
Standardne kombinacione mreže

Logičke funkcije o kojima je do sada bilo reč dele jedno zajedničko svojstvo, a to je da stanja na izlazima zavise isključivo od trenutnih stanja ulaznih signala, bez obzira na to kakva su ta stanja bila u prošlosti. Drugim rečima, mreže kojima se ostvaruju ovakve funkcije nemaju mogućnost pamćenja stanja. Ovakve mreže nazivaju se kombinacionim mrežama. Pored aritmetičkih kola, u ovu klasu mreža spadaju još neka važna i u praksi često korišćena kola čija funkcionalnost će ovde biti opisana.

Komparatori

Komparatori su standardne kombinacione mreže koje vrši se poređenje dva binarna broja, a izlazi komparatora daju informaciju o tome da li su brojevi jednaki ili nisu i o tome koji je odnos među njima (tj. koji broj je veći). Pri tome je broj bita, odnosno širina binarnih brojeva koje komparator može da poredi određen unapred (npr. 4-bitni komparator).

Prilikom poređenja dva n-bitna binarna broja $A = a_{n-1}a_{n-2}..a_1a_0$ i $B = b_{n-1}b_{n-2}..b_1b_0$, poređenje kreće od para najviših bita, a_{n-1} i b_{n-1} . Ukoliko je $a_{n-1} = 1$, a $b_{n-1} = 0$, broj A je veći od broja B, bez obzira na vrednosti njihovih nižih bita $a_{n-2}..a_0$ i $b_{n-2}..b_0$. Slično razmatranje važi i za obrnut slučaj, kada je $a_{n-1} = 0$, a $b_{n-1} = 1$. Međutim, ukoliko je $a_{n-1} = b_{n-1}$, odnos između brojeva zavisi od vrednosti bita nižeg značaja. Prema tome, u ovom slučaju poređenje brojeva A i B se prenosi na poređenje sledećeg para bita po značaju (bita a_{n-2} i b_{n-2}), itd.

Dakle, u cilju dizajniranja n-bitnog, prvo će biti realizovan jednobitni komparator koji omogućava poređenje jednog para bita, a n-bitni komparator će biti dobijen kaskadiraanjem n jednobitnih komparatora. Logika rada jednobitnog komparatora je sledeća. Izlazi $Out < i Out >$ daju informaciju o tome da li je $A < B$, odnosno $B > A$. Pri odlučivanju, prvo se gledaju stanja na ulazima $In < i In >$, koja su diktirana od stranje prethodnog komparatora, koji je izvršio poređenje para bita većeg značaja (A_{n+1} i B_{n+1}). U zavisnosti od stanja na ovim ulazima, moguća su 4 slučaja:

1. $(In <) = 0, (In >) = 0$ označava da je $A = B$ na osnovu poređenja viših bita.
2. $(In <) = 0, (In >) = 1$ označava da je $A > B$ na osnovu poređenja viših bita.
3. $(In <) = 1, (In >) = 0$ označava da je $A < B$ na osnovu poređenja viših bita.
4. $(In <) = 1, (In >) = 1$ je nedozvoljena kombinacija ulaza.

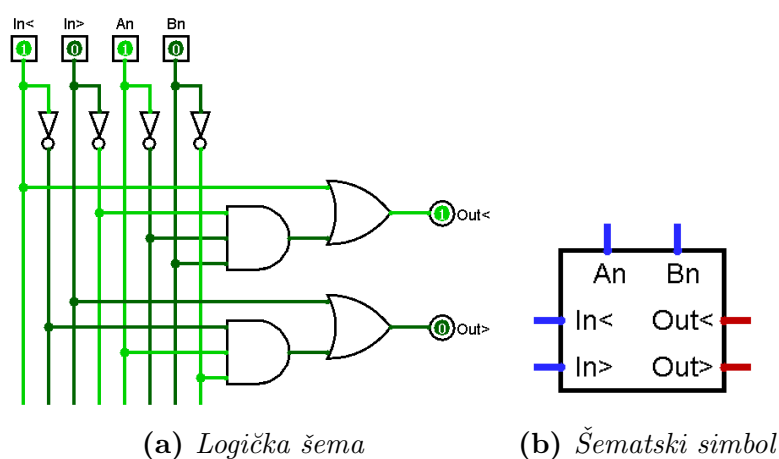
U slučajevima 2 i 3, odluka o odnosu između brojeva A i B je već donesena na višem stupnju logike (nekom od prethodnih komparatora), pa se prenosi na izlaze bez obzira na stanja na ulazima A_n i B_n . U slučaju 1, odluka se donosi na osnovu stanja na ulazima A_n i B_n . Tabela 1 je funkcionalna tabela ovako realizovanog komparatora.

| $In <$ | $In >$ | A_n | B_n | $Out <$ | $Out >$ |
|--------|--------|-------|-------|---------|---------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | X | X |
| 1 | 1 | 0 | 1 | X | X |
| 1 | 1 | 1 | 0 | X | X |
| 1 | 1 | 1 | 1 | X | X |

Tabela 1: Funkcionalna tabela jednobitnog komparatora

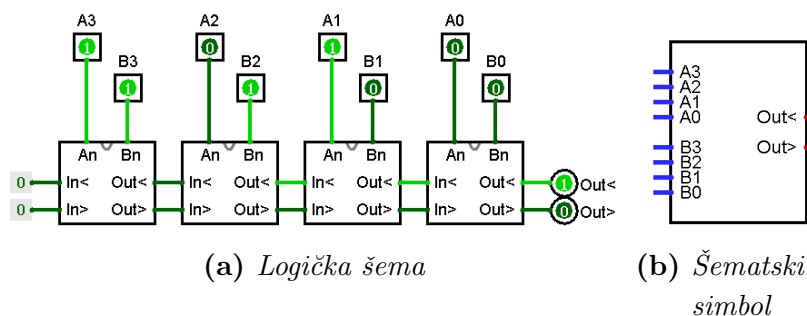
Minimizacijom funkcija $Out <$ i $Out >$ dobija se:

$$\begin{aligned} (Out <) &= (In <) + \overline{In >} \cdot \overline{A_n} \cdot B_n \\ (Out >) &= (In >) + \overline{In <} \cdot A_n \cdot \overline{B_n} \end{aligned}$$



Slika 1: Jednobitni komparator

Mreža koja ostvaruje logiku jednobitnog komparatora prikazana je na slici 1. Kaskadnim povezivanjem 4 jednobitna komparatora, dobija se 4-bitni komparator, prikazan na slici 2.

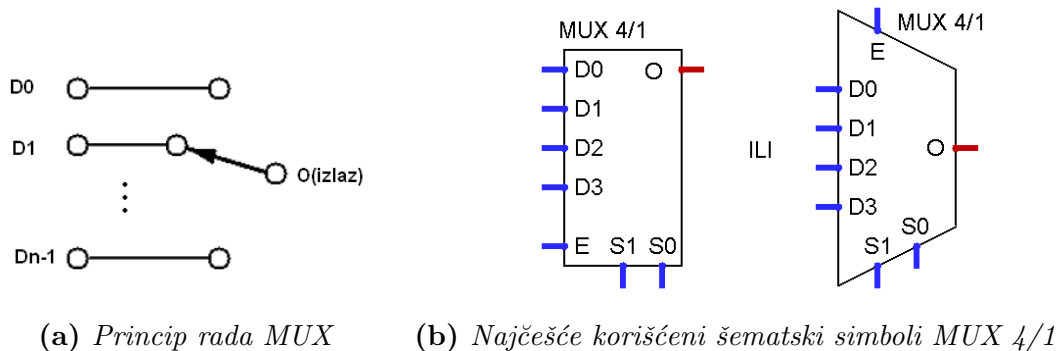


Slika 2: 4-bitni komparator

Zadatak Koristeći integrisanu komponentu 4-bitnog komparatora čiji je simbol prikazan na slici 2b i dodatna logička kola po potrebi, realizovati 4-bitni komparator sa izlazima $Out \leq$, $Out \geq$ i $Out =$, koji daju informaciju o odnosima između ulaza "A je manje ili jednako B", "A je veće ili jednako B" i "A je jednako B".

Zadatak Koristeći potreban broj integrisanih 4-bitnih komparatora i dodatnih logičkih kola, realizovati komparator koji na svom izlazu daje logičku jedinicu ukoliko je ulazni četvorobitni broj unutar opsega $3 \leq A \leq 7$.

Multiplekseri i demultiplekseri



Slika 3: Multiplekser

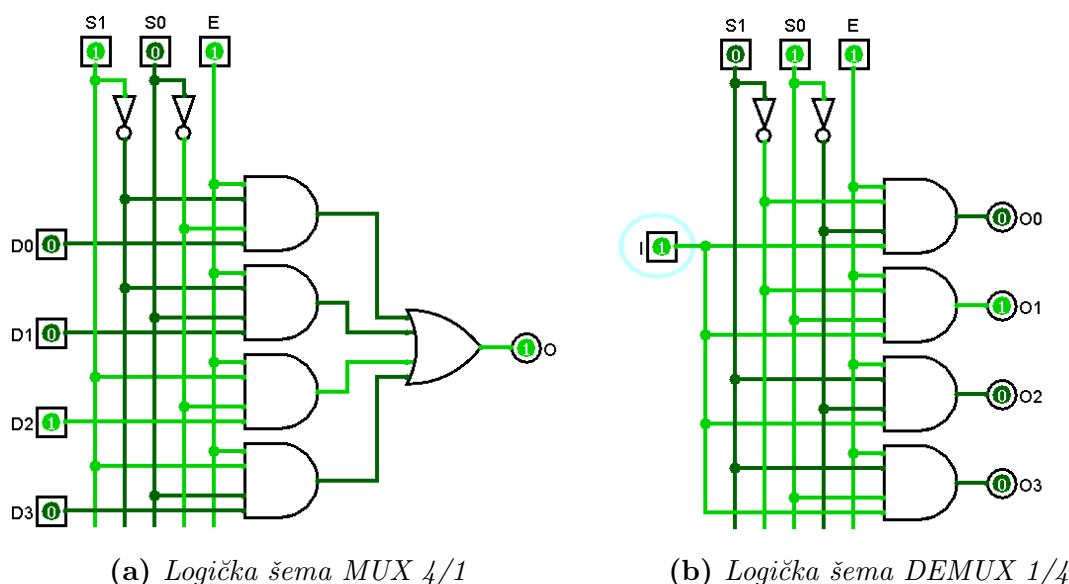
Multiplekser je kombinaciona mreža koja ostvaruje funkciju višepoložajnog prekidača, kao što je prikazano na slici 3a: u svakom trenutku se stanje sa tačno jednog od ulaza prosleđuje na izlaz. Ulazni signali multipleksera su:

- Informacioni ulazi $D_{n-1}, D_{n-2}, \dots, D_1, D_0$, pri čemu je broj ulaza $n = 2^m$
- Selekcioni ulazi S_{m-1}, \dots, S_1, S_0
- Signal dozvole E

Multiplexer se označava oznakom MUX $n/1$, gde n označava broj informacionih ulaza. Taj broj je po pravilu stepen dvojke, tj. $n = 2^m$, gde je m broj selekcionih ulaza. Drugim rečima, broj informacionih ulaza je uvek onoliko koliko postoji mogućih kombinacija na selekcionim ulazima. Logika rada multiplexera je takva da se u svakom trenutku na izlaz prosleđuje signal sa onog informacionog ulaza koji je određen trenutnom kombinacijom na selekcionim ulazima (kombinacija na selekcionim ulazima se interpretira kao binarni broj koji odgovara indeksu odgovarajućeg informacionog ulaza), pod uslovom da je aktivan signal dozvole ($E = 1$). Ako je signal dozvole jednak nuli, na izlaz se prosleđuje nula bez obzira na stanja na selekcionim i informacionim ulazima, pa se kaže da je u tom slučaju izlaz blokiran. U opštem slučaju, stanje na izlazu multiplexera se opisuje sledećom funkcijom:

$$O = E \cdot (\overline{S_{n-1}} \cdot \overline{S_{n-2}} \cdot \dots \cdot \overline{S_1} \cdot \overline{S_0} \cdot D_0 + \overline{S_{n-1}} \cdot \overline{S_{n-2}} \cdot \dots \cdot \overline{S_1} \cdot S_0 \cdot D_1 + \dots + S_{n-1} \cdot S_{n-2} \cdot \dots \cdot S_1 \cdot \overline{S_0} \cdot D_{n-2} + S_{n-1} \cdot S_{n-2} \cdot \dots \cdot S_1 \cdot S_0 \cdot D_{n-1})$$

Realizacija MUX $4/1$ pomoću logičkih kola prikazana je na slici 4a. U ovom slučaju, pošto postoje 2 selekciona i 4 informaciona ulaza, prenosna funkcija poprima oblik $O = E \cdot (\overline{S_1} \cdot \overline{S_0} \cdot D_0 + \overline{S_1} \cdot S_0 \cdot D_1 + S_1 \cdot \overline{S_0} \cdot D_2 + S_1 \cdot S_0 \cdot D_3)$.



Slika 4: Primer multiplexera i demultiplexera

Mreža koja obavlja suprotnu funkciju od multiplexera naziva se demultiplexer, a obeležava se sa DEMUX. Demultiplexer ima 1 ulaz i n izlaza, a stanje sa ulaza se prenosi na onaj izlaz koji je određen kombinacijom na selekcionim ulazima, dok su ostali izlazi neaktivni i na njima se pojavljuje 0. Na slici 4b data je šema DEMUX $1/4$, čije ponašanje je opisano izlaznim funkcijama:

$$O_0 = E \cdot \overline{S_1} \cdot \overline{S_0} \cdot I$$

$$O_1 = E \cdot \overline{S_1} \cdot S_0 \cdot I$$

$$O_2 = E \cdot S_1 \cdot \overline{S_0} \cdot I$$

$$O_3 = E \cdot S_1 \cdot S_0 \cdot I$$

Zadatak Realizovati MUX 16/1 koristeći potreban broj integrisanih multipleksera MUX 4/1.

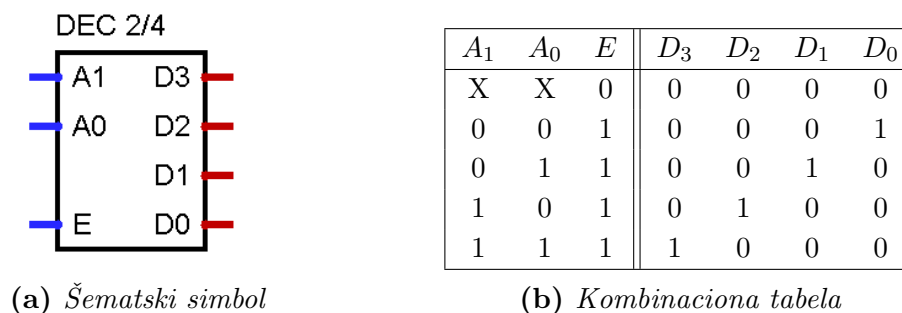
Multiplekseri se često koriste u okviru programabilnih logičkih komponenti, pošto je njihovim korišćenjem moguće realizovati proizvoljnu logičku funkciju. U idealnom slučaju, ako je na raspolaganju multiplekser koji ima onoliko selekcionih ulaza koliko ima ulaznih promenljivih, ulazne promenljive se povezuju upravo na selekzione ulaze, dok se na informacione ulaze dovode vrednosti funkcije (0 ili 1), za svaku kombinaciju ulaznih promenljivih. U slučaju da je broj promenljivih veći od broja selekcionih ulaza, promenljive koje su "višak" (one koje nije bilo moguće dovesti na selekzione ulaze), kombinuju se sa informacionim ulazima, uz korišćenje dodatnih logičkih kola, po potrebi.

Zadatak Funkcija 4 promenljive $F(A, B, C, D) = \sum(3, 4, 6, 7, 9, 11, 12, 13)$ je zadata skupom indeksa. Realizovati funkciju F korišćenjem:

- Jednog MUX 16/1.
- Jednog MUX 8/1 i dodatnih logičkih kola.

Dekoderi

Dekoder je kombinaciona mreža sa m selekcionih (adresnih)¹ ulaza i $n = 2^m$ izlaza. Logika rada dekodera je takva da je u svakom trenutku tačno jedan izlaz aktivan i to onaj koji je određen kombinacijom na adresnim ulazima, dok su svi ostali izlazi neaktivni. Izlaz se smatra aktivnim ako se na njemu pojavljuje logička jedinica, odnosno neaktivnim ako je na njemu logička nula. Da bi dekodeo obavljao svoju funkciju, signal dozvole mora biti postavljen na jedinicu, u suprotnom će svi izlazi biti blokirani (odnosno postavljeni na nulu).



Slika 5: Dekoder DEC 2/4

Na slici 5 prikazani su šematski simbol i kombinaciona tabela dekodera DEC 2/4. Izlazi takvog dekodera opisani su logičkim funkcijama:

¹Kod dekodera selekциони ulazi se često nazivaju i *adresnim* ulazima. Ovakav naziv dobija pun smisao ako se ima u vidu jedna od najčešćih primena dekodera, pri adresiranju lokacija u memorijskim sistemima. O tome će biti više reči kada bude proučavana struktura RAM i ROM memorije.

$$D_0 = E \cdot \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = E \cdot \overline{A_1} \cdot A_0$$

$$D_2 = E \cdot A_1 \cdot \overline{A_0}$$

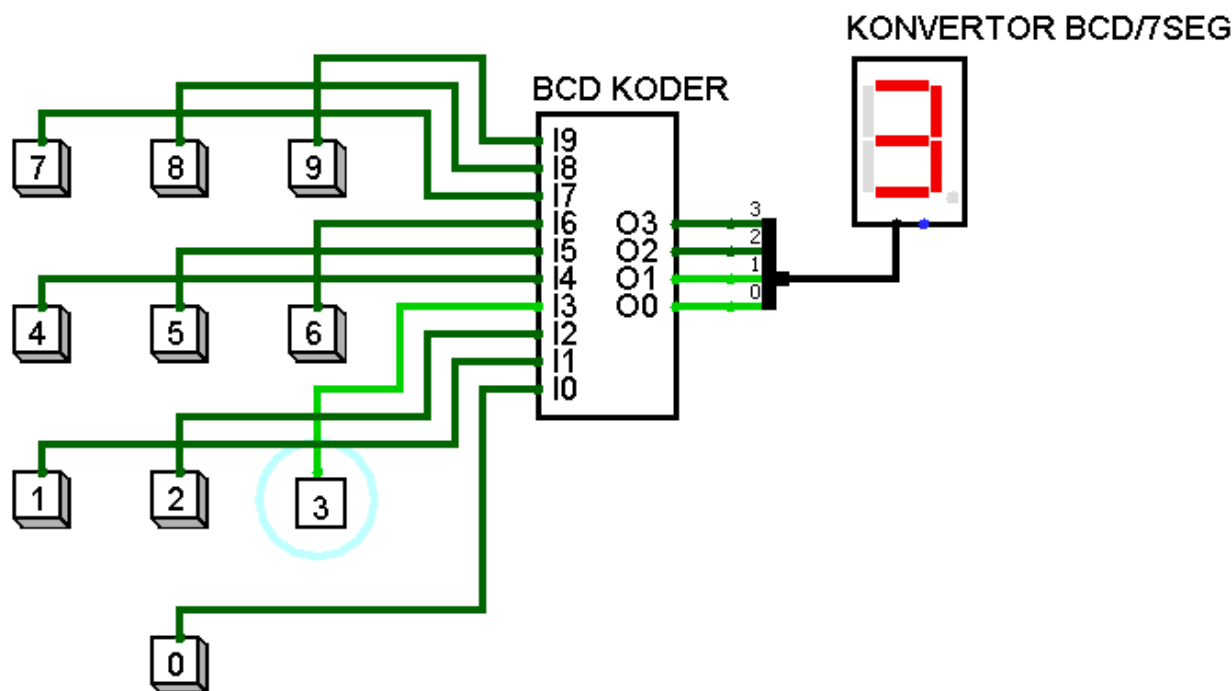
$$D_3 = E \cdot A_1 \cdot A_0$$

Zadatak Realizovati komponentu DEC 3/8 koristeći isključivo NILI logička kola. Simulirati rad mreže u Logisim-u.

Zadatak Realizovati dekodler DEC 2/4 koristeći komponentu DEMUX 2/4.

Koderi

Koder je kombinaciona mreža koja obavlja suprotnu operaciju od dekodera: na izlazima generiše kod koji odgovara onom ulazu koji je u datom trenutku aktivan. Ako koder ima 2^n ulaza i n izlaza, kaže se da je potpun. Pored potpunih postoje i nepotpuni koderi, kod kojih je broj ulaza manji od 2^n . Tipičan primer takvog kodera je BCD² koder. Primena BCD kodera je prikazana na slici 6. Tastatura sa ukupno 10 tastera je povezana na ulaze BCD kodera. U primeru sa slike pritisnut je taster br.3, čime je aktiviran ulaz I_3 . Na izlazima se generiše kod $0011_2 = 3_{10}$, koji se dalje vodi na mrežu koja upravlja 7-segmentnim LED displejom, koji prikazuje cifru koja odgovara generisanom kodu. Konstrukcija ovakve mreže prikazana je u sledećem odeljku (*Konvertori koda*).



Slika 6: Primer primene BCD kodera

²BCD (engl. Binary Coded Decimal) je kod koji se koristi za predstavljanje decimalnih cifara, u kojem je svaka decimalna cifra predstavljena sa tačno 4 bita.

Problem vezan za upotrebu kodera je u pretpostavci da je u svakom trenutku aktivan samo jedan od ulaza. Ako to nije slučaj (npr. u primeru sa slike 6 su pritisnuta 2 tastera istovremeno), postavlja se pitanje kako kodovati takvu situaciju. Ovakva situacija se rešava upotrebom prioriternog kodera, kod kojeg je svakom od ulaza unapred dodeljen jedinstvan nivo prioriteta. Dakle, onda kada je više ulaza istovremeno aktivno, na izlazu će se generisati kod koji odgovara ulazu koji je najviši po prioritetu. Ovakav koder obično ima još jedan izlaz koji se obeležava sa DV (engl. Data Valid). Svrha ovog izlaza je signalizacija da je kod na izlazima validan, odnosno da jedinica postoji na bar jednom od ulaza. Tabela 2 je funkcionalna tabela prioriternog kodera sa 4 ulaza. MSB (ulaz I_3) predstavlja ulaz najvišeg, a LSB (ulaz I_0) najnižeg prioriteta.

| I_3 | I_2 | I_1 | I_0 | O_1 | O_0 | DV |
|-------|-------|-------|-------|-------|-------|----|
| 0 | 0 | 0 | 0 | X | X | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | X | 0 | 1 | 1 |
| 0 | 1 | X | X | 1 | 0 | 1 |
| 1 | X | X | X | 1 | 1 | 1 |

Tabela 2: Funkcionalna tabela prioriternog kodera sa 4 ulaza

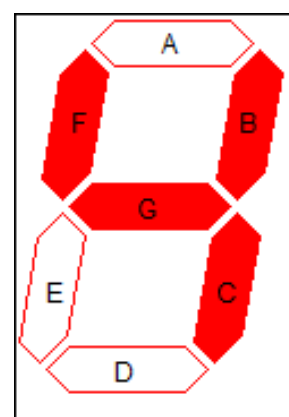
Zadatak Realizovati prioriterni koder sa 4 ulaza pomoću logičkih kola i izvršiti simulaciju u Logisim-u.

Konvertori koda

Konvertori koda su kombinacione mreže sa proizvoljnim brojem ulaza i izlaza, koje vrše konverziju iz jednog binarnog koda u drugi. Tipičan primer je mreža iz prethodnog primera sa 4 ulaza i 7 izlaza, koja BCD kod na ulazu konvertuje u kod koji aktivira odgovarajuće segmente na sedmosegmentnom LED displeju (slika 7b).

| Cifra | D_3 | D_2 | D_1 | D_0 | A | B | C | D | E | F | G |
|-------|-------|-------|-------|-------|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

(a) Kombinaiona tabela



(b) 7SEG displej

Slika 7: Konvertor koda BCD/7SEG

Primer prikazan na slici 7 odgovara situaciji kada se na ulaze dovodi kombinacija $0100_2 = 4_{10}$. Tada se aktiviraju izlazi B, C, F i G, čime se postiže ispis decimalne cifre 4 na displeju.

Projektovanje ovakvog konvertora koda podrazumeva realizaciju 7 izlaznih funkcija (A..G), od kojih svaka ima 4 ulazne promenljive ($D_3..D_0$). Ovde će biti prikazan način realizacije na primeru izlazne funkcije za upravljanje segmentom A. Za početak, potrebno je uneti vrednosti funkcije A iz kombinacione u Karnoovu tabelu, a zatim minimizovati funkciju³:

| | | D_1D_0 | | | |
|----------|----|----------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| D_3D_2 | 00 | 1 | 0 | 1 | 1 |
| | 01 | 0 | 1 | 1 | 1 |
| | 11 | X | X | X | X |
| | 10 | 1 | 1 | X | X |

$$A = \overline{D_2} \cdot \overline{D_0} + D_2 \cdot D_0 + D_1 + D_3$$

Polja u Karnoovoj tabeli sa indeksima od 10 do 15 predstavljaju zabranjena stanja. Ova stanja ne spadaju u BCD kod, pa važi pretpostavka da se neće pojavljivati na ulazu. Zabranjena stanja se obeležavaju simbolom X u Karnoovoj tabeli i pri minimizaciji se koriste kao "džokeri": po potrebi može im se dodeliti vrednost 1 ili 0, u cilju prekrivanja jedinica (odnosno nula) u tabeli što je moguće većim maskama. Pri tome nije neophodno prekriti maskama sva polja koja sadrže simbol X, nego se takva polja koriste samo u slučajevima kada je to od pomoći.

Zadatak Realizovati konvertor koda BCD/7SEG pomoću logičkih kola i izvršiti simulaciju u Logisim-u.

³U ovom primeru, funkcija je realizovana u DNF (standardnom dvostepenom I-ILI logikom), u cilju demonstracije. Na čitaocu je da proveriti da li je ekonomičnija ovakva realizacija, ili realizacija na bazi KNF (standardnom dvostepenom I-ILI logikom)