

**DIGITALNA TEHNIKA**

**skripta za III razred**

**zanimanje: Tehničar računarstva**

**prof. Biljana Vidaković dipl.el.ing**

SADRŽAJ

[SADRŽAJ 2](#_Toc417465851)

[1. Vrste informacija 5](#_Toc417465852)

[1.1.Pojam informacija i njihovo kodovanje 5](#_Toc417465853)

[1.2.Vrste informacija 5](#_Toc417465854)

[1.3.Digitalna kola i mreže 6](#_Toc417465855)

[2. Brojni sistemi 9](#_Toc417465856)

[2.1.Decimalni brojni sistem 9](#_Toc417465857)

[2.2 Binarni brojni sistem 10](#_Toc417465858)

[2.3 Oktalni brojni sistem 11](#_Toc417465859)

[2.4. Heksadecimalni brojni sistem 12](#_Toc417465860)

[ZADACI ZA VJEŽBU 15](#_Toc417465861)

[3. Konverzija brojeva iz jednog u drugi brojni sistem 16](#_Toc417465862)

[3.1. Konverzija decimalnih brojeva u binarne i obrnuto 16](#_Toc417465863)

[3.2. Konverzija oktalnih brojeva u decimalne i obrnuto 17](#_Toc417465864)

[3.3. Konverzija heksadecimalnih brojeva u decimalne i obrnuto 19](#_Toc417465865)

[3.4. Konverzija binarnih brojeva u oktalne i obrnuto 20](#_Toc417465866)

[3.4. Konverzija binarnih brojeva u heksadecimalne i obrnuto 21](#_Toc417465867)

[3.6. Osnovne aritmetičke operacije u binarnom brojnom sistemu 22](#_Toc417465868)

[3.6.1.Sabiranje u binarnom brojnom sistemu 22](#_Toc417465869)

[3.6.2.Oduzimanje u binarnom brojnom sistemu 23](#_Toc417465870)

[3.6.3.Množenje u binarnom brojnom sistemu 24](#_Toc417465871)

[3.6.4.Dijeljenje u binarnom brojnom sistemu 25](#_Toc417465872)

[ZADACI ZA VJEŽBU 26](#_Toc417465873)

[PITANJA I ZADACI 28](#_Toc417465874)

[4. Kodiranje i kodovi 29](#_Toc417465875)

[4.1. Kôd BCD 8421 30](#_Toc417465876)

[4.2. Kôd BCD „više 3“ 31](#_Toc417465877)

[4.3. Grejov kôd 31](#_Toc417465878)

[4.4. Specijalni kodovi 32](#_Toc417465879)

[4.5. Kontrola kodovanja 33](#_Toc417465880)

[4.6. Kodovi za otkrivanje i kontrolu greške 34](#_Toc417465881)

[4.7. Kod sa kontrolom parnosti 35](#_Toc417465882)

[4.8. Primjeri kodiranja 36](#_Toc417465883)

[5. Prekidačka (Bulova) algebra 37](#_Toc417465884)

[5.1. Zakoni,teoreme i pravila prekidačke algebre 38](#_Toc417465885)

[5.2. Tabelarno predstavljanje prekidačkih funkcija 39](#_Toc417465886)

[5.3. Analitičko predstavljanje prekidačkih funkcija 41](#_Toc417465887)

[5.4. Primjeri-vježbanje 43](#_Toc417465888)

[5.5. Minimizacija prekidačkih funkcija 44](#_Toc417465889)

[5.5.1. Veič-Karnoova metoda minimizacije 44](#_Toc417465890)

[5.8. Primjeri-vježbanje 48](#_Toc417465891)

[6. Logička kola 50](#_Toc417465892)

[6.1. Logičke operacije i logička kola 50](#_Toc417465893)

[6.4. Specijalne logičke operacije i kola 55](#_Toc417465894)

[VJEŽBA: 55](#_Toc417465895)

[7. Kombinacione mreže 57](#_Toc417465896)

[7.1. Sinteza prekidačkih funkcija 57](#_Toc417465897)

[7.2. Koderi 57](#_Toc417465898)

[7.3. Dekoderi 58](#_Toc417465899)

[7.4. Konvertori koda 8421/+3 59](#_Toc417465900)

[7.5. Konvertori koda: Grej u binarni 59](#_Toc417465901)

[7.6. Multiplekseri (selektori) 59](#_Toc417465902)

[7.7. Demultiplekseri (distributori) 60](#_Toc417465903)

[8. Aritmetička kola 61](#_Toc417465904)

[8.1. Predstavljanje brojeva u digitalnim mrežama 61](#_Toc417465905)

[8.2.Binarni komparatori 61](#_Toc417465906)

[8.3. Binarni sabirači 61](#_Toc417465907)

[8.3.1. Binarni polusabirač 61](#_Toc417465908)

[Sl. 8.2. Binarni polusabirač 61](#_Toc417465909)

[8.3.2. Binarni potpuni sabirač 62](#_Toc417465910)

[8.4. Množači 63](#_Toc417465911)

1. Vrste informacija

1.1.Pojam informacija i njihovo kodovanje

Riječ informacija u svakodnevnom životu uglavnom znači isto što i obavještenje. Pri procjeni količine informacija,nije bitan broj riječi od kojih se obavještenje sastoji, već sadržaj obavještenja, odnosno koliko novosti ono nosi.

Svako obavještenje o događaju koji se sa sigurnošću očekuje, tj. čija je vjerovatnoća ostvarivanja vrlo velika, sadrži malo informacija. I obrnuto,saopštenje o neočekivanom događaju, čija je vjerovatnoća ostvarivanja vrlo mala, sadrži znatno više informacija. Pojam informacija u elektronici je prvo korišten za prenos vijesti telekomunikacionim sistemima. Danas taj pojam ima šire značenje i odnosi se na obradu informacija, tj. podataka.

Predstavljanje informacija pomoću simbola koji su elementi nekog unaprijed određenog skupa,tkz. kodne azbuke, naziva se ***kodovanje informacija***. Binarno kodovane informacije su sastavljene od dva različita simbola 1 i 0 tj. kažemo da su kodovane u binarnom brojnom sistemu.

Uvedena je jedinica za mjerenje količine informacija 1 BIT ( binary unit).

Dakle, jedan bit je jedinica mjere za količinu informacije koja može imati samo jednu od dvije moguće vrijednosti – 1 ili 0. U žargonu digitalne tehnike pod terminom bit se udomaćio naziv za jednu cifru binarnog broja 1 ili 0, odnosno za impuls kojim je ta cifra predstavljena, bez obzira na vjerovatnoću sa kojom se pojedina cifra, tj. impuls pojavljuje.

1.2.Vrste informacija

Sve vrste podataka, naredbi, slova i znakova zapisuju se i obrađuju u digitalnim uređajima samo u obliku binarnih brojeva. To je „prirodni“ jezik računara. Najlakše je tehnički realizovati elektronske elemente sa dva različita stabilna stanja. Tako su bistabilni elementi postali najvažnije komponente u savremenim digitalnim računarima, a binarni brojni sistem najvažniji brojni sistem u svijetu elektronike.

Kod digitalnih uređaja razlikujemo tri vrste informacija:

* + - naredbe (komande)
    - podaci i
    - adrese.

***Podaci*** su informacije koje je računar primio od vanjskog svijeta, ili ono što je dobijeno kao rezultat neke obrade u samom računaru. Prije uvođenja u računar podaci se prevode u binarni oblik da bi ih računar mogao obrađivati. Nakon obrade dobijaju se neke nove informacije.

***Instrukcije*** nalažu računaru šta treba da radi sa podacima. Niz instrukcija koji čini logičku cjelinu predstavlja program po kome računar radi.

***Adrese*** nam govore na kom mjestu se nalaze određni podaci i gdje ih smještamo nakon završene obrade.

1.3.Digitalna kola i mreže

Prema funkciji koju obavljaju, svi elementi u digitalnim uređajima, mogu se svrstati u tri grupe:

* + - elementi koji generišu i uobličavaju električne impulse
    - logički elementi ili elementi kombinacione tehnike
    - memorijski elementi.

***Elementi koji generišu i uobličavaju impulse*** su standardna kola impulsne tehnike, kao što su razne vrste multivibratora, komparatori, kola za diferenciranje i integriranje i druga.

***Logički elementi*** ili logička kola služe za realizovanje logičkih stanja, odnosno njihovih kombinacija bez mogućnosti pamćenja. Stanje na izlazu logičkog kola se zadržava samo dotle dok postoje signali koji su ga izazvali. Električne mreže realizovane sa ovim elementima se zovu kombinacione logičko-prekidačke mreže. Opšti oblik je prikazan na slici 1.1.

***Memorijski elementi*** služe za pamćenje prethodnih stanja tj. za čuvanje informacija. Električne mreže realizovane sa ovakvim elementima zovu se sekvencijalne logičko-prekidačke mreže. Na slici 1.2. prikazana je uopštena strukturna šema takve mreže. U posmatranom trenutku na ulaz mreže dolaze vanjski signali koji odgovaraju tom trenutku i koji se mogu nazvati sadašnje ulazne promjenljive. Iz mreže se uzimaju sadašnji izlazni signali, dok se iz memorijskog dijela interpretiraju sadašnja stabilna stanja, a iz kombinacionog u memorijski dio ulaze signali koji će promijeniti postojeća-sadašnja stabilna stanja memorijskih elemenata i prouzrokovati nova koja će se koristiti u narednim vremenskim intervalima.

ULAZI IZLAZI

KOMBINACIONI DIO MREŽE

X1 Y1

KOMBINACIONA LOGIČKO – PREKIDAČKA MREŽA

X2 Y2

Xn Yn

MEMORIJSKI DIO MREŽE

Sl.1.1. Kombinaciona logičko-prekidačka mreža Sl.1.2. Sekvencijalna logičko-prekidačka mreža

Impulsi u jednom digitalnom uređaju mogu biti:

* + - ***takt – impulsi***, koji se dobijaju iz posebnog takt generatora i daju osnovni ritam rada cijelom uređaju.
    - ***informacioni impulsi***, pomoću kojih se izražavaju informacije u vidu binarno kodovanih podataka, kao što su brojevi, slova i simboli.
    - ***kontrolni impulsi***, koji upravljaju redoslijedom izvršavanja pojedinih operacija u uređaju.

Analiza logičko – prekidačke mreže podrazumijeva da se za zadate ulazne komande i podatke, odrede vrijednosti izlaznih funkcija koje mreža realizuje. Pri sintezi logičko – prekidačke mreže treba prvo definisati blok – šemu uređaja, zadatke pojedinih blokova, kapacitete memorija pojedinih blokova i izvršiti izbor logičkih i memorijskih elemenata za realizaciju uređaja.

Zadatak koji digitalni uređaj treba da izvrši mora biti izražen u obliku redoslijedno- sekvencijalnog niza aritmetičkih operacija koji se zove ***algoritam****.*

Redoslijedni niz elementarnih operacija, instrukcija ili naredbi koje uređaj treba da izvrši da bi tehnički realizovao neki algoritam naziva se ***program***.

Riječ impuls je latinskog porekla i označava udarac, podsticaj, a u elektrotehnici kratak udar energije, odnosno naglu promjenu amplitude napona ili struje. Cjelokupni sistem digitalne obrade podataka zasniva se na primjeni impulsa. Njima se, prije svega, predstavljaju brojne vrijednosti podataka u nekom pogodnom brojnom sistemu. Sve instrukcije u vezi sa obradom, kontrolom i sinhronizacijom procesa pri obradi podataka izvode se pomoću impulsa.

U analizi električnih kola i prenosnih sistema najčešće se koriste teorijski idealizovani oblici impulsa. Impuls na slici 1.3.a ima trenutnu promjenu amplitude i poznat je kao Hevisajdova funkcija, odskočna funkcija ili step impuls. Ako se amlituda step-impulsa označi sa 1, ovakav impuls se naziva i jedinični.

Kada se odskočnoj funkciji poslije vremena Tp doda još jedna takva funkcija sa negativnim predznakom, dobija se idealan pravougaoni impuls kao na slici 1.3.b. Specijalan oblik idealnog pravougaonog impulsa prikazan je na slici 1.3.c. To je potpuno teorijski impuls, koji bi postojao samo u jednom trenutku, i to sa beskonačno velikom amplitudom. Ovakav impuls se zove Dirakova funkcija ili delta impuls.

Ako se pojedinačni pravougaoni impulsi periodično ponavljaju, onda nastaje povorka impulsa.

V V V

1 1 ∞

t t t

a) b) c)

Sl.1.3. Idealizovani oblici tipičnih impulsa

V

t

TP T0

T

Sl. 1.4. Povorka pravougaonih impulsa

Tp - trajanje impulsa

T0 - pauza između impulsa

T - period ponavljanja impulsa

U realnim uslovima idealni oblik pravougaonog impulsa praktično je neostvarljiv.

2. Brojni sistemi

Ljudi su svoje ruke i prste od davnina koristili kao pomoćno sredstvo za brojanje. Ipak, takav brojni sistem nije najefikasniji niti najpogodniji za korištenje u računarskim mašinama.

Informacije se u digitalnoj tehnici predstavljaju pomoću brojeva, pa se postavlja pitanje koji je brojni sistem najpogodniji za to. Brojni sistemi su sistemi simbola za označavanje skupova.

Uopšte za osnovu brojnog sistema može se uzeti bilo koji broj veći od 1. Tako, pored decimalnog brojnog sistema sa osnovom 10, koji predstavlja „prirodni“ brojni sistem za čovjeka, postoje i oktalni brojni sistem sa osnovom 8, heksadecimalni sa osnovom 16 i drugi.

U digitalnoj tehnici je najpogodniji za primjenu binarni brojni sistem sa osnovom 2 koji predstavlja „prirodni“ jezik računara. Najpogodniji je zbog jednostavnosti tehničke realizacije i pouzdanosti, jer je dovoljno razlikovati samo dva stanja koja se predstavljaju naponskim ili strujnim nivoom.

Može se reći i to da ni jedan brojni sistem nema apsolutnu prednost nad ostalim sistemima. Zato su i razvijeni različiti brojni sistemi i svaki od njih se koristi u određenim uslovima, kada je u izvjesnoj prednosti od drugih sistema.

2.1.Decimalni brojni sistem

Decimalni brojni sistem je sistem koji se danas najviše koristi u matematici. Svi proračuni koje razumije čovjek polaze od ovog sistema. Takođe i operacije nad opštimbrojevima se takođe obavljaju ovim sistemom.

Decimalni brojni sistem ima osnovu 10***.*** To znači da ovaj sistem sadrži 10 simbola (0,1,2,3,4,5,6,7,8,9), koje još nazivamo *arapskim ciframa.* Ovaj sistem koristi *pozicionu notaciju* (za razliku od npr. rimskih brojeva čija se vrijednost određuje sabiranjem cifara)

**primjer:** 12+14=26

XII+XIV=XXVI

**n-**cifreni cijeli brojevi u decimalnom brojnom sistemu sa pozicionom notacijom se zapisuju na sljedeći način:

**(1)**

Prva cifra zdesna ima vrijednost ili „težinu“ jedinice a0.100, gdje je a0 decimalna cifra koja označava broj jedinica. Druga cifra ima „težinu“ desetica ili a1.101, treća težinu stotica ili a2.102 itd.

**primjer:** Rastaviti broj 3408902 prema formuli za zapisivanje decimalnih brojeva kao što je to prikazano u formuli **(1)**



Koeficijenti a0, a1, a2,....,an uzimaju vrijednosti iz skupa 0,1,2,3,...,9.

Na isti način za razlomljeni decimalni broj važi:

**primjer:** 19,74 = 1 . 10 1 + 9 . 10 2 + 7 . 10 -1 + 4 . 10 -2 = 10 + 9 + 7/10 + 4/100

U opštem slučaju za neki broj sa *n* cijelih i *m* razlomljenih mjesta važi:

**(2)**

gdje ai označava brojnu vrijednost cifre na nekoj poziciji, a bi predstavlja osnovu ili bazu brojnog sistema. Stepenovanjem broja b u svakom sabirku dobija se poziciona vrijednost, odnosno „težina“ cifre u tom sabirku.

To znači da decimalni brojni sistem spada u tzv. pozicione kodove, pošto se svakoj cifri pripisuje težina čija je vrijednost određena mjestom te cifre u odnosu na pozicioni zarez.

2.2 Binarni brojni sistem

Poput dekadnog brojnog sistema koji se udomaćio u svakodnevnoj upotrebi, tako je u računarskom svijetu uobičajen binarni sistem. Binarni sistem se bazira na istim principima kao i dekadni brojni sistem. Spada u težinske pozicione brojne sisteme. On ima osnovu 2 i samo dva cifarska simbola 0 i 1.

Za binarni brojni sistem važi:

**(3)**

Svaki član u ovom nizu ima težinu dvostruku od prethodnog člana. Koeficijenti a0, a1, a2,....,an imaju vrijednost 0 ili 1 i predstavljaju cifre binarnog broja.

**primjer :** Rastaviti broj 10011101 prema formuli za zapisivanje binarnih brojeva kao što je to prikazano u formuli **(3)**

**7 6 5 4 3 2 1 0**

10011101=  

**primjer :** Rastaviti broj 11011100 prema formuli za zapisivanje binarnih brojeva kao što je to prikazano u formuli **(3)**

**7 6 5 4 3 2 1 0**

11011100=  

**primjer :** Rastaviti broj 11101 prema formuli za zapisivanje binarnih brojeva kao što je to prikazano u formuli **(3)**

**4 3 2 1 0**

11101=  

**primjer :** Rastaviti broj 110110 prema formuli za zapisivanje binarnih brojeva kao što je to prikazano u formuli **(3)**

**5 4 3 2 1 0**

110110= 



Brojevi koji nisu cijeli mogu se izraziti u istom sistemu na sledeći način:

**primjer:** (1,8125)10 = 1 . 20 + 1 . 2-1 + 1 . 2-2 + 0 . 2-3 + 1 . 2-4 = 1,1101

2.3 Oktalni brojni sistem



Pored binarnog i decimalnog brojnog sistema postoje i drugi brojni sistemi sa različitim osnovama. Međutim za računarsku tehniku posebno su interesantna još dva brojna sistema i to:

1. Oktalni brojni sistem
2. Heksadecimalni brojni sistem

Oktalni brojni sistem ima bazu 8, i simbole (0,1,2,3,4,5,6,7) i koristi pozicionu notaciju. Oktalni brojni sistem se bazira na istim principima kao i dekadni brojni sistem.

**n-**cifreni cijeli brojevi u oktalnom brojnom sistemu sa pozicionom notacijom se zapisuju na sljedeći način:

**(4)**

**primjer :** Rastaviti broj 26735 prema formuli za zapisivanje oktalnih brojeva kao što je to prikazano u formuli **(4)**

**4 3 2 1 0**

26735= 

**primjer :** Rastaviti broj 253 prema formuli za zapisivanje oktalnih brojeva kao što je to prikazano u formuli **(4)**

**2 1 0**

253= 

**primjer :** Rastaviti broj 26735 prema formuli za zapisivanje oktalnih brojeva kao što je to prikazano u formuli **(4)**

**4 3 2 1 0**

34105= 

**primjer :** Rastaviti broj 3116 prema formuli za zapisivanje oktalnih brojeva kao što je to prikazano u formuli **(4)**

**3 2 1 0**

3116= 

Oktalni brojevi manji od nule se vrlo rijetko upotrebljavaju, pa zbog toga nisu ni prikazani.

2.4. Heksadecimalni brojni sistem

Princip predstavljanja brojeva u heksadecimalnom brojnom sistemu isti je kao i kod prethodno opisanih. Osnova ili baza heksadecimalnog brojnog sistema je broj 16. Koristi se 16 cifara: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Jasno je da se velikim slovima abecede pripisuju sledeće vrijednosti:

A=10, B=11, C=12, D=13, E=14, F=15.

Ova slova se uvode da bi se izbjegle dvocifarske oznake koje otežavaju pisanje. Heksadecimalni brojni sistem se bazira na istim principima kao dekadni, binarni i oktalni brojni sistem.

**n-**cifreni cijeli brojevi uhexadecimalnom brojnom sistemu sa pozicionom notacijom se zapisuju na sljedeći način:

**(5)**

**primjer :** Rastaviti broj CAFE prema formuli za zapisivanje heksadecimalnih brojeva kao što je to prikazano u formuli **(5)**

**3 2 1 0**

CAFE= 

**primjer :** Rastaviti broj FACA prema formuli za zapisivanje hexadecimalnih brojeva kao što je to prikazano u formuli **(5)**

**3 2 1 0**

FACA= 

**primjer :** Rastaviti broj 3A7 prema formuli za zapisivanje hexadecimalnih brojeva kao što je to prikazano u formuli **(5)**

**2 1 0**

3A7= 

**primjer :** Rastaviti broj ABA prema formuli za zapisivanje hexadecimalnih brojeva kao što je to prikazano u formuli **(5)**

**2 1 0**

ABA= 

Kao i oktalni, heksadecimalni brojni sistem je samo pomoć programeru da lakše upamti i napiše program, odnosno prihvati ga iz računara. Postavlja se pitanje zbog čega je bilo potrebno uvesti heksadecimalni brojni sistem kada on treba da obavlja potpuno istu funkciju kao i oktalni.

Razlog je pogodnost primjene heksadecimalnog sistema u radu sa binarnim brojevima. Jedan heksadecimalni broj se može prikazati sa četiri bita, dva broja se mogu prikazati sa osam bita itd. Mikroprocesori rade sa binarnim brojevima upravo takvog formata, pa se heksadecimalni brojni sistem najviše upotrebljava u radu sa mikroprocesorima. U tabeli 1 prikazani su ekvivalenti decimalnih brojeva do 16 u binarnom, oktalno i heksadecimalnom brojnom sistemu.

|  |  |  |  |
| --- | --- | --- | --- |
| DECIMALNI | BINARNI | OKTALNI | HEKSADECIMALNI |
| 0 | 0000 | 0 | 0 |
| 1 | 0001 | 1 | 1 |
| 2 | 0010 | 2 | 2 |
| 3 | 0011 | 3 | 3 |
| 4 | 0100 | 4 | 4 |
| 5 | 0101 | 5 | 5 |
| 6 | 0110 | 6 | 6 |
| 7 | 0111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |

*Tabela 2.1.*

ZADACI ZA VJEŽBU

1. Rastaviti decimalne brojeve:
   1. 23
   2. 26534
   3. 54987
   4. 6654
   5. 54
2. Rastaviti binarne brojeve i ustanoviti o kojem broju dekadnog sistema se radi:
   1. 10
   2. 1101
   3. 1000110
   4. 110001
   5. 101100101
3. Rastaviti oktalne brojeve i ustanoviti o kojem broju dekadnog sistema se radi:
   1. 371
   2. 32
   3. 2765
   4. 333
   5. 1123
4. Rastaviti hexadecimalne brojeve i ustanoviti o kojem broju dekadnog sistema se radi:
   1. ACE
   2. 2A5
   3. 12
   4. 11101
   5. 22FA

Rješenja:

2. a) 2, b) 13, c) 70, d) 49, e) 357

3. a) 249, b) 26, c) 1525, d) 219, e) 595

4. a) 2766, b) 677, c) 18, d) 69889, e) 8954

3. Konverzija brojeva iz jednog u drugi brojni sistem

Binarno predstavljanje brojeva je teško zamislivo kao dio ljudskog izražavanja, ali je zato prirodni jezik digitalnih računara,čiji svi elementi imaju samo dva stanja. U praksi se koriste drugi brojni sistemi. Stoga se često postavlja problem prevođenja brojeva iz jednog brojnog sistema u drugi.

Kao što smo vidjeli iz predhodnih lekcija, svaki brojni sistem definisan je osnovom (radixom,bazom), što implicira da je on i relevantan parametar kod prelaska sa jednog brojnog sistema na drugi, što se naziva ***konverzija brojnog sistema.*** Kao što formule **(1), (2), (3), (4) i (5)** iz predhodnih lekcija pokazuju da je broj predstavljen zbirom proizvoda cifre (digita) i njoj odgovarajućeg exponenta, to je za prelazak na drugi brojni sistem dovoljno izvršiti dijeljenje sa željenom osnovom.

3.1. Konverzija decimalnih brojeva u binarne i obrnuto

Prevođenje cijelih decimalnih brojeva u binarne vrši se metodom uzastopnog dijeljenja. Metoda se sastoji u tome što se broj koji se konvertuje dijeli sa bazom sistema, a ostatak dijeljenja posebno zapisuje. Konvertovani broj se dobija kada se napišu cifre ostatka obrnutim redom od onog kako su dobijene. Na primjer, za decimalni broj 642 postupak prevođenja u binarni sistem je sledeći:

**primjer:**

642 : 2 = 321 (0)

321 : 2 = 160 (1)

160 : 2 = 80 (0)

80 : 2 = 40 (0)

40 : 2 = 20 (0)

20 : 2 = 10 (0)

10 : 2 = 5 (0)

5 : 2 = 2 (1)

2 : 2 = 1 (0)

1 : 2 = 0 (1) (642)10=(1010000010)2

Decimalni broj manji od jedinice konvertuje se u binarni tako što se množi bazom 2. Cjelobrojni dio rezultata ulazi u binarni broj, a razlomljeni se dalje množi bazom i postupak se nastavlja dok razlomljeni dio ne postane nula.

**primjer:**

**cjelobrojni dio**

0,375 . 2 = 0,75 (0)

0,75 . 2 = 1,5 (1)

0,5 . 2 = 1,0 (1) (0,375)10=(0,011)2

Konverzija mješovitih decimalnih brojeva vrši se tako što se posebno konvertuje cjelobrojni dio, a posebno dio sa razlomljenim vrijednostima, pa se dobijeni rezultati sabiraju. Treba naglasiti da se cijeli brojevi mogu izraziti precizno u binarnoj formi, dok se konverzija brojeva koji nisu cijeli ne može uvijek izvesti do kraja, što podrazumijeva primjenu aproksimacija. To ne znači da je binarni sistem manje tačan od decimalnog, već da on samo zahtijeva veći broj cifara za izražavanje određene veličine željenom preciznošću.

Za obrnut postupak konverzije binarnih brojeva u decimalne može se koristiti direktno sumiranje članova prema formuli **(3)**:

**primjer:**

(10011101,01)2 =



**primjer:**

8 7 6 5 4 3 2 1 0

(100110101)2 =  256 + 32 + 16 + 4 + 1 = (309)10

3.2. Konverzija oktalnih brojeva u decimalne i obrnuto

Oktalni broj se pretvara u decimalni slično kao i binarni, što je ilustrovano sledećim primjerom:

**primjer:**

(1267) 8 = 1 . 83 + 2 . 82 + 6 . 81 + 7 . 80 = 512 + 128 + 48 = (695)10

Dakle, (1267) 8 = (695) 10

Da se ova množenja ne bi uvijek iznova radila, može se napraviti pomoćna tabela sa prikazanim proizvodima težinskih vrijednosti odgovarajućih pozicija cifre i oktalnih cifara.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POZICIJA CIFRE  5 4 3 2 1 0 | | | | | | OKTALNA CIFRA |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32768 | 4096 | 512 | 64 | 8 | 1 | 1 |
| 65536 | 8192 | 1024 | 128 | 16 | 2 | 2 |
| 98304 | 12288 | 1536 | 192 | 24 | 3 | 3 |
| 131072 | 16384 | 2048 | 256 | 32 | 4 | 4 |
| 163840 | 20480 | 2560 | 320 | 40 | 5 | 5 |
| 196608 | 24576 | 3072 | 384 | 48 | 6 | 6 |
| 229376 | 28672 | 3584 | 448 | 56 | 7 | 7 |

*Tabela 3.1.*

Obrnuti postupak se može izvesti matodom uzastopnog dijeljenja sa brojem 8 slično kao kod prevođenja iz decimalnog u binarni broj.

**primjer:**

356 : 8 = 44 (4)

44 : 8 = 5 (4)

5 : 8 = 0 (5) (356)10 = (544)8

Jednostavnija metoda je metoda oduzimanja stepena 8, koji je najbliži zadatom decimalnom broju. Za to može poslužiti tabela 2. U tabeli treba potražiti najbliži broj,ali manji od decimalnog. To je broj 512 koji se nalazi u koloni 3, što znači da će oktalni broj imati 4 cifre. Broj 512 se nalazi u vrsti 1, a to znači da je 1 odgovarajuća oktalna cifra na četvrtoj poziciji,odnosno 1 itd.

**primjer:** ( 1 2 0 3 )8

(643)10

- 512 = 1 . 83

131

- 128 = 2 . 82

3

- 0 = 0 . 81

3

- 3 = 3 . 80

0

3.3. Konverzija heksadecimalnih brojeva u decimalne i obrnuto

Kao i kod oktalnih brojeva, i ovde se može napraviti tabela sa proizvodima odgovarajućih vrijednosti heksadecimalnih cifara.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| POZICIJA CIFRE  3 2 1 0 | | | | HEKSADECIMALNA CIFRA |
| 0 | 0 | 0 | 0 | 0 |
| 4096 | 256 | 16 | 1 | 1 |
| 8192 | 512 | 32 | 2 | 2 |
| 12288 | 768 | 48 | 3 | 3 |
| 16384 | 1024 | 64 | 4 | 4 |
| 20480 | 1280 | 80 | 5 | 5 |
| 24576 | 1536 | 96 | 6 | 6 |
| 28672 | 1792 | 112 | 7 | 7 |
| 32768 | 2048 | 128 | 8 | 8 |
| 36864 | 2304 | 144 | 9 | 9 |
| 40960 | 2560 | 160 | 10 | A |
| 45056 | 2816 | 176 | 11 | B |
| 49152 | 3072 | 192 | 12 | C |
| 53248 | 3328 | 208 | 13 | D |
| 57344 | 3584 | 224 | 14 | E |
| 61440 | 3840 | 240 | 15 | F |

*Tabela 3.2.*

**primjer:**

( 1 E 9 B )16

nulta cifra = B, iz tabele: 11

prva cifra = 9, iz tabele: 144

druga cifra = E, iz tabele: 3584

treća cifra = 1, iz tabele: + 4096

(7835)10

I u slučaju konverzije decimalnog broja u heksadecimalni može se primijeniti tabela 3, kao i za konverziju u oktalni sistem. Međutim,ovom prilikom se metoda uzastopnog dijeljenja može prikazati na primjeru pretvaranja decimalnog broja 47653 u odgovarajući heksadecimalni.

**primjer:**

CJELOBROJNI DIO OSTATAK

(47653)10 : 16 = 2978 5

2978 : 16 = 186 2

186 : 16 = 11 10

11 : 16 = 0 11

= (B A 2 5)16

3.4. Konverzija binarnih brojeva u oktalne i obrnuto

Pošto je 23 = 8, to znači da za jedan jednocifren oktalni broj treba tri bita. Prema tome, binarni brojevi se mogu podijeliti u grupe po tri bita,počevši od pozicionog zareza. Svakoj takvoj grupi može se pripisati jedan oktalni broj.

**primjer:**

(1000100100011110)2 => ( ? )8

**(**1'000'100'100'011'110)2 = (104436)8

1 0 4 4 3 6

I oktalni broj se vrlo jednostavno pretvara u binarni.

**primjer:**

**(**572)8 => ( ? )2

|  |  |  |  |
| --- | --- | --- | --- |
|  | **4 (22)** | **2 (21)** | **1 (20)** |
| **5** | 1 | 0 | 1 |
| **7** | 1 | 1 | 1 |
| **2** | 0 | 1 | 0 |

( 101 111 010)2

Jasno je, na osnovu ovih primjera, da se broj cifara u oktalnom broju smanjuje tri puta u odnosu na isti taj broj u binarnom sistemu. Takođe je jasno da se sa malo vježbe binarni broj može napamet pretvoriti u oktalni i obrnuto. To je velika pomoć programerima kada pišu program i prihvataju podatke iz računara, jer umjesto dugog niza nula i jedinica koji se teško pamti mogu bilježiti podatke s mnogo manje cifara.

3.4. Konverzija binarnih brojeva u heksadecimalne i obrnuto

Pošto je osnova heksadecimalnog sistema 16 = 24, to se svaki binarni broj koji treba pretvoriti u heksadecimalni dijeli u grupe po četiri bita i svakoj grupi posebno dodjeljuje odgovarajući haksadecimalni ekvivalent.

**primjer:**

(1001101000011111)2 = 1001 1010 0001 1111

( 9 A 1 F )16

Dakle, dugi niz od 16 binarnih cifara prikazuje se sa četiri heksadecimalne cifre, pa se takav broj lakše pamti i piše.

**primjer:**

(11'1101'0010'0110'1010'0011)2 = 0011 1101 0010 0110 1010 0011 = (3D26A3)16

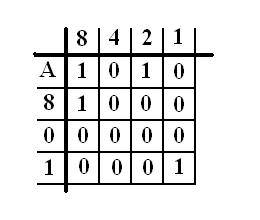
Za povratak u binarni sistem, svakoj heksadecimalnoj cifri pripisuje se binarni ekvivalent sa četiri cifre.

**primjer:**

( E6A2 )16 = E 6 A 2 = (1110011010100010)2

1110 0110 1010 0010

**primjer: (**A801)16 => ( )2



(1010 1000 0000 0001)2

3.6. Osnovne aritmetičke operacije u binarnom brojnom sistemu

Da bi računar mogao da izvršava proračune određenih funkcija potrebno je definisati osnovne matamatičke operacije. Te osnovne operacije su sabiranje, oduzimanje, množenje i dijeljenje.

3.6.1.Sabiranje u binarnom brojnom sistemu

Binarni sistem koristi pozicionu notaciju prilikom zapisivanja brojeva, što znači da se sabiranje binarnih brojeva obavlja slično kao i sabiranje decimalnih brojeva, tako što se sabiraju biti sa istim težinskim vrijednostima.

Ukoliko eventualno dođe do dvocifrenog zbira, isto kao kod decimalnih brojeva, zapisuje se cifra manje težinske vrijednosti, a cifra veće težinske vrijednosti predstavlja prenos i prenosi se u lijevu kolonu.

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | Suma | Prenos |
| 0 | 0 | 0 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 | 1 |

*Tabela 3.3.*

**primjer:**

185 010111001

+ 141 010001101

326 1 1 1 1 1 *prenos*

101000110

**primjer:**

11101,01 29,25

+ 1001,101 = 9,625

11 1 *prenos*

100110,111 38,875

3.6.2.Oduzimanje u binarnom brojnom sistemu

Oduzimanje binarnih brojeva se takođe vrši po analogiji sa decimalnim brojnim sistemom. Tabela oduzimanja ima sledeći izgled:

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | Razlika | Pozajmljeno |
| 0 | 0 | 0 |  |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |

*Tabela 3.4.*

**primjer:**

01100 12

- 00111 = - 7

1 1 1 *prenos*

00101 5

Idući od najnižeg prema višim razredima, izvedene su sledeće operacije:

- pošto se u prvom razredu ne može oduzeti 1 od 0 i pošto je i u drugom razredu umanjenik takođe 0, onda se pozajmljuje cifra 1 iz trećeg razreda, koja ima težinsku vrijednost 22 = 21 + 21. Jedna cifra 1 sa težinom 21 se zadržava u drugom razredu, a preostalih 21 =10 se prenosi u prvi razred. Najzad, binarna razlika u prvom razredu 10 – 1 daje vrijednost 1;

- sada u drugom razredu već postoji pozajmljena cifra 1 sa odgovarajućom težinom, pa će biti 1 – 1 = 0;

- pošto je iz trećeg razreda umanjenika već pozajmljena cifra 1, sada se mora pozajmiti 1 iz četvrtog razreda, pa će biti 10 – 1 = 1;

- najzad, u četvrtom razredu je 0 – 0 = 0.

Razlomljeni brojevi se oduzimaju na isti način:

1010,1

111,1

011,0

Primjer:

16 10000 11000 24

- 3 - 11 -10101 - 21

13 01101 00011 3

3.6.3.Množenje u binarnom brojnom sistemu

Pošto binarni sistem ima osnovu 2, množenje sa dva se izvodi pomijeranjem zareza udesno, dok se pri dijeljenju sa dva zarez pomjeri za jedno mjesto ulijevo. Ako se posmatra binarni broj 1001,00 = 9 i ako se zarez pomjeri za jedno mjesto udesno,biće:

10010,0 = 1 . 24 + 0 . 23 + 0 . 22 + 1 . 21 + 0 . 20 = 16 + 0 + 0 + 2 + 0 = 18, što znači da je zaista izvršeno množenje sa 2. U istom primjeru pomijeranje zareza za jedno mjesto ulijevo daje:

100,1 = 1 . 22 + 0 . 21 + 0 . 20 + 1 . 2-1 = 4 + 0 + 0 + 1/2 = 4,5.

Ovim je dokazano da je izvršeno dijeljenje sa 2.

Množenje binarnih brojeva se takođe obavlja na isti način kao i množenje decimalnih brojeva, gdje se svaki bit jednog broja množi sa svakim bitom drugog broja koji sudjeluje u množenju. Tablica koja se koristi za množenje binarnih brojeva ima sljedeći izgled.

|  |  |  |
| --- | --- | --- |
| a | b | Proizvod |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

*Tabela 3.5.*

Zatim se vrši sabiranje umnožaka na sledeći način

10011 x 111 7 x 19=13310

10011

10011

+ 10011

1 11 11 1 *prenos*

(10000101)2 Proizvod

**primjer:** 101,1 . 10,01

1011

0000

0000

1011

1 1 *prenos*

1100,011

3.6.4.Dijeljenje u binarnom brojnom sistemu

Binarno dijeljenje je slično decimalnom, s tim što se i međuoperacije množenja i oduzimanja izvode, naravno, u binarnom sistemu,uz pravila:

0 : 1 = 0

1 : 1 = 1, Dijeljenje nulom je , naravno, besmisleno.

(1010,1: 111,1)2 (10,5 : 7,5)10

Da bismo se oslobodili zareza, i dijeljenik i djelilac moraju se proširiti sa 2 (binarno 10), pa će biti:

10101 : 1111 = 1,011 (21)10 : (15)10 = (1,4)10

- 1111

0011000

- 1111

010010

- 1111

0011

Iz primjera se vidi da je rezultat dijeljenja u binarnom brojnom sistemu periodičan broj, dok u decimalnom brojnom sistemu to nije slučaj. To proističe iz činjenice da se decimalni broj 0,4 ne može prevesti u binarni broj u konačnom obliku.

Dijeljenje binarnih brojeva takođe je analogno onom u decimalnom sistemu

**primjer:**

101101 : 101 = 1001 45 : 5 = 9

-101

000101

- 101

000

ZADACI ZA VJEŽBU

1. Izvršiti sabiranje brojeva:
   1. 1101101011101

1111001001101

* 1. 1010101010101

1111111111111

1. Izvršiti oduzimanje brojeva:
   1. 1010001010

101011101

* 1. 1101000101

110011011

1. Izvršiti množenje brojeva:
   1. 110010101101101 x 11
   2. 10111010101010 x 101
2. Izvršiti dijeljenje brojeva:
   1. 110010101101101 : 1010
   2. 10111010101010 : 1111

|  |  |
| --- | --- |
| TABLICA ZA SABIRANJE  A + B | |
| 0 + 0 | 0 |
| 0 + 1 | 1 |
| 1 + 0 | 1 |
| 1 + 1 | 0 prenos 1 |

|  |  |
| --- | --- |
| TABLICA ZA ODUZIMANJE  A - B | |
| 0 - 0 | 0 |
| 0 - 1 | 1 pozajmljeno 1 |
| 1 - 0 | 1 |
| 1 - 1 | 0 |

|  |  |
| --- | --- |
| TABLICA ZA MNOŽENJE  A . B | |
| 0 . 0 | 0 |
| 0 . 1 | 0 |
| 1 . 0 | 0 |
| 1 . 1 | 1 |

|  |  |
| --- | --- |
| TABLICA ZA SABIRANJE  A : B | |
| 0 : 0 | ? |
| 0 : 1 | 1 |
| 1 : 0 | ? |
| 1 : 1 | 1 |

PITANJA I ZADACI

1. Konvertovati razlomljeni decimalni broj 72,125 u binarni
2. Izvršiti oduzimanje sledećih binarnih brojeva: 100010 i 1010
3. Ako se paran decimalni broj konvertuje u binarni brojni sistem, kakva mora biti cifra najnižeg razreda?
4. Koliko je cifara potrebno da se decimalni broj 18 izrazi u binarnom brojnom sistemu?
5. Koju vrijednost ima cifra najveće težine (razreda) u binarnim brojevima od pet cifara koji odgovaraju decimalnim brojevima između 16 i 31?
6. Oktalni broj 37501 konvertovati u decimalni.
7. Heksadecimalni broj 9D1B konvertovati u decimalni.
8. Decimalni broj 256 konvertovati u binarni.
9. Decimalni broj 64,375 konvertovati u binarni.
10. Decimalni broj 336 konvertovati u heksadecimalni.

4. Kodiranje i kodovi

Znakovi kojima se ljudi služe u međusobnoj komunikaciji, pa i u komunikaciji s računarom, dakle oni koji se nalaze na tastaturi računara moraju poprimiti binarni oblik prije obrade u računaru. Elektronski elementi od kojih je građen računar mogu poprimiti samo dva različita stanja od kojih se jedno označava sa **1**, a drugo sa **0**. To je veza s binarnim brojnim sistemom, ali se zapisi podataka u računaru ne tumače uvijek kao binarni brojevi, već kao binarni zapis.

***Kodiranje*** je način predstavljanja informacija pomoću simbola koji su elementi nekog unaprijed određenog skupa. Taj skup simbola je ***kodna azbuka***. Grupa simbola kojom se predstavlja-koduje neki pojam naziva se *kodna riječ*, a broj simbola u kodnoj riječi je ***dužina kodne******riječi***. Ako su sve riječi jednog koda iste dužine, onda se radi o tzv. ***ravnomjernom kodu***.

Ukoliko neki kod sadrži simbole koji ne označavaju nikakvu,ili nikakvu novu informaciju, onda se za njega kaže da je ***redundantan***. Primjer za ovo je označavanje glasova sa dva znaka (lj,nj,dž).



Slika 1. Procesi kodiranja i dekodiranja podataka

Binarni brojni kodovi se mogu podijeliti na težinske i redoslijedne. Kod težinskih svakom binarnom simbolu se pripisuje određena težina npr. binarni brojni sistem. Kodne kombinacije se mogu praviti na različite načine, prema drugim težinskim zakonitostima. Razvijeno jemnoštvo binarnih kodova za specijalne namjene. Pri konstrukciji težinskih kodova mora se voditi računa da težinska vrijednost nekog mjesta ne bude za više od jedan veća od zbira težinskih vrijednosti svih prethodnih binarnih mjesta. U protivnom se svi brojevi neće moći predstaviti u tom kodu.

Svi kodovi koji nisu težinski spadaju u grupu redoslijednih kodova. Binarne cifre nemaju odgovarajuću težinu, već je bitan samo njihov redoslijed u kodnoj riječi.

4.1. Kôd BCD 8421

Da bi se iskoristile dobre osobine binarnog i decimalnog sistema razvijeni su ***binarno kodovani decimalni*** ili **BCD** kodovi.

Posebno se koduje svaka cifra decimalnog broja tako da je prikazujemo sa četiri bita. To se obavlja prema nekom utvrđenom zakonu. Ovi kodovi spadaju u redundantne kodove i moguće je napraviti mnogo različitih binarno kodovanih decimalnih sistema. Ukupno je moguće napraviti približno 2,9 . 109 četverobitnih BCD kodova.

Kôd BCD 8421, koji se još zove i prirodni BCD kôd (NBCD), uzima prvih deset kombinacija od 4 bita za kodovanje decimalnih simbola. Ovo se vrši po težinskom principu 8421, kao i kod prirodnih binarnih brojeva. Njegova je prednost da može koristiti normalnu binarnu tehniku pa se često koristi u digitalnim uređajima.

Konverzija decimalnog sistema u kod 8421 je veoma jednostavna. Za svaku decimalnu cifru nađe se četverocifreni binarni ekvivalent.

**primjer:**

Konverzija decimalnog broja 107:

1 0 7

0001 0000 0111

U prirodnom binarnom sistemu broj 107 se predstavlja kao 1101011.

Za predstavljanje decimalnog broja ovaj kôd očito zahtijeva više cifarskih mjesta nego prirodni binarni kod.

Nedostatak BCD 8421 kôda je što se na njega ne mogu uvijek primijeniti pravila binarnog sabiranja. Ako je zbir decimalnih cifara u jednoj koloni iste pozicione vrijednosti veći od 9, pa se pomenuta kolona decimalnih brojeva konvertuje u BCD 8421 kôd i sabere, onda će se u rezultatu pojaviti grupa od 4 bita koja ne pripada kôdu.

npr. 4 7

0100 0111

+

2 4

0010 0100

0110 1011

Rezultat bi trebao biti: 7 1

0111 0001

Dakle, pojavljuje se grupa 1011 koja ne postoji u ovom kôdu.

4.2. Kôd BCD „više 3“

Problemi iz prethodnog hoda vezani za binarno sabiranje riješeni su kôdom „više 3“.

Kôd „više 3“ se dobija tako što se svakoj cifri decimalnog broja prvo doda vrijednost 3, pa se tako dobijeni broj koduje u BCD 8421 sistemu. Ovaj kôd nema težinski karakter kao raniji kodovi.

**primjer:**

Kodirati broj 643 u „više 3“ kôdu:

643 ( 6+3;4+3;3+3) = (976) **1001 0111 0110**

4.3. Grejov kôd

Grejov kôd ima osobinu da se njegove susjedne kodne grupe razlikuju samo za jedan bit. On nema težinski karakter, tj. njegove cifre nemaju određene pozicione vrijednosti. Pretvaranje prirodnog binarnog sistema u Grejov kôd vrši se postupnim ispisivanjem cifara, idući slijeva na desno. Bit najviše pozicione vrijednosti se prepisuje i sledeća cifra se dobija sabiranjem susjednih binarnih cifara najviših razreda. Postupak se nastavlja do kraja binarnog broja, da bi se dobile ostale cifre Grejovog kôda. Ukoliko se pri sabiranju susjednih cifara pojave jedinice za prenos, one se izostavljaju.

+ + +

510 = 0 1 0 1 BINARNI SISTEM

0 1 1 1 GREJOV KÔD

Obrnuti postupak, konverzija Grejovog kôda u binarni vrši se tako što se bit najveće pozicione vrijednosti prepiše, pa se sabira sa cifrom sledećeg razreda u Grejovom kôdu. Dobijeni bit binarnog sistema se ponovo sabere sa cifrom sledeće pozicione vrijednosti Grejovog kôda, itd. I ovdje se eventualne jedinice za prenos izostavljaju.

1110 = 1 1 1 0 GREJOV KÔD

+ + +

1 0 1 1 BINARNI SISTEM

|  |  |  |
| --- | --- | --- |
| DECIMALNI  SISTEM | BINARNI  SISTEM | GREJOV  KÔD |
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |

*Tabela 4.1.*

4.4. Specijalni kodovi

Ako se pored brojeva žele kodirati slova i drugisimboli, BCD kôd od 4 bita po simbolu postaje nedovoljan. Zato se koristi alfa-numerički BCD kôd sa sedam bita po simbolu, poznat kao ASCII kôd (engl. American Standard Code for Information Interchange). On omogućava kodovanje 128 različitih simbola ( 2 7 = 128) i može se koristiti i za prenos jednostavnijih crteža. Ovaj kôd je standardizovan za potrebe prenosa informacija između perifernih uređaja, terminala i računara.

Npr. simbolu # odgovara kodna riječ:

b7 b6 b5 b4 b3 b2 b1 = 0100011,

očigledno je da je ovo BCD kôd sa 7 bita po kodnoj riječi.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B I T I | 0 0 0 0 1 1 1 1  0 0 1 1 0 0 1 1  0 1 0 1 0 1 0 1 | | | | | | | | b7  b6  b5 |
| b4 b3 b2 b1 |
| 0 0 0 0 | NUL | DLE | SP | 0 | @ | P | / | p | |
| 0 0 0 1 | SOH | DC1 | ! | 1 | A | Q | a | q | |
| 0 0 1 0 | STX | DC2 | " | 2 | B | R | b | r | |
| 0 0 1 1 | ETX | DC3 | # | 3 | C | S | c | s | |
| 0 1 0 0 | EDT | DC4 | $ | 4 | D | T | d | t | |
| 0 1 0 1 | ENO | NAK | % | 5 | E | U | e | u | |
| 0 1 1 0 | ASK | SYN | / | 6 | F | V | f | v | |
| 0 1 1 1 | BEL | ETB | ' | 7 | G | W | g | w | |
| 1 0 0 0 | BS | CAN | ( | 8 | H | X | h | x | |
| 1 0 0 1 | HT | EM | ) | 9 | I | Y | i | y | |
| 1 0 1 0 | LF | SUB | \* | : | J | Z | j | z | |
| 1 0 1 1 | VT | ESC | + | ; | K | [ | k | { | |
| 1 1 0 0 | FF | FS | , | < | L | \ | l | | | |
| 1 1 0 1 | CR | GS | - | = | M | J | m | } | |
| 1 1 1 0 | SO | RS | . | > | N | ^ | n | ~ | |
| 1 1 1 1 | SI | US | / | ? | O | - | o | DEL | |

*Tabela 4.2.*

4.5. Kontrola kodovanja

Pri prenosu binarno kodovanih informacija postoji vjerovatnoća da će seodređeni broj bita kodne grupe registrovati sa greškom. Uzroci smetnji na prenosnom putu su različiti (npr. atmosferske smetnje, termički šumovi,preslušavanje itd.) i na njih se uglavnom ne može uticati.

Pri projektovanju digitalnih sistema posvećuje se velika pažnja kontroli kodovanih podataka tj. zaštiti od smetnji, kao i detekciji i korekciji grešaka koje one prouzrokuju.

Greške se mogu podijeliti na slučajne i grupne. Slučajne se pojavljuju usamljeno, bilo gdje u grupi kodnih riječi, a grupne su zgusnute kao grupa grešaka u određenom segmentu.

Detekcija grešaka na prijemnoj strani nikada nije sama sebi cilj jer nakon detekcije slijedi i korekcija grešaka. Najjednostavniji način detekcije grešaka je ponavljanje pojedinih riječi gdje se na prijemu upoređuju direktna i ponovljena informacija. Ukoliko postoji razlika onda se registruje greška.Ovo je jednostavan i jeftin postupak ali upola sporiji. Zbog toga se koristi detekcija grešaka pomoću visokoredundantnih kodova.

Razlikujemo dva metoda korekcije grešaka, ARQ (engl. Automatic Repeat Request) i FEC (engl. Forward Error Control). Prvi se zasniva na ponavljanju pogrešno primljenih informacija, kada se to zatraži od otpremne strane, povratnim kanalom.

Druga koristi visokoredundantne kodove pomoću kojih se određuje lokacija jednog ili više pogrešno primljenih bita. Ovde nema povratnog kanala a koder i dekoder su mnogo kompleksniji nego kod ARQ metoda.

Kod oba sistema smanjuje se brzina prenosa i to sve više što je stepen pouzdanosti veći.

ARQ sistem na otpremnoj strani dijeli inf ormacije na blokove kodnih riječi i svakom dodjeljuje kontrolni bit za detekciju greške na prijemu. Preko povratnog kanala traži se ponovno slanje bloka u kome se desila greška.

ARQ sistemi se dijele na dvije grupe. Prva je Stop and Wait ARQ kod koje se poslije otpreme bloka čeka na pozitivno ili negativno obavještenje od prijemnika. Nakon toga ili šalje sledeći blok ili ponavlja prethodno poslati.

Drugi je kontinualni ARQ gdje otpremnik poslije slanja bloka ne čeka na obavještenje od prijemne strane već odmah emituje sledeći blok. Povratni kanal signalizira samo ukoliko je neki blok pogrešno primljen. Otpremnik tada ponovo emituje pomenuti blok i sve blokove koji su emitovani poslije njega. Prednost ARQ postupka je u njegovoj jednostavnosti, a nedostatak je potreba za povratnim kanalom.

4.6. Kodovi za otkrivanje i kontrolu greške

U svakom realnom digitalnom sistemu postoji vjerovatnoća da će se određeni biti registrovati sa greškom. Zbog toga su napravljeni kodovi kojima se može detektovati greška na prijemu ili čak detektovati i korigovati, a to su zaštitni kodovi.

Kodne grupe ili kodne riječi se u prirodnom binarnom kodu razlikuju u najmanje jednoj poziciji bita. To se zove distanca kodne riječi.Kada je distanca jedan greška se ne može detektovati.Da bi se mogla detektovati greška minimalna kodna distanca treba da bude bar dva.

Npr. ako je kodna kombinacija 1010 koja odgovara vrijednosti signala 2, primljena nekorektno kao 1011, prijemnik će detektovati grešku, jer riječ 1011 nije korištena za kodovanje. Ukoliko je broj pogrešno primljenih bita paran tada greška neće biti detektovana, jer je nova kodna kombinacija korištena u kodovanju. Da bi se omogućilo da sistem detektuje i koriguje grešku, distanca između kodnih grupa mora se povećati na najmanje tri.

Osnovni nedostatak ovakvog načina kodovanja je da se smanjuje brzina rada sistema, odnosno brzina prenosa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a3 | a2 | a1 | a0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 1 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 | 1 |

*Tabela 4.3.*

4.7. Kod sa kontrolom parnosti

Najčešće se koristi kod sa kontrolom parnosti za povećanje kodne distance. Binarna kodna riječ je parna ako sadrži paran broj jedinica a neparna ako sadrži neparan broj jedinica. Da bi dobili da sve kodne riječi budu parne ili neparne koristi se dodatni bit, koji se dodaje svakoj kodnoj riječi.

Ako je korišten samo jedan dodatni bit govorimo o kodu sa jednom kontrolom parnosti i kod njega je minimalna distanca dva. Kod ovog koda se mogu detektovati samo greške u neparnom broju bita i efikasan je samo ako je zanemarljiva vjerovatnoća greške u dva bita.

Greške nastale zbog pogrešno primljenih bita mogu se detektovati i korigovati primjenom poznatog Hemingovog (Hamming) koda. Kod njega je minimalna kodna distanca tri.Zasniva se na sistemu više kontrola parnosti. Rfealizuje se tako da omogućava izolovanje pogrešnog bita, bilo da je u pitanju bit koji nosi informaciju, bilo da je kontrolni bit.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DECIMALNI BROJ | BINARNI BROJ | | | | KONTROLNI BIT |
| 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 | 1 |

*Tabela 4.4.*

|  |
| --- |
| 8 |
|  |

4.8. Primjeri kodiranja

5. Prekidačka (Bulova) algebra

Obrada podataka u digitalnom računaru se realizuje pomoću električnih veličina (napon, struja), odnosno elektronski sklopovi računara obrađuju električne veličine kojima su predstavljeni podaci. Najpogodnije je podatke binarno kodirati, odnosno predstavljati ih pomoću dva definisana stanja elektronskih sklopova, koji se stoga nazivaju digitalni sklopovi, a pošto se radi o elektronskim kolima češće se koristi termin digitalna kola.

Dva moguća stanja digitalnog kola su najčešće dva nivoa napona U1 i U2.

Recimo, U1=0V, a U2=5V. Fizičkim stanjima 0V i 5V odgovaraju dvije logičke vrijednosti ⊥ (laž) i T (istina) koja se u digitalnoj elektronici označavaju kao logička nula (0) i logička jedinica (1).



Sl. 5.1. Logičko i fizičko stanje digitalnog kola

Data korespondencija između fizičkih i logičkih stanja odgovara tzv. pozitivnoj logici. Moguće je suprotno, nižem naponu U1 dodijeliti logičku 1, a višem naponu U2 logičku 0 i tada se radi o negativnoj logici. Stanja i funkcije digitalnih kola se dakle mogu opisati pomoću logičkih vrijednosti i logičkih operacija, pa se zato umjesto termina digitalno kolo najčešće koristi termin logičko kolo. Ponašanje logičkih kola može se opisati pomoću prekidačkih ili Bulovih funkcija koje su predmet izučavanja Bulove (ili prekidačke) algebre.

Bulova algebra se zasniva na aksiomima pomoću kojih se dalje dokazuju teoreme.

Neka skup S = {X,Y,Z,U,W,...} sadrži minimalno dva različita elementa. Na ovom skupu se definišu dvije interne binarne operacije, koje se označavaju KAO " + " i " • ". Da bi skup S i operacije " + " i " • " činili Bulovu algebru, potrebno je da budu zadovoljeni aksiomi nazvani aksiomi Hantingtona. To su:

A.1. Binarne interne operacije su komutativne na skupu S i imaju osobinu distributivnosti jedne prema drugoj, tj. ako za svako X•Y•Z ∈S važi:

X + Y = Y + X ; X • Y = Y • X

X • (Y + Z) = (X • Y) + (X • Z); X + (Y • Z) = (X + Y) • (X + Z)

A.2. Binarne interne operacije imaju na skupu S različite neutralne elemente 1 i 0, tako da za svako X ∈ S postoji element 0, 0 ∈ S, tako da je:

X + 0 = X

i za svako X ∈ S postoji 1 ∈ S, tako da je:

X • 1 = X

A.3. Na skupu S svaki element, X, X ∈ S, ima jedinstven inverzan element , ∈ S, tako da je:

X + = 1 X • = 0

Važna osobina Bulove algebre, koja slijedi direktno iz aksioma, je princip dualnosti. Svi su aksiomi dati u parovima, posebno za operaciju " + " , posebno za operaciju " • ". Ako se izvrši zamjena " + " sa " • " i 1 sa 0, onda se, polazeći od aksioma za " + " operaciju, dobijaju njima dualni tj. simetrični aksiomi za " • " operaciju.

5.1. Zakoni,teoreme i pravila prekidačke algebre

Iz navedenih aksioma Hantingtona izvode se teoreme koje se dokazuju pomoću ovih aksioma.Teoreme su takođe date u parovima (dualno) i dokazuju se simetričnim – dualnim aksiomima.

T.1. *Teorema o idempotentnosti*

X + X = X ; X • X = X

dokaz:

X + X = (X + X) • 1 = (X + X) • (X + ) = X + (X • ) = X + 0 = X

X • X = (X • X) + 0 = (X • X) + (X • ) = X • (X + ) = X • 1 = X

T.2. *Teorema o "0" elementima*

X + 1 = 1 ; X • 0 = 0

dokaz:

X + 1 = (X + 1) • 1 = (X + 1) • (X + ) = X + ( • 1) = X + = 1

X • 0 = (X • 0) + (X • ) = X • (0 + ) = X • = 0

T.3. *Teorema o involuciji*

() = X

dokaz:

X + = 1 i X • = 0 slijedi:

) + () = 1 i ) • () = 0,

što znači da je X jedinstvena negacija veličine .

T.4. *Teorema o apsorpciji*

X + (X • Y) = X; X • (X + Y) = X

dokaz:

X + (X • Y) = (X • 1) + (X • Y) = X • (1 + Y) = X • 1 = X

X • (X + Y) = (X + 0) • (X + Y) = X + (0 • Y) = X + 0 = X

T.5. *Teorema o asocijativnosti*

X + (Y + Z) = (X + Y) + Z; X • (Y • Z) = (X • Y) • Z

T.6. *De Morganovi zakoni*

() = • ; () = +

Dokaz, s obzirom na A.3. dokazuje se najprije da je:

(X + Y) + ( • ) = 1 (X + Y) • (• ) = 0

odnosno: (X + Y) + ( • = {(X + Y) + } • {(X + Y) +} = {(X + ) + Y} • {X + (Y + )}

= {1 + Y} • {1 + X} = 1 • 1 = 1

(X + Y) • ( • = X • ( • + Y • ( • = (X • • + (Y •) • = 0 • + 0 • = 0 + 0 = 0 ;

što znači da je: (X + Y) + ( • = 1 = (X + Y + (

(X + Y) • ( • = 0 = (X + Y • (

odakle slijedi da je:

= .

Drugi dio teoreme se može dokazati analognim putem, korištenjem principa dualnosti.

5.2. Tabelarno predstavljanje prekidačkih funkcija

Zašto je Bulova algebra važna? Bulova algebra je matematički aparat pomoću koga se matematički opisuju procesi obrade binarnih informacija. Ranije je objašnjeno zašto su kod digitalnih sistema zastupljeni binarni signali, koji su, očigledno, po svojoj prirodi diskretni, u smislu, da mogu da imaju samo dvije unapijred određene vrijednosti, koje su kodirane sa 0 i 1, i nijednu drugu vrijednost. Takođe, već je rečeno, da su s tim u vezi digitalni sistemi izgrađeni od elemenata koji mogu da imaju samo dva različita stanja, dok je prelaz između ta dva stanja kratkotrajan i može se zanemariti, tj. taj prelaz idealno posmatrano je beskonačno kratak.

Element, koji se upravo ponaša na opisani način, i koji može da posluži da se pomoću njega fizički interpretiraju navedene logičke operacije, kako bi se lakše shvatilo njihovo značenje, je tz. kontakt ili iz običnog života poznatiji pojam, prekidač.

Na slici 5.2 je prikazan simbol prekidača, koji očigledno pokazuje da prekidač može da ima samo dva stabilna stanja, tj. da je otvoren (a) ili da je zatvoren (b), s tim da je prelaz između ova dva stanja vrlo kratkotrajan, idealno posmatrano, beskonačno kratak.



a) b)

Slika 5.2. Simbol prekidača



Slika 5.3. Fizička interpretacija logičkih operacija I i ILI pomoću strujnog kola

Binarna promjenljiva koja se fizički realizuje pomoću prikazanog prekidača ima vrijednost 0 kada je prekidač otvoren i vrijednost 1 kada je prekidač zatvoren. Zbog toga se promjenljiva x piše pored simbola za prikazani prekidač čime se simbolično pokazuje da se ta promjenljiva fizički realizuje pomoću tog prekidača. Prikazani prekidač se naziva normalno otvoreni ili radni prekidač.

Na slici 5.3.b. je prikazan simbol takozvanog normalno zatvorenog ili mirnog prekidača. Ovaj prekidač je zatvoren kada nije aktiviran i obrnuto otvoren kada je aktiviran, tako da radi inverzno u odnosu na normalno otvoreni prekidač.Promjenljiva koja se ostvaruje pomoću ovog prekidača je negacija odnosno komplement promjenljive x; u već ranije datoj oznaci . Pored ovog prekidača se piše oznaka što simbolično označava da se baš ta binarna promjenljiva ostvaruje pomoću ovog prekidača.

Na slici 5.3.a. je prikazano strujno kolo koje sadrži dva prekidača i sijalicu, vezane na red, pri čemu ovo strujno kolo fizički interpretira I logičku operaciju.

Slično kao što prekidači realizuju binarne promjenljive tako i sijalica realizuje binarnu promjenljivu čija je vrijednost jednaka 1 kada sijalica svijetli i 0 kada sijalica ne svijetli. Očigledno, sijalica će da svijetli kada su oba prekidača u isto vrijeme zatvorena, a u svim drugim slučajevima sijalica neće da svijetli. Drugim riječima, binarna promjenljiva z; koja se fizički realizuje pomoću sijalice, imaće vrijednost 1 jedino kada binarne promjenljive x i y; koje se fizički realizuju pomoću prekidača, imaju vrijednost 1 obe u isto vrijeme, a u svim drugim slučajevima promjenljiva z će biti jednaka 0. To nije ništa drugo do operacija logičkog množenja čime je razjašnjeno kakvo je njeno fizičko tumačenje.

Dakle, redna veza prekidača predstavlja tehničku realizaciju funkcije konjunkcije, odnosno I funkcije:

Z = X • Y

Na slici 5.3.b. je prikazano strujno kolo koje sadrži, takođe, dva prekidača, ali sada vezanih paralelno i sijalicu koja je vezana redno sa pomenutom paralelnom spregom prekidača. Očigledno, sijalica neće da svijetli kada su oba prekidača u isto vrijeme otvorena, a u svim drugim slučajevima sijalica će da svijetli. Drugim riječima, binarna promjenljiva z imaće vrijednost 0 jedino kada binarne promjenljive x i y imaju vrijednost 0, obe u isto vrijeme, a u svim drugim slučajevima promjenljiva z će biti jednaka 1. To nije ništa drugo do operacija logičkog sabiranja čime je razjašnjeno kakvo je njeno fizičko tumačenje.

Dakle, paralelna veza prekidača predstavlja tehničku realizaciju funkcije disjunkcije, odnosno ILI funkcije:

Z = X + Y

Ovo se može i tabelarno prikazati. Ovakve tabele se zovu kombinacione tabele ili tabele vrijednosti logičkih stanja ili tabele istinitosti. Tabela 5.1. se odnosi na I funkciju a tabela 5.2. na ILI funkciju.

|  |  |  |
| --- | --- | --- |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Tabela 5.1. Tabela 5.2.

5.3. Analitičko predstavljanje prekidačkih funkcija

Analitički funkcije se mogu izražavati u dva oblika tj. pomoću dvije forme: disjunktivne forme i konjunktivne forme.

Disjunktivna forma predstavlja logičku sumu logičkih proizvoda a konjunktivna forma logički proizvod logičkih suma.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| dec. broj | X1 | X2 | X3 | Z |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

Tabela 5.3.

Tabelarno zadata prekidačka funkcija predstavlja se analitički pomoću disjunktivne forme na taj način što se napiše logički zbir onoliko elementarnih proizvoda koliko u tabeli ima jediničnih vrijednosti funkcije. Zatim se u elementarnim proizvodima negiraju one promjenljive koje u odgovarajućem redu imaju vrijednosti nula. Pod elementarnim proizvodima podrazumijevamo proizvod nezavisno promjenljivih. Neka je funkcija zadata tabelom 5.3. Ona ima vrijednost 1 za vrijednosti nezavisno promjenljivih navedenih u trećoj, četvrtoj, šestoj i sedmoj vrsti tj. redu. Sada se pišu elementarni proizvodi za svaku od njih:

Z =  1•X2•X3 + X1•2•3 + X1•X2•3 + X1•X2•X3

Analitičko predstavljanje prekidačke funkcije pomoću konjunktivne forme, na osnovu zadate tabele, postiže se pisanjem logičkog proizvoda onoliko elementarnih suma koliko u tabeli ima redova sa vrijednošću 0. Ovde se negiraju promjenljive koje imaju vrijednost 1 u elementarnim sumama. Za isti primjer tabelarno date funkcije (tabela 5.3.) pišu se elementarne sume za redove br. 0,1,2,5, jer za te vrijednosti nezavisno promjenljivih funkcija ima vrijednost nula. Dakle, konjunktivna forma funkcije je sledeća:

Z =( X 1+X2+X3) • (X1+X2+3) •(X1+2+X3) •(1+X2+3)

Disjunktivna forma (DF) i konjunktivna forma (KF) označavaju se decimalnom notacijom tako što se umjesto elementarnog proizvoda,odnosno sume piše decimalna vrijednost binarnog broja kome odgovara taj red u tabeli. Tako se prethodni primjer može napisati kao:

disjunktivna forma (DF) Z = {3,4,6,7}

konjunktivna forma (KF) Z = {0,1,2,5}

Pravilo je da se koristi ona forma koja daje manje elementarnih članova jer je pogodnija za upotrebu u računanju i realizaciji.

5.4. Primjeri-vježbanje

Riješiti zadatke:

1. Predstaviti funkciju tabelarno: 
2. Za zadatu tabelu  napisati analitički oblik prekidačke funkcije.
3. Za zadatu tabelu  napisati analitički oblik prekidačke funkcije.

5.5. Minimizacija prekidačkih funkcija

Minimizacija je postupak u kojem se logička funkcija tj. algebarski izraz svodi na njoj jednostavniji oblik. Аlgebarski metod gdje se uz pomoć pravila, osobina i zakona Bulove algebre i De Morganovih pravila dolazi do jednostavnijeg oblika logičke funkcije. Ova metoda za četiri i više promjenjivih predstavlja težak i mukotrpan posao, tako da su razvijene druge metode minimizacije logičkih funkcija kao na primjer Veič- Karnoova metoda minimizacije.

5.5.1. Veič-Karnoova metoda minimizacije

Prema ovoj metodi minimizacija se izvodi grafičkim putem. Ona je jednostavna i praktična, a zasniva se na upisivanju funkcije u specijalnu tabelu. Neka je data funkcija tri promjenljive. Vrijednosti funkcije P0, P1, P2, ... , P7 koje mogu biti 0 ili 1, unose se u Veič-Karnoov dijagram redoslijedom prikazanim u tabeli 5.4. Na stranicama dijagrama naznačene su oblasti nenegiranih promjenljivih (afirmacija) i njihovih komplemenata (negacija). Grafički se minimizacija izvodi tako što se u tabeli zaokruže dvije susjedne jedinice (tabela 5.6. b), za funkciju zadatu tabelom 5.6. a). Treba obraditi pažnju na raspored promjenljivih i njihovih negacija na stranicama dijagrama. Posmatra se koja od promjenljivih u intervalu zaokruženog para jedinica ne prelazi iz afirmacije u negaciju. To su u datom primjeru 2 i 3. Njihov logički proizvod će predstavljati minimalnu disjunktivnu formu funkcije:

F = 2 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ULAZI | | | IZLAZI | Standardni logički proizvod |
| X1 | X2 | X3 | F |
| 0 | 0 | 0 | 0 | P0 |
| 0 | 0 | 1 | 0 | P1 |
| 0 | 1 | 0 | 1 | P2 |
| 0 | 1 | 1 | 1 | P3 |
| 1 | 0 | 0 | 0 | P4 |
| 1 | 0 | 1 | 0 | P5 |
| 1 | 1 | 0 | 1 | P6 |
| 1 | 1 | 1 | 0 | P7 |

Tabela 5.4.

X1 1

|  |  |  |  |
| --- | --- | --- | --- |
| P6 | P7 | P3 | P2 |
| P4 | P5 | P1 | P0 |

X2

2

3 X3 3

Tabela 5.5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ULAZI | | | IZLAZI | Standardni logički proizvod |
| X1 | X2 | X3 | F |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |

Tabela 5.6. a)

X1 1

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |

X2

2

3 X3 3

Tabela 5.6. b)

U slučaju četiri susjedne jedinice primjenjuje se isti postupak – zaokružuju se kvarteti jedinica, pa se za članove logičkog proizvoda uzimaju samo one promjenljive koje unutar kvarteta ne prelaze iz afirmacije u negaciju.

Na primjer, ako treba minimizirati funkciju

 = 

na osnovu decimalne notacije ove funkcije može se zaključiti na kojim mjestima Veič – Karnoovog dijagrama (tabela 5.7) treba da se nalaze jedinice (0,1,2,3). Vidi se da promjenljive Y i Z prelaze u intervalu kvarteta iz afirmacije u negaciju, pa je minimalni oblik funkcije

F =

X

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |

Y

Z

Tabela 5.7.

Ako se pažljivije posmatra Veič – Karnoova tabela, može se uočiti da se područje nezavisno promjenljive Z proteže direktno slijeva na desnu stranu tabele. To se može zamisliti kao da je tabela nacrtana na valjku, čiji je omotač po visini razvije u ravan.

Ukoliko u tabeli postoji više grupa susjednih jedinica, za svaku grupu se piše logički proizvod promjenljivih, koje unutar grupe ne prelaze iz afirmacije u negaciju. minimalna disjunktivna forma (MDF) funkcije je logički zbir tih proizvoda, na primjer:

 = 

Kada se ovo unese u dijagram (tabela 5.8.), dobijaju se tri para susjednih jedinica. Par zaokružen dvostrukom linijom ima logički proizvod , par zaokružen isprekidanom linijom Y Z i par zaokružen punom linijom , pa je MDF funkcije:

F(X,Y,Z) = 

Za funkciju četiri promjenljive redoslijed ređanja elemenata je dat u Veič – Karnoovoj tabeli 5.8. Pravila minimizacije ostaju ista kao i za funkciju tri promjenljive, s tim što se ovdje, osim parova i kvarteta, zaokružuju još i okteti jedinica. Ovu tabelu treba zamisliti kao da je nacrtana na torusu, pa razvijena u ravan. Drugim riječima, njena lijeva ivica se direktno naslanja na desnu, a donja ivica na gornju. Ovo treba imati u vidu prilikom zaokruživanja.

X1 1

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | 14 | 6 | 4 |
| 13 | 15 | 7 | 5 |
| 9 | 11 | 3 | 1 |
| 8 | 10 | 2 | 0 |

4

X2

X4

2

4

3 X3 3  Tabela 5.8.

Za nalaženje minimalne konjunktivne forme (MKF) funkcije, u tabeli treba zaokružiti parove, kvartete i oktete nula, s tim što afirmacije i negacije promjenljivih na stranicama dijagrama međusobno mijenjaju mjesta. Ovaj slučaj je prikazan u tabeli 5.10. Tabela 5.9. je za dobijanje disjunktivne, a 5.10. za dobijanje konjunktivne forme funkcije.

Minimalna disjunktivna forma je:

F = 123 + 3 4 + 1 2 4 + 123 4

MKF ima onoliko logičkih suma koliko u tabeli postoji zaokruženih parova, kvarteta i okteta nula. U svakoj od tih suma učestvuju one promjenljive koje u intervalu zaokruženih grupa ne prelaze iz afirmacije u negaciju. Kvartet nula zaokružen punom linijom daje logičku sumu 2 + 3. Od para zaokruženog dvostrukom linijom dobija se član 1 + 3 + 4, od kvarteta zaokruženog tačkasto 1 + 4 i od para zaokruženog debljom linijom 2 + 3 + 4. Finalni oblik MKF dobija se kada se napravi logički proizvod svih suma:

F = (2 + 3) (1 + 3 +4) (1 + 4) (2 + 3 + 4)

X1 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |

4

X2

X4

2

4

3 X3 3

Tabela 5.9.

X1 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |

4

X2

X4

2

4

3 X3 3 Tabela 5.10.

Veič – Karnoova metoda minimizacije je veoma praktična za mali broj nezavisno promjenljivih. Međutim, za veći broj promjenljivih tabela postaje glomazna i nepraktična. Ova metoda je isto tako nepovoljna ako je raspored jedinica „šahovski“ ili njemu sličan. U ovim slučajevima se primjenjuju druge metode minimizacije.

5.8. Primjeri-vježbanje

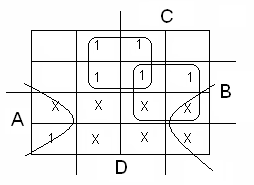
**Primjer:** a osnovu kombinacione tabele izvesti Karnoovu mapu i napisati odgovarajuće Booleove jednačine za izlaze a, b i c.

**Rješenje:**

NAPOMENA:U tabeli je zadnjih 6 izlaza označeno sa X, jer ćemo se bazirati samo na prvih 10 od 16 kombinacija. Inače, postoje slučajevi kada, za odeređena stanja na ulazu, izlazno stanje nije definisano. Takva stanja se zovu **nevažna stanja,** pošto o njima ne moramo voditi računa. Takva stanja se u Karnoovoj mapi mogu označiti kao 0 ili 1, zavisno od potrebe u tom trenutuku.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ULAZI** | | | | **IZLAZI** | | |
| **A** | **B** | **C** | **D** | **a** | **b** | **c** |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X |

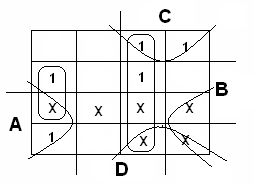
Uradimo prvo za izlaz a:



Prva kontura (8,12,10,14) daje , druga kontura (1,3,5,7) daje  i treća kontura (6,7,14,15) daje BC, pa za izlaz a imamo:

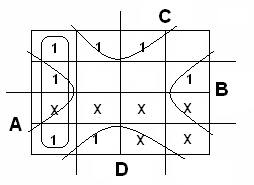
a = ++BC

Sada za izlaz b:

Prva kontura (8,12,10,14) daje , druga kontura (2,3,10,11) daje , treća kontura (3,7,11,15) daje CD, četvrta kontura (4,12) daje  pa za izlaz b imamo:

b = ++CD+

Za izlaz c ćemo imati:



Prva kontura (4,6,12,14) daje , treća kontura (1,3,9,11,) daje, četvrta kontura (0,4,8,12) daje  pa za izlaz c imamo:

c = ++

PITANJA I ZADACI:

1. Predstaviti funkciju tabelarno: 
2. Za zadatu tabelu  napisati analitički oblik prekidačke funkcije.
3. Za zadatu tabelu  napisati analitički oblik prekidačke funkcije.
4. Izvršiti minimizaciju funkcije Veič – Karnoovom metodom 
5. Izvršiti minimizaciju funkcije Veič – Karnoovom metodom 
6. Izvršiti minimizaciju funkcije Veič – Karnoovom metodom 

6. Logička kola

6.1. Logičke operacije i logička kola

Poznavanje funkcija logičkih struktura je neophodno za razumijevanje kako zaista funkcionišu digitalni sklopovi, a samim tim i digitalni računar kao poseban kompleksan sklop.

Digitalna tehnologija se razvija i mijenja, tako utiče na promjenu shvatanja digitalnih sistema kao i efikasnih načina njihovog projektovanja. Elementi konceptualizacije digitalnih računara su funkcionalne cjeline, kao procesor, memorijski sistem, sabirnice, U/I sistemi, ali u praksi se logičko projektovanje oslanja na dvije osnovne vrste gradivnih elemenata:

1. Kombinacione strukture ( kola kod kojih su logički izlazi određeni logičkim funkcijama izvršenim nad logičkim ulazima u datom vremenskom trenutku)
2. Sekvencijalne strukture ( su kola kod kojih izlazi zavise od ulaza kao i od predhodnih stanja)

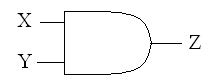
Obe vrste logičkih struktura koriste logička kola za kombinovanje logičkih ulaza u željene izlaze. Digitalni svijet je baziran na binarnom brojnom sistemu, jer signali u digitalnim sklopovima imaju samo dvije vrijednosti **0** i **1**, za razliku od analognih kola koja rade sa signalima koji mogu imati beskonačno različitih vrijednosti. Osnovni razlog što su logičke operacije bazirane na binarnom brojnom sistemu je to što tako mogu projektovati jednostavna i stabilna električna kola koja razlikuju samo dva stanja. Druga vrsta sklopova koji se mogu graditi na osnovu pomenutog binarnog principa su kola koja mogu pamtiti stanja.

Ovo je polazna osnova za građenje uređaja – digitalnog računara, koji može prolaziti kroz različite binarne sekvence stanja određujući svako naredno stanje na osnovu predhodnog i vrijednosti upravljačkih funkcija.

Logičko kolo je elektronska struktura koja od jednog ili više ulaznih signala pravi jedan izlazni signal.

*I funkcija ili konjunkcija* predstavlja logički proizvod nezavisno promjenljivih. Električna kola koja realizuju ovu funkciju su logička I kola (engl. AND gate). Iz kombinacione tabele 6.1. se vidi da ako su svi ulazi I kola nula i izlaz mora biti nula. Ako je bilo koji izlaz nula i izlaz je takođe nula. Samo kada su svi ulazi na nivou logičke jedinice i izlaz je na nivou logičke jedinice. Na slici 6.1. je prikazan grafički simbol I kola. I kolo se koristi prvenstveno kao kontrolni element, pri čemu se pomoću jednog ulaza reguliše propuštanje informacija s drugih ulaza.

|  |  |  |
| --- | --- | --- |
| ULAZ | | IZLAZ |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

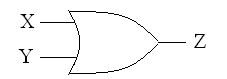


Slika 6.1. **«I»** Logičko kolo

Tabela 6.1.

*ILI funkcija ili disjunkcija* predstavlja logički zbir nezavisno promjenljivih. Električna kola koja realizuju ovu funkciju su logička ILI kola (engl. OR gate). Iz kombinacione tabele 6.2. se vidi da je izlaz ILI kola na nivou logičke jedinice ako bilo koji od njegovih ulaza ima vrijednost 1. Na slici 6.2. je prikazan grafički simbol ILI kola.

|  |  |  |
| --- | --- | --- |
| ULAZ | | IZLAZ |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



Slika 6.2. **«ILI»** Logičko kolo

Tabela 6.2.

*Negacija ili invertovanje* je značajna operacija u logičko-prekidačkim mrežama. Pored navedenih binarnih operatora postoje i unarni. Takav je komplement i označava se sa . Tako znači «Uzmi komplement od X» (invertuj X). Operacija komlement se fizički realizuje logičkim «NE» kolom zvanim Invertor.

|  |  |
| --- | --- |
| ULAZ | IZLAZ |
| X | Y |
| 0 | 1 |
| 1 | 0 |

X

Y

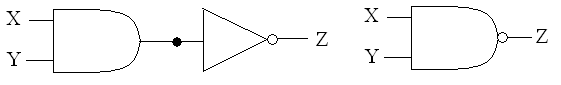
Slika 6.3. **«NE»** Logičko kolo Tabela 6.3.

*Šeferova funkcija*  po definiciji ima vrijednost 0 samo ako su sve nezavisno promjenljive 1. U svim ostalim slučajevima funkcija ima vrijednost 1. Na osnovu toga možemo napraviti kombinacionu tabelu ( tabela 6.4.). Iz ove tabele se može napisati analitički oblik funkcije pomoću konjunktivne forme kao

F = ili F = ako se primjene De Morganovi zakoni.

Kolo koje realizuje Šeferovu funkciju zove se NI kolo (engl. NAND gate). NI kolo se može realizovati i pomoću I kola i invertora, po čemu je i dobilo ime. Analitički oblik ove funkcije pokazuje da se ona može tehnički realizovati kao paralelna veza zatvorenih prekidača (slika 6.5.).

NI kolo je ekvivalentno I kolu sa invertovanim izlazom (I kolo u seriji sa NE kolom)



Slika 6.4. **«NAND»** Logičko kolo

|  |  |  |
| --- | --- | --- |
| ULAZ | | IZLAZ |
| X | Y | Z |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



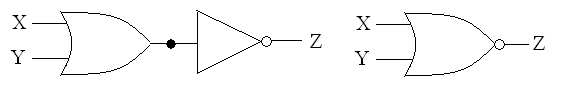


Slika 6.5. Relejno NI kolo Tabela 6.4.

*Pirsova funkcija* po definiciji ima vrijednost 1 samo ako su sve nezavisno promjenljive nula. Ako bilo koja od promjenljivih ima vrijednost 1, funkcja je nula. Tabela 6.5. predstavlja kombinacionu tabelu Pirsove funkcije iz koje se može napisati disjunktivna forma:

F = ili primjenom De Morganovih zakona

F =



Slika 6.6. **«NOR»** Logičko kolo

|  |  |  |
| --- | --- | --- |
| ULAZ | | IZLAZ |
| X | Y | Z |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Tabela 6.5.

Pirsova funkcija se može realizovati pomoću ILI kola i invertora, pa se zato kolo zove NILI kolo (engl. NOR gate). Analitički oblik ove funkcije pokazuje da se ona može tehnički realizovati kao redna veza zatvorenih prekidača (slika 6.7.).



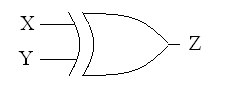
Slika 6.7. Relejno NILI kolo

*Ekskluzivna disjunkcija* ima vrijednost 1 kada samo jedna od nezavisno promjenljivih ima vrijednost 1. U svim ostalim slučajevima je 0. To se prikazuje u kombinacionoj tabeli 6.6. i disjunktivna forma funkcije će biti:

F =

Na slici 6.8.a) je prikazana realizacija ove funkcije.

Ekskluzivno ILI kolo ili EXILI (engl. EXOR gate) je kolo koje realizuje funkciju ekskluzivne disjunkcije i označava se posebnim simbolom (slika 6.8.b).



a) b)

Slika 6.8. **«EXOR»** Logičko kolo

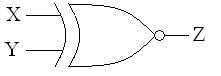
|  |  |  |
| --- | --- | --- |
| ULAZ | | IZLAZ |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Tabela 6.6.

*Funkcija ekvivalencije* dobija se negacijom funkcije ekskluzivne disjunkcije.

F =

Ona ima vrijednost 1 kada su nezavisno promjenljive iste. Kolo koje realizuje funkciju ekvivalencije naziva se **komparator**. Kolo se još zove i NEXILI kolo (engl. NEXOR gate) njegov simbol je prikazan na slici 6.9.b).



a) b)

**Slika 6.9.«NEXOR»** Logičko kolo

|  |  |  |
| --- | --- | --- |
| ULAZ | | IZLAZ |
| X | Y | Z |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Tabela 6.7.