varijabla B se ne mijenja, a zaokruženo područje je mjesto gdje varijabla B ima vrijednost 1. Ako se varijabla na pravokutnom području mijenja (može biti i 0 i 1), tada logička funkcija ne ovisi o toj varijabli. Prema tome, minimizirana logička funkcija prema slici 4.4 bit će $f(A,B) = B$. Isto bismo rješenje dobili algebarskom metodom minimizacije:

$$f(A,B) = \bar{A}B + AB = (\underbrace{\bar{A} + A}_{1})B = B.$$

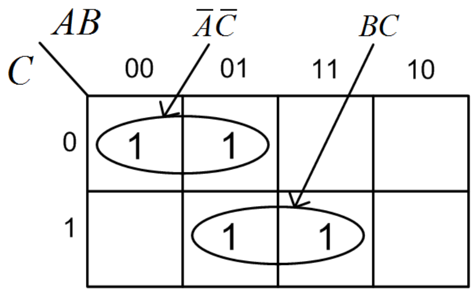
b) Za logičku funkciju $f(A,B,C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC$ vrijedi:

$$f(A,B,C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC = m_0 + m_2 + m_3 + m_7.$$

U K-tablicu sa slike 4.3b upišu se vrijednosti 1 na mjestu minterma koji definiraju logičku funkciju $f(A,B,C)$. Popunjena K-tablica prikazana je na slici 4.5a. U popunjenoj K-tablici potrebno je pronaći najmanji broj najvećih mogućih pravokutnih područja koja su prekrivena vrijednostima 1. Pravokutna područja moraju imati dimenzije koje su potencije broja 2 (2^i, $i = 0, 1, 2, 3, ...$). Pronađeno pravokutno područje potrebno je zaokružiti (slika 4.5b).



(a) Upisivanje vrijednosti 1 na mjesto minterma



(b) Grupiranje minterma

Slika 4.5: K-tablica za logičku funkciju $f(A,B,C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC$

U zaokruženim pravokutnim područjima (slika 4.5b) potrebno je uočiti koje se logičke varijable mijenjaju, a koje su konstantne na tim područjima. Promotrit ćemo najprije područje kojeg čine mintermi m_0 i m_2. U tom području konstantne su varijable $\overline{A}$ i $\overline{C}$, dok se varijabla B mijenja. Prema tome, dio logičke funkcije kojeg definiraju mintermi m_0 i m_2 ne ovisi o varijabli B. Zbog toga vrijedi: $\overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} = \overline{A}(\overline{B} + B)\overline{C} = \overline{A}\overline{C}$. Prema tome, minimalni oblik područja kojeg čine mintermi m_0 i m_2 je $\overline{A}\overline{C}$ (slika 4.5b).

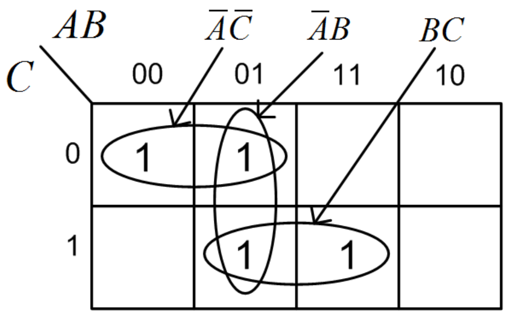
U području kojeg čine mintermi m_3 i m_7 konstantne su varijable B i C, dok se varijabla A mijenja. Prema tome, dio logičke funkcije kojeg definiraju mintermi m_3 i m_7 ne ovisi o varijabli A. Zbog toga vrijedi: $\overline{A}BC+ABC = (\overline{A}+A)BC = BC$. Prema tome, minimalni oblik područja kojeg čine mintermi m_3 i m_7 je BC (slika 4.5b).

Minimizirana logička funkcija prema slici 4.5 bit će $f(A,B,C) = \overline{A}\overline{C}+BC$. Isto bismo rješenje dobili algebarskom metodom minimizacije:

$$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + ABC = \bar{A}(\underbrace{\bar{B} + B}_{1})\bar{C} + (\underbrace{\bar{A} + A}_{1})BC = \bar{A}\bar{C} + BC.$$

Primijetimo da smo pravokutna područja mogli zaokružiti na način prikazan na slici 4.6. Način zaokruživanja jest ispravan, ali rezultat minimizacije neće biti minimalni oblik, već

$f(A,B,C) = \overline{A}\,\overline{C} + \overline{A}B + BC$. Dakle, bitno je slijediti upute da se zaokruži najmanji broj najvećih pravokutnih područja.

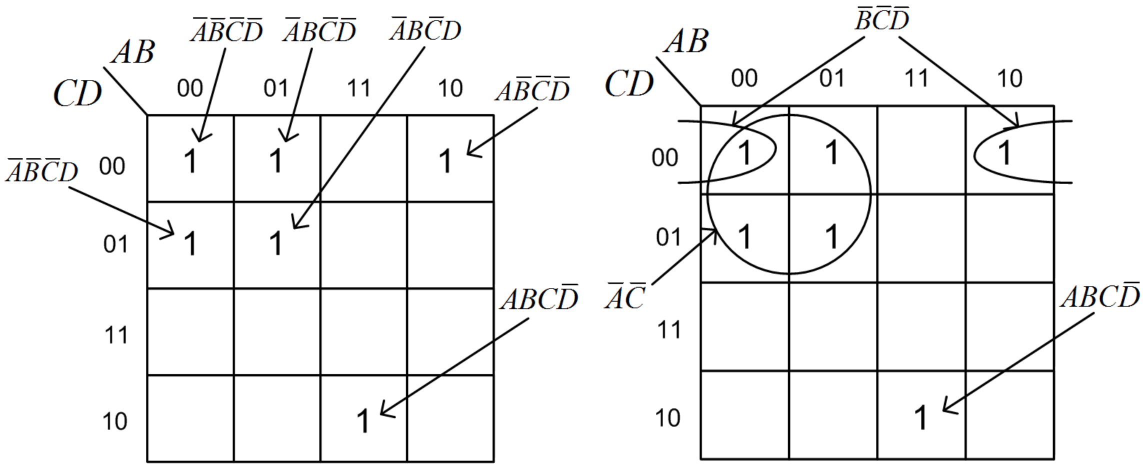


Slika 4.6: K-tablica za logičku funkciju $f(A,B,C) = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + ABC$ - neminimalni oblik

c) Za logičku funkciju $f(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + ABC\overline{D}$ vrijedi:

$$f(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + ABC\overline{D} =$$
$$= m_0 + m_1 + m_4 + m_5 + m_8 + m_{14}.$$

U K-tablicu sa slike 4.3c upišu se vrijednosti 1 na mjestu minterma koji definiraju logičku funkciju $f(A,B,C,D)$. Popunjena K-tablica prikazana je na slici 4.7a. U popunjenoj K-tablici potrebno je pronaći najmanji broj najvećih mogućih pravokutnih područja koja su prekrivena vrijednostima 1. Pravokutna područja moraju imati dimenzije koje su potencije broja 2 (2^i, $i = 0, 1, 2, 3, ...$). Pronađeno pravokutno područje potrebno je zaokružiti (slika 4.7b).



(a) Upisivanje vrijednosti 1 na mjesto minterma

(b) Grupiranje minterma

Slika 4.7: K-tablica za logičku funkciju
$f(A,B,C,D) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}B\overline{C}\,\overline{D} + \overline{A}B\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + ABC\overline{D}$

U zaokruženim pravokutnim područjima (slika 4.7b) potrebno je uočiti koje se logičke varijable mijenjaju, a koje su konstantne na tim područjima. Promotrit ćemo najprije područje kojeg čine mintermi m_0, m_1, m_4 i m_5. U tom području konstantne su varijable $\overline{A}$ i $\overline{C}$, dok se varijable B i D mijenjaju. Prema tome, dio logičke funkcije kojeg definiraju mintermi m_0, m_1, m_4 i m_5 ne ovisi o varijablama B i D. Minimalni oblik područja kojeg čine mintermi m_0, m_1, m_4 i m_5 je $\overline{A}\overline{C}$ (slika 4.7b).

Mintermi m_0 i m_8 također su susjedni mintermi i čine pravokutno područje. U ovom području konstantne su varijable $\overline{B}$, $\overline{C}$ i $\overline{D}$, dok se varijabla A mijenja. Prema tome, dio logičke funkcije kojeg definiraju mintermi m_0 i m_8 ne ovisi o varijabli A. Minimalni oblik područja kojeg čine mintermi m_0 i m_8 je $\overline{B}\overline{C}\overline{D}$ (slika 4.7b).

Minterm m_{14} nema susjednih elementa s kojima može činiti pravokutno područje te je njegov minimalni oblik on sam.

Minimizirana logička funkcija prema slici 4.7 bit će $f(A,B,C) = \overline{A}\overline{C} + \overline{B}\overline{C}\overline{D} + ABC\overline{D}$. U ovom slučaju algebarska metoda je složenija pa je nećemo niti navoditi.

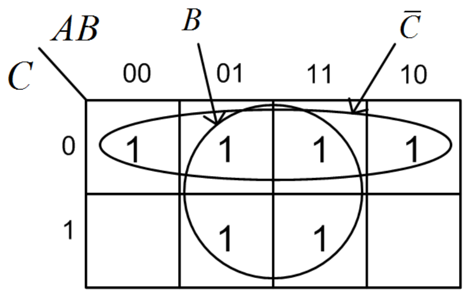
☞ Primjer 4.2.2

Zadane logičke funkcije minimizirajte K-tablicom:
a) $f(A,B,C) = \sum(0,2,3,4,6,7)$,
b) $f(A,B,C) = \prod(2,3,6,7)$,
c) $f(A,B,C,D) = \sum(2,4,5,6,7,10,13,15)$,
d) $f(A,B,C,D) = \prod(5,7,13,15)$.

✍ **Rješenje:** Zadane funkcije napisane su u skraćenom obliku iz kojeg se direktno mogu pročitati mintermi i makstermi.

a) Minimizacija logičke funkcije $f(A,B,C) = \sum(0,2,3,4,6,7)$ K-tablicom prikazana je na slici 4.8, a minimalni oblik je:

$$f(A,B,C) = B + \overline{C}.$$



Slika 4.8: K-tablica za logičku funkciju $f(A,B,C) = \sum(0,2,3,4,6,7)$

b) Neposredno prije minimizacije potrebno je funkciju iz oblika produkta maksterma zapisati u oblik sume minterma:

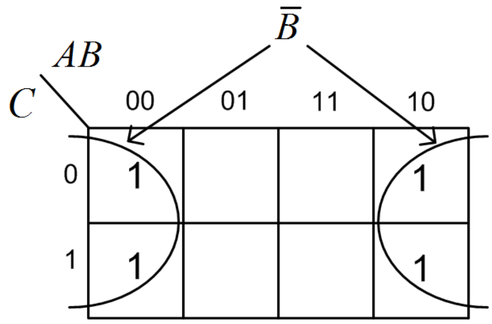
$$f(A,B,C) = \prod(2,3,6,7) = \sum(0,1,4,5).$$

Minimizacija logičke funkcije $f(A,B,C) = \prod(2,3,6,7) = \sum(0,1,4,5)$ K-tablicom prikazana je na slici 4.9, a minimalni oblik je:

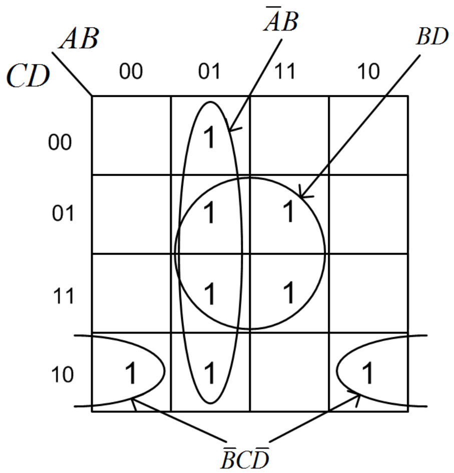
$$f(A,B,C) = \overline{B}.$$

c) Minimizacija logičke funkcije $f(A,B,C,D) = \sum(2,4,5,6,7,10,13,15)$ K-tablicom prikazana je na slici 4.10, a minimalni oblik je:

$$f(A,B,C,D) = \overline{A}B + BD + \overline{B}C\overline{D}.$$



Slika 4.9: K-tablica za logičku funkciju $f(A,B,C) = \prod(2,3,6,7)$



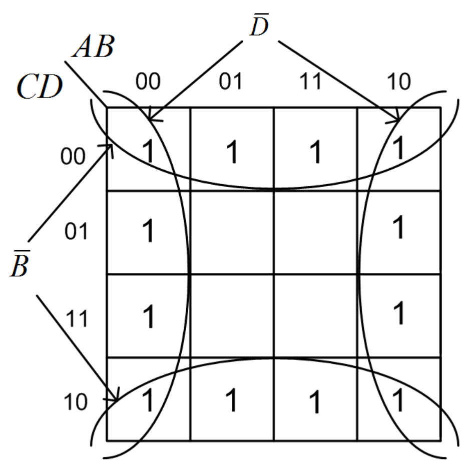
Slika 4.10: K-tablica za logičku funkciju $f(A,B,C,D) = \sum(2,4,5,6,7,10,13,15)$

d) Neposredno prije minimizacije potrebno je funkciju iz oblika produkta maksterma zapisati u oblik sume minterma:

$$f(A,B,C,D) = \prod(5,7,13,15) = \sum(0,1,2,3,4,6,8,9,10,11,12,14).$$

Minimizacija logičke funkcije $f(A,B,C,D) = \prod(5,7,13,15) = \sum(0,1,2,3,4,6,8,9,10,11,12,14)$ K-tablicom prikazana je na slici 4.11, a minimalni oblik je:

$$f(A,B,C,D) = \overline{B} + \overline{D}.$$



Slika 4.11: K-tablica za logičku funkciju $f(A,B,C,D) = \prod(5,7,13,15)$

☞ Primjer 4.2.3

Minimizirajte logičke funkcije upisane u K-tablice sa slike 4.12.

CD \ AB	00	01	11	10
00	1			1
01		1		
11	1	1	1	1
10	1			1

(a) K-tablica za $f_1(A,B,C,D)$

CD \ AB	00	01	11	10
00	1		1	
01	1			1
11	1	1	1	1
10	1			

(b) K-tablica za $f_2(A,B,C,D)$

CD \ AB	00	01	11	10
00		1	1	
01	1		1	1
11	1			1
10		1		

(c) K-tablica za $f_3(A,B,C,D)$

Slika 4.12: K-tablice za logičke funkcije $f_1(A,B,C,D)$, $f_2(A,B,C,D)$ i $f_3(A,B,C,D)$

✍ Rješenje:

a) Minimizacija logičke funkcije $f_1(A,B,C,D)$ K-tablicom prikazana je na slici 4.13a, a minimalni oblik je:

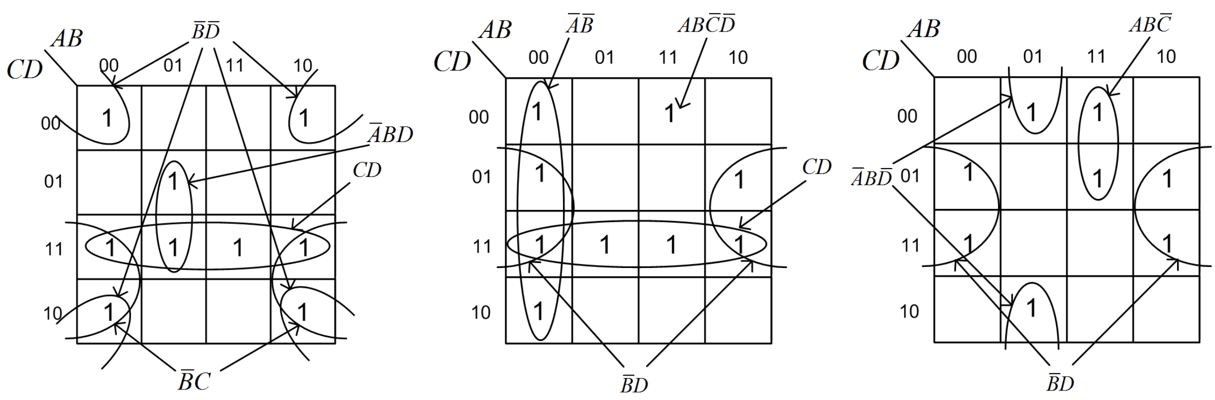
$$f(A,B,C,D) = \overline{B}\overline{D} + \overline{B}C + CD + \overline{A}BD.$$

b) Minimizacija logičke funkcije $f_2(A,B,C,D)$ K-tablicom prikazana je na slici 4.13b, a minimalni oblik je:

$$f(A,B,C,D) = \overline{A}\overline{B} + \overline{B}D + CD + AB\overline{C}\overline{D}.$$

c) Minimizacija logičke funkcije $f_2(A,B,C,D)$ K-tablicom prikazana je na slici 4.13c, a minimalni oblik je:

$$f(A,B,C,D) = \overline{B}D + \overline{A}B\overline{D} + AB\overline{C}.$$



(a) K-tablica za $f_1(A,B,C,D)$ (b) K-tablica za $f_2(A,B,C,D)$ (c) K-tablica za $f_3(A,B,C,D)$

Slika 4.13: K-tablice za logičke funkcije $f_1(A,B,C,D)$, $f_2(A,B,C,D)$ i $f_3(A,B,C,D)$ - zaokružena pravokutna područja

☞ Primjer 4.2.4

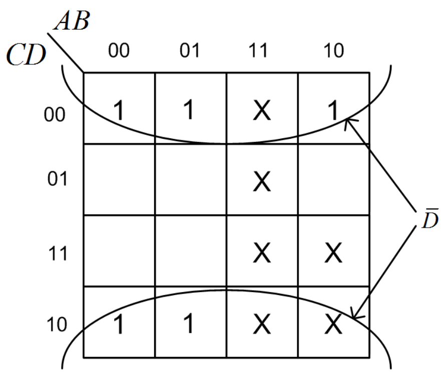
Minimizirajte logičku funkciju koja će na izlazu dati vrijednost 1 ako su prvih 10 kombinacija 4-bitnog binarnog broja djeljive s 2. Ostalih 6 kombinacija nespecificirane su.

✍ Rješenje:

Tablica kombinacija za logičku funkciju $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako su prvih 10 kombinacija 4-bitnog binarnog broja djeljive s 2, dok su ostale kombinacije nespecificirane prikazana je tablicom 4.1. Nespecificirane kombinacije označili smo s X. Prilikom popunjavanja K-tablice unose se mintermi za koje logička funkcija ima vrijednost 1. Za nespecificirane minterme funkcija može poprimiti vrijednost 1 ili 0. U K-tablicu nespecificirani se mintermi mogu unijeti s oznakom X. Kod formiranja pravokutnih područja, minterm s oznakom X može se promatrati kao vrijednost 1 te tako omogućiti stvaranje većih pravokutnih područja. U postupku stvaranja pravokutnih područja bitno je obuhvatiti sve vrijednosti 1 te one vrijednosti X koje pomažu u formiranju većih pravokutnih područja.

Tablica 4.1: Tablica kombinacija za logičku funkciju $f(A,B,C,D)$ koja na izlazu daje vrijednost 1 ako su prvih 10 kombinacija 4-bitnog binarnog broja djeljive s 2 dok su ostale kombinacije nespecificirane

i	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	X
11	1	0	1	1	X
12	1	1	0	0	X
13	1	1	0	1	X
14	1	1	1	0	X
15	1	1	1	1	X



Slika 4.14: K-tablica za logičku funkciju definiranu tablicom 4.1

Minimizacija logičke funkcije definirane tablicom 4.1 pomoću K-tablice prikazana je na slici 4.14, a minimalni oblik je:

$$f(A,B,C,D) = \overline{D}.$$

4.2.1 Zadaci za vježbu

✎ Zadatak 4.2.1

Zadane logičke funkcije minimizirajte K-tablicom:
a) $f(A,B) = A\overline{B} + AB$,
b) $f(A,B,C) = \overline{A}\overline{B}C + \overline{A}BC + A\overline{B}\overline{C} + AB\overline{C}$,
c) $f(A,B,C,D) = \overline{A}\overline{B}CD + \overline{A}BCD + A\overline{B}CD + AB\overline{C}D + ABC\overline{D} + ABCD$.

✎ Zadatak 4.2.2

Zadane logičke funkcije minimizirajte K-tablicom:
a) $f(A,B,C) = \sum(0,1,3,5,7)$,
b) $f(A,B,C) = \prod(3,6,7)$,
c) $f(A,B,C,D) = \sum(1,3,5,7,11,12,15)$,
d) $f(A,B,C,D) = \prod(1,3,5,7,11,12,15)$.

✎ Zadatak 4.2.3

Minimizirajte logičke funkcije upisane u K-tablice sa slike 4.15.

CD \ AB	00	01	11	10
00		1		
01		1		1
11	1		1	1
10	1			

(a) K-tablica za $f_1(A,B,C,D)$

CD \ AB	00	01	11	10
00	1			
01		1	1	1
11		1	1	1
10	1			

(b) K-tablica za $f_2(A,B,C,D)$

CD \ AB	00	01	11	10
00		1	1	
01		1		
11	1	1		
10		1	1	

(c) K-tablica za $f_3(A,B,C,D)$

Slika 4.15: K-tablice za logičke funkcije $f_1(A,B,C,D)$, $f_2(A,B,C,D)$ i $f_3(A,B,C,D)$

✎ Zadatak 4.2.4

Minimizirajte logičku funkciju koja će na izlazu dati vrijednost 1 ako su prvih 8 kombinacija 4-bitnog binarnog broja djeljive s 2. Ostale 4 kombinacije nespecificirane su.

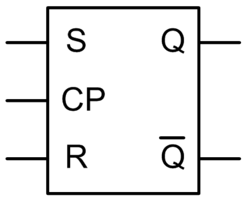
Poglavlje 5

Bistabili

Elektronički sklopovi sastavljeni od dva invertirajuća pojačala povezana u petlju pozitivne povratne veze zovu se bistabilni multivibratori ili bistabili. U nastavku ćemo pokazati simbole, tablice stanja, tablice uzbuda i dijagrame stanja sljedećih bistabila:

- SR bistabil,
- JK bistabil,
- T bistabil,
- D bistabil.

Na slici 5.1 prikazan je simbol SR bistabila. SR bistabil ima dva ulaza (S, R), ulaz za takt impuls (CP) te dva izlaza (Q, $\overline{Q}$). Izlaz $\overline{Q}$ invertirani je izlaz Q. SR bistabil na temelju ulaza S i R mijenja stanje svog izlaza na svaki impuls takta CP prema tablici 5.1.



Slika 5.1: Simbol SR bistabila

Tablica 5.1: Tablica kombinacija za SR bistabil

(a) Proširena tablica

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

(b) Sažeta tablica

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	X

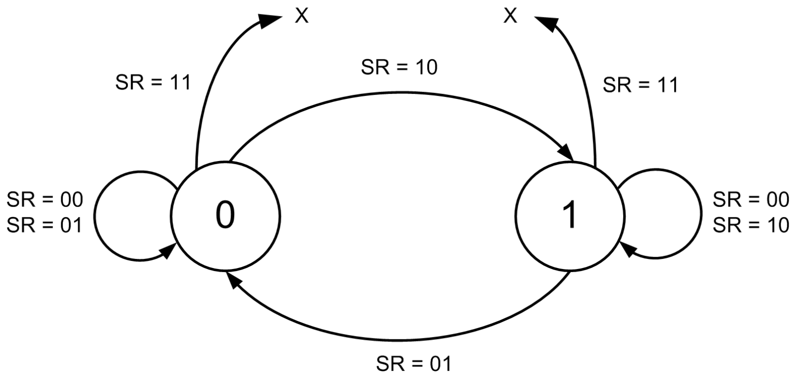
U tablicama 5.1a i 5.1b prikazana je i nedopuštena kombinacija ulaza $SR = 11$ pri čemu je izlaz bistabila označen s X (može poprimiti vrijednost 0 ili 1). Izlaz je u tablicama 5.1a i 5.1b označen na dva načina:

- Q_n - trenutno stanje izlaza SR bistabila (korak n),
- Q_{n+1} - sljedeće stanje izlaza SR bistabila (korak $n + 1$).

Na isti će se način izlazi označavati i kod ostalih bistabila. Jednadžba sljedećeg stanja izlaza dobivena na temelju trenutnog stanja izlaza SR bistabila i njegovih ulaza je:

$$Q_{n+1} = S + Q_n\overline{R}. \tag{5.1}$$

Ponašanje SR bistabila može se grafički prikazati pomoću dijagrama stanja. Pri tome se stanja prikazuju kao čvorovi (u kružnici) povezani usmjerenim linijama iznad kojih se upisuju vrijednosti ulaza koje izazivaju prikazanu promjenu. Za SR bistabil dijagram stanja je prikazan na slici 5.2.



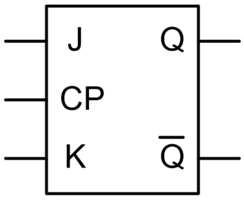
Slika 5.2: Dijagram stanja SR bistabila

Kada se bistabil koristi u složenijem sklopu važno je znati koje će sadašnje stanje ulaza (S, R) i izlaza (Q_n) bistabila dovesti do traženog sljedećeg stanja (Q_{n+1}), odnosno koja će pobuda dovesti do tražene promjene na izlazu. Tablica koja na taj način opisuje ponašanje sklopa zove se tablica uzbude (eng. *excitation table*). Tablica uzbude SR bistabila prikazana je tablicom 5.2, a izvedena je iz tablice 5.1a. X u tablici označava nespecificirano stanje (0 ili 1).

Tablica 5.2: Tablica uzbude za SR bistabil

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Nedopuštena stanja SR bistabila dopuštena su u JK bistabilu. Na slici 5.3 prikazan je simbol JK bistabila. JK bistabil ima dva ulaza (J, K), ulaz za takt impuls (CP) te dva izlaza (Q, $\overline{Q}$). Izlaz $\overline{Q}$ je invertirani izlaz Q. JK bistabil na temelju ulaza J i K mijenja stanje svog izlaza na svaki impuls takta CP prema tablici 5.3.



Slika 5.3: Simbol JK bistabila

Tablica 5.3: Tablica kombinacija za JK bistabil

(a) Proširena tablica

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

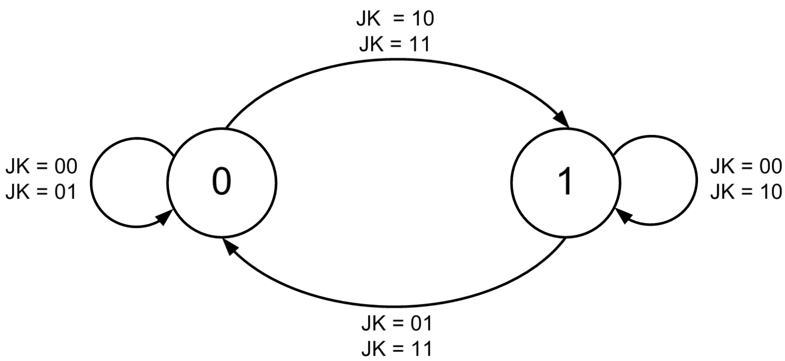
(b) Sažeta tablica

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q}_n$

U tablicama 5.3a i 5.3b vidimo da JK bistabil za ulaz $JK = 11$ mijenja svoje stanje. Jednadžba sljedećeg stanja izlaza dobivena na temelju trenutnog stanja izlaza JK bistabila i njegovih ulaza je:

$$Q_{n+1} = J\overline{Q}_n + \overline{K}Q_n. \tag{5.2}$$

Dijagram stanja JK bistabila za razliku od SR bistabila ima definirane sve prijelaze i prikazan je na slici 5.4.



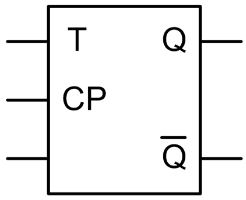
Slika 5.4: Dijagram stanja JK bistabila

Tablica uzbude koristi se kod projektiranja složenijih logičkih sklopova koji koriste JK bistabile za stvaranje uvjeta za prijelaz bistabila u sljedeće stanje. Iz nje se može iščitati što je potrebno dovesti na ulaze bistabila da bi se dolaskom impulsa takta CP dogodio prijelaz u sljedeće stanje Q_{n+1}. Tablica uzbude JK bistabila prikazana je tablicom 5.4, a izvedena je iz tablice 5.3a.

Tablica 5.4: Tablica uzbude za JK bistabil

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Na slici 5.5 prikazan je simbol T bistabila. T bistabil ima jedan ulaz (T), ulaz za takt impuls (CP) te dva izlaza (Q, $\overline{Q}$). Izlaz $\overline{Q}$ je invertirani izlaz Q. T bistabil na temelju ulaza T mijenja stanje svog izlaza na svaki impuls takta CP prema tablici 5.5.



Slika 5.5: Simbol T bistabila

Tablica 5.5: Tablica kombinacija za T bistabil

(a) Proširena tablica

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

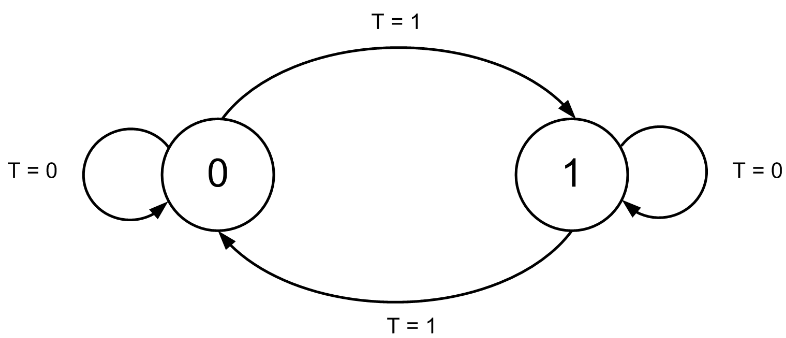
(b) Sažeta tablica

T	Q_{n+1}
0	Q_n
1	$\overline{Q}_n$

Jednadžba sljedećeg stanja izlaza dobivena na temelju trenutnog stanja izlaza T bistabila i njegovih ulaza je:

$$Q_{n+1} = T\overline{Q}_n + \overline{T}Q_n. \tag{5.3}$$

Dijagram stanja T bistabila prikazan je na slici 5.6.



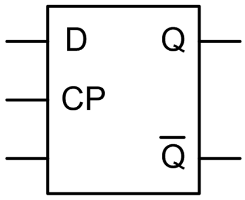
Slika 5.6: Dijagram stanja T bistabila

Tablica uzbude koristi se kod projektiranja složenijih logičkih sklopova koji koriste T bistabile za stvaranje uvjeta za prijelaz bistabila u sljedeće stanje. Tablica uzbude T bistabila prikazana je tablicom 5.6, a izvedena je iz tablice 5.5a.

Tablica 5.6: Tablica uzbude za T bistabil

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Na slici 5.7 prikazan je simbol D bistabila. D bistabil ima jedan ulaz (D), ulaz za takt impuls (CP) te dva izlaza (Q, $\overline{Q}$). Izlaz $\overline{Q}$ je invertirani izlaz Q. D bistabil na temelju ulaza D mijenja stanje svog izlaza na svaki impuls takta CP prema tablici 5.7.



Slika 5.7: Simbol D bistabila

Tablica 5.7: Tablica kombinacija za D bistabil

(a) Proširena tablica

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

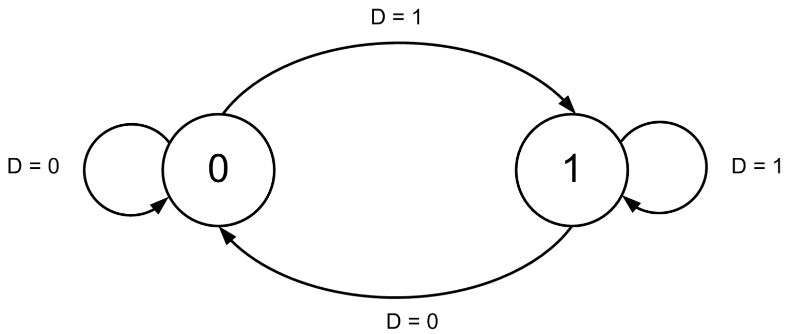
(b) Sažeta tablica

D	Q_{n+1}
0	0
1	1

Jednadžba sljedećeg stanja izlaza dobivena na temelju trenutnog stanja izlaza D bistabila i njegovih ulaza je:

$$Q_{n+1} = D. \tag{5.4}$$

Dijagram stanja D bistabila prikazan je na slici 5.8.



Slika 5.8: Dijagram stanja D bistabila

Tablica uzbude koristi se kod projektiranja složenijih logičkih sklopova koji koriste D bistabile za stvaranje uvjeta za prijelaz bistabila u sljedeće stanje. Tablica uzbude D bistabila prikazana je tablicom 5.8, a izvedena je iz tablice 5.7a.

Tablica 5.8: Tablica uzbude za D bistabil

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

☞ Primjer 5.1.1

Pomoću SR bistabila i minimalnog broja osnovnih logičkih sklopova ostvarite JK bistabil, a zatim pomoću JK bistabila i minimalnog broja osnovnih logičkih sklopova ostvarite T bistabil.

✍ Rješenje:

Pomoću SR bistabila i minimalnog broja osnovnih logičkih sklopova potrebno je ostvariti JK bistabil. Kako bismo pomoću SR bistabila ostvarili JK bistabil, potrebno je prvo napisati tablicu kombinacija za JK bistabil (tablica 5.3). Ulazi u novi bistabil bit će J, K i Q_n. U tu istu tablicu kombinacija potrebno je dodati stupce za ulaze SR bistabila S i R kako bi se pomoću tablice uzbude SR bistabila 5.2 ostvario prijelaz JK bistabila iz stanja Q_n u Q_{n+1}. Tablica kombinacija za JK bistabil proširena uzbudom SR bistabila prikazana je tablicom 5.9.

Tablica 5.9: Tablica kombinacija za JK bistabil proširena uzbudom SR bistabila

i	J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	0	X
1	0	0	1	1	X	0
2	0	1	0	0	0	X
3	0	1	1	0	0	1
4	1	0	0	1	1	0
5	1	0	1	1	X	0
6	1	1	0	1	1	0
7	1	1	1	0	0	1

U realizaciji JK bistabila pomoću SR bistabila, ulaze S i R promatramo kao logičke funkcije od J, K i Q_n ($S(J,K,Q_n)$, $R(J,K,Q_n)$). Potrebno je pronaći minimalne oblike logičkih funkcija $S(J,K,Q_n)$ i $R(J,K,Q_n)$ pomoću tablice 5.9. Minimizaciju ćemo provesti pomoću K-tablice.

$S(J,K,Q_n)$

Q_n \ JK	00	01	11	10
0			1	1
1	X			X

(a) K-tablica za logičku funkciju $S(J,K,Q_n)$

$R(J,K,Q_n)$

Q_n \ JK	00	01	11	10
0	X	X		
1		1	1	

(b) K-tablica za logičku funkciju $R(J,K,Q_n)$

Slika 5.9: K-tablica za logičke funkcije $S(J,K,Q_n)$ i $R(J,K,Q_n)$

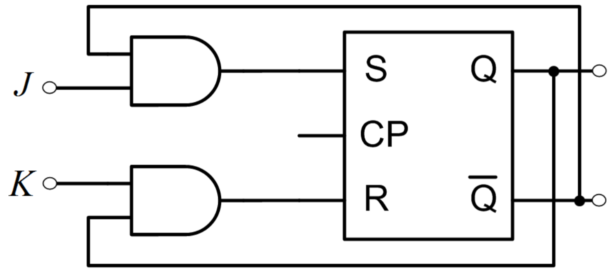
Minimizacija logičke funkcije $S(J,K,Q_n)$ K-tablicom prikazana je na slici 5.9a, a minimalni oblik je:

$$S(J,K,Q_n) = J\overline{Q}_n.$$

Minimizacija logičke funkcije $R(J,K,Q_n)$ K-tablicom prikazana je na slici 5.9b, a minimalni oblik je:

$$S(J,K,Q_n) = KQ_n.$$

Realizacija JK bistabila pomoću SR bistabila i minimalnog broja osnovnih logičkih sklopova prema izvedenim logičkim funkcijama $S(J,K,Q_n)$ i $R(J,K,Q_n)$ prikazana je na slici 5.10.



Slika 5.10: Realizacija JK bistabila pomoću SR bistabila i minimalnog broja osnovnih logičkih sklopova

Kako bismo pomoću JK bistabila ostvarili T bistabil, potrebno je prvo napisati tablicu kombinacija za T bistabil (tablica 5.5). Ulazi u novi bistabil bit će T i Q_n. U tu istu tablicu kombinacija potrebno je dodati stupce za ulaze JK bistabila J i K kako bi se pomoću tablice uzbude JK bistabila 5.4 ostvario prijelaz T bistabila iz stanja Q_n u Q_{n+1}. Tablica kombinacija za T bistabil proširena uzbudom JK bistabila prikazana je tablicom 5.10.

Tablica 5.10: Tablica kombinacija za T bistabil proširena uzbudom JK bistabila

i	T	Q_n	Q_{n+1}	J	K
0	0	0	0	0	X
1	0	1	1	X	0
2	1	0	1	1	X
3	1	1	0	X	1

(a) K-tablica za logičku funkciju $J(T,Q_n)$ (b) K-tablica za logičku funkciju $K(T,Q_n)$

Slika 5.11: K-tablica za logičke funkcije $J(T,Q_n)$ i $K(T,Q_n)$

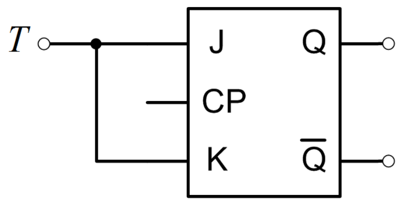
Minimizacija logičke funkcije $J(T,Q_n)$ K-tablicom prikazana je na slici 5.11a, a minimalni oblik je:

$$J(T,Q_n) = T.$$

Minimizacija logičke funkcije $K(T,Q_n)$ K-tablicom prikazana je na slici 5.11b, a minimalni oblik je:

$$K(T,Q_n) = T.$$

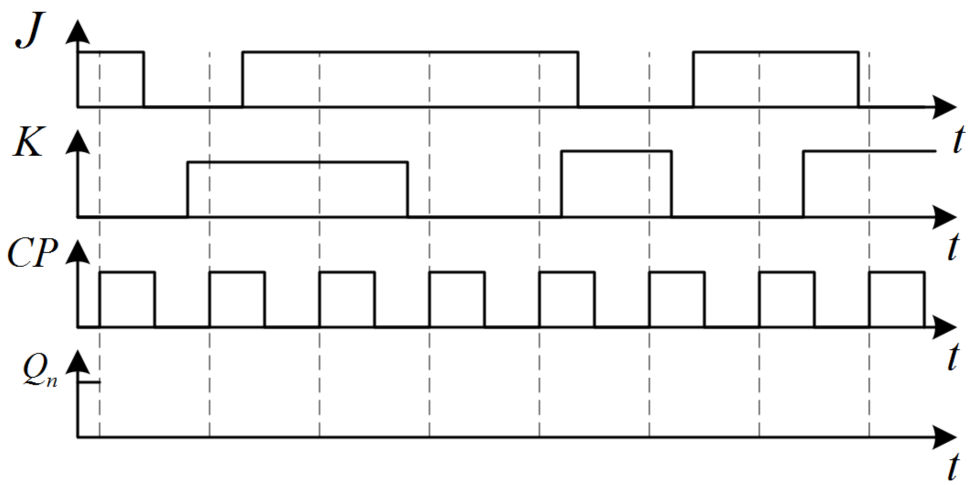
T bistabil dobit ćemo tako da kratko spojimo ulaze JK bistabila. Realizacija T bistabila pomoću JK bistabila i minimalnog broja osnovnih logičkih sklopova prema izvedenim logičkim funkcijama $J(T,Q_n)$ i $K(T,Q_n)$ prikazana je na slici 5.10.



Slika 5.12: Realizacija T bistabila pomoću JK bistabila i minimalnog broja osnovnih logičkih sklopova

☞ Primjer 5.1.2

Na slici 5.13 prikazan je vremenski dijagram JK bistabila.



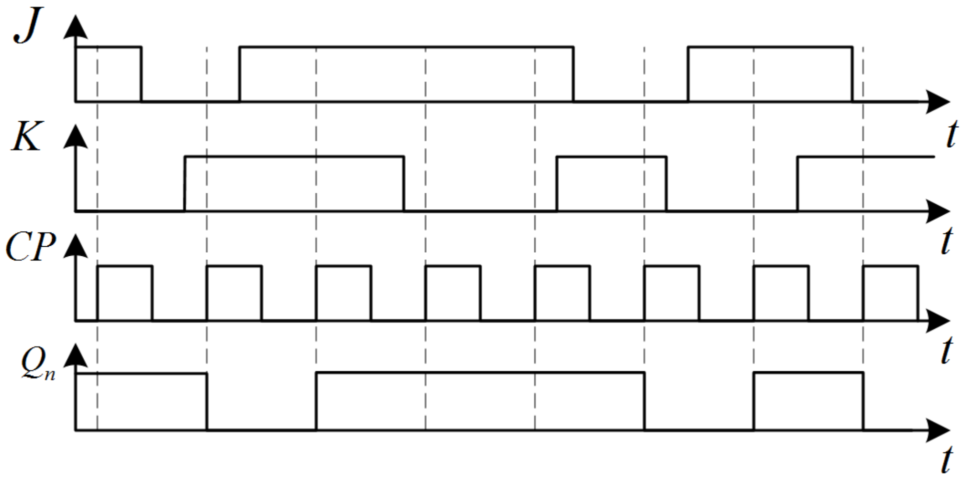
Slika 5.13: Vremenski dijagram JK bistabila

Pod pretpostavkom da pozitivni brid impulsa takta CP uzrokuje promjenu stanja JK bistabila, odredite vremenski odziv izlaza Q_n.

✍ **Rješenje:**

Vremenski odziv izlaza JK bistabila na slici 5.13 određuje se pomoću tablice stanja JK bistabila (tablica 5.3). Pri tome treba voditi računa kada nastupa pozitivan brid impulsa takta CP. Ako se ulazi JK bistabila mijenjaju za vrijeme trajanja CP impulsa, izlaz se neće mijenjati dok god ne nastupi novi pozitivan brid impulsa takta CP.

Prema vremenskom dijagramu na slici 5.13, početno stanje izlaza Q_n bilo je 1. Za vrijeme prvog impulsa takta CP ulaz JK bistabila bio je $JK = 10$. U tom slučaju izlaz bistabila Q_n mora poprimiti vrijednost 1 (u ovom se slučaju stanje izlaza ne mijenja). Za vrijeme drugog impulsa takta CP ulaz JK bistabila bio je $JK = 01$ pa će izlaz poprimiti vrijednost 0. Kod trećeg impulsa takta CP ulaz JK bistabila bio je $JK = 11$ pa će izlaz promijeniti svoju vrijednost, odnosno iz stanja 0 će preći u stanje 1. Po istom obrascu potrebno je završiti vremenski odziv izlaza bistabila Q_n. Vremenski dijagram JK bistabila s vremenskim odzivom izlaza Q_n prikazan je na slici 5.14.



Slika 5.14: Vremenski dijagram JK bistabila s vremenskim odzivom izlaza Q_n

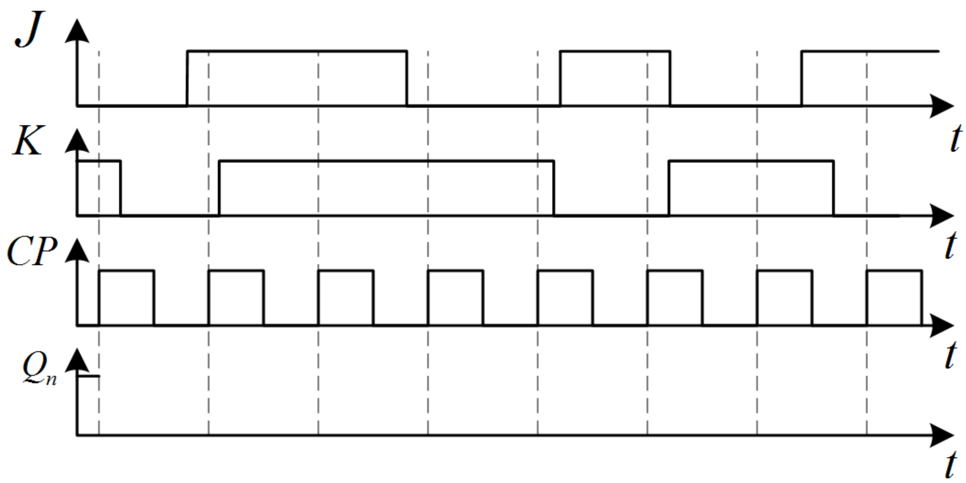
5.1.2 Zadaci za vježbu

✎ **Zadatak 5.1.1**

Pomoću JK bistabila i minimalnog broja osnovnih logičkih sklopova ostvarite D bistabil.

✎ **Zadatak 5.1.2**

Na slici 5.15 prikazan je vremenski dijagram JK bistabila.



Slika 5.15: Vremenski dijagram JK bistabila

Pod pretpostavkom da pozitivni brid impulsa takta CP uzrokuje promjenu stanja JK bistabila, odredite vremenski odziv izlaza Q_n.

Poglavlje 6

Sekvencijski sklopovi

Sekvencijski sklopovi imaju sposobnost memorije. Najjednostavniji sekvencijski sklopovi su bistabili koje smo opisali u prethodnom poglavlju. Upravo će bistabili, zbog svojstva memoriranja jednog bita, imati glavnu ulogu u gradnji sljedećih sekvencijskih sklopova koje ćemo opisati:

- prstenasto brojilo,
- Johnsonovo brojilo,
- binarno sinkrono brojilo,
- binarno asinkrono brojilo,
- binarno sinkrono dekadsko brojilo,
- generator niza,
- detektor niza.

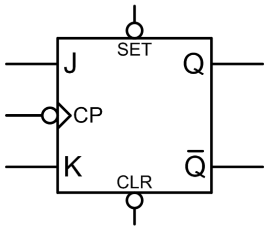
Sve navedene sekvencijske sklopove opisat ćemo primjerima.

6.1 Prstenasto brojilo

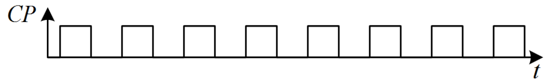
Prstenasto brojilo izvodi se tako da se n bistabila (npr. JK bistabila) serijski spoji na način da se izlazi prvog bistabila spoje na ulaze drugog ($Q_0 \to J_1$, $\overline{Q}_0 \to K_1$), izlazi drugog bistabila spoje na ulaze trećeg ($Q_1 \to J_2$, $\overline{Q}_1 \to K_2$) itd. Izlaz zadnjeg bistabila u serijskom spoju spaja se na ulaze prvog bistabila ($Q_{n-1} \to J_0$, $\overline{Q}_{n-1} \to K_0$).

☞ Primjer 6.1.1

Nacrtajte shemu 4-bitnog prstenastog brojila pomoću JK bistabila prikazanog na slici 6.1a. Za takt impulse CP na slici 6.1b skicirajte oblik napona (vremenski dijagram) izlaza iz bistabila. Napišite tablicu stanja 4-bitnog prstenastog brojila. Proučite ponašanje sklopa za sve početne 4-bitne kombinacije.



(a) JK bistabil za prstenasto brojilo



(b) CP impulsi takta

Slika 6.1: JK bistabil za prstenasto brojilo i CP impulsi takta

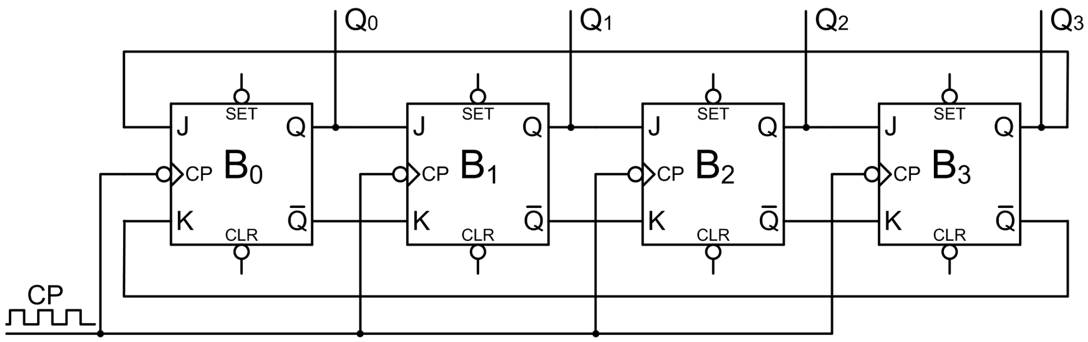
---

✍ Rješenje:

JK bistabil sa slike 6.1a ima dva ulaza J i K koji djeluju sinkrono s takt impulsima CP. Oznaka $>$ na CP ulazu JK bistabila označava da će bistabil mijenjati stanje na brid impulsa takta CP. Ako je na ulazu CP oznaka $\circ$ (kružić)[1], tada će bistabil mijenjati stanje izlaza na padajući brid takt impulsa CP, a ako na ulazu CP nema kružića, tada će bistabil mijenjati stanje izlaza na rastući brid takt impulsa CP.

JK bistabil sa slike 6.1a ima i dva dodatna ulaza: SET i CLR. Ulaz SET služi za postavljanje bistabila u vrijednost 1 neovisno o J i K ulazima, dok ulaz CLR (eng. *clear*) služi za postavljanje bistabila u vrijednost 0 neovisno o J i K ulazima. Ovi ulazi mogu biti izvedeni tako da djeluju asinkrono (odmah po pojavi aktivnog signala) ili sinkrono s impulsima takta CP. Ako SET i CLR ulazi imaju oznaku $\circ$, aktivni signal je logička vrijednost 0, a ako nemaju oznaku $\circ$, aktivni signal je logička vrijednost 1.

Shema 4-bitnog prstenastog brojila prikazana je na slici 6.2. JK bistabili spojeni su u prsten. Informacija koja je upisana u prstenasto brojilo cirkulira kroz brojilo pod utjecajem impulsa takta CP koji se istodobno dovode na sve CP ulaze bistabila. Zadani JK bistabil (slika 6.1a) mijenja stanje na padajući brid takt impulsa.



Slika 6.2: Shema 4-bitnog prstenastog brojila

Ponašanje izlaza Q_0, Q_1, Q_2 i Q_3 prstenastog brojila ovisi o tome koja je početna kombinacija logičkih vrijednosti 1 i 0 upisana u JK bistabile. Ako se u bistabile upiše kombinacija 1000 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$), prstenasto brojilo prolazit će kroz četiri stanja prikazana u tablici 6.1.

---

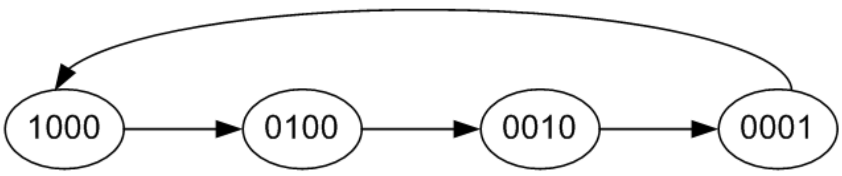
[1]Kružić $\circ$ označava negaciju.

Tablica 6.1: Tablica stanja prstenastog brojila uz početnu kombinaciju 1000

CP	Stanja bistabila			
n	Q_0	Q_1	Q_2	Q_3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

Prstenasto brojilo je sinkrono brojilo jer se promjene stanja događaju sinkronizirano s impulsima takta CP. U početnom stanju za izlaze bistabila vrijedi: $Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$. Invertirani izlazi bistabila su u početnom stanju: $\overline{Q}_0 = 0$, $\overline{Q}_1 = 1$, $\overline{Q}_2 = 1$, $\overline{Q}_3 = 1$. Promjena stanja izlaza prstenastog brojila uslijedit će na prvi padajući brid takt impulsa CP. Važno je znati da će do promjena stanja svih bistabila doći u isto vrijeme[2], na padajući brid takt impulsa CP. Neposredno prije prvog padajućeg brida takt impulsa CP, stanja ulaza bistabila su sljedeća: $J_1 = Q_0 = 1$, $J_2 = Q_1 = 0$, $J_3 = Q_2 = 0$, $J_0 = Q_3 = 0$, $K_1 = \overline{Q}_0 = 0$, $K_2 = \overline{Q}_1 = 1$, $K_3 = \overline{Q}_2 = 1$ i $K_0 = \overline{Q}_3 = 1$. Budući da bistabil B_1 na svom ulazu ima kombinaciju $JK = 10$, a ostali bistabili (B_0, B_2 **i** B_3) imaju kombinaciju $JK = 01$, dolaskom prvog padajućeg brida takt impulsa CP prstenasto brojilo iz početnog stanja 1000 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$) prelazi u stanje 0100 ($Q_0 = 0$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$). Drugi padajući brid takt impulsa postavlja prstenasto brojilo u stanje 0010 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 0$), treći u stanje 0001 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 1$), a četvrti ponovno u početno stanje 1000 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$). Prstenasto brojilo sa četiri bistabila broji do 4. Općenito, prstenasto brojilo sačinjeno od n bistabila broji do n.

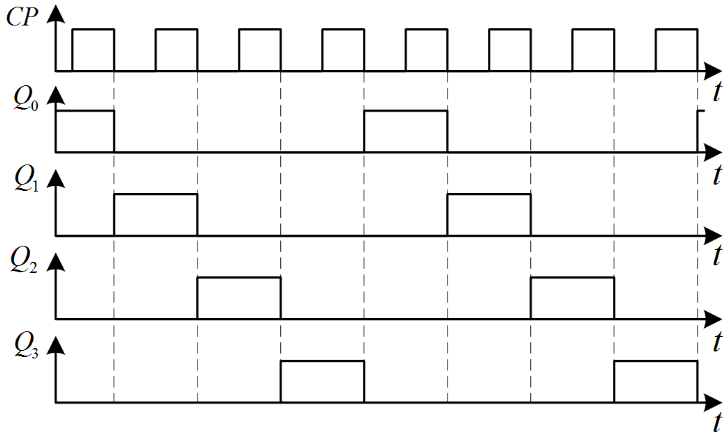
Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1000 prikazan je na slici 6.3. Opisano ponašanje prstenastog brojila vrijedi ako se u brojilo početno upiše bilo koja od kombinacija: 1000, 0100, 0010, 0001.



Slika 6.3: Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1000

Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1000 prikazan je na slici 6.4. Na slici 6.4 možemo primijetiti da se za svaka četiri CP impulsa na svakom od izlaza bistabila dobije jedan impuls, što znači da sklop dijeli frekvenciju takt impulsa CP sa četiri.

[2]Promjena stanja izlaza prvog bistabila u koraku n neće utjecati na promjenu stanja drugog bistabila u koraku n.



Slika 6.4: Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1000

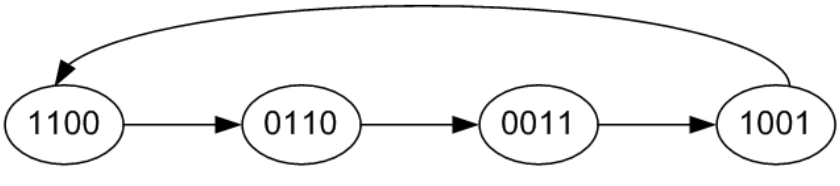
Ako se u bistabile upiše kombinacija 1100 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$), prstenasto brojilo prolazit će kroz četiri stanja prikazana u tablici 6.2.

Tablica 6.2: Tablica stanja prstenastog brojila uz početnu kombinaciju 1100

CP	Stanja bistabila			
n	Q_0	Q_1	Q_2	Q_3
0	1	1	0	0
1	0	1	1	0
2	0	0	1	1
3	1	0	0	1
4	1	1	0	0

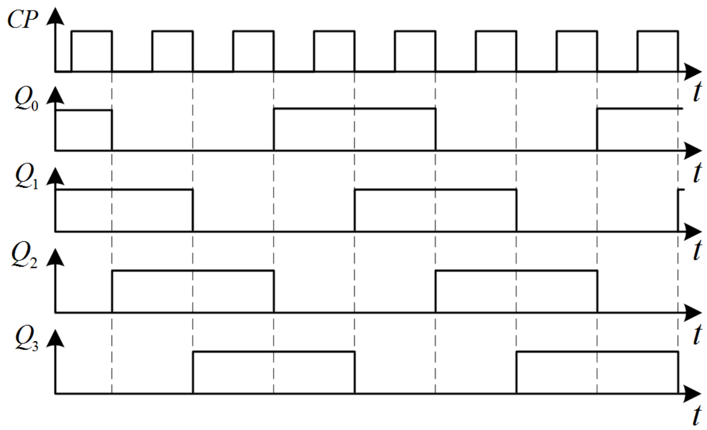
U početnom stanju za izlaze bistabila vrijedi: $Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$. Invertirani su izlazi bistabila u početnom stanju: $\overline{Q}_0 = 0$, $\overline{Q}_1 = 0$, $\overline{Q}_2 = 1$, $\overline{Q}_3 = 1$. Promjena stanja izlaza prstenastog brojila uslijedit će na prvi padajući brid takt impulsa CP. Neposredno prije prvog padajućeg brida takt impulsa CP, stanja ulaza bistabila su sljedeća: $J_1 = Q_0 = 1$, $J_2 = Q_1 = 1$, $J_3 = Q_2 = 0$, $J_0 = Q_3 = 0$, $K_1 = \overline{Q}_0 = 0$, $K_2 = \overline{Q}_1 = 0$, $K_3 = \overline{Q}_2 = 1$ i $K_0 = \overline{Q}_3 = 1$. Budući da bistabili B_1 i B_2 na svom ulazu imaju kombinaciju $JK = 10$, a ostali bistabili (B_0 i B_3) imaju kombinaciju $JK = 01$, dolaskom prvog padajućeg brida takt impulsa CP prstenasto brojilo iz početnog stanja 1100 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$) prelazi u stanje 0110 ($Q_0 = 0$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 0$). Drugi padajući brid takt impulsa postavlja prstenasto brojilo u stanje 0011 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 1$), treći u stanje 1001 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 1$), a četvrti ponovno u početno stanje 1100 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$).

Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1100 prikazan je na slici 6.5. Opisano ponašanje prstenastog brojila vrijedi ako se u brojilo početno upiše bilo koja od kombinacija: 1100, 0110, 0011, 1001.



Slika 6.5: Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1100

Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1100 prikazan je na slici 6.6. Na slici 6.6 možemo primijetiti da se za svaka četiri CP impulsa na svakom od izlaza bistabila dobije jedan impuls, što znači da sklop dijeli frekvenciju takt impulsa CP sa četiri.



Slika 6.6: Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1100

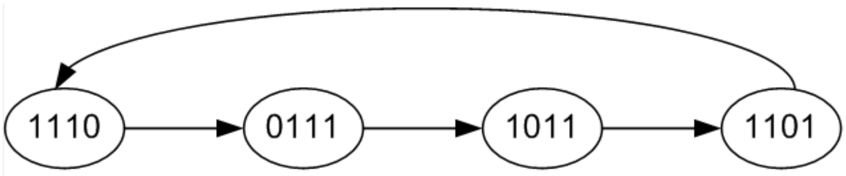
Ako se u bistabile upiše kombinacija 1110 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 0$), prstenasto brojilo prolazit će kroz četiri stanja prikazana u tablici 6.3.

Tablica 6.3: Tablica stanja prstenastog brojila uz početnu kombinaciju 1110

CP	Stanja bistabila			
n	Q_0	Q_1	Q_2	Q_3
0	1	1	1	0
1	0	1	1	1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	0

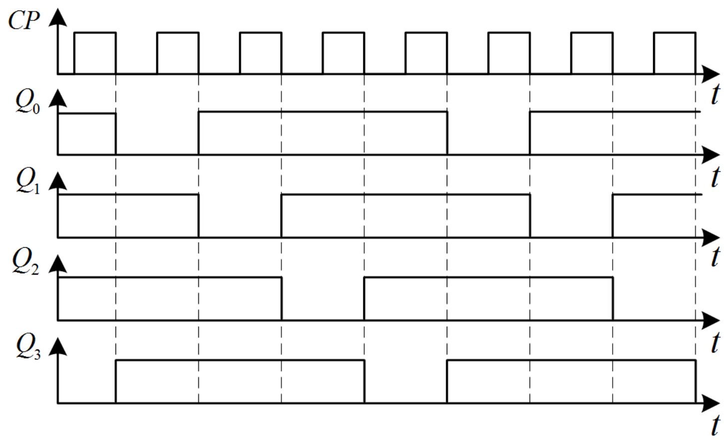
U početnom stanju za izlaze bistabila vrijedi: $Q_0 = 1$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 0$. Invertirani su izlazi bistabila u početnom stanju: $\overline{Q}_0 = 0$, $\overline{Q}_1 = 0$, $\overline{Q}_2 = 0$, $\overline{Q}_3 = 1$. Promjena stanja izlaza prstenastog brojila uslijedit će na prvi padajući brid takt impulsa CP. Neposredno prije prvog padajućeg brida takt impulsa CP, stanja ulaza bistabila su sljedeća: $J_1 = Q_0 = 1$, $J_2 = Q_1 = 1$, $J_3 = Q_2 = 1$, $J_0 = Q_3 = 0$, $K_1 = \overline{Q}_0 = 0$, $K_2 = \overline{Q}_1 = 0$, $K_3 = \overline{Q}_2 = 0$ i $K_0 = \overline{Q}_3 = 1$. Budući da bistabili B_1, B_2 i B_3 na svom ulazu imaju kombinaciju $JK = 10$, a bistabil B_0 ima kombinaciju $JK = 01$, dolaskom prvog padajućeg brida takt impulsa CP prstenasto brojilo iz početnog stanja 1110 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 0$) prelazi u stanje 0111 ($Q_0 = 0$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 1$). Drugi padajući brid takt impulsa postavlja prstenasto brojilo u stanje 1011 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 1$), treći u stanje 1101 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 1$), a četvrti ponovno u početno stanje 1110 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 0$).

Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1110 prikazan je na slici 6.7. Opisano ponašanje prstenastog brojila vrijedi ako se u brojilo početno upiše bilo koja od kombinacija: 1110, 0111, 1011, 1101.



Slika 6.7: Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1110

Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1110 prikazan je na slici 6.8. Na slici 6.8 možemo primijetiti da se za svaka četiri CP impulsa na svakom od izlaza bistabila dobije jedan impuls, što znači da sklop dijeli frekvenciju takt impulsa CP sa četiri.



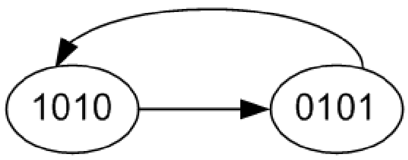
Slika 6.8: Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1110

Ako se u bistabile upiše kombinacija 1010 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 0$), prstenasto brojilo prolazit će kroz dva stanja prikazana u tablici 6.3.

Tablica 6.4: Tablica stanja prstenastog brojila uz početnu kombinaciju 1010

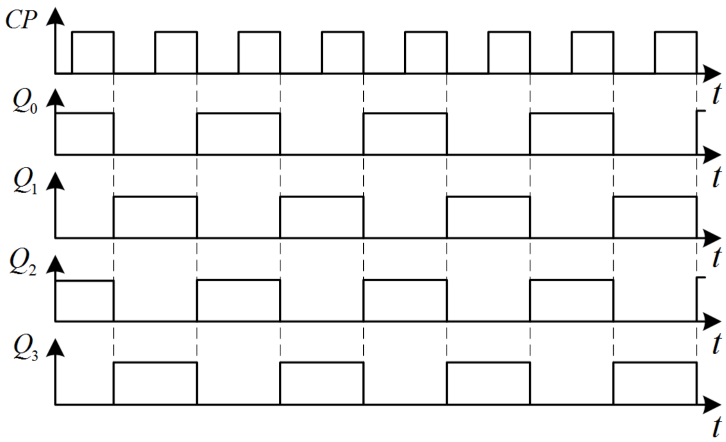
CP	Stanja bistabila			
n	Q_0	Q_1	Q_2	Q_3
0	1	0	1	0
1	0	1	0	1
2	1	0	1	0

U početnom stanju za izlaze bistabila vrijedi: $Q_0 = 1$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 0$. Invertirani su izlazi bistabila u početnom stanju: $\overline{Q}_0 = 0$, $\overline{Q}_1 = 1$, $\overline{Q}_2 = 0$, $\overline{Q}_3 = 1$. Promjena stanja izlaza prstenastog brojila uslijedit će na prvi padajući brid takt impulsa CP. Neposredno prije prvog padajućeg brida takt impulsa CP, stanja ulaza bistabila su sljedeća: $J_1 = Q_0 = 1$, $J_2 = Q_1 = 0$, $J_3 = Q_2 = 1$, $J_0 = Q_3 = 0$, $K_1 = \overline{Q}_0 = 0$, $K_2 = \overline{Q}_1 = 1$, $K_3 = \overline{Q}_2 = 0$ i $K_0 = \overline{Q}_3 = 1$. Budući da bistabili B_1 i B_3 na svom ulazu imaju kombinaciju $JK = 10$, a ostali bistabili (B_0 **i** B_2) imaju kombinaciju $JK = 01$, dolaskom prvog padajućeg brida takt impulsa CP prstenasto brojilo iz početnog stanja 1010 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 0$) prelazi u stanje 0101 ($Q_0 = 0$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 1$). Drugi padajući brid takt impulsa postavlja prstenasto brojilo u početno stanje 1010 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 1$, $Q_3 = 0$). Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1010 prikazan je na slici 6.9. Opisano ponašanje prstenastog brojila vrijedi ako se u brojilo početno upiše bilo koja od kombinacija: 1010, 0101.



Slika 6.9: Dijagram stanja 4-bitnog prstenastog brojila za početnu kombinaciju 1010

Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1010 prikazan je na slici 6.10. Na slici 6.10 možemo primijetiti da se za svaka četiri CP impulsa na svakom od izlaza bistabila dobije jedan impuls, što znači da sklop dijeli frekvenciju takt impulsa CP s dva.



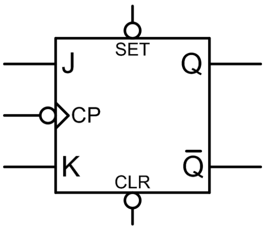
Slika 6.10: Vremenski dijagram napona na izlazima bistabila 4-bitnog prstenastog brojila uz početnu kombinaciju 1010

Ako se u prstenasto brojilo početno upiše kombinacija 0000 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$) ili 1111 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 1$), dolaskom takt impulsa neće se dogoditi nikakva promjena jer svi bistabili na ulazu imaju kombinaciju $JK = 01$, odnosno $JK = 10$, što osigurava ostanak u istom stanju.

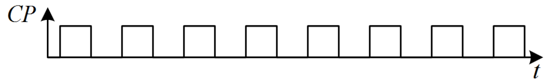
6.1.1 Zadaci za vježbu

✎ Zadatak 6.1.1

Nacrtajte shemu 3-bitnog prstenastog brojila pomoću JK bistabila prikazanog na slici 6.11a. Za takt impulse CP na slici 6.11b skicirajte oblik napona (vremenski dijagram) izlaza iz bistabila. Napišite tablicu stanja 3-bitnog prstenastog brojila. Proučite ponašanje sklopa za sve početne 3-bitne kombinacije.



(a) JK bistabil za prstenasto brojilo



(b) CP impulsi takta

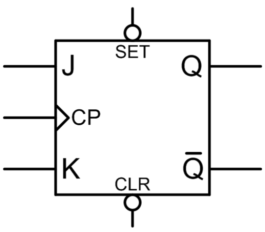
Slika 6.11: JK bistabil za prstenasto brojilo i CP impulsi takta

6.2 Johnsonovo brojilo

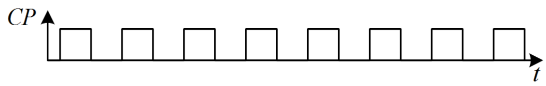
Johnsonovo brojilo izvodi se tako da se n bistabila (npr. JK bistabila) serijski spoji na način da se izlazi prvog bistabila spoje na ulaze drugog ($Q_0 \rightarrow J_1$, $\overline{Q}_0 \rightarrow K_1$), izlazi drugog bistabila na ulaze trećeg ($Q_1 \rightarrow J_2$, $\overline{Q}_1 \rightarrow K_2$) itd. Izlaz zadnjeg bistabila u serijskom spoju spaja se ukršteno na ulaze prvog bistabila ($Q_{n-1} \rightarrow K_0$, $\overline{Q}_{n-1} \rightarrow J_0$). U literaturi se ovo brojilo naziva još i ukršteno brojilo.

☞ Primjer 6.2.1

Nacrtajte shemu 4-bitnog Johnsonovog brojila pomoću JK bistabila prikazanog na slici 6.12a. Za takt impulse CP na slici 6.12b skicirajte oblik napona (vremenski dijagram) izlaza iz bistabila. Napišite tablicu stanja 4-bitnog Johnsonovog brojila. Proučite ponašanje sklopa za sve početne 4-bitne kombinacije.



(a) JK bistabil za Johnsonovo brojilo

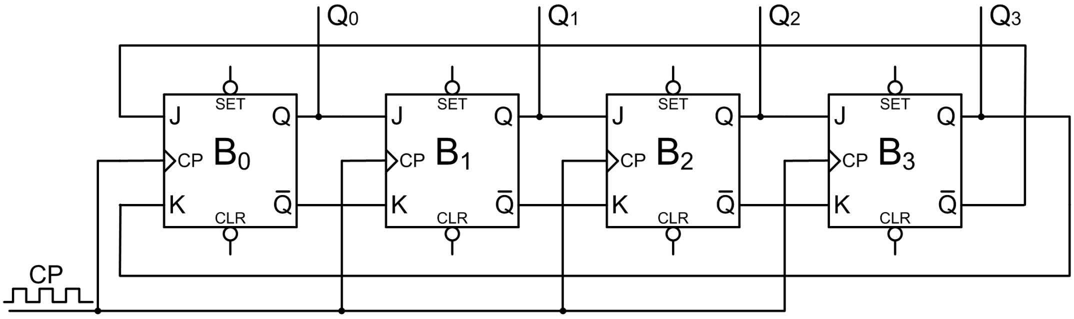


(b) CP impulsi takta

Slika 6.12: JK bistabil za Johnsonovo brojilo i CP impulsi takta

✍ Rješenje:

JK bistabil sa slike 6.12a na ulazu CP nema oznaku ○ (kružić) pa će bistabil mijenjati stanje izlaza na rastući brid takt impulsa CP. Shema 4-bitnog Johnsonovog brojila prikazana je na slici 6.13. JK bistabili spojeni su u ukršteni prsten. Informacija koja je upisana u Johnsonovo brojilo cirkulira kroz brojilo pod utjecajem impulsa takta CP koji se istodobno dovode na sve CP ulaze bistabila.



Slika 6.13: Shema 4-bitnog Johnsonovog brojila

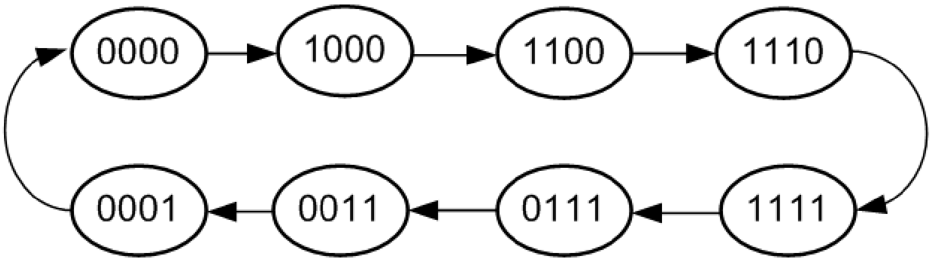
Ponašanje izlaza Q_0, Q_1, Q_2 i Q_3 Johnsonovog brojila ovisi o tome koja je početna kombinacija logičkih vrijednosti 1 i 0 upisana u JK bistabile. Ako se u bistabile upiše kombinacija 0000 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$), prstenasto brojilo prolazit će kroz osam stanja prikazanih u tablici 6.5.

Tablica 6.5: Tablica stanja Johnsonovog brojila uz početnu kombinaciju 0000

CP	Stanja bistabila			
n	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

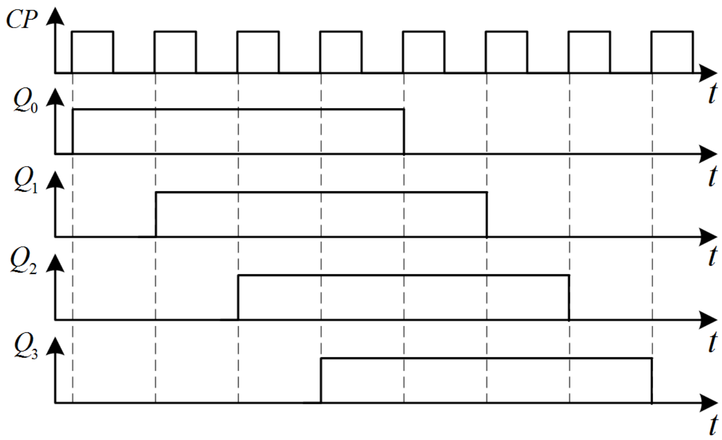
U početnom stanju za izlaze bistabila vrijedi: $Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$. Invertirani su izlazi bistabila u početnom stanju: $\overline{Q}_0 = 1$, $\overline{Q}_1 = 1$, $\overline{Q}_2 = 1$, $\overline{Q}_3 = 1$. Promjena stanja izlaza Johnsonovog brojila uslijedit će na prvi rastući brid takt impulsa CP. Neposredno prije prvog rastućeg brida takt impulsa CP, stanja ulaza bistabila su sljedeća: $J_1 = Q_0 = 0$, $J_2 = Q_1 = 0$, $J_3 = Q_2 = 0$, $J_0 = Q_3 = 1$, $K_1 = \overline{Q}_0 = 1$, $K_2 = \overline{Q}_1 = 1$, $K_3 = \overline{Q}_2 = 1$ i $K_0 = \overline{Q}_3 = 0$. Budući da bistabil B_0 na svom ulazu ima kombinaciju $JK = 10$, a ostali bistabili (B_1, B_2 i B_3) imaju kombinaciju $JK = 01$, dolaskom prvog rastućeg brida takt impulsa CP Johnsonovo brojilo iz početnog stanja 0000 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$) prelazi u stanje 1000 ($Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$). Drugi rastući brid takt impulsa postavlja Johnsonovo brojilo u stanje 1100 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$), treći u stanje 1110 ($Q_0 = 1$, $Q_1 = 1$, $Q_2 = 1$, $Q_3 = 0$) itd. Nakon sedmog rastućeg brida takt impulsa, Johnsonovo brojilo postavlja se u stanje 0001 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 1$), a nakon osmog rastućeg brida takt impulsa u početno stanje 0000 ($Q_0 = 0$, $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$). Johnsonovo brojilo sa četiri bistabila broji do 8. Općenito, Johnsonovo brojilo sačinjeno od n bistabila broji do $2n$.

Dijagram stanja 4-bitnog Johnsonovog brojila za početnu kombinaciju 0000 prikazan je na slici 6.14. Opisano ponašanje Johnsonovog brojila vrijedi ako se u brojilo početno upiše bilo koja od kombinacija: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001.



Slika 6.14: Dijagram stanja 4-bitnog Johnsonovog brojila za početnu kombinaciju 0000

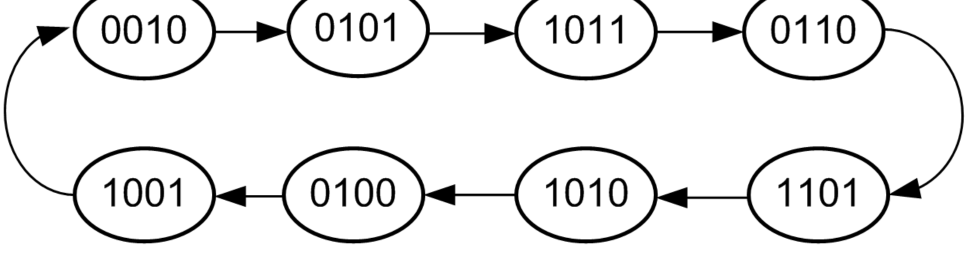
Vremenski dijagram napona na izlazima bistabila 4-bitnog Johnsonovog brojila uz početnu kombinaciju 0000 prikazan je na slici 6.8. Sa slike 6.8 možemo primijetiti da se za svakih osam CP impulsa na svakom od izlaza bistabila dobije jedan impuls, što znači da sklop dijeli frekvenciju takt impulsa CP s osam.



Slika 6.15: Vremenski dijagram napona na izlazima bistabila 4-bitnog Johnsonovog brojila uz početnu kombinaciju 0000

Dijagram stanja 4-bitnog Johnsonovog brojila za početnu kombinaciju 0010 prikazan je na slici 6.16. Ovo je drugi ciklus od osam stanja kroz koje Johnsonovo brojilo može prolaziti. Provjerite!

Stanja na slici 6.16 nazivaju se neregularna stanja, a Johnsonovo se brojilo u njima može naći ako neki bistabil promjeni svoje stanje zbog vanjskih smetnji. Dijagram stanja sa slike 6.14 predstavlja regularna stanja Johnsonovog brojila. Ako se Johnsonovo brojilo nađe u nekom od neregularnih stanja prikazanih na slici 6.16, potrebno je registrirati to stanje te asinkronim ulazima Johnsonovo brojilo postaviti u stanje 0000 ili 1111. Takav način asinkronog djelovanja na brojilo naziva se osiguranje sigurnog starta.



Slika 6.16: Dijagram stanja 4-bitnog Johnsonovog brojila za početnu kombinaciju 0010

☞ Primjer 6.2.2

Osigurajte siguran start 4-bitnog Johnsonovog brojila sa slike 6.13 tako da neregularna stanja prijeđu u stanje 0000.

✍ Rješenje:

Tablica regularnih i neregularnih stanja 4-bitnog Johnsonovog brojila prikazana je tablicom 6.6. U tablici 6.6 nalazi se stupac s logičkom funkcijom $N(Q_0,Q_1,Q_2,Q_3)$ koja poprima vrijednost 1 ako je Johnsonovo brojilo u neregularnom stanju (stanja prikazana dijagramom sa slike 6.16).

Tablica 6.6: Tablica regularnih i neregularnih stanja 4-bitnog Johnsonovog brojila

Indeks minterma	Stanja bistabila				Neregularno stanje
i	Q_0	Q_1	Q_2	Q_3	N
0	0	0	0	0	0
8	1	0	0	0	0
12	1	1	0	0	0
14	1	1	1	0	0
15	1	1	1	1	0
7	0	1	1	1	0
3	0	0	1	1	0
1	0	0	0	1	0
2	0	0	1	0	1
5	0	1	0	1	1
11	1	0	1	1	1
6	0	1	1	0	1
13	1	1	0	1	1
10	1	0	1	0	1
4	0	1	0	0	1
9	1	0	0	1	1

Siguran start osigurat ćemo detekcijom neregularnih stanja. Detekciju neregularnih stanja radi logička funkcija $N(Q_0,Q_1,Q_2,Q_3)$ koja ima vrijednost 1 za osam minterma. Potrebno je minimizirati logičku funkciju $N(Q_0,Q_1,Q_2,Q_3)$. Minimizacija je prikazana na slici 6.17.

Q_2Q_3 \ Q_0Q_1	00	01	11	10
00		1		
01		1	1	1
11				1
10	1	1		1

Slika 6.17: Minimizacija logičke funkcije $N(Q_0,Q_1,Q_2,Q_3)$ K-tablicom

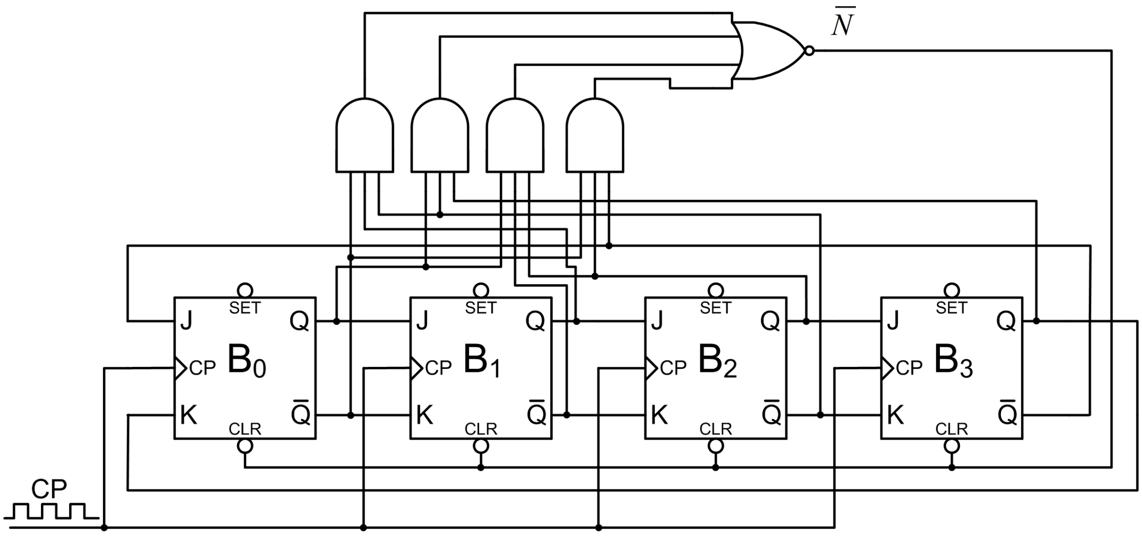
Minimalni oblik logičke funkcije $N(Q_0,Q_1,Q_2,Q_3)$ je:

$$N = \overline{Q}_0 Q_1 \overline{Q}_2 + Q_0 \overline{Q}_2 Q_3 + Q_0 \overline{Q}_1 Q_2 + \overline{Q}_0 Q_2 \overline{Q}_3.$$

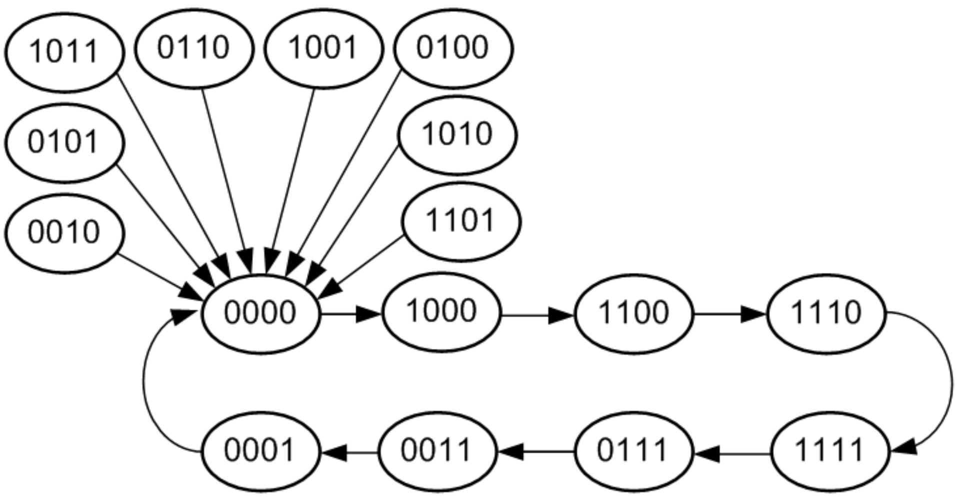
Izvedeni minimalni oblik logičke funkcije daje vrijednost 1 kada je sklop u neregularnom stanju. Asinkroni ulaz CLR ima $\circ$ (kružić) pa je vrijednost 0 ona vrijednost koja bistabil postavlja u stanje 0. Prema tome, minimalni oblik potrebno je komplementirati:

$$\overline{N} = \overline{\overline{Q}_0 Q_1 \overline{Q}_2 + Q_0 \overline{Q}_2 Q_3 + Q_0 \overline{Q}_1 Q_2 + \overline{Q}_0 Q_2 \overline{Q}_3}.$$

Produkte koji se nalaze u minimalnom obliku dovest ćemo na ulaz NILI sklopa. Shema 4-bitnog Johnsonovog brojila sa sklopom za siguran start prikazana je na slici 6.18. Invertirane vrijednosti izlaza bistabila uzete su direktno s bistabila. Dijagram stanja 4-bitnog Johnsonovog brojila sa sklopom za siguran start prikazan je na slici 6.19.



Slika 6.18: Shema 4-bitnog Johnsonovog brojila sa sklopom za siguran start

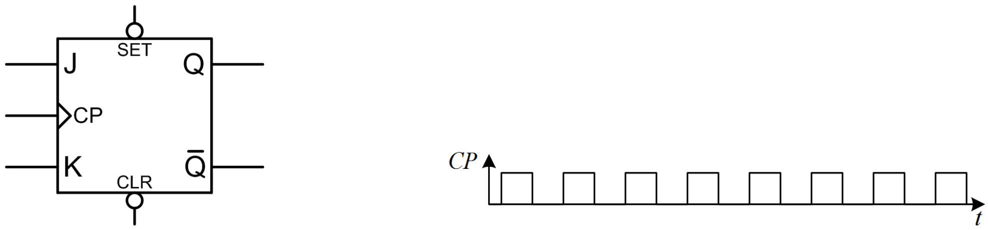


Slika 6.19: Dijagram stanja 4-bitnog Johnsonovog brojila sa sklopom za siguran start

6.2.1 Zadaci za vježbu

✎ Zadatak 6.2.1

Nacrtajte shemu 3-bitnog Johnsonovog brojila pomoću JK bistabila prikazanog na slici 6.20. Za takt impulse CP na slici 6.20b skicirajte oblik napona (vremenski dijagram) izlaza iz bistabila. Napišite tablicu stanja 3-bitnog Johnsonovog brojila. Proučite ponašanje sklopa za sve početne 3-bitne kombinacije.



(a) JK bistabil za Johnsonovo brojilo (b) CP impulsi takta

Slika 6.20: JK bistabil za Johnsonovo brojilo i CP impulsi takta

---

✎ Zadatak 6.2.2

Osigurajte siguran start 3-bitnog Johnsonovog brojila tako da neregularna stanja prijeđu u stanje 111.

6.3 Binarno sinkrono brojilo

Binarno sinkrono brojilo je brojilo koje prolazi kroz binarne kombinacije. Za 4-bitno brojilo te kombinacije bile bi: 0000, 0001, 0010, ..., 1110, 1111. Budući da se radi o binarnom sinkronom brojilu, kombinacije se izmjenjuju sinkrono s takt impulsom CP.

☞ Primjer 6.3.1

Projektirajte 4-bitno binarno sinkrono brojilo koristeći JK bistabile sa slike 6.12a. Koji bistabili su najpogodniji za realizaciju binarnih brojila?

✍ **Rješenje:** Tablica prijelaza 4-bitnog binarnog sinkronog brojila i uzbude JK bistabila prikazana je tablicom 6.7. U tablici 6.7 prikazana su sadašnja stanja bistabila u koraku n i sljedeća stanja bistabila u koraku $n+1$. U stupcu za korak n nalaze se 4-bitne kombinacije kroz koje prolazi binarno sinkrono brojilo. Stupac za korak $n+1$ prikazuje sljedeću kombinaciju u koju se bistabili moraju postaviti pri rastućem bridu takt impulsa CP. Na primjer, ako je u koraku n binarno sinkrono brojilo u stanju 0000, tada će u koraku $n+1$ biti u stanju 0001. Ako je u koraku n binarno sinkrono brojilo u stanju 0001, tada će u koraku $n+1$ biti u stanju 0010 itd. Konačno, kada se binarno sinkrono brojilo u koraku n nalazi u stanju 1111, tada će u koraku $n+1$ biti ponovno u stanju 0000. Ovo brojilo broji do 16. Općenito, n-bitno binarno sinkrono brojilo broji do 2^n. Da bi JK bistabil prešao iz stanja u koraku n u novo stanje u koraku $n+1$, na ulaze J i

K potrebno je dovesti pobudu koja će na rastući brid takt impulsa CP izazvati traženi prijelaz. Pomoću tablice uzbude JK bistabila (tablica 5.4) popunjeni su stupci uzbude za bistabile B_0, B_1, B_2 i B_3 u tablici 6.7. Na primjer, pri prijelazu iz stanja 0000 ($Q_3 = 0$, $Q_2 = 0$, $Q_1 = 0$, $Q_0 = 0$) u stanje 0001 ($Q_3 = 0$, $Q_2 = 0$, $Q_1 = 0$, $Q_0 = 1$) bistabil B_0 mora promijeniti stanje iz 0 u 1, a bistabili B_1, B_2 i B_3 ostaju u stanju 0. Da bi bistabil B_0 prešao iz stanja 0 u stanje 1, na ulaze mu je prema tablici uzbude 5.4 potrebno dovesti kombinaciju $JK = 1X$. Razlog tome je što će bistabil B_0 iz stanja 0 u stanje 1 prijeći i za kombinaciju $JK = 10$ koja uvijek JK bistabil postavlja u stanje 1 i za kombinaciju $JK = 11$ koja uvijek mijenja stanje JK bistabila (iz stanja 0 u 1). Bistabili B_1, B_2 i B_3 ostaju u stanju 0 pa je na njihove ulaze prema tablici uzbude 5.4 potrebno dovesti kombinaciju $JK = 0X$.

Tablica 6.7: Tablica prijelaza 4-bitnog binarnog sinkronog brojila i uzbude JK bistabila

	Sadašnje stanje bistabila, n				Sljedeće stanje bistabila, $n+1$				Uzbuda za bistabile							
									B_3		B_2		B_1		B_0	
i	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
1	0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
2	0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
3	0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
4	0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
5	0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
6	0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
7	0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
8	1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
9	1	0	0	1	1	0	1	0	X	0	0	X	1	X	X	1
10	1	0	1	0	1	0	1	1	X	0	0	X	X	0	1	X
11	1	0	1	1	1	1	0	0	X	0	1	X	X	1	X	1
12	1	1	0	0	1	1	0	1	X	0	X	0	0	X	1	X
13	1	1	0	1	1	1	1	0	X	0	X	0	1	X	X	1
14	1	1	1	0	1	1	1	1	X	0	X	0	X	0	1	X
15	1	1	1	1	0	0	0	0	X	1	X	1	X	1	X	1

Stupci uzbude bistabila u tablici 6.7 logičke su funkcije varijabli Q_3, Q_2,Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaze JK bistabila. Minimizacija je prikazana na slikama 6.21, 6.22, 6.23 i 6.24. Radi lakšeg popunjavanja K-tablice u tablici 6.7 prvi stupac je stupac indeksa minterma i.

Minimalni oblik logičke funkcije $J_0(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.21a:

$$J_0 = 1.$$

Minimalni oblik logičke funkcije $K_0(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.21b:

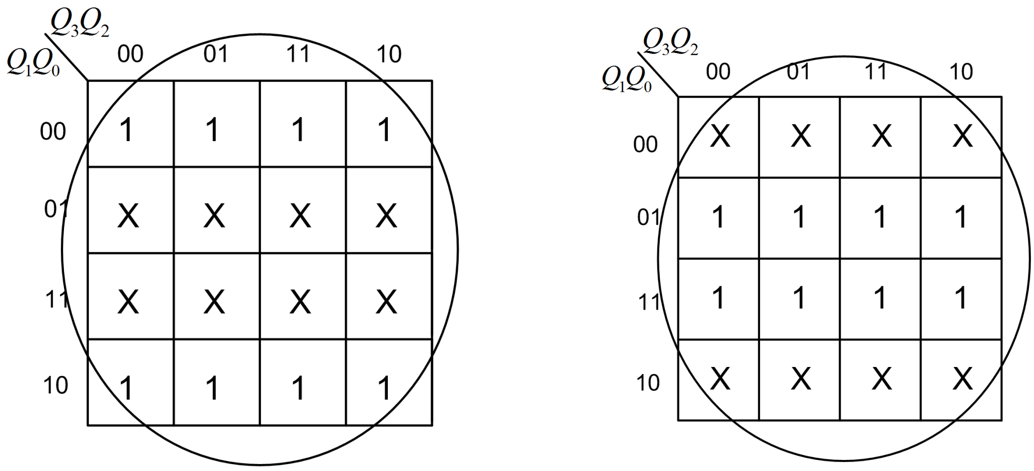
$$K_0 = 1.$$

Minimalni oblik logičke funkcije $J_1(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.22a:

$$J_1 = Q_0.$$

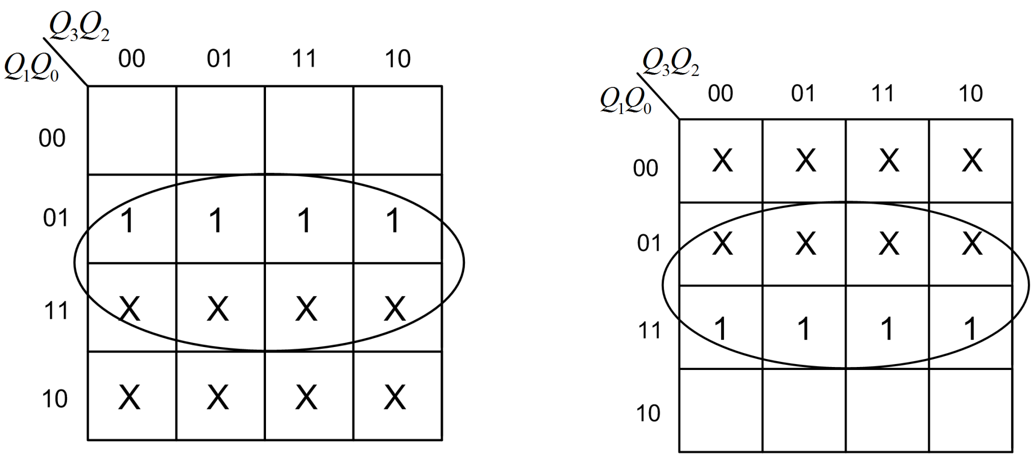
Minimalni oblik logičke funkcije $K_1(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.22b:

$$K_1 = Q_0.$$



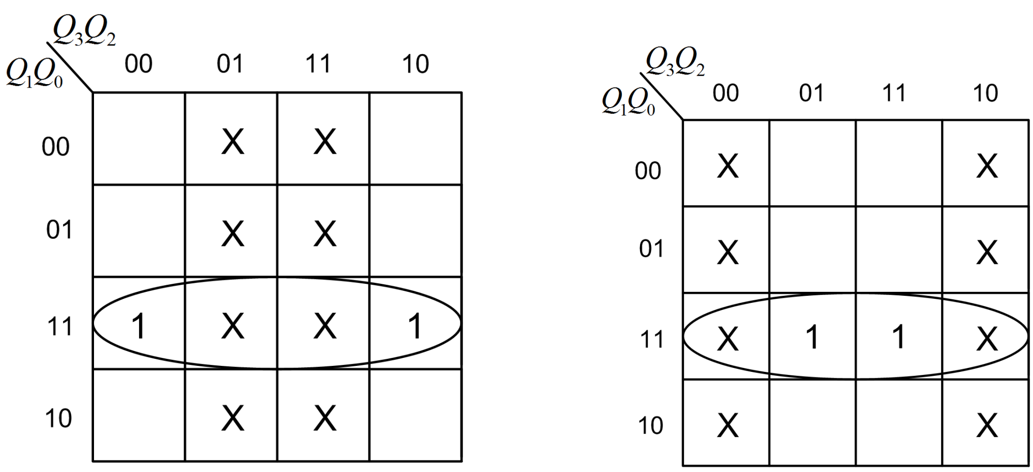
(a) K-tablica za $J_0(Q_3,Q_2,Q_1,Q_0)$ (b) K-tablica za $K_0(Q_3,Q_2,Q_1,Q_0)$

Slika 6.21: K-tablice za ulaze JK bistabila B_0 binarnog sinkronog brojila



(a) K-tablica za $J_1(Q_3,Q_2,Q_1,Q_0)$ (b) K-tablica za $K_1(Q_3,Q_2,Q_1,Q_0)$

Slika 6.22: K-tablice za ulaze JK bistabila B_1 binarnog sinkronog brojila



(a) K-tablica za $J_2(Q_3,Q_2,Q_1,Q_0)$ (b) K-tablica za $K_2(Q_3,Q_2,Q_1,Q_0)$

Slika 6.23: K-tablice za ulaze JK bistabila B_2 binarnog sinkronog brojila

Q_1Q_0 \ Q_3Q_2	00	01	11	10
00			X	X
01			X	X
11			X	X
10		1	X	X

(a) K-tablica za $J_3(Q_3,Q_2,Q_1,Q_0)$

Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	X	X		
01	X	X		
11	X	X		
10	X	X	1	

(b) K-tablica za $K_3(Q_3,Q_2,Q_1,Q_0)$

Slika 6.24: K-tablice za ulaze JK bistabila B_3 binarnog sinkronog brojila

Minimalni oblik logičke funkcije $J_2(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.23a:

$$J_2 = Q_1Q_0.$$

Minimalni oblik logičke funkcije $K_2(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.23b:

$$K_2 = Q_1Q_0.$$

Minimalni oblik logičke funkcije $J_3(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.24a:

$$J_3 = Q_2Q_1Q_0.$$

Minimalni oblik logičke funkcije $K_3(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.24b:

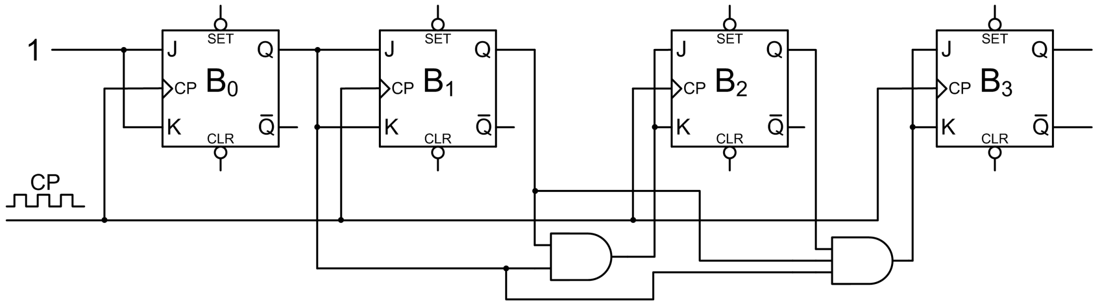
$$K_3 = Q_2Q_1Q_0.$$

Prema izvedenim minimalnim oblicima za prethodne logičke funkcije za n-bitno binarno sinkrono brojilo stanja ulaza i - tog bistabila bila bi uz $J_0 = 1$ i $K_0 = 1$:

$$J_i = \prod_{k=0}^{i-1} Q_k, (i = 1,2,...,n-1),$$

$$K_i = \prod_{k=0}^{i-1} Q_k, (i = 1,2,...,n-1)$$

Na temelju izvedenih logičkih funkcija za J i K ulaze bistabila dobivena je shema 4-bitnog binarnog sinkronog brojila koja je prikazana na slici 6.25.

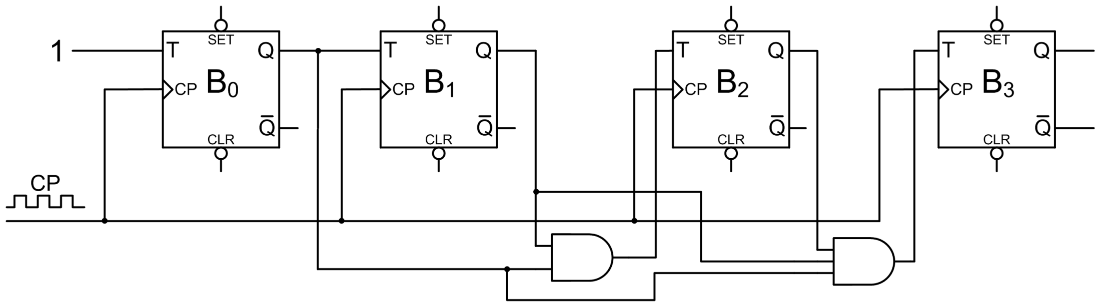


Slika 6.25: Shema 4-bitnog binarnog sinkronog brojila realiziranog JK bistabilima

Primijetimo da su u svim slučajevima J i K ulazi jednaki pa se ovo brojilo može izvesti pomoću T bistabila za koji smo pokazali da se može izvesti iz JK bistabila ako mu se kratko spoje ulazi. Za logičke funkcije ulaza T bistabila vrijedi:

$$
\begin{aligned}
T_0 &= 1 \\
T_1 &= Q_0 \\
T_2 &= Q_1 Q_0 \\
T_3 &= Q_2 Q_1 Q_0 .
\end{aligned}
$$

Shema 4-bitnog binarnog sinkronog brojila realiziranog T bistabilima prikazana je na slici 6.26. Realizacija binarnog sinkronog brojila pomoću T bistabila najpogodnija je.



Slika 6.26: Shema 4-bitnog binarnog sinkronog brojila realiziranog T bistabilima

☞ Primjer 6.3.2

Projektirajte 4-bitno binarno sinkrono brojilo koje broji unazad pomoću T bistabila koji mijenjaju stanje na rastući brid takt impulsa.

✍ Rješenje:

Tablica prijelaza 4-bitnog binarnog sinkronog brojila unazad i uzbude T bistabila prikazana je tablicom 6.8. U tablici 6.8 prikazana su sadašnja stanja bistabila u koraku n i sljedeća stanja bistabila u koraku $n+1$. U stupcu za korak n nalaze se 4-bitne kombinacije kroz koje prolazi binarno sinkrono brojilo unazad. Stupac za korak $n+1$ prikazuje sljedeću kombinaciju u koju se bistabili moraju postaviti pri rastućem bridu takt impulsa CP. Na primjer, ako je u koraku n binarno sinkrono brojilo u stanju 1111, tada će u koraku $n+1$ biti u stanju 1110. Ako je u koraku n binarno sinkrono brojilo u stanju 1110, tada će u koraku $n+1$ biti u stanju 1101 itd. Konačno, kada se binarno sinkrono brojilo u koraku n nalazi u stanju 0000, tada će u koraku $n+1$ ponovno biti u stanju 1111. Ovo brojilo broji do 16. Da bi T bistabil prešao iz stanja u koraku n u novo stanje u koraku $n+1$, na ulaz T potrebno je dovesti pobudu koja će na rastući brid takt impulsa CP izazvati traženi prijelaz. Pomoću tablice uzbude T bistabila (tablica 5.6) popunjeni su stupci uzbude za bistabile B_0, B_1, B_2 i B_3 u tablici 6.8. Na primjer, pri prijelazu iz stanja 1111 ($Q_3 = 1$, $Q_2 = 1$, $Q_1 = 1$, $Q_0 = 1$) u stanje 1110 ($Q_3 = 1$, $Q_2 = 1$, $Q_1 = 1$, $Q_0 = 0$) bistabil B_0 mora promijeniti stanje iz 1 u 0, a bistabili B_1, B_2 i B_3 ostaju u stanju 1. Da bi bistabil B_0 prešao iz stanja 1 u stanje 0, na ulaz T mu je prema tablici uzbude 5.6 potrebno dovesti vrijednost 1. Bistabili B_1, B_2 i B_3 ostaju u stanju 1 pa je na njihov ulaz T prema tablici uzbude 5.6 potrebno dovesti vrijednost 0.

Tablica 6.8: Tablica prijelaza 4-bitnog binarnog sinkronog brojila unazad i uzbude T bistabila

	Sadašnje stanje bistabila, n				Sljedeće stanje bistabila, $n+1$				Uzbuda za bistabile			
i	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	T_3	T_2	T_1	T_0
15	1	1	1	1	1	1	1	0	0	0	0	1
14	1	1	1	0	1	1	0	1	0	0	1	1
13	1	1	0	1	1	1	0	0	0	0	0	1
12	1	1	0	0	1	0	1	1	0	1	1	1
11	1	0	1	1	1	0	1	0	0	0	0	1
10	1	0	1	0	1	0	0	1	0	0	1	1
9	1	0	0	1	1	0	0	0	0	0	0	1
8	1	0	0	0	0	1	1	1	1	1	1	1
7	0	1	1	1	0	1	1	0	0	0	0	1
6	0	1	1	0	0	1	0	1	0	0	1	1
5	0	1	0	1	0	1	0	0	0	0	0	1
4	0	1	0	0	0	0	1	1	0	1	1	1
3	0	0	1	1	0	0	1	0	0	0	0	1
2	0	0	1	0	0	0	0	1	0	0	1	1
1	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	1	1	1	1	1	1	1	1

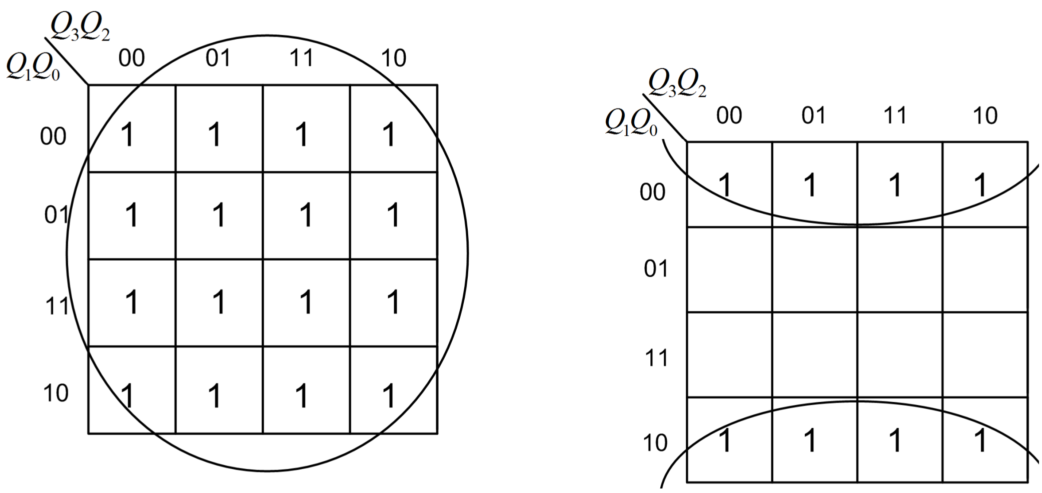
Stupci uzbude bistabila u tablici 6.8 logičke su funkcije varijabli Q_3, Q_2,Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaz T bistabila. Minimizacija je prikazana na slikama 6.27 i 6.28. Radi lakšeg popunjavanja K-tablice u tablici 6.8 prvi stupac je stupac indeksa minterma i.

Minimalni oblik logičke funkcije $T_0(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.27a:

$$T_0 = 1.$$

Minimalni oblik logičke funkcije $T_1(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.27b:

$$T_1 = \overline{Q}_0.$$



(a) K-tablica za $T_0(Q_3,Q_2,Q_1,Q_0)$ (b) K-tablica za $T_1(Q_3,Q_2,Q_1,Q_0)$

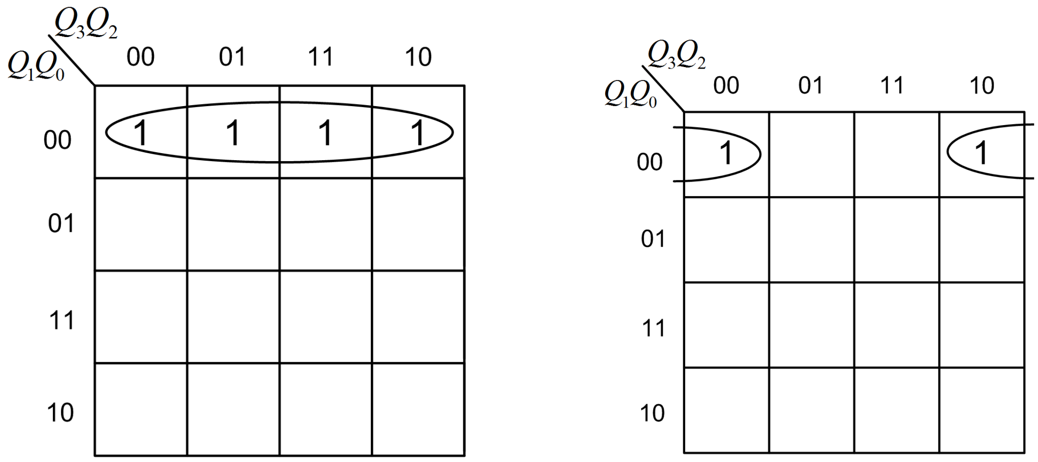
Slika 6.27: K-tablice za ulaz T bistabila B_0 i B_1 binarnog sinkronog brojila unazad

Minimalni oblik logičke funkcije $T_2(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.28a:

$$T_2 = \overline{Q}_1\overline{Q}_0.$$

Minimalni oblik logičke funkcije $T_3(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.28b:

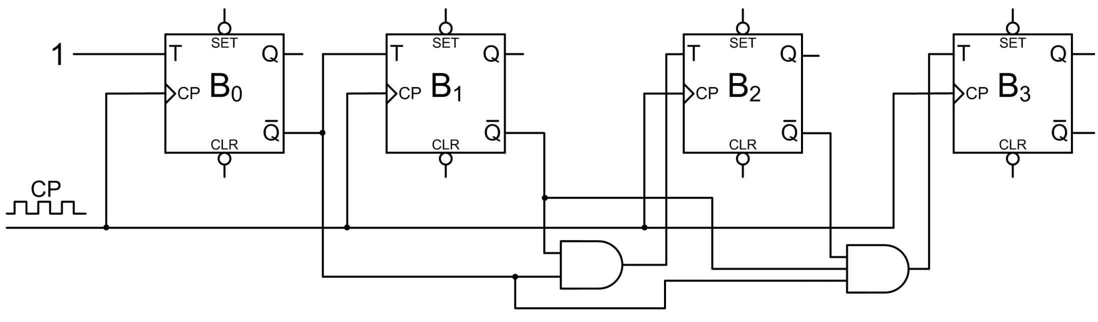
$$T_3 = \overline{Q}_2\overline{Q}_1\overline{Q}_0.$$



(a) K-tablica za $T_2(Q_3,Q_2,Q_1,Q_0)$ (b) K-tablica za $T_3(Q_3,Q_2,Q_1,Q_0)$

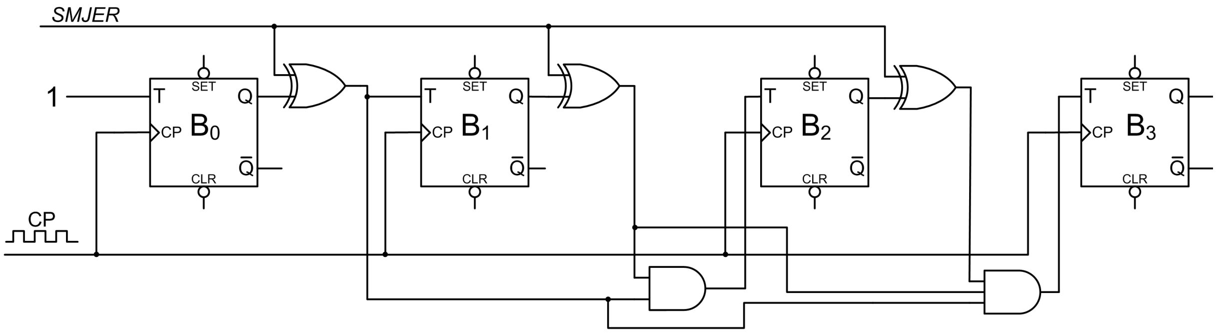
Slika 6.28: K-tablice za ulaz T bistabila B_2 i B_3 binarnog sinkronog brojila unazad

Na temelju izvedenih logičkih funkcija za T ulaze bistabila dobivena je shema 4-bitnog binarnog sinkronog brojila unazad koja je prikazana na slici 6.29.



Slika 6.29: Shema 4-bitnog binarnog sinkronog brojila unazad realiziranog T bistabilima

Ako se ovaj primjer usporedi s prethodnim, primijetit ćemo da je jedina razlika u smjeru brojanja (naprijed i nazad) korištenje izlaza bistabila Q ili $\overline{Q}$. Kada se broji unaprijed, I funkcije u minimiziranim oblicima koriste izlaz Q, a kada se broji unazad, I funkcije u minimiziranim oblicima koriste izlaz $\overline{Q}$. Moguće je u tom slučaju napraviti sklop koji će moći brojati i naprijed i nazad tako da uvedemo novu varijablu koja će se zvati $SMJER$. Shema 4-bitnog binarnog sinkronog brojila naprijed/nazad realiziranog T bistabilima prikazana je na slici 6.30. Ako varijabla $SMJER$ ima vrijednost 0, EX-ILI sklop proslijedit će na I funkcije izlaz Q. Ako varijabla $SMJER$ ima vrijednost 1, EX-ILI sklop proslijedit će na I funkcije izlaz $\overline{Q}$. Na taj način sklop može brojati naprijed ($SMJER = 0$) i nazad ($SMJER = 1$).



Slika 6.30: Shema 4-bitnog binarnog sinkronog brojila naprijed/nazad realiziranog T bistabilima

☞ Primjer 6.3.3

Projektirajte binarno sinkrono dekadsko brojilo koristeći T bistabile. Provjerite ima li ovo brojilo siguran start.

✍ Rješenje:

Tablica prijelaza 4-bitnog binarnog sinkronog dekadskog brojila uzbude T bistabila prikazana je tablicom 6.9. Dekadsko brojilo specifično je po tome što prolazi kroz 10 stanja (od 0 do 9). U tablici 6.9 dodan je izlaz Y koji se postavlja u vrijednost 1 kada brojilo dođe u stanje 9. Taj izlaz može poslužiti kod povezivanja brojila u kaskadu kao takt impuls za brojenje na višem brojnom mjestu. Brojilo se nakon stanja 1001 (9_{10}) mora vratiti u stanje 0000 (0_{10}). Binarna stanja od 1010 (10_{10}) do 1111 (15_{10}) neregularna su stanja i brojilo u ispravnom stanju neće ući u nijedno od njih. Zbog toga se u svrhu bolje minimizacije kod njih za uzbudu može staviti nespecificirano stanje X.

Tablica 6.9: Tablica prijelaza 4-bitnog binarnog sinkronog dekadskog brojila uzbude T bistabila

	Sadašnje stanje bistabila, n				Sljedeće stanje bistabila, $n+1$				Uzbuda za bistabile				Izlaz
i	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	T_3	T_2	T_1	T_0	Y
0	0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	1	0	0	1	0	0	0	1	1	0
2	0	0	1	0	0	0	1	1	0	0	0	1	0
3	0	0	1	1	0	1	0	0	0	1	1	1	0
4	0	1	0	0	0	1	0	1	0	0	0	1	0
5	0	1	0	1	0	1	1	0	0	0	1	1	0
6	0	1	1	0	0	1	1	1	0	0	0	1	0
7	0	1	1	1	1	0	0	0	1	1	1	1	0
8	1	0	0	0	1	0	0	1	0	0	0	1	0
9	1	0	0	1	0	0	0	0	1	0	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X	X	0
11	1	0	1	1	X	X	X	X	X	X	X	X	0
12	1	1	0	0	X	X	X	X	X	X	X	X	0
13	1	1	0	1	X	X	X	X	X	X	X	X	0
14	1	1	1	0	X	X	X	X	X	X	X	X	0
15	1	1	1	1	X	X	X	X	X	X	X	X	0

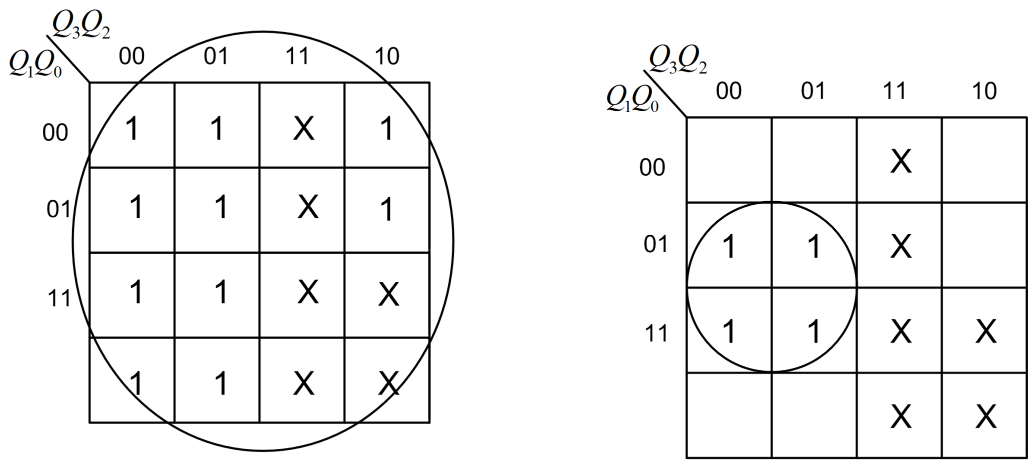
Stupci uzbude bistabila u tablici 6.9 logičke su funkcije varijabli Q_3, Q_2,Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaz T bistabila. Minimizacija je prikazana na slikama 6.31 i 6.32. Radi lakšeg popunjavanja K-tablice u tablici 6.9 prvi stupac je stupac indeksa minterma i (ujedno je indeks i dekadski broj kojeg brojilo prikazuje).

Minimalni oblik logičke funkcije $T_0(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.31a:

$$T_0 = 1.$$

Minimalni oblik logičke funkcije $T_1(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.31b:

$$T_1 = \overline{Q}_3 Q_0.$$



(a) K-tablica za $T_0(Q_3,Q_2,Q_1,Q_0)$ (b) K-tablica za $T_1(Q_3,Q_2,Q_1,Q_0)$

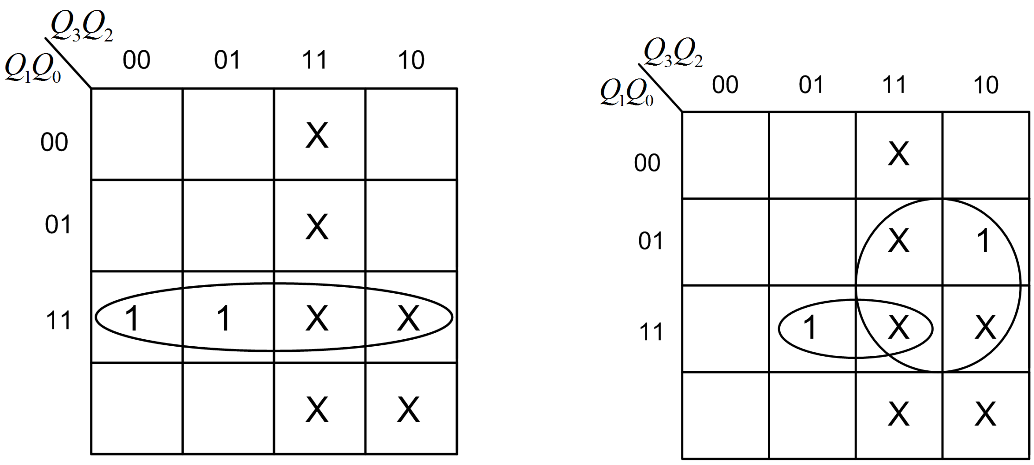
Slika 6.31: K-tablice za ulaz T bistabila B_0 i B_1 binarnog sinkronog dekadskog brojila

Minimalni oblik logičke funkcije $T_2(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.32a:

$$T_2 = Q_1 Q_0.$$

Minimalni oblik logičke funkcije $T_3(Q_3,Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.32b:

$$T_3 = Q_2 Q_1 Q_0 + Q_3 Q_0.$$



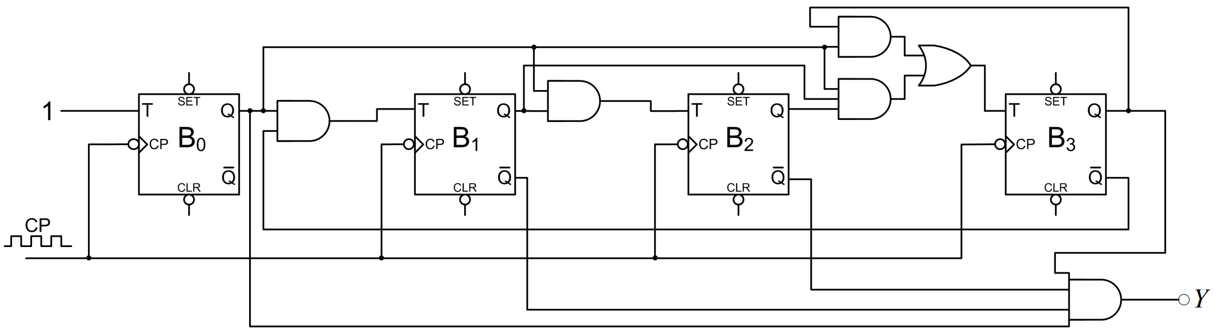
(a) K-tablica za $T_2(Q_3,Q_2,Q_1,Q_0)$ (b) K-tablica za $T_3(Q_3,Q_2,Q_1,Q_0)$

Slika 6.32: K-tablice za ulaz T bistabila B_2 i B_3 binarnog sinkronog dekadskog brojila

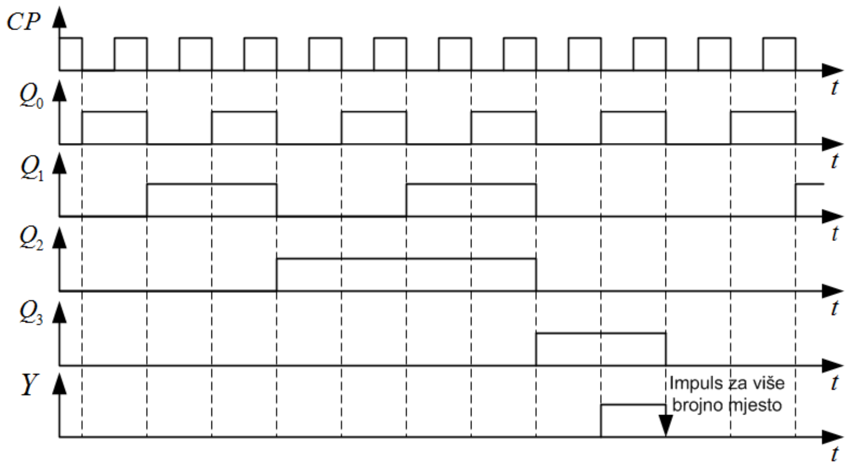
Izlaz $Y(Q_3,Q_2,Q_1,Q_0)$ opisan je samo jednim mintermom:

$$Y = Q_3\overline{Q}_2\overline{Q}_1Q_0.$$

Na temelju izvedenih logičkih funkcija za T ulaze bistabila dobivena je shema 4-bitnog binarnog sinkronog dekadskog brojila koja je prikazana na slici 6.33. Za realizaciju ovog brojila korišten je T bistabil koji stanje mijenja na padajući brid. Razlog tome je mogućnost povezivanja ovog brojila u kaskadu kako bi brojilo moglo brojiti višeznamenkaste dekadske brojeve. Na slici 6.34 prikazan je vremenski dijagram 4-bitnog binarnog sinkronog dekadskog brojila realiziranog T bistabilima. Izlaz Y koristi se kao takt impuls za brojilo na višem brojnom mjestu. Kada bi se koristili T bistabili koji stanje mijenjaju na rastući brid, tada bi se više brojno mjesto uvećalo za 1 kada niže brojno mjesto dođe u stanje 9. Više brojno mjesto mora se uvećati kada se izlazi iz stanja 9, a to je moguće T bistabilima koji stanje mijenjaju na padajući brid.



Slika 6.33: Shema 4-bitnog binarnog sinkronog dekadskog brojila realiziranog T bistabilima



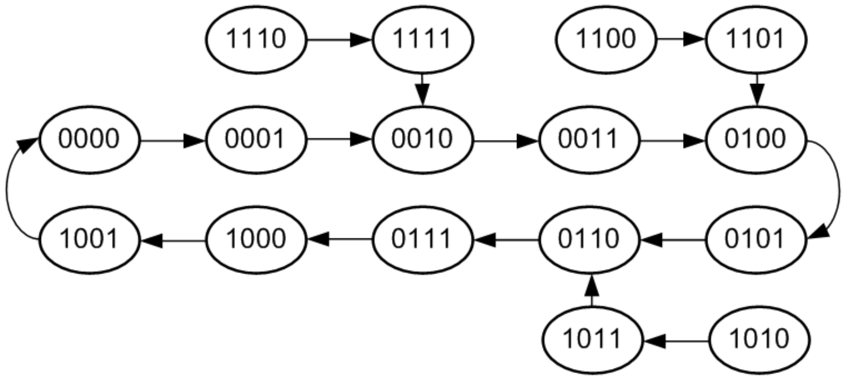
Slika 6.34: Vremenski dijagram 4-bitnog binarnog sinkronog dekadskog brojila realiziranog T bistabilima

Ako se iz nekog razloga u bistabilima pojavi neregularna kombinacija, na temelju ulaznih funkcija bistabila dobivenih minimizacijom može se popuniti tablica prijelaza za ta stanja. Na primjer, ako je brojilo u neregularnom stanju 1011:

- Bistabil B_0 promijenit će svoje stanje jer je na njegovom ulazu $T_0 = 1$, $Q_0 = 0$,
- Bistabil B_1 promijenit će ostaje u stanju 1 jer je $T_1 = \overline{Q}_3Q_0 = \overline{1} \cdot 1 = 0$, $Q_1 = 1$,
- Bistabil B_2 promijenit će ostaje u stanju 1 jer je $T_2 = Q_1Q_0 = 1 \cdot 1 = 1$, $Q_2 = 1$,

- Bistabil B_3 promijenit će svoje stanje u 0 jer je na njegovom ulazu $T_3 = Q_2Q_1Q_0 + Q_3Q_0 = 0 \cdot 1 \cdot 1 + 1 \cdot 1 = 0 + 1 = 1$, $Q_3 = 0$.

Nakon neregularnog stanja 1011, sljedeće je stanje brojila 0110 i ono je regularno. Provjerite za ostala neregularna stanja. Dijagram prijelaza binarnog sinkronog dekadskog brojila sa sigurnim startom prikazan je na slici 6.35 na kojoj se vidi da će brojilo uvijek ući u ciklus brojanja. Ovo brojilo ima siguran start.



Slika 6.35: Dijagram prijelaza binarnog sinkronog dekadskog brojila sa sigurnim startom

6.3.1 Zadaci za vježbu

✎ Zadatak 6.3.1

Projektirajte 5-bitno binarno sinkrono brojilo pomoću T bistabila.

✎ Zadatak 6.3.2

Projektirajte 4-bitno binarno sinkrono brojilo koje broji unazad pomoću JK bistabila.

✎ Zadatak 6.3.3

Projektirajte binarno sinkrono dekadsko brojilo unazad koristeći T bistabile. Provjerite ima li ovo brojilo siguran start.

6.4 Modulo m sinkrono brojilo

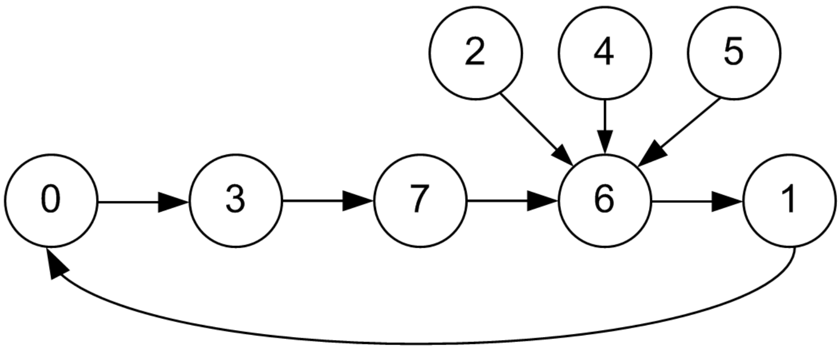
Ako postoji potreba za brojanjem kroz m stanja (npr. $m = 5$), potrebno je koristiti n bistabila tako da vrijedi $m < 2^n$. Prema tome, $2^n - m$ stanja brojila su neiskorištena i nazivamo ih nespecificirana stanja. Za sinkrono brojilo koje prolazi kroz m stanja potrebno je napraviti tablicu kombinacija s uzbudama bistabila koje će osigurati prijelaze kroz željena stanja.

☞ Primjer 6.4.1

Projektirajte sinkrono brojilo koje prolazi kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$. Osigurajte siguran start brojila tako da se sva nespecificirana stanja prebace u stanje 6. Nacrtajte dijagram stanja ovog brojila, a zatim:
a) projektirajte sklop s minimalnim brojem JK bistabila i osnovnih logičkih sklopova.
b) projektirajte sklop s minimalnim brojem T bistabila i osnovnih logičkih sklopova.
c) projektirajte sklop s minimalnim brojem D bistabila i osnovnih logičkih sklopova.

✍ Rješenje:

Dijagram prijelaza sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 prikazan je na slici 6.36. Nespecificirana stanja 2, 4 i 5 prelaze u stanje 6 čime je ostvaren siguran start.



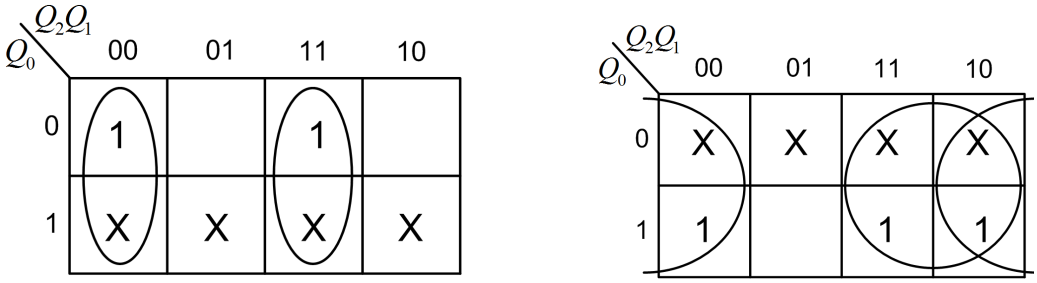
Slika 6.36: Dijagram prijelaza sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6

a) Tablica prijelaza sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 i uzbude JK bistabila prikazana je tablicom 6.10. U kreiranju stupaca uzbude bistabila B_0, B_1 i B_2 korištena je tablica uzbude JK bistabila (tablica 5.4).

Tablica 6.10: Tablica prijelaza sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 i uzbude JK bistabila

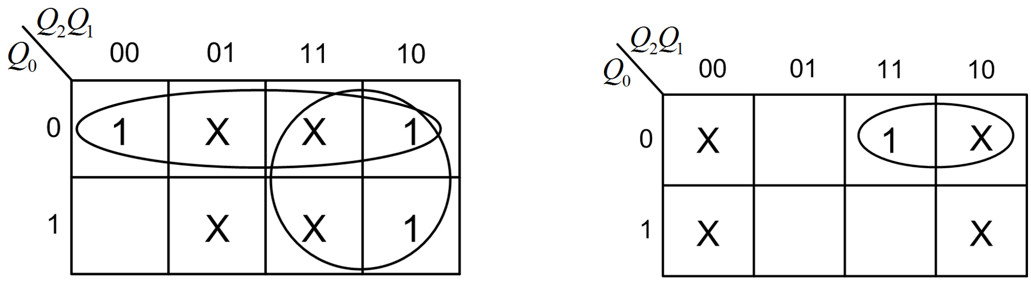
	Sadašnje stanje bistabila, n			Sljedeće stanje bistabila, $n+1$			Uzbuda za bistabile					
							B_2		B_1		B_0	
i	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	1	0	X	1	X	1	X
1	0	0	1	0	0	0	0	X	0	X	X	1
2	0	1	0	1	1	0	1	X	X	0	0	X
3	0	1	1	1	1	1	1	X	X	0	X	0
4	1	0	0	1	1	0	X	0	1	X	0	X
5	1	0	1	1	1	0	X	0	1	X	X	1
6	1	1	0	0	0	1	X	1	X	1	1	X
7	1	1	1	1	1	0	X	0	X	0	X	1

Stupci uzbude bistabila u tablici 6.10 logičke su funkcije varijabli Q_2, Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaze JK bistabila. Radi lakšeg popunjavanja K-tablice, u tablici 6.10 prvi stupac je stupac indeksa minterma i. K-tablice za ulaze JK bistabila B_0, B_1 i B_2 sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ s prijelazom nespecificiranih stanja u stanje 6 prikazane su na slikama 6.37, 6.38 i 6.39.



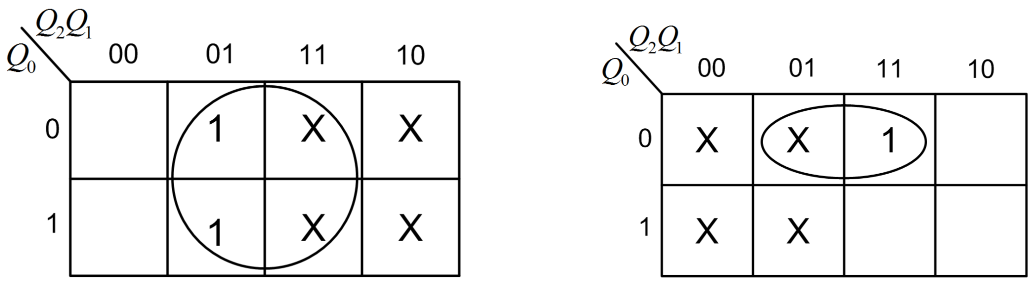
(a) K-tablica za $J_0(Q_2,Q_1,Q_0)$ (b) K-tablica za $K_0(Q_2,Q_1,Q_0)$

Slika 6.37: K-tablice za ulaze JK bistabila B_0 sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ s prijelazom nespecificiranih stanja u stanje 6



(a) K-tablica za $J_1(Q_2,Q_1,Q_0)$ (b) K-tablica za $K_1(Q_2,Q_1,Q_0)$

Slika 6.38: K-tablice za ulaze JK bistabila B_1 sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ s prijelazom nespecificiranih stanja u stanje 6



(a) K-tablica za $J_2(Q_2,Q_1,Q_0)$ (b) K-tablica za $K_2(Q_2,Q_1,Q_0)$

Slika 6.39: K-tablice za ulaze JK bistabila B_2 sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ s prijelazom nespecificiranih stanja u stanje 6

Minimalni oblik logičke funkcije $J_0(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.37a:

$$J_0 = \overline{Q}_2\overline{Q}_1 + Q_2Q_1.$$

Minimalni oblik logičke funkcije $K_0(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.37b:

$$K_0 = \overline{Q}_1 + Q_2.$$

Minimalni oblik logičke funkcije $J_1(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.38a:

$$J_1 = \overline{Q}_0 + Q_2.$$

Minimalni oblik logičke funkcije $K_1(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.38b:

$$K_1 = Q_2\overline{Q}_0.$$

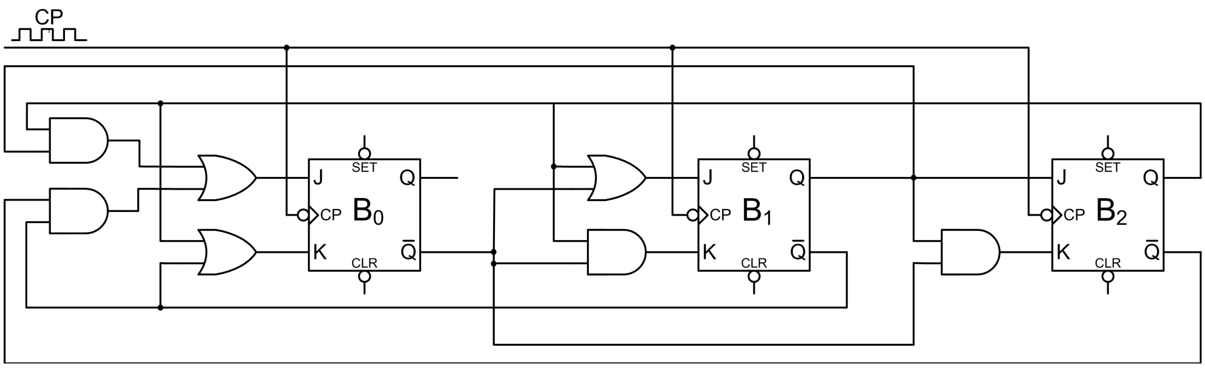
Minimalni oblik logičke funkcije $J_2(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.39a:

$$J_2 = Q_1.$$

Minimalni oblik logičke funkcije $K_2(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.39b:

$$K_2 = Q_1\overline{Q}_0.$$

Na temelju izvedenih logičkih funkcija za J i K ulaze bistabila dobivena je shema sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ s prijelazom nespecificiranih stanja u stanje 6 realiziranog pomoću JK bistabila. Shema je prikazana na slici 6.40.



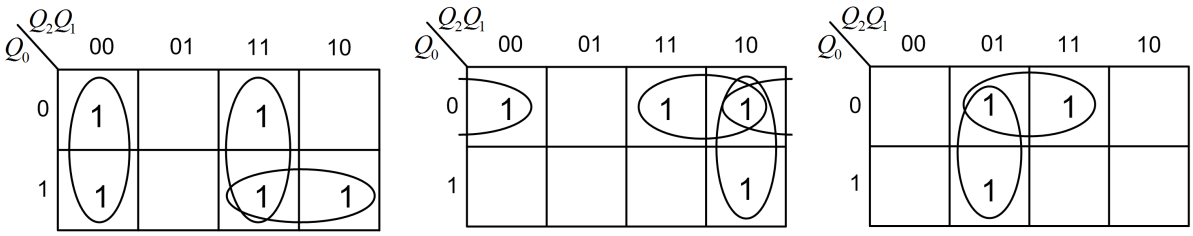
Slika 6.40: Shema sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ s prijelazom nespecificiranih stanja u stanje 6 realiziranog pomoću JK bistabila

b) Tablica prijelaza sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ i uzbude T bistabila prikazana je tablicom 6.11. U kreiranju stupaca uzbude bistabila B_0, B_1 i B_2 korištena je tablica uzbude T bistabila (tablica 5.6).

Tablica 6.11: Tablica prijelaza sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ i uzbude T bistabila

	Sadašnje stanje bistabila, n			Sljedeće stanje bistabila, $n+1$			Uzbuda za bistabile		
i	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0
0	0	0	0	0	1	1	0	1	1
1	0	0	1	0	0	0	0	0	1
2	0	1	0	1	1	0	1	0	0
3	0	1	1	1	1	1	1	0	0
4	1	0	0	1	1	0	0	1	0
5	1	0	1	1	1	0	0	1	1
6	1	1	0	0	0	1	1	1	1
7	1	1	1	1	1	0	0	0	1

Stupci uzbude bistabila u tablici 6.11 logičke su funkcije varijabli Q_2, Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaze T bistabila. Radi lakšeg popunjavanja K-tablice u tablici 6.11 prvi stupac je stupac indeksa minterma i. K-tablice za ulaze T bistabila B_0, B_1 i B_2 sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 prikazane su na slici 6.41.



(a) K-tablica za $T_0(Q_2,Q_1,Q_0)$ (b) K-tablica za $T_1(Q_2,Q_1,Q_0)$ (c) K-tablica za $T_2(Q_2,Q_1,Q_0)$

Slika 6.41: K-tablice za ulaze T bistabila B_0, B_1 i B_2 sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6

Minimalni oblik logičke funkcije $T_0(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.41a:

$$T_0 = \overline{Q}_2\overline{Q}_1 + Q_2Q_1 + Q_2Q_0.$$

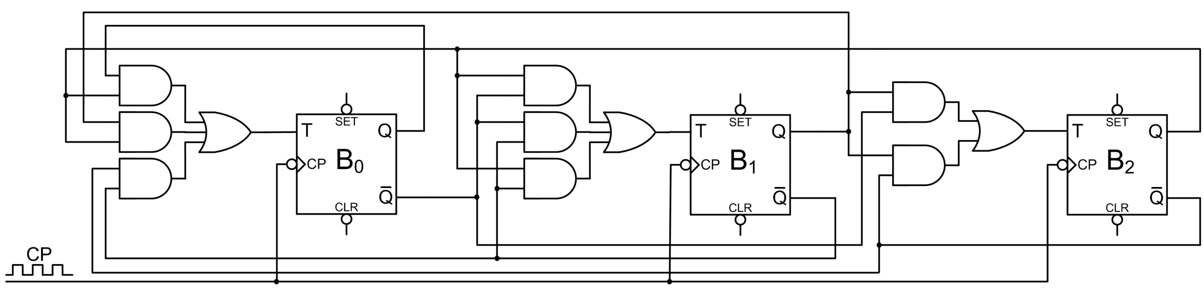
Minimalni oblik logičke funkcije $T_1(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.41b:

$$T_1 = \overline{Q}_1\overline{Q}_0 + Q_2\overline{Q}_0 + Q_2\overline{Q}_1.$$

Minimalni oblik logičke funkcije $T_2(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.41c:

$$T_2 = \overline{Q}_2Q_1 + Q_1\overline{Q}_0.$$

Na temelju izvedenih logičkih funkcija za T ulaze bistabila dobivena je shema sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 realiziranog pomoću T bistabila. Shema je prikazana na slici 6.42.



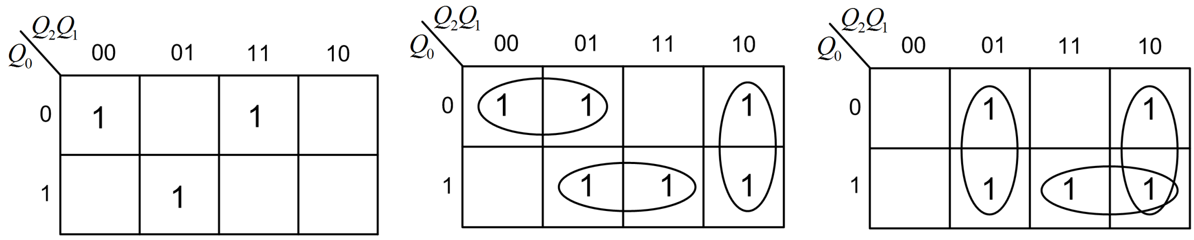
Slika 6.42: Shema sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 realiziranog pomoću T bistabila

c) Tablica prijelaza sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ i uzbude D bistabila prikazana je tablicom 6.12. U kreiranju stupaca uzbude bistabila B_0, B_1 i B_2 korištena je tablica uzbude D bistabila (tablica 5.8).

Tablica 6.12: Tablica prijelaza sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ i uzbude D bistabila

	Sadašnje stanje bistabila, n			Sljedeće stanje bistabila, $n+1$			Uzbuda za bistabile		
i	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0	0	0	1	1	0	1	1
1	0	0	1	0	0	0	0	0	0
2	0	1	0	1	1	0	1	1	0
3	0	1	1	1	1	1	1	1	1
4	1	0	0	1	1	0	1	1	0
5	1	0	1	1	1	0	1	1	0
6	1	1	0	0	0	1	0	0	1
7	1	1	1	1	1	0	1	1	0

Stupci uzbude bistabila u tablici 6.12 logičke su funkcije varijabli Q_2, Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaze D bistabila. Radi lakšeg popunjavanja K-tablice u tablici 6.12 prvi stupac je stupac indeksa minterma i. K-tablice za ulaze D bistabila B_0, B_1 i B_2 sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 prikazane su na slici 6.43.



(a) K-tablica za $D_0(Q_2,Q_1,Q_0)$ (b) K-tablica za $D_1(Q_2,Q_1,Q_0)$ (c) K-tablica za $D_2(Q_2,Q_1,Q_0)$

Slika 6.43: K-tablice za ulaze D bistabila B_0, B_1 i B_2 sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6

Minimalni oblik logičke funkcije $D_0(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.43a:

$$D_0 = \overline{Q}_2\overline{Q}_1\overline{Q}_0 + \overline{Q}_2 Q_1 Q_0 + Q_2 Q_1 \overline{Q}_0.$$

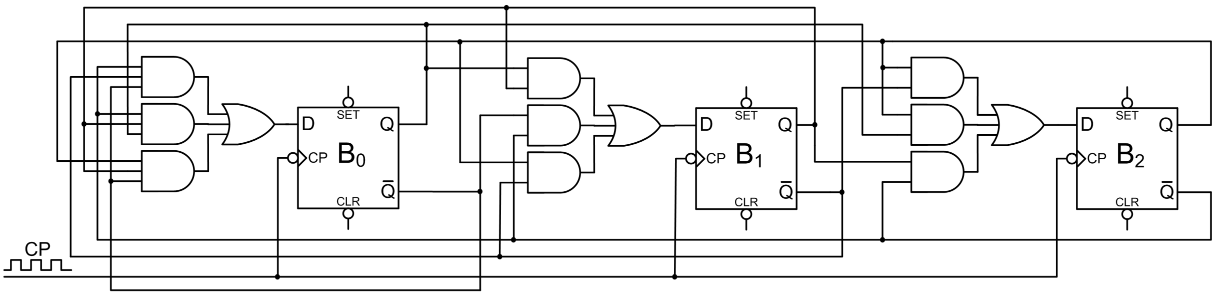
Minimalni oblik logičke funkcije $D_1(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.43b:

$$D_1 = \overline{Q}_2\overline{Q}_0 + Q_1 Q_0 + Q_2\overline{Q}_1.$$

Minimalni oblik logičke funkcije $D_2(Q_2,Q_1,Q_0)$ je prema K-tablici na slici 6.43c:

$$D_2 = \overline{Q}_2 Q_1 + Q_2 Q_0 + Q_2\overline{Q}_1.$$

Na temelju izvedenih logičkih funkcija za D ulaze bistabila dobivena je shema sinkronog brojila kroz stanja $0 \to 3 \to 7 \to 6 \to 1 \to 0$ s prijelazom nespecificiranih stanja u stanje 6 realiziranog pomoću T bistabila. Shema je prikazana na slici 6.44.



Slika 6.44: Shema sinkronog brojila kroz stanja $0 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 0$ s prijelazom nespecificiranih stanja u stanje 6 realiziranog pomoću D bistabila

6.4.1 Zadaci za vježbu

✎ Zadatak 6.4.1

Projektirajte sinkrono brojilo koje prolazi kroz stanja $0 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 7 \rightarrow 6 \rightarrow 0$. Osigurajte siguran start brojila tako da se sva nespecificirana stanja prebace u stanje 0. Nacrtajte dijagram stanja ovog brojila, a zatim:
a) projektirajte sklop s minimalnim brojem JK bistabila i osnovnih logičkih sklopova.
b) projektirajte sklop s minimalnim brojem T bistabila i osnovnih logičkih sklopova.
c) projektirajte sklop s minimalnim brojem D bistabila i osnovnih logičkih sklopova.

6.5 Binarno asinkrono brojilo

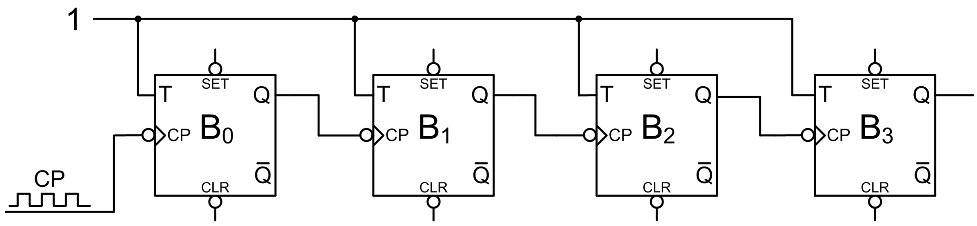
Binarno asinkrono brojilo je brojilo kod kojeg se stanja bistabila ne mijenjaju istovremeno na takt impulsa CP, već se stanja mijenjaju pod utjecajem izlaza prethodnog bistabila. Takt impuls CP dovodi se samo na prvi bistabil u nizu. Za realizaciju binarnog asinkronog brojila koriste se T bistabili (ili JK bistabil s kratko spojenim ulazima). Na sve ulaze T bistabila dovodi se logička vrijednost 1.

☞ Primjer 6.5.1

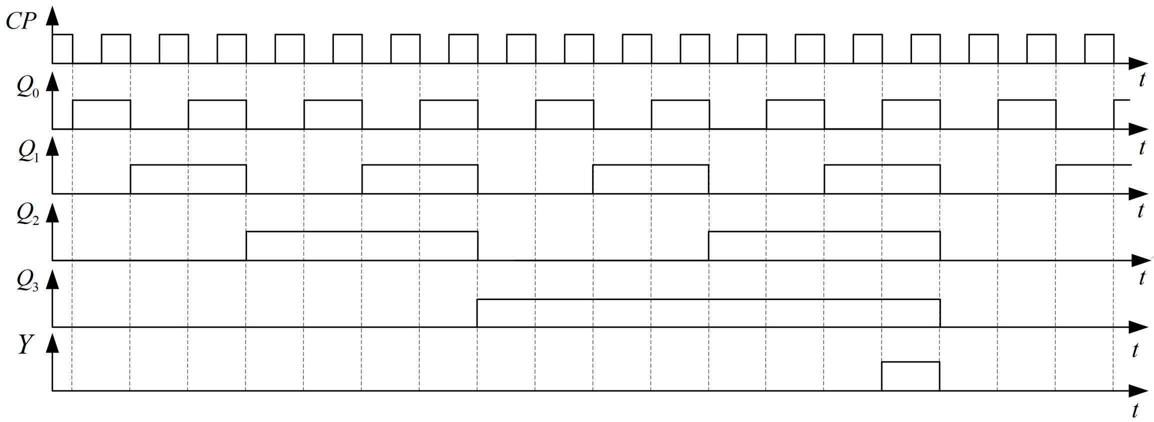
Projektirajte 4-bitno asinkrono brojilo pomoću T bistabila. Kako ovo brojilo dijeli frekvenciju takt impulsa CP? Pomoću osnovnih logičkih sklopova realizirajte sklop koji će na izlazu dati logičku vrijednost 1 svakih 16 impulsa takta CP.

✍ **Rješenje:** Shema 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila prikazana je na slici 6.45. Svi bistabili na slici 6.45 na ulazu T imaju vrijednost 1. T bistabil s tako definiranim ulazom mijenja stanje na svaki padajući brid impulsa dovedenog na ulaz CP. Na bistabil B_0 doveden je impuls takta CP. Izlaz Q_0 spojen je na ulaz CP T bistabila B_1. U takvu kaskadu moguće je spojiti n bistabila koji bi prolazili kroz 2^n stanja.

Vremenski dijagram 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila za prvih 19 impulsa takta CP prikazan je na slici 6.46. Tablica prijelaza 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila prikazana je tablicom 6.13. Tablica 6.13 izvedena je iz vremenskog dijagrama sa slike 6.46, a sadrži i logičku funkciju Y koja generira vrijednost 1 svakih 16 takt impulsa.



Slika 6.45: Shema 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila



Slika 6.46: Vremenski dijagram 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila s izlazom Y

Tablica 6.13: Tablica prijelaza 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila s izlazom Y

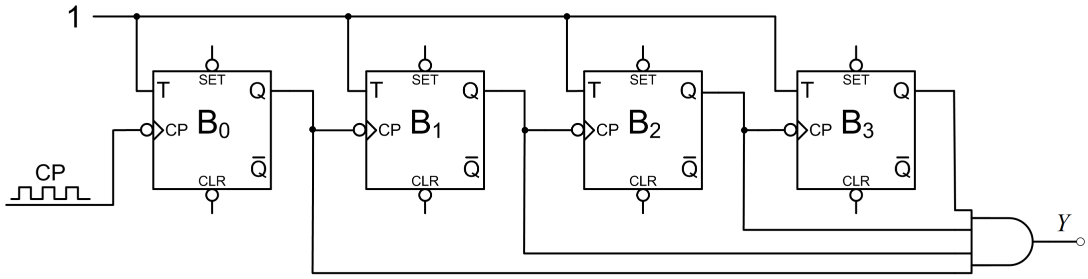
CP (broj impulsa)	Stanje brojila					Izlaz
n	i	Q_3	Q_2	Q_1	Q_0	Y
0	0	0	0	0	0	0
1	1	0	0	0	1	0
2	2	0	0	1	0	0
3	3	0	0	1	1	0
4	4	0	1	0	0	0
5	5	0	1	0	1	0
6	6	0	1	1	0	0
7	7	0	1	1	1	0
8	8	1	0	0	0	0
9	9	1	0	0	1	0
10	10	1	0	1	0	0
11	11	1	0	1	1	0
12	12	1	1	0	0	0
13	13	1	1	0	1	0
14	14	1	1	1	0	0
15	15	1	1	1	1	1
16	0	0	0	0	0	0
17	1	0	0	0	1	0
18	2	0	0	1	0	0
19	3	0	0	1	1	0

Svaki bistabil frekvenciju takt impulsa CP dijeli s dva što se može vidjeti na slici 6.46. Tako prema slici 6.46 za sklop sa slike 6.45 vrijedi:

- izlaz Q_0 ima dva puta manju frekvenciju od takt impulsa CP ($f_{Q_0} = f_{CP}/2$),
- izlaz Q_1 ima četiri puta manju frekvenciju od takt impulsa CP ($f_{Q_1} = f_{CP}/4$),
- izlaz Q_2 ima osam puta manju frekvenciju od takt impulsa CP ($f_{Q_2} = f_{CP}/8$),
- izlaz Q_3 ima 16 puta manju frekvenciju od takt impulsa CP ($f_{Q_3} = f_{CP}/16$).

Binarno asinkrono brojilo koristi se kao djelitelj frekvencije u mikrokontrolerima. Općenito, ako je binarno asinkrono brojilo sačinjeno od n bistabila, tada će izlaz zadnjeg bistabila imati 2^n puta manju frekvenciju od frekvencije takt impulsa CP ($f_{Q_{n-1}} = f_{CP}/2^n$).

Shema 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila s izlazom Y koji generira vrijednost 1 svakih 16 takt impulsa prikazana je na slici 6.47. Izlaz Y generiran je pomoću I logičke funkcije svih izlaza bistabila, odnosno mintermom: $Y = Q_3Q_2Q_1Q_0$. Moguće je koristiti bilo koji minterm (m_0 do m_{15}) za realizaciju logičke funkcije Y.



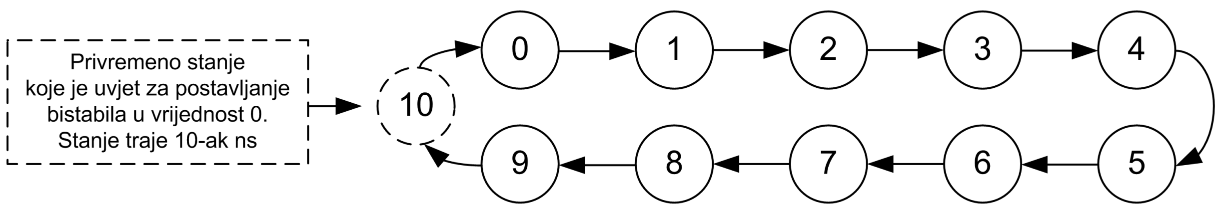
Slika 6.47: Shema 4-bitnog binarnog asinkronog brojila realiziranog pomoću T bistabila s izlazom Y koji generira vrijednost 1 svakih 16 takt impulsa

☞ Primjer 6.5.2

Projektirajte asinkrono brojilo pomoću JK bistabila i potrebnog broja NI sklopova koje će prolaziti kroz 10 stanja tako da:
a) koristite asinkrone ulaze za postavljanje u vrijednost 0 (CLR),
b) koristite asinkrone ulaze za postavljanje u vrijednost 1 (SET).

✍ Rješenje:

a) Dijagram prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0 prikazan je na slici 6.48.



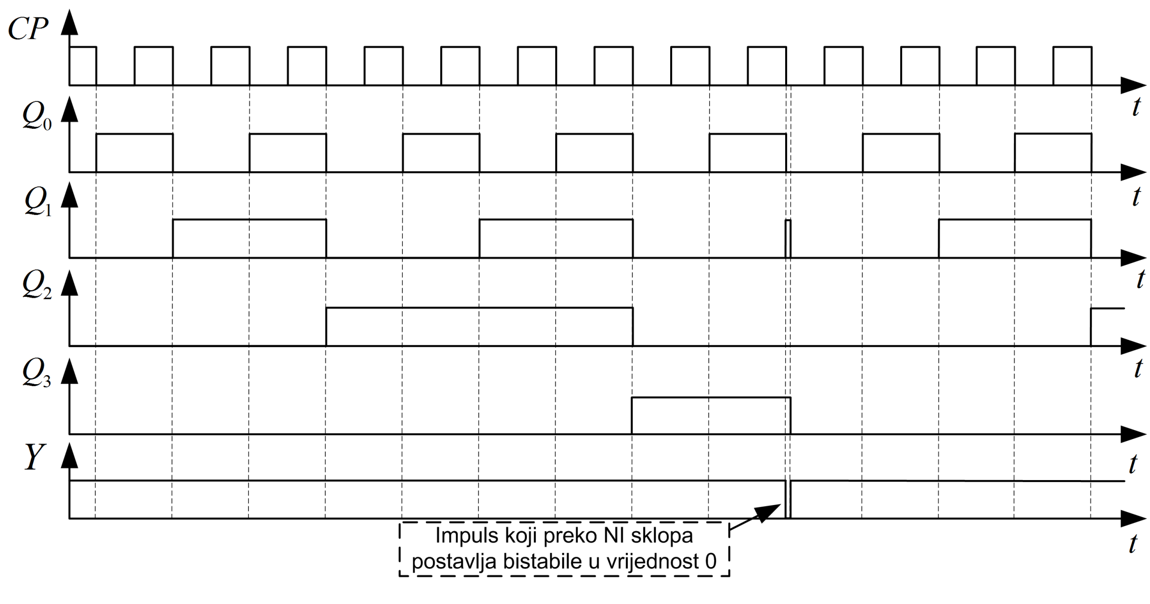
Slika 6.48: Dijagram prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0

Ako koristimo asinkroni ulaz bistabila za postavljanje u vrijednost 0, tada je stanje 0000 (0_{10}) obavezno stanje kroz koje ovo brojilo mora proći. Kako bi asinkrono brojilo moglo prolaziti kroz 10 stanja, potrebno je zadovoljiti uvjet: $10 < 2^n$ gdje je n broj bistabila. Najmanji n za koji vrijedi da je $10 < 2^n$ je $n = 4$. U dijagramu prijelaza sa slike 6.48 prikazana su stanja kroz koje asinkrono brojilo prolazi. Kada brojilo dođe u stanje 10, sve je bistabile potrebno postaviti u vrijednost 0. Stanje 10 zapravo je jedanaesto stanje u koje ovo brojilo uđe, ali u ovom stanju brojilo ostaje jako kratko vrijeme (10-ak ns). Kada brojilo uđe u stanje 10, NI sklopom potrebno je asinkrono postaviti bistabile u vrijednost 0. Tablica prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0 prikazana je tablicom 6.14. U tablici 6.14 prikazano je privremeno stanje 10 u kojem se brojilo nalazi neposredno prije postavljanja bistabila u vrijednost 0.

Tablica 6.14: Tablica prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0

	Sadašnje stanje bistabila, n				Sljedeće stanje bistabila, $n+1$				
i	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	0	0	0	1	
1	0	0	0	1	0	0	1	0	
2	0	0	1	0	0	0	1	1	
3	0	0	1	1	0	1	0	0	
4	0	1	0	0	0	1	0	1	
5	0	1	0	1	0	1	1	0	
6	0	1	1	0	0	1	1	1	
7	0	1	1	1	1	0	0	0	
8	1	0	0	0	1	0	0	1	
9	1	0	0	1	1	0	1	0	← Privremeno stanje 1010 koje traje 10-ak ns
					0	0	0	0	← Asinkrono postavljanje bistabila u vrijednost 0

Vremenski dijagram asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0 prikazan je na slici 6.49.



Slika 6.49: Vremenski dijagram asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0

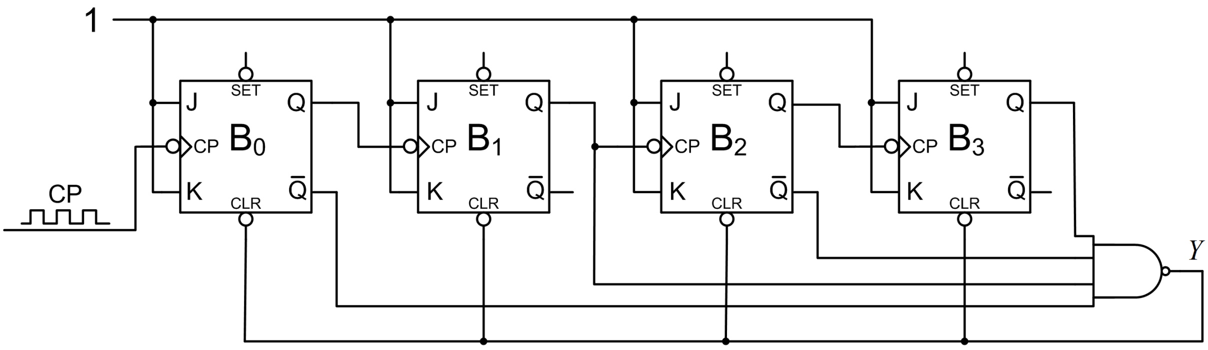
Na vremenskom dijagramu sa slike 6.49 prikazan je trenutak u kojem se bistabili pomoću asinkronih ulaza CLR postavljaju u vrijednost 0. Prema dijagramu prijelaza sa slike 6.48 i tablice 6.14, minterm koji moramo detektirati je:

$$m_{10} = Q_3\overline{Q}_2Q_1\overline{Q}_0.$$

Budući da je vrijednost 0 aktivna na asinkronim ulazima CLR, prethodni minterm moramo komplementirati:

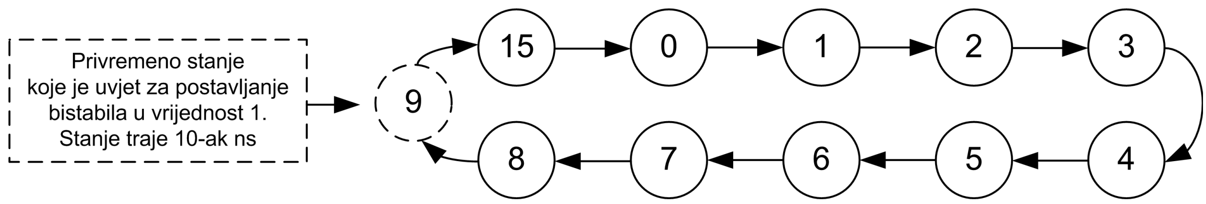
$$Y = \overline{m_{10}} = \overline{Q_3\overline{Q}_2Q_1\overline{Q}_0}.$$

Izvedena logička funkcija Y pogodna je za realizaciju pomoću jednog NI sklopa sa četiri ulaza. Shema asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0 prikazana je na slici 6.50.



Slika 6.50: Shema asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 0

Dijagram prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1 prikazana je na slici 6.51.



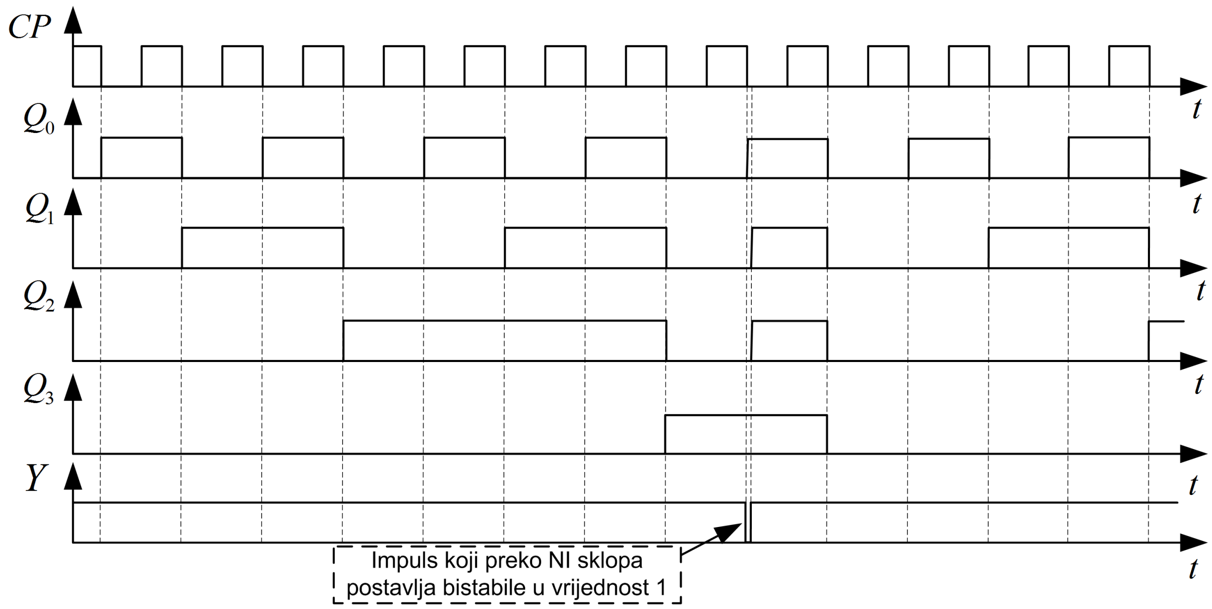
Slika 6.51: Dijagram prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1

Ako koristimo asinkroni ulaz bistabila za postavljanje u vrijednost 1, tada je stanje 1111 (15_{10}) obavezno stanje kroz koje ovo brojilo mora proći. U dijagramu prijelaza sa slike 6.51 prikazana su stanja kroz koje asinkrono brojilo prolazi. Kada brojilo dođe u stanje 9, sve bistabile je potrebno postaviti u vrijednost 1. Stanje 9 zapravo je jedanaesto stanje u koje ovo brojilo uđe, ali u ovom stanju brojilo ostaje jako kratko vrijeme (10-ak ns). Kada brojilo uđe u stanje 9, NI sklopom potrebno je asinkrono postaviti bistabile u vrijednost 1. Tablica prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1 prikazana je tablicom 6.15. U tablici 6.15 prikazano je privremeno stanje 9 u kojem se brojilo nalazi neposredno prije postavljanja bistabila u vrijednost 1.

Tablica 6.15: Tablica prijelaza asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1

<table>
<tr><th></th><th colspan="4">Sadašnje stanje bistabila, n</th><th colspan="4">Sljedeće stanje bistabila, $n+1$</th><th></th></tr>
<tr><th>i</th><th>Q_3</th><th>Q_2</th><th>Q_1</th><th>Q_0</th><th>Q_3</th><th>Q_2</th><th>Q_1</th><th>Q_0</th><th></th></tr>
<tr><td>15</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td></td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td></td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td></td></tr>
<tr><td>4</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td></td></tr>
<tr><td>5</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td></td></tr>
<tr><td>6</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td></td></tr>
<tr><td>7</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td></td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>← Privremeno stanje 1001 koje traje 10-ak ns</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>← Asinkrono postavljanje bistabila u vrijednost 1</td></tr>
</table>

Vremenski dijagram asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1 prikazan je na slici 6.52.



Slika 6.52: Vremenski dijagram asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1

Na vremenskom dijagramu sa slike 6.52 prikazan je trenutak u kojem se bistabili pomoću asinkronih ulaza *SET* postavljaju u vrijednost 1. Prema dijagramu prijelaza sa slike 6.51 i tablice 6.15, minterm koji moramo detektirati je:

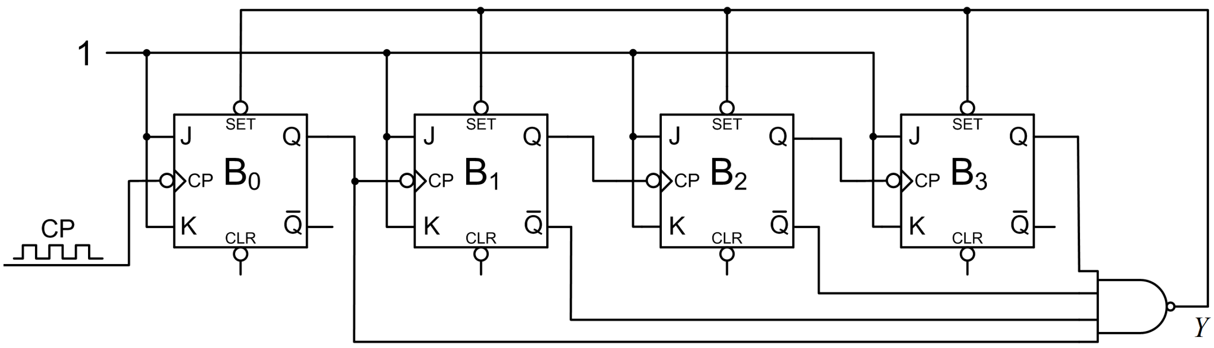
$$m_9 = Q_3\overline{Q}_2\overline{Q}_1Q_0.$$

Budući da je vrijednost 0 aktivna na asinkronim ulazima *SET*, prethodni minterm moramo komplementirati:

$$Y = \overline{m_9} = \overline{Q_3\overline{Q}_2\overline{Q}_1Q_0}.$$

Izvedena logička funkcija Y pogodna je za realizaciju pomoću jednog NI sklopa sa četiri ulaza.

Shema asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1 prikazana je na slici 6.53.



Slika 6.53: Shema asinkronog brojila koje prolazi kroz 10 stanja tako da se koriste asinkroni ulazi za postavljanje u vrijednost 1

6.5.1 Zadaci za vježbu

✎ **Zadatak 6.5.1**

Projektirajte 3-bitno asinkrono brojilo pomoću T bistabila. Kako ovo brojilo dijeli frekvenciju takt impulsa CP? Pomoću osnovnih logičkih sklopova realizirajte sklop koji će na izlazu dati logičku vrijednost 1 svakih osam impulsa takta CP.

✎ **Zadatak 6.5.2**

Projektirajte asinkrono brojilo pomoću JK bistabila i potrebnog broja NI sklopova koje će prolaziti kroz 13 stanja tako da:
a) koristite asinkrone ulaze za postavljanje u vrijednost 0 (CLR),
b) koristite asinkrone ulaze za postavljanje u vrijednost 1 (SET).

6.6 Sinkroni generator niza

Sinkroni generator niza izvodi se pomoću sinkronog brojila i logičke funkcije koja će u svakom stanju sinkronog brojila generirati traženu binarnu vrijednost. Dužina niza l koji se generira određuje broj stanja kroz koje sinkrono brojilo mora prolaziti. Na primjer:

- prstenasto brojilo s n bistabila može generirati niz duljine $l = n$,
- Johnsonovo brojilo s n bistabila može generirati niz duljine $l = 2n$,
- binarno sinkrono brojilo s n bistabila može generirati niz duljine $1 \leq l \leq 2^n$.

Od svih ponuđenih rješenja, prstenasto brojilo ima najveći utrošak bistabila za generiranje niza dužine l. Prednost ovog brojila je ta da sr ne moraju koristiti dodatni logički sklopovi za prijelaz između stanja.

Johnsonovo brojilo također ima prednost da ne mora koristiti dodatne logičke sklopove za prijelaz između stanja. Binarno sinkrono brojilo može generirati niz bilo koje duljine, ali za osiguranje prijelaza iz jednog stanja u drugo mora koristiti dodatne logičke sklopove.

☞ Primjer 6.6.1

Projektirajte sinkroni generator niza 10111101 pomoću Johnsonovog brojila. Nacrtajte logički sklop primjenom minimalnog broja JK bistabila i logičkih sklopova. Sklop mora biti zaštićen sklopom za siguran start u skladu s primjerom 6.2.2.

✍ Rješenje:

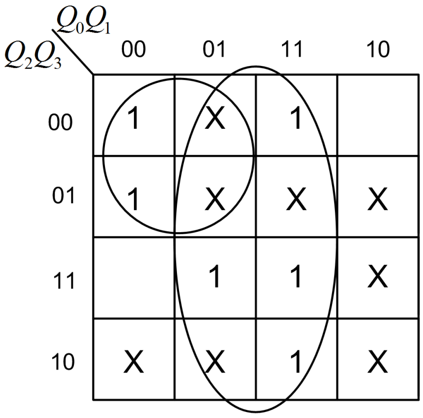
Niz 10111101 kojeg je potrebno generirati ima ukupno osam bitova ($l = 8$). Za generiranje osam bitova potrebno je brojilo koje broji do osam, odnosno brojilo koje prolazi kroz osam stanja. Johnsonovo brojilo prolazi kroz $2n$ stanja gdje je n broj bistabila. Prema tome, za realizaciju zadanog generatora niza potrebno je 4-bitno Johnsonovo brojilo. Tablica regularnih i neregularnih stanja 4-bitnog Johnsonovog brojila sinkronog generatora niza 10111101 prikazana je tablicom 6.16. Svako od osam regularnih stanja kroz koje Johnsonovo brojilo prolazi generirat će po jedan bit zadanog niza 10111101. Taj niz bitova nalazi se u tablici 6.16 u stupcu Y i njime je opisana logička funkcija Y. Za neregularna stanja Johnsonovog brojila logička funkcija Y poprimit će nespecificiranu vrijednost X.

Tablica 6.16: Tablica regularnih i neregularnih stanja 4-bitnog Johnsonovog brojila sinkronog generatora niza 10111101

Indeks minterma	Stanja bistabila				Generirani niz
i	Q_0	Q_1	Q_2	Q_3	Y
0	0	0	0	0	1
8	1	0	0	0	0
12	1	1	0	0	1
14	1	1	1	0	1
15	1	1	1	1	1
7	0	1	1	1	1
3	0	0	1	1	0
1	0	0	0	1	1
2	0	0	1	0	X
5	0	1	0	1	X
11	1	0	1	1	X
6	0	1	1	0	X
13	1	1	0	1	X
10	1	0	1	0	X
4	0	1	0	0	X
9	1	0	0	1	X

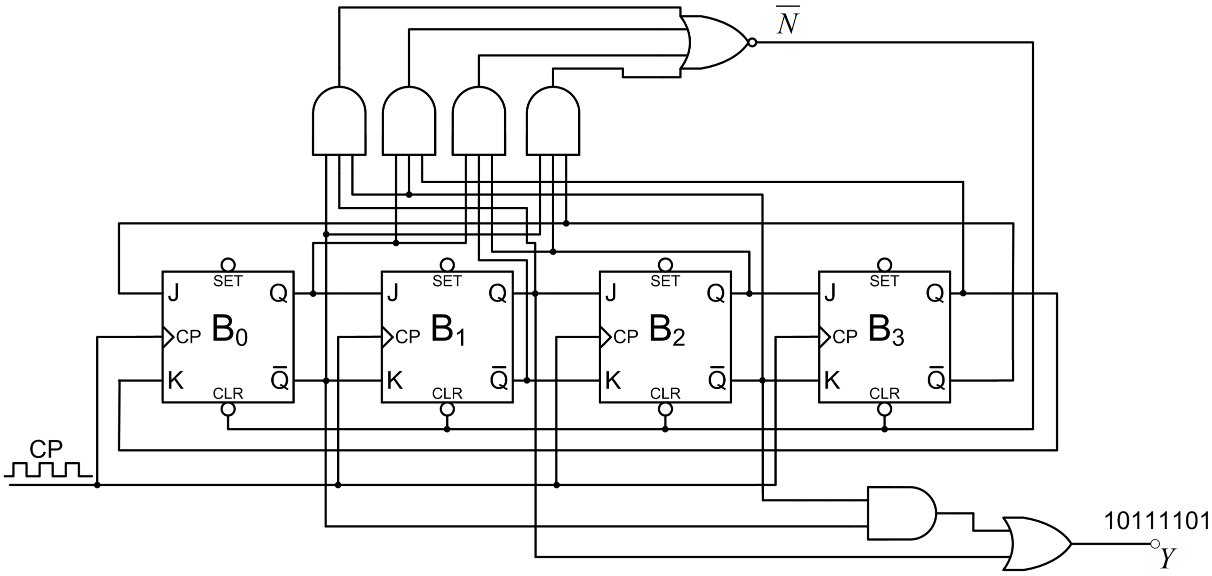
Osiguranje sigurnog starta Johnsonovog brojila prikazano je u primjeru 6.2.2. Logička funkcija Y je funkcija varijabli Q_0, Q_1, Q_2 i Q_3. Za logičku funkciju Y potrebno je pronaći minimalni oblik pomoću K-tablice. K-tablica logičke funkcije $Y(Q_0,Q_1,Q_2,Q_3)$ za sinkroni generator niza 10111101 prikazana je na slici 6.54. Minimalni oblik logičke funkcije $Y(Q_0,Q_1,Q_2,Q_3)$ je prema K-tablici na slici 6.54:

$$Y = Q_1 + \overline{Q}_2\overline{Q}_0.$$



Slika 6.54: K-tablica logičke funkcije $Y(Q_0,Q_1,Q_2,Q_3)$ za sinkroni generator niza 10111101

Shema sinkronog generatora niza 10111101 realiziranog Johnsonovim brojilom prikazana je na slici 6.55.



Slika 6.55: Shema sinkronog generatora niza 10111101 realiziranog Johnsonovim brojilom

☞ Primjer 6.6.2

Projektirajte sinkroni generator niza 110011 pomoću Johnsonovog brojila. Nacrtajte logički sklop primjenom minimalnog broja JK bistabila i logičkih sklopova. Sinkroni generator niza nije potrebno zaštititi sklopom za siguran start.

✍ Rješenje:

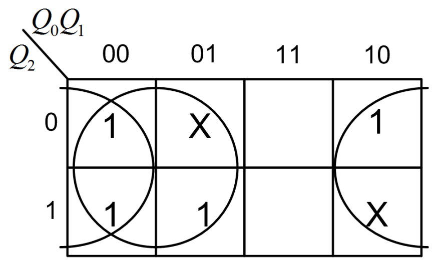
Niz 110011 kojeg je potrebno generirati ima ukupno šest bitova ($l = 6$). Za generiranje šest bitova potrebno je brojilo koje broji do šest, odnosno brojilo koje prolazi kroz šest stanja. Johnsonovo brojilo prolazi kroz $2n$ stanja gdje je n broj bistabila. Prema tome, za realizaciju zadanog generatora niza potrebno je 3-bitno Johnsonovo brojilo. Tablica regularnih i neregularnih stanja 3-bitnog Johnsonovog brojila sinkronog generatora niza 110011 prikazana je tablicom 6.17. Svako od šest regularnih stanja kroz koje Johnsonovo brojilo prolazi generirat će po jedan bit zadanog niza 110011. Taj niz bitova nalazi se u tablici 6.17 u stupcu Y i njime je opisana logička funkcija Y. Za neregularna stanja Johnsonovog brojila logička funkcija Y poprimit će nespecificiranu vrijednost X.

Tablica 6.17: Tablica regularnih i neregularnih stanja 3-bitnog Johnsonovog brojila sinkronog generatora niza 110011

Indeks minterma	Stanja bistabila			Generirani niz
i	Q_0	Q_1	Q_2	Y
0	0	0	0	1
4	1	0	0	1
6	1	1	0	0
7	1	1	1	0
3	0	1	1	1
1	0	0	1	1
2	0	1	0	X
5	1	0	1	X

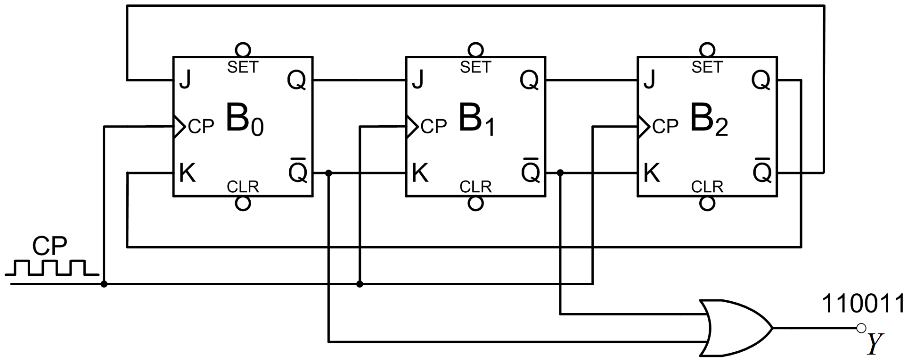
Logička funkcija Y je funkcija varijabli Q_0, Q_1 i Q_2. Za logičku funkciju Y potrebno je pronaći minimalni oblik pomoću K-tablice. K-tablica logičke funkcije $Y(Q_0,Q_1,Q_2)$ za sinkroni generator niza 110011 prikazana je na slici 6.56. Minimalni oblik logičke funkcije $Y(Q_0,Q_1,Q_2)$ je prema K-tablici na slici 6.56:

$$Y = \overline{Q}_1 + \overline{Q}_0.$$



Slika 6.56: K-tablica logičke funkcije $Y(Q_0,Q_1,Q_2)$ za sinkroni generator niza 110011

Shema sinkronog generatora niza 110011 realiziranog Johnsonovim brojilom prikazana je na slici 6.57.



Slika 6.57: Shema sinkronog generatora niza 110011 realiziranog Johnsonovim brojilom

6.6.1 Zadaci za vježbu

✎ Zadatak 6.6.1

Projektirajte sinkroni generator niza 01001111 pomoću Johnsonovog brojila. Nacrtajte logički sklop primjenom minimalnog broja JK bistabila i logičkih sklopova. Sklop mora biti zaštićen sklopom za siguran start u skladu s primjerom 6.2.2.

✎ Zadatak 6.6.2

Projektirajte sinkroni generator niza 011010 pomoću Johnsonovog brojila. Nacrtajte logički sklop primjenom minimalnog broja JK bistabila i logičkih sklopova. Sinkroni generator niza nije potrebno zaštititi sklopom za siguran start.

6.7 Sinkroni detektor niza

Sinkroni detektor niza opisat ćemo pomoću digitalnog automata. Automat za sinkroni detektor definiraju:

- skup stanja automata: $S = \{$a, b, c, d,... $\}$,
- ulaz automata: $x = \{0, 1\}$,
- izlaz automata: $y = \{0, 1\}$,
- funkcija prijelaza koja ovisno o stanju i ulazu automata kreira novo stanje automata,
- početno stanje automata: $S_0 =$ a.

Ovako definirani automat naziva se Mealyev automat. Automat ćemo realizirati pomoću bistabila.

☞ Primjer 6.7.1

Konstruirajte digitalni automat za detektor niza 0100:
a) bez preklapanja,
b) s preklapanjem.

✍ **Rješenje:** Detekcija niza 0100 bez preklapanja prikazana je na slici 6.58.

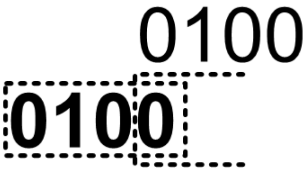
$$1\boxed{0100}10010111\boxed{0100}1\boxed{0100}10\boxed{0100}111\boxed{0100}$$

Slika 6.58: Detekcija niza 0100 bez preklapanja

U detekciji niza bez preklapanja svaki novi detektirani niz ne može biti podniz već detektiranog niza. Detekcija niza 0100 s preklapanjem prikazana je na slici 6.59.

Slika 6.59: Detekcija niza 0100 s preklapanjem

U detekciji niza s preklapanjem svaki novi detektirani niz može biti podniz već detektiranog niza. Preklapanje u nizu 0100 prikazano je na slici 6.60. U nizu kojeg želimo detektirati s preklapanjem potrebno je početak niza pronaći na kraju niza. Na slici 6.60 vidimo da je početak niza 0, a kraj niza također 0. Prema tome, kraj niza može biti kandidat za novi detektirani niz s preklapanjem. Na primjer, ako želimo detektirati niz 00100 s preklapanjem, tada se početak niza 00 pronalazi na kraju niza. Detekcija niza 1000 bit će jednaka s preklapanjem i bez preklapanja jer početak niza nije moguće pronaći na kraju niza.



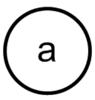
Slika 6.60: Preklapanje u nizu 0100

a) Za detekciju niza 0100 potrebno je konstruirati digitalni automat. Broj stanja digitalnog automata jednak je broju bitova niza kojeg je potrebno detektirati. U slučaju detekcije niza 0100 imat ćemo četiri stanja.

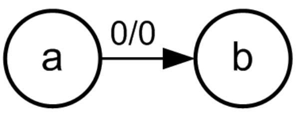
Početno stanje automata je stanje `a` i prikazano je na slici 6.61a. Stanje `a` prazno je stanje ($\mathtt{a} = \emptyset$), odnosno stanje u kojem nije detektiran nijedan bit niza 0100.

Ako je automat u stanju `a` i na ulazu se pojavi vrijednost 0, automat će prijeći u novo stanje, stanje `b` (slika 6.61b). Funkcija prijelaza označena je strelicom koja ima smjer iz stanja `a` u stanje `b`. Na strelici se nalazi trenutno stanje ulaza/trenutno stanje izlaza ($x/y = 0/0$). Izlaz će imati vrijednost 1 samo kada niz 0100 bude detektiran. Stanje `b` odgovara stanju u kojem je pronađen prvi bit zadanog niza 0100 ($\mathtt{b} = 0$).

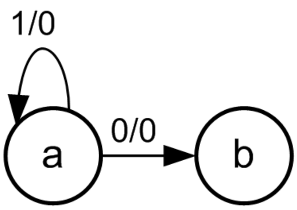
Ako je automat u stanju `a` i na ulazu se pojavi vrijednost 1, automat će ostati u stanju `a` (slika 6.61c) jer vrijednost 1 nije početak niza 0100. Funkcija prijelaza u ovom je slučaju označena strelicom koja ima smjer iz stanja `a` u stanje `a`, a na strelici se nalazi 1/0 ($x/y = 1/0$).



(a) Početno stanje automata `a`



(b) Funkcija prijelaza za ulaz 0 i stanje `a`



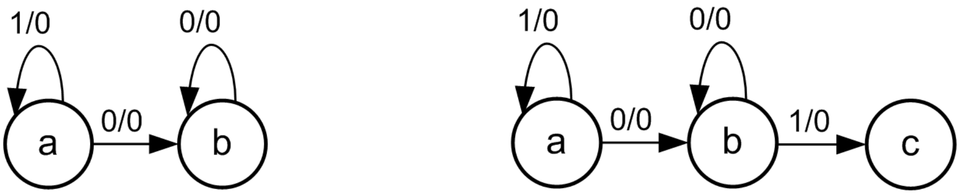
(c) Funkcija prijelaza za ulaz 1 i stanje `a`

Slika 6.61: Funkcije prijelaza za stanje automata `a` - detektor niza 0100 bez preklapanja

Ako je automat u stanju `b` i na ulazu se pojavi vrijednost 0, to odgovara ulaznom nizu 00. Budući da traženi niz počinje s 01, automat će ostati u stanju `b` (slika 6.62a) jer je zadnja vrijednost ulaza bila 0 što upravo odgovara stanju `b`. Funkcija prijelaza u ovom je slučaju označena strelicom koja ima smjer iz stanja `b` u stanje `b`, a na strelici se nalazi 0/0 ($x/y = 0/0$).

Ako je automat u stanju `b` i na ulazu se pojavi vrijednost 1, automat će prijeći u novo stanje,

stanje c (slika 6.62b). Funkcija prijelaza označena je strelicom koja ima smjer iz stanja b u stanje c, a na strelici se nalazi 1/0 ($x/y = 1/0$). Stanje c odgovara stanju u kojem su pronađena prva dva bita zadanog niza 0100 (c = 01).

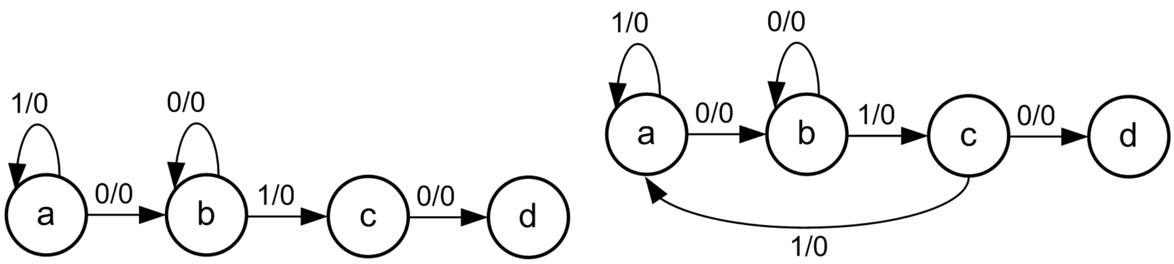


(a) Funkcija prijelaza za ulaz 0 i stanje b   (b) Funkcija prijelaza za ulaz 1 i stanje b

Slika 6.62: Funkcije prijelaza za stanje automata b - detektor niza 0100 bez preklapanja

Ako je automat u stanju c i na ulazu se pojavi vrijednost 0, automat će prijeći prijeći u novo stanje, stanje d (slika 6.63a). Funkcija prijelaza označena je strelicom koja ima smjer iz stanja c u stanje d, a na strelici se nalazi 0/0 ($x/y = 0/0$). Stanje d odgovara stanju u kojem su pronađena prva tri bita zadanog niza 0100 (d = 010).

Ako je automat u stanju c i na ulazu se pojavi vrijednost 1 to odgovara ulaznom nizu 011. Budući da traženi niz počinje s 010, automat će prijeći u stanje a (slika 6.63b) jer je zadnja vrijednost ulaza bila 1 što ne odgovara prethodnim stanjima. Funkcija prijelaza u je ovom slučaju označena strelicom koja ima smjer iz stanja c u stanje a, a na strelici se nalazi 1/0 ($x/y = 1/0$).



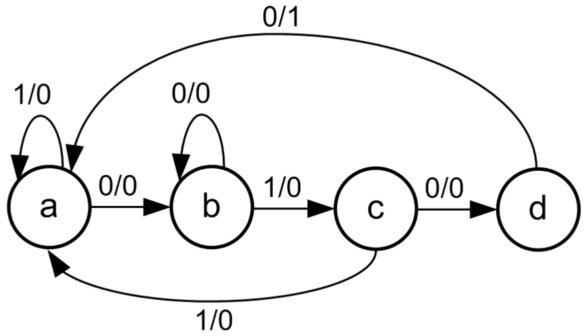
(a) Funkcija prijelaza za ulaz 0 i stanje c   (b) Funkcija prijelaza za ulaz 1 i stanje c

Slika 6.63: Funkcije prijelaza za stanje automata c - detektor niza 0100 bez preklapanja

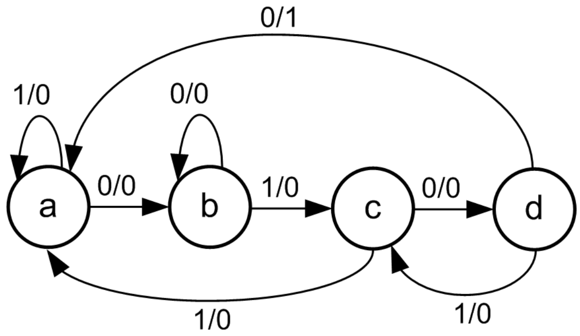
Ako je automat u stanju d i na ulazu se pojavi vrijednost 0, ulazni je niz jednak traženom nizu 0100. U detekciji niza bez preklapanja, nakon detektiranog niza 0100, sljedeće stanje je stanje a (slika 6.64a). Funkcija prijelaza označena je strelicom koja ima smjer iz stanja d u stanje a, a na strelici se nalazi 0/1 ($x/y = 0/1$). Vrijednost 1 generirana na izlazu automata označava da je traženi niz 0100 detektiran. Stanje a odgovara praznom stanju (a = $\emptyset$).

Ako je automat u stanju d i na ulazu se pojavi vrijednost 1, to odgovara ulaznom nizu 0101. Budući da je traženi niz 0100, automat će prijeći u stanje c (slika 6.64b) jer su zadnje dvije vrijednosti ulaza bile 01 što upravo odgovara stanju c. Funkcija prijelaza u ovom je slučaju označena strelicom koja ima smjer iz stanja d u stanje c, a na strelici se nalazi 1/0 ($x/y = 1/0$).

Detektor niza 0100 bez preklapanja prikazan je na slici 6.64a.



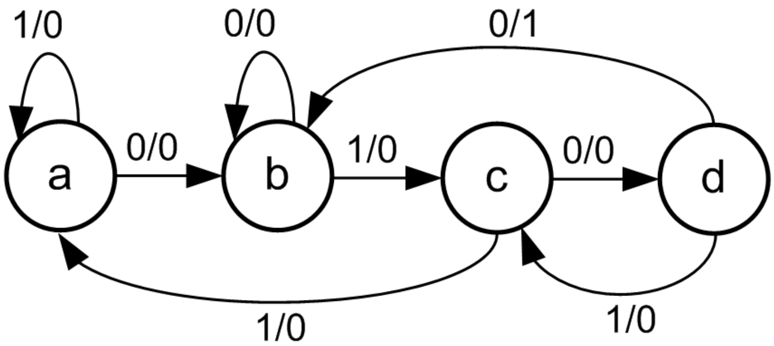
(a) Funkcija prijelaza za ulaz 0 i stanje `d`



(b) Funkcija prijelaza za ulaz 1 i stanje `d`

Slika 6.64: Funkcije prijelaza za stanje automata `d` - detektor niza 0100 bez preklapanja

b) Detektor niza 0100 s preklapanjem prikazan je na slici 6.65. Jedina razlika detektora s preklapanjem (slika 6.65) i detektora bez preklapanja (slika 6.64a) je u funkciji prijelaza kada u stanju `d` na ulazu automata pristigne vrijednost 0. U nizu 0100 na kraju se niza nalazi početak niza, kako je prikazano na slici 6.60. Zadnja vrijednost 0 koja je pristigla na ulaz automata kada se on nalazio u stanju `d` kandidat je za početak niza 0100 pa automat stoga prelazi iz stanja `d` u stanje `b`. Stanje `b` je stanje u kojoj je pronađen prvi bit zadanog niza 0100.



Slika 6.65: Detektor niza 0100 s preklapanjem

☞ Primjer 6.7.2

Konstruirajte digitalni automat za detektor niza s preklapanjem i bez preklapanja za sljedeće nizove:
a) 1110,
b) 1001,
c) 101101,
d) 10101010.

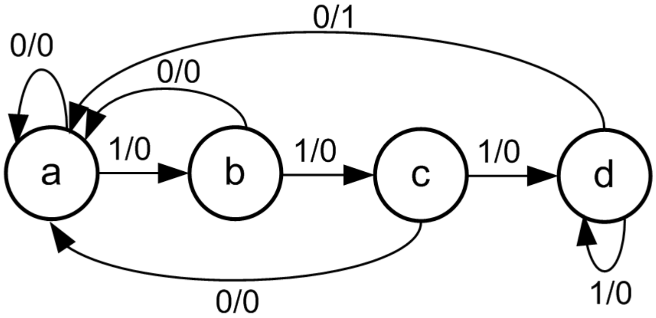
✍ Rješenje:

a) Detektor niza 1110 s preklapanjem i bez preklapanja prikazan je na slici 6.66. U slučaju kada kraj niza nije moguće pronaći na početku niza, detektor niza s preklapanjem i bez preklapanja bit će isti. Niz 1110 ima ukupno četiri bita, a sukladno tome i sljedeća četiri stanja:

- `a` $= \emptyset$ - prazno stanje,
- `b` $= 1$ - stanje koje odgovara detekciji prvog bita niza 1110,

- `c` = 11 - stanje koje odgovara detekciji prva dva bita niza 1110,
- `d` = 111 - stanje koje odgovara detekciji prva tri bita niza 1110.

Iz stanja `d` automat se vraća u stanje `a` za ulaznu vrijednost 0 i generira izlaznu vrijednost 1.

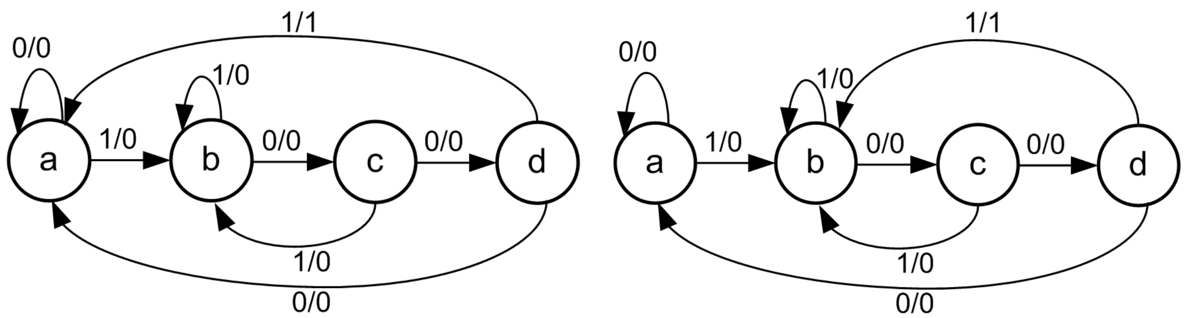


Slika 6.66: Detektor niza 1110 s preklapanjem i bez preklapanja

b) Detektor niza 1001 bez preklapanja prikazan je na slici 6.67a, dok je detektor niza 1001 s preklapanjem prikazan na slici 6.67b. U slučaju niza 1001, zadnji bit niza ujedno je i prvi bit niza pa postoji preklapanje. Niz 1001 ima ukupno četiri bita, a sukladno tome i sljedećih četiri stanja:

- `a` = $\emptyset$ - prazno stanje,
- `b` = 1 - stanje koje odgovara detekciji prvog bita niza 1001,
- `c` = 10 - stanje koje odgovara detekciji prva dva bita niza 1001,
- `d` = 100 - stanje koje odgovara detekciji prva tri bita niza 1001.

Automat za detektor niza 1001 bez preklapanja (slika 6.67a) vraća se iz stanja `d` u stanje `a` za ulaznu vrijednost 1 i generira izlaznu vrijednost 1. Automat za detektor niza 1001 s preklapanjem (slika 6.67b) vraća se iz stanja `d` u stanje `b` (`b` = 1) za ulaznu vrijednost 1 i generira izlaznu vrijednost 1.

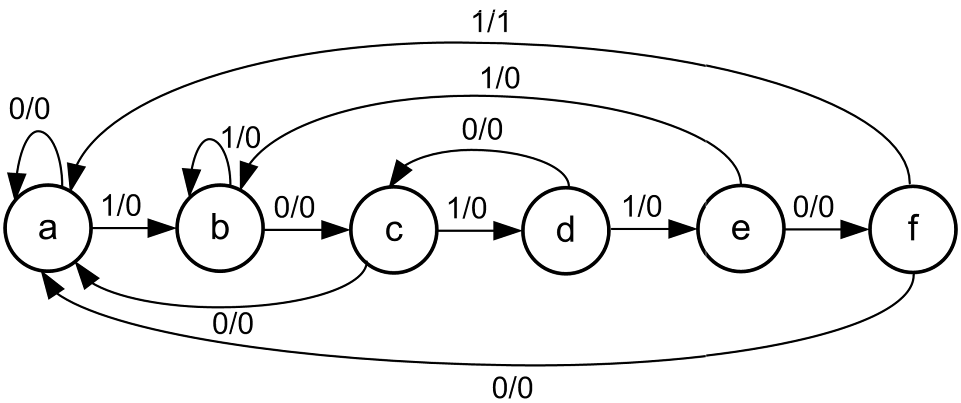


(a) Detektor niza 1001 bez preklapanja (b) Detektor niza 1001 s preklapanjem

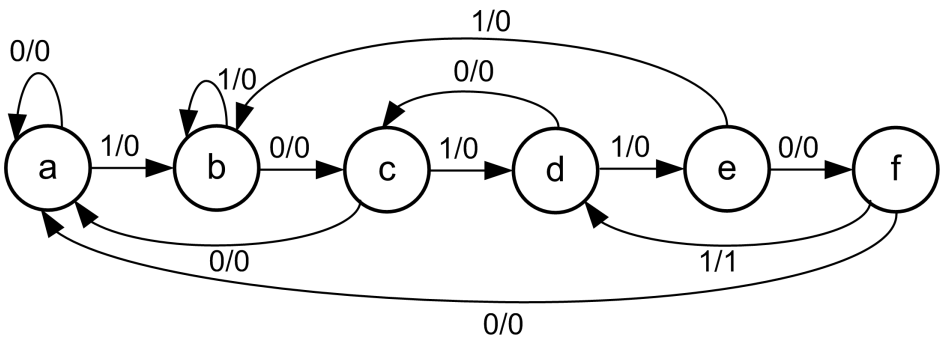
Slika 6.67: Detektori niza 1001

c) Detektor niza 101101 bez preklapanja prikazan je na slici 6.68, dok je detektor niza 101101 s preklapanjem prikazan na slici 6.69. U slučaju niza 101101, zadnja tri bita (101) niza ujedno su početak niza 101101 pa postoji preklapanje. Niz 101101 ima ukupno šest bitova, a sukladno tome i sljedećih šest stanja:

- `a` = $\emptyset$ - prazno stanje,
- `b` = 1 - stanje koje odgovara detekciji prvog bita niza 101101,
- `c` = 10 - stanje koje odgovara detekciji prva dva bita niza 101101,
- `d` = 101 - stanje koje odgovara detekciji prva tri bita niza 101101,
- `e` = 1011 - stanje koje odgovara detekciji prva četiri bita niza 101101,
- `f` = 10110 - stanje koje odgovara detekciji prvih pet bitova niza 101101.



Slika 6.68: Detektor niza 101101 bez preklapanja



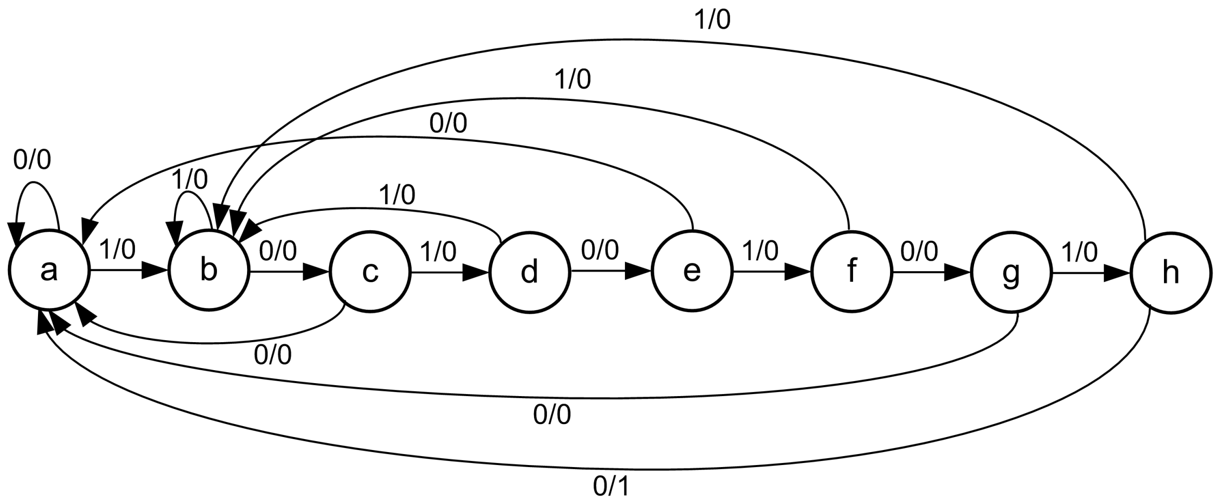
Slika 6.69: Detektor niza 101101 s preklapanjem

Automat za detektor niza 101101 bez preklapanja (slika 6.68) vraća se iz stanja `f` u stanje `a` za ulaznu vrijednost 1 i generira izlaznu vrijednost 1. Automat za detektor niza 101101 s preklapanjem (slika 6.69) vraća se iz stanja `f` u stanje `c` (`c` = 101) za ulaznu vrijednost 1 i generira izlaznu vrijednost 1.

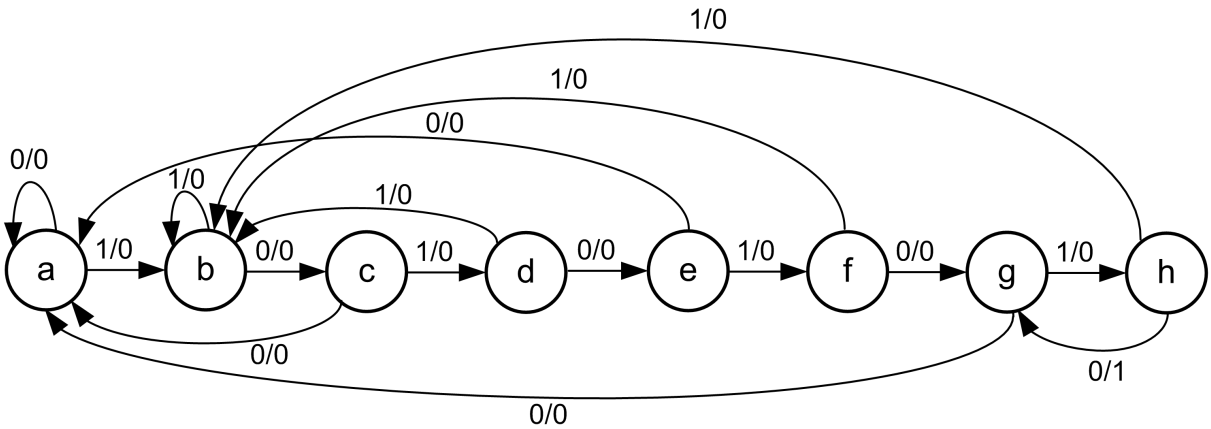
d) Detektor niza 10101010 bez preklapanja prikazan je na slici 6.70, dok je detektor niza 10101010 s preklapanjem prikazan na slici 6.71. U slučaju niza 10101010, zadnjih šest bitova (101010) niza ujedno su početak niza 10101010 pa postoji preklapanje. Niz 10101010 ima ukupno osam bitova, a sukladno tome i sljedećih osam stanja:

- `a` = $\emptyset$ - prazno stanje,
- `b` = 1 - stanje koje odgovara detekciji prvog bita niza 10101010,
- `c` = 10 - stanje koje odgovara detekciji prva dva bita niza 10101010,

- `d` = 101 - stanje koje odgovara detekciji prva tri bita niza 10101010,
- `e` = 1010 - stanje koje odgovara detekciji prva četiri bita niza 10101010,
- `f` = 10101 - stanje koje odgovara detekciji prvih pet bitova niza 10101010,
- `g` = 101010 - stanje koje odgovara detekciji prvih šest bitova niza 10101010,
- `h` = 1010101 - stanje koje odgovara detekciji prvih sedam bitova niza 10101010.



Slika 6.70: Detektor niza 10101010 bez preklapanja



Slika 6.71: Detektor niza 10101010 s preklapanjem

Automat za detektor niza 10101010 bez preklapanja (slika 6.70) vraća se iz stanja `h` u stanje `a` za ulaznu vrijednost 0 i generira izlaznu vrijednost 1. Automat za detektor niza 10101010 s preklapanjem (slika 6.71) vraća se iz stanja `h` u stanje `g` (`g` = 101010) za ulaznu vrijednost 0 i generira izlaznu vrijednost 1.

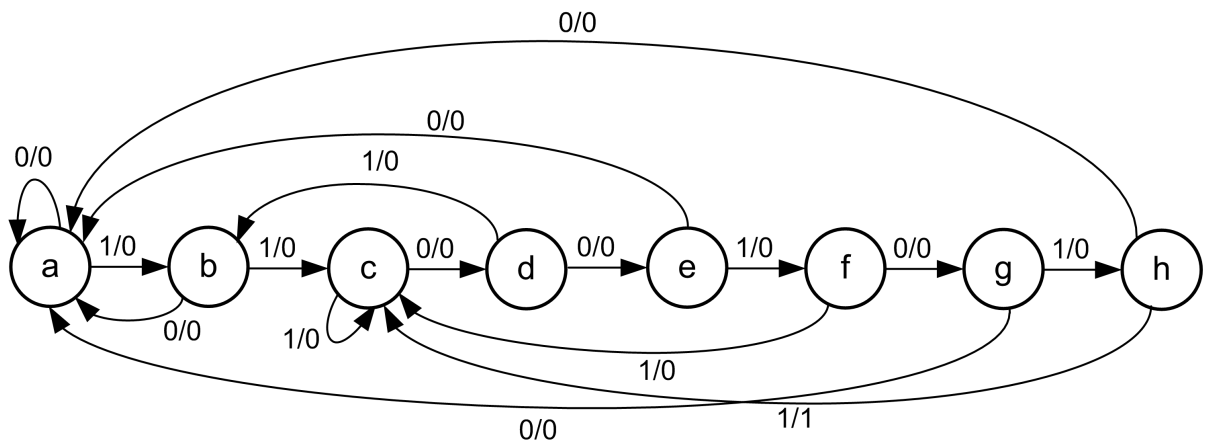
☞ Primjer 6.7.3

Konstruirajte digitalni automat za detektor niza 11001011, a zatim definirajte i ispišite znakovnu tablicu prijelaza stanja za taj detektor.

✍ Rješenje:

Detektor niza 11001011 s preklapanjem prikazan je na slici 6.72. U slučaju niza 11001011, zadnja dva bita (11) niza ujedno su početak niza 11001011 pa postoji preklapanje. Niz 11001011 ima ukupno osam bitova, a sukladno tome i sljedećih osam stanja:

- `a` = $\emptyset$ - prazno stanje,
- `b` = 1 - stanje koje odgovara detekciji prvog bita niza 11001011,
- `c` = 11 - stanje koje odgovara detekciji prva dva bita niza 11001011,
- `d` = 110 - stanje koje odgovara detekciji prva tri bita niza 11001011,
- `e` = 1100 - stanje koje odgovara detekciji prva četiri bita niza 11001011,
- `f` = 11001 - stanje koje odgovara detekciji prvih pet bitova niza 11001011,
- `g` = 110010 - stanje koje odgovara detekciji prvih šest bitova niza 11001011,
- `h` = 1100101 - stanje koje odgovara detekciji prvih sedam bitova niza 11001011.



Slika 6.72: Detektor niza 11001011 s preklapanjem

Automat za detektor niza 11001011 s preklapanjem (slika 6.72) vraća se iz stanja `h` u stanje `c` (`c` = 11) za ulaznu vrijednost 1 i generira izlaznu vrijednost 1.

Znakovna tablica stanja je tablica u kojoj se stanjima digitalnog automata pridružuju stanja sinkronog brojila. Automat za detektor niza 11001011 ima osam stanja pa su za njegovu realizaciju pomoću sinkronog brojila potrebna tri bistabila. Znakovna tablica stanja detektora niza 11001011 prikazana je tablicom 6.18. Stanja su redom dodijeljena binarnim kombinacijama 3-bitnog sinkronog brojila.

Tablica 6.18: Znakovna tablica stanja detektora niza 11001011

	Bistabili			Stanje automata
i	Q_2	Q_1	Q_0	S
0	0	0	0	a
1	0	0	1	b
2	0	1	0	c
3	0	1	1	d
4	1	0	0	e
5	1	0	1	f
6	1	1	0	g
7	1	1	1	h

Znakovna tablica prijelaza stanja detektora niza 11001011 konstruira se pomoću znakovne tablice stanja (tablica 6.18) i automata za detektor niza 11001011 s preklapanjem (slika 6.72). Znakovna tablica prijelaza stanja detektora niza 11001011 prikazana je tablicom 6.19.

Tablica 6.19: Znakovna tablica prijelaza stanja detektora niza 11001011

Ulaz	Sadašnje stanje automata n	Sljedeće stanje automata, $n+1$	Izlaz
x	S	S	y
0	a	a	0
0	b	a	0
0	c	d	0
0	d	e	0
0	e	a	0
0	f	g	0
0	g	a	0
0	h	a	0
1	a	b	0
1	b	c	0
1	c	c	0
1	d	b	0
1	e	f	0
1	f	c	0
1	g	h	0
1	h	c	1

U tablici 6.19 nalazi se stupac ulaznih vrijednosti automata x i izlaznih vrijednosti automata y. Na primjer, ako je automat u stanju `c` i na ulazu se nalazi vrijednost 0, prema slici 6.72 sljedeće će stanje biti stanje `d`, a na izlazu će se generirati vrijednost 0. Na isti je način za vrijednosti ulaza 0 i 1 te za svako stanje u kojem se automat nalazi potrebno ispisati prijelaze u sljedeća stanja te vrijednosti izlaza za te prijelaze (tablica 6.19).

Ako se u znakovnoj tablici prijelaza stanja detektora niza 11001011 napravi zamjena znakovnih stanja sa stanjima bistabila prema tablici 6.18, dobit ćemo proširenu znakovnu tablicu prijelaza stanja detektora niza 11001011 (tablica 6.20). Ova se tablica može koristiti za kreiranje uzbude bilo kojeg bistabila kako bi se ostvarili traženi prijelazi iz jednog u drugo stanje sinkronog brojila. Kod minimizacije logičkih funkcija ulaza bistabila u obzir treba uzeti i logičku varijablu x kao ulaz u automat. Na primjer, ako koristimo JK bistabil za realizaciju detektora niza, ulaz J_0

bistabila B_0 bit će logička funkcija varijabli x, Q_2, Q_1 i Q_0 ($J_0(x, Q_2, Q_1, Q_0)$).

Tablica 6.20: Proširena znakovna tablica prijelaza stanja detektora niza 11001011

	Ulaz	Sadašnje stanje bistabila, n			Sljedeće stanje bistabila, $n+1$			Izlaz
i	x	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	y
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
2	0	0	1	0	0	1	1	0
3	0	0	1	1	1	0	0	0
4	0	1	0	0	0	0	0	0
5	0	1	0	1	1	1	0	0
6	0	1	1	0	0	0	0	0
7	0	1	1	1	0	0	0	0
8	1	0	0	0	0	0	1	0
9	1	0	0	1	0	1	0	0
10	1	0	1	0	0	1	0	0
11	1	0	1	1	0	0	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	0	1	0	0
14	1	1	1	0	1	1	1	0
15	1	1	1	1	0	1	0	1

☞ Primjer 6.7.4

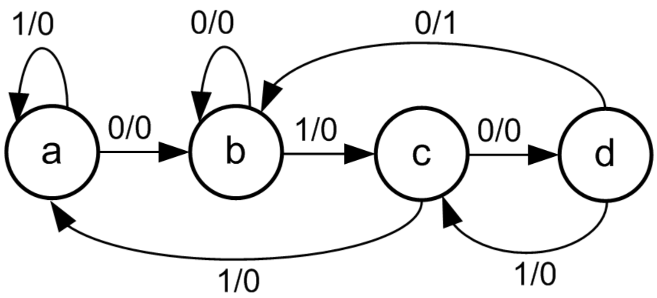
Projektirajte sinkroni detektor niza 0100 s preklapanjem primjenom:
a) minimalnog broja JK bistabila i osnovnih logičkih sklopova,
b) minimalnog broja T bistabila i osnovnih logičkih sklopova,
c) minimalnog broja D bistabila i osnovnih logičkih sklopova.
Detektor niza mora na izlazu y generirati vrijednost 1 svaki puta kada detektira niz 0100. Definirajte i ispišite znakovnu tablicu prijelaza stanja za zadani detektor.

✍ Rješenje:

Detektor niza 0100 s preklapanjem prikazan je na slici 6.73. U slučaju niza 0100, zadnji bit (0) niza ujedno je početak niza 0100 pa postoji preklapanje. Niz 0100 ima ukupno četiri bita, a sukladno tome i sljedeća četiri stanja:

- `a` $= \emptyset$ - prazno stanje,
- `b` $= 0$ - stanje koje odgovara detekciji prvog bita niza 0100,
- `c` $= 01$ - stanje koje odgovara detekciji prva dva bita niza 0100,
- `d` $= 010$ - stanje koje odgovara detekciji prva tri bita niza 0100.

Automat za detektor niza 0100 s preklapanjem (slika 6.73) vraća se iz stanja `d` u stanje `b` (`b` $= 0$) za ulaznu vrijednost 0 i generira izlaznu vrijednost 1.



Slika 6.73: Detektor niza 0100 s preklapanjem

Za realizaciju automata za detektor niza 0100 pomoću sinkronog brojila potrebna su dva bistabila jer su moguća četiri stanja automata sa slike 6.73. Znakovna tablica stanja detektora niza 0100 prikazana je tablicom 6.21. Stanja su redom dodijeljena binarnim kombinacijama 2-bitnog sinkronog brojila.

Tablica 6.21: Znakovna tablica stanja detektora niza 0100

	Bistabili		Stanje automata
i	Q_1	Q_0	S
0	0	0	a
1	0	1	b
2	1	0	c
3	1	1	d

Znakovna tablica prijelaza stanja detektora niza 0100 konstruira se pomoću znakovne tablice stanja (tablica 6.21) i automata za detektor niza 0100 s preklapanjem (slika 6.73). Znakovna tablica prijelaza stanja detektora niza 0100 prikazana je tablicom 6.22.

Tablica 6.22: Znakovna tablica prijelaza stanja detektora niza 0100

Ulaz	Sadašnje stanje automata n	Sljedeće stanje automata, $n+1$	Izlaz
x	S	S	y
0	a	b	0
0	b	b	0
0	c	d	0
0	d	b	1
1	a	a	0
1	b	c	0
1	c	a	0
1	d	c	0

U tablici 6.22 nalazi se stupac ulaznih vrijednosti automata x i izlaznih vrijednosti automata y. Na primjer, ako je automat u stanju `b` i na ulazu se nalazi vrijednost 1, prema slici 6.73 sljedeće će stanje biti stanje `c`, a na izlazu će se generirati vrijednost 0. Na isti je način za vrijednosti ulaza 0 i 1 te za svako stanje u kojem se automat nalazi potrebno ispisati prijelaze u sljedeća stanja te vrijednosti izlaza za te prijelaze (tablica 6.22).

Ako se u znakovnoj tablici prijelaza stanja detektora niza 0100 napravi zamjena znakovnih stanja sa stanjima bistabila prema tablici 6.21, dobit ćemo proširenu znakovnu tablicu prijelaza stanja detektora niza 0100 (tablica 6.23). Ovu tablicu koristit ćemo za kreiranje uzbude JK, T i D bistabila kako bi se ostvarili traženi prijelazi iz jednog u drugo stanje sinkronog brojila. Kod minimizacije logičkih funkcija ulaza bistabila u obzir treba uzeti i logičku varijablu x kao ulaz u automat.

Tablica 6.23: Proširena znakovna tablica prijelaza stanja detektora niza 0100

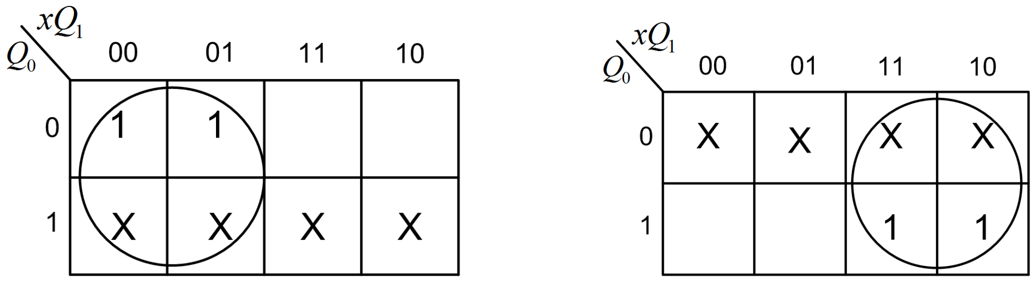
	Ulaz	Sadašnje stanje bistabila, n		Sljedeće stanje bistabila, $n+1$		Izlaz
i	x	Q_1	Q_0	Q_1	Q_0	y
0	0	0	0	0	1	0
1	0	0	1	0	1	0
2	0	1	0	1	1	0
3	0	1	1	0	1	1
4	1	0	0	0	0	0
5	1	0	1	1	0	0
6	1	1	0	0	0	0
7	1	1	1	1	0	0

a) Tablica uzbude JK bistabila za detektor niza 0100 prikazana je tablicom 6.24. U kreiranju stupaca uzbude bistabila B_0 i B_1 korištena je tablica uzbude JK bistabila (tablica 5.4).

Tablica 6.24: Tablica uzbude JK bistabila za detektor niza 0100

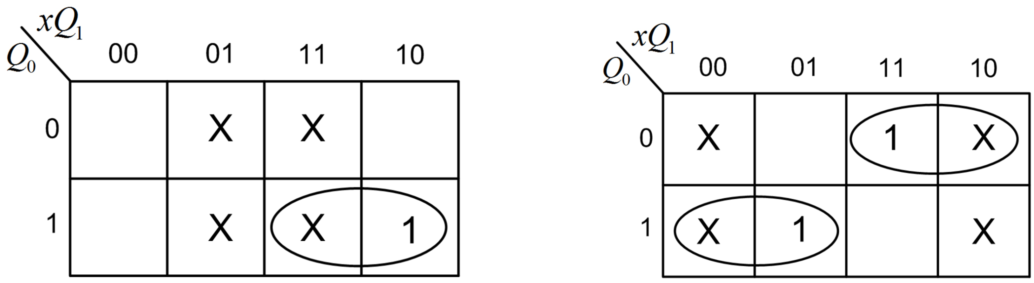
	Ulaz	Sadašnje stanje bistabila, n		Sljedeće stanje bistabila, $n+1$		Uzbuda bistabila				Izlaz
						B_1		B_0		
i	x	Q_1	Q_0	Q_1	Q_0	J_1	K_1	J_0	K_0	y
0	0	0	0	0	1	0	X	1	X	0
1	0	0	1	0	1	0	X	X	0	0
2	0	1	0	1	1	X	0	1	X	0
3	0	1	1	0	1	X	1	X	0	1
4	1	0	0	0	0	0	X	0	X	0
5	1	0	1	1	0	1	X	X	1	0
6	1	1	0	0	0	X	1	0	X	0
7	1	1	1	1	0	X	0	X	1	0

Stupci uzbude bistabila u tablici 6.24 logičke su funkcije varijabli x, Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaze JK bistabila. Radi lakšeg popunjavanja K-tablice u tablici 6.24 prvi stupac je stupac indeksa minterma i. K-tablice za ulaze JK bistabila B_0 i B_1 detektora niza 0100 prikazane su na slikama 6.74 i 6.75.



(a) K-tablica za $J_0(x,Q_1,Q_0)$ (b) K-tablica za $K_0(x,Q_1,Q_0)$

Slika 6.74: K-tablice za ulaze JK bistabila B_0 detektora niza 0100



(a) K-tablica za $J_1(x,Q_1,Q_0)$ (b) K-tablica za $K_1(x,Q_1,Q_0)$

Slika 6.75: K-tablice za ulaze JK bistabila B_1 detektora niza 0100

Minimalni oblik logičke funkcije $J_0(x,Q_1,Q_0)$ je prema K-tablici na slici 6.74a:

$$J_0 = \bar{x}.$$

Minimalni oblik logičke funkcije $K_0(x,Q_1,Q_0)$ je prema K-tablici na slici 6.74b:

$$K_0 = x.$$

Minimalni oblik logičke funkcije $J_1(x,Q_1,Q_0)$ je prema K-tablici na slici 6.75a:

$$J_1 = xQ_0.$$

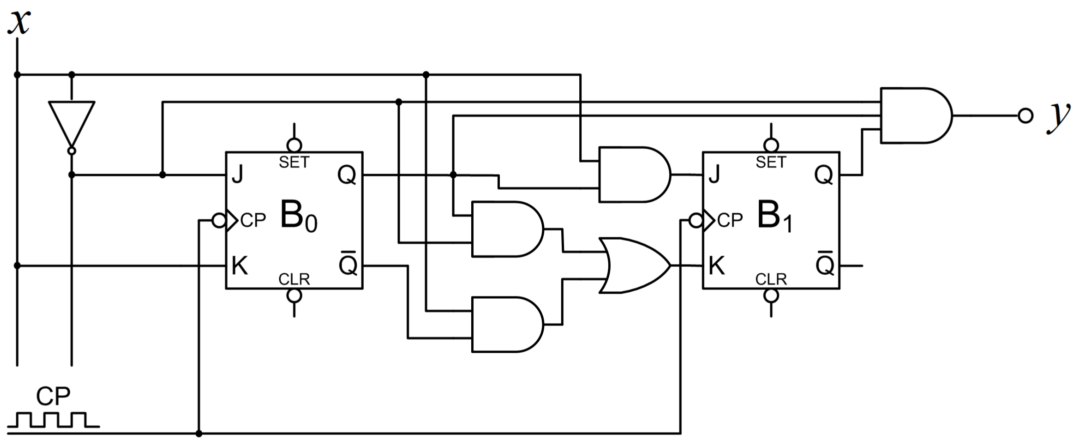
Minimalni oblik logičke funkcije $K_1(x,Q_1,Q_0)$ je prema K-tablici na slici 6.75b:

$$K_1 = x\overline{Q_0} + \bar{x}Q_0.$$

Izlaz y detektora niza 0100 u tablici 6.24 ima vrijednost 1 za minterm m_3 pa vrijedi:

$$y = \bar{x}Q_1Q_0.$$

Na temelju izvedenih logičkih funkcija za J i K ulaze bistabila te logičke funkcije izlaza y dobivena je shema detektora niza 0100 izvedenog pomoću JK bistabila. Shema je prikazana na slici 6.76.



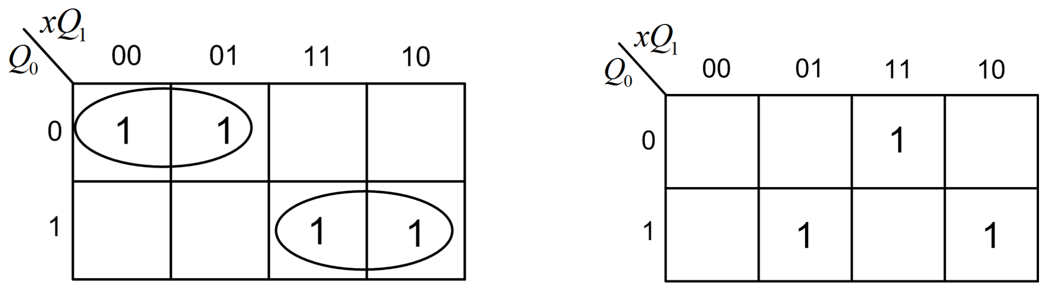
Slika 6.76: Shema detektora niza 0100 izvedenog pomoću JK bistabila

b) Tablica uzbude T bistabila za detektor niza 0100 prikazana je tablicom 6.25. U kreiranju stupaca uzbude bistabila B_0 i B_1 korištena je tablica uzbude T bistabila (tablica 5.6).

Tablica 6.25: Tablica uzbude T bistabila za detektor niza 0100

	Ulaz	Sadašnje stanje bistabila, n		Sljedeće stanje bistabila, $n+1$		Uzbuda bistabila		Izlaz
i	x	Q_1	Q_0	Q_1	Q_0	T_1	T_0	y
0	0	0	0	0	1	0	1	0
1	0	0	1	0	1	0	0	0
2	0	1	0	1	1	0	1	0
3	0	1	1	0	1	1	0	1
4	1	0	0	0	0	0	0	0
5	1	0	1	1	0	1	1	0
6	1	1	0	0	0	1	0	0
7	1	1	1	1	0	0	1	0

Stupci uzbude bistabila u tablici 6.25 logičke su funkcije varijabli x, Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaze T bistabila. Radi lakšeg popunjavanja K-tablice u tablici 6.25 prvi stupac je stupac indeksa minterma i. K-tablice za ulaze T bistabila B_0 i B_1 detektora niza 0100 prikazane su slici 6.77.



(a) K-tablica za $T_0(x,Q_1,Q_0)$ (b) K-tablica za $T_1(x,Q_1,Q_0)$

Slika 6.77: K-tablice za ulaze T bistabila B_0 i B_1 detektora niza 0100

Minimalni oblik logičke funkcije $T_0(x,Q_1,Q_0)$ je prema K-tablici na slici 6.77a:

$$T_0 = \bar{x}\overline{Q_0} + xQ_0.$$

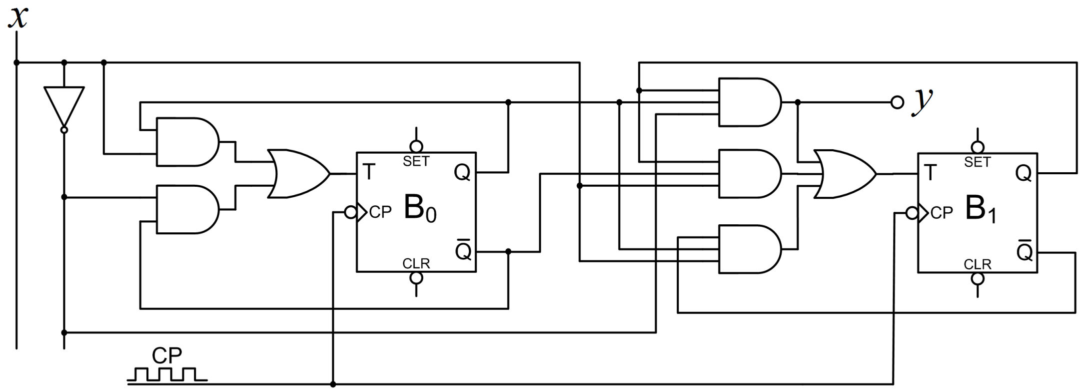
Minimalni oblik logičke funkcije $T_1(x,Q_1,Q_0)$ je prema K-tablici na slici 6.77b:

$$T_1 = \bar{x}Q_1Q_0 + xQ_1\overline{Q_0} + x\overline{Q_1}Q_0.$$

Izlaz y detektora niza 0100 u tablici 6.25 ima vrijednost 1 za minterm m_3 pa vrijedi:

$$y = \bar{x}Q_1Q_0.$$

Na temelju izvedenih logičkih funkcija za T ulaze bistabila te logičke funkcije izlaza y dobivena je shema T bistabila. Shema je prikazana na slici 6.78.



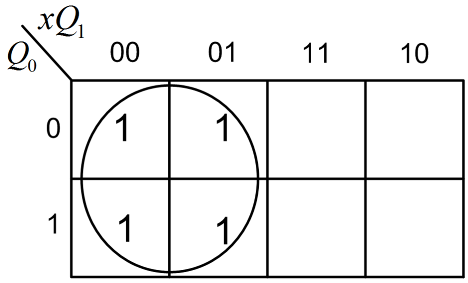
Slika 6.78: Shema detektora niza 0100 izvedenog pomoću T bistabila

c) Tablica uzbude D bistabila za detektor niza 0100 prikazana je tablicom 6.26. U kreiranju stupaca uzbude bistabila B_0 i B_1 korištena je tablica uzbude D bistabila (tablica 5.8).

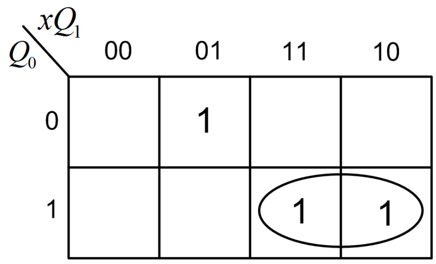
Tablica 6.26: Tablica uzbude D bistabila za detektor niza 0100

	Ulaz	Sadašnje stanje bistabila, n		Sljedeće stanje bistabila, $n+1$		Uzbuda bistabila		Izlaz
i	x	Q_1	Q_0	Q_1	Q_0	D_1	D_0	y
0	0	0	0	0	1	0	1	0
1	0	0	1	0	1	0	1	0
2	0	1	0	1	1	1	1	0
3	0	1	1	0	1	0	1	1
4	1	0	0	0	0	0	0	0
5	1	0	1	1	0	1	0	0
6	1	1	0	0	0	0	0	0
7	1	1	1	1	0	1	0	0

Stupci uzbude bistabila u tablici 6.26 logičke su funkcije varijabli x, Q_1 i Q_0. Te logičke funkcije potrebno je minimizirati kako bi se mogle ostvariti minimalnim brojem osnovnih logičkih sklopova i spojiti na ulaze D bistabila. Radi lakšeg popunjavanja K-tablice u tablici 6.26 prvi stupac je stupac indeksa minterma i. K-tablice za ulaze D bistabila B_0 i B_1 detektora niza 0100 prikazane su slici 6.79.



(a) K-tablica za $D_0(x,Q_1,Q_0)$



(b) K-tablica za $D_1(x,Q_1,Q_0)$

Slika 6.79: K-tablice za ulaze D bistabila B_0 i B_1 detektora niza 0100

Minimalni oblik logičke funkcije $D_0(x,Q_1,Q_0)$ je prema K-tablici na slici 6.79a:

$$D_0 = \bar{x}.$$

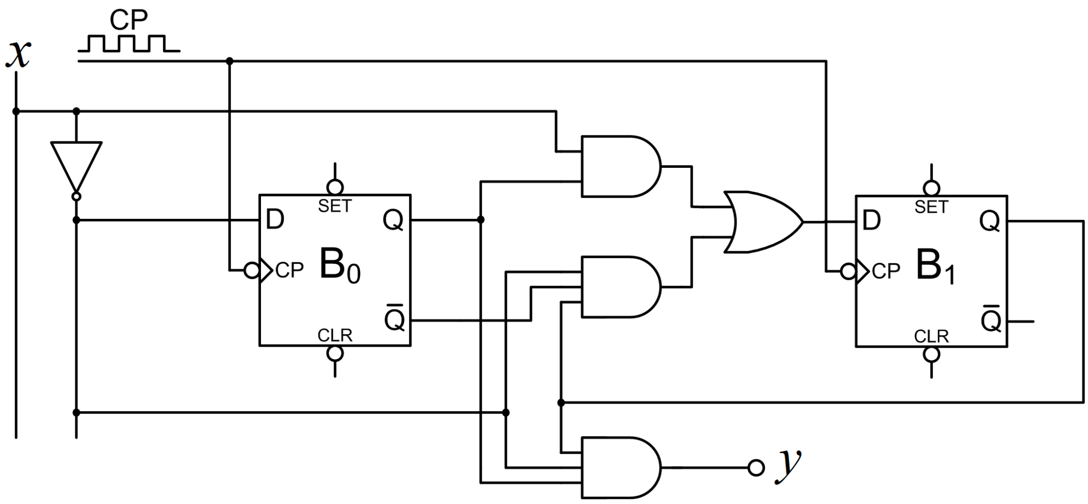
Minimalni oblik logičke funkcije $D_1(x,Q_1,Q_0)$ je prema K-tablici na slici 6.79b:

$$D_1 = xQ_0 + \bar{x}Q_1\overline{Q_0}.$$

Izlaz y detektora niza 0100 u tablici 6.26 ima vrijednost 1 za minterm m_3 pa vrijedi:

$$y = \bar{x}Q_1Q_0.$$

Na temelju izvedenih logičkih funkcija za D ulaze bistabila te logičke funkcije izlaza y dobivena je shema D bistabila. Shema je prikazana na slici 6.80.



Slika 6.80: Shema detektora niza 0100 izvedenog pomoću D bistabila

6.7.1 Zadaci za vježbu

✎ Zadatak 6.7.1

Konstruirajte digitalni automat za detektor niza 0100:
a) bez preklapanja,
b) s preklapanjem.

✎ Zadatak 6.7.2

Konstruirajte digitalni automat za detektor niza s preklapanjem i bez preklapanja za sljedeće nizove:
a) 0111,
b) 0110,
c) 010010,
d) 01000101.

✎ Zadatak 6.7.3

Konstruirajte digitalni automat za detektor niza 10110111, a zatim definirajte i ispišite znakovnu tablicu prijelaza stanja za taj detektor.

✎ Zadatak 6.7.4

Projektirajte sinkroni detektor niza 1011 s preklapanjem primjenom:
a) minimalnog broja JK bistabila i osnovnih logičkih sklopova,
b) minimalnog broja T bistabila i osnovnih logičkih sklopova,
c) minimalnog broja D bistabila i osnovnih logičkih sklopova.
Detektor niza mora na izlazu y generirati vrijednost 1 svaki puta kada detektira niz 1011. Definirajte i ispišite znakovnu tablicu prijelaza stanja za zadani detektor.

Bibliografija

[1] Šumiga, I., *Digitalna logika i elektronika*, Veleučilište u Varaždinu, Varaždin 2013.

[2] Peruško, U.; Glavinić, V., *Digitalni sustavi*, Školska knjiga, Zagreb, 2005.

[3] Čupić, M., *Digitalna elektronika i digitalna logika - zbirka riješenih zadataka*, Kigen, Zagreb, 2006.

Visoka Tehnička Škola u Bjelovaru

www.vtsbj.hr

ISBN 978-953-7676-20-9

9 789537 676209