GLAVA I

UVOD

Makroskopski posmatrano, većina prirodnih pojava je kontinualnog (neprekidnog) karaktera, menja se u vremenu, bez trenutnih skokova kao, na primer, temperatura, pritisak, prirodna svetlost itd. Tehnički uređaji čije se ulazne i izlazne veličine - signali - menjaju po analogiji (sličnosti) s promenama prirodnih - kontinualnih - pojava nazivaju se analogni uređaji. Ako se, pak, neka veličina menja skokovito s jedne vrednosti na drugu, onda je to diskretna veličina. Uređaji koji rade sa diskretnim signalima nazivaju se digitalnim uređajima. Diskretni signali su obično dati u vidu diskretnih naponskih nivoa. Termin digitalni potiče od latinske reči *digitus* - prst. Međutim, u ovom slučaju bi adekvatnije značenje bilo "računanje na prste", koje je, u stvari, i bilo prvi način računanja u ljudskom društvu. Za primer analognog i digitalnog načina rada može se uzeti običan ručni časovnik. Ako se kazaljke časovnika pomeraju kontinualno, onda je to analogni uređaj. Međutim, ako se kazaljke časovnika pokreću u skokovima od po 1 sekundu ili 1 minut, ili 5 minuta itd., onda je to digitalni uređaj. Na sličan način se i teret izražava kao kontinualna veličina - sumom diskretnih vrednosti tegova od po 1 g ili 1 kg, itd. Ako se želi veća preciznost merenja vremena, odnosno težine, mora se usvojiti manja jedinica mere, kvant ili nivo diskretizacije kontinualne veličine. To znači da se izražavanjem kontinualnih veličina pomoću diskretnih svesno čini manja ili veća greška, koja se naziva greška diskretizacije ili greška kvantizacije.

Iako je analogni način rada na izgled tačniji od digitalnog, preciznost kontinualnih signala je retko moguće potpuno iskoristiti zbog toga što se oni ne mogu meriti, očitavati, obrađivati ili na bilo koji drugi način interpretirati sa tako visokom tačnošću. S druge strane, digitalni podaci se javljaju u numeričkoj formi, pa se mogu lakše kvanititativno izražavati, obrađivati, memorisati i očitavati. Osim toga, signali izraženi u digitalnoj (impulsnoj) formi manje su osetljivi na šum i smetnje nego analogni signali, tako da se mogu prenositi na daljinu mnogo pouzdanije.

Primenjuje se, takođe, i hibridna obrada podataka korišćenjem dobrih strana i analognog i digitalnog načina rada. Najčešće se realni analogni ulazni signali prvo pretvaraju - konvertuju - u "veštačke" diskretne impulsne signale pomoću analogno-digitalnih ili A/D konvertora, pa se vrši digitalna obrada tih podataka. Kada je potrebno, na izlazu se obrađeni digitalni signali ponovo vraćaju u analogni oblik pomoću digitalno-analognih konvertora (D/A). Zahvaljujući ovim konvertorima, omogućena je znatno šira primena digitalnih uređaja, sa svim njihovim prednostima i u onim oblastima gde je signal u suštini analogne prirode.

Ideja o izgradnji digitalnih uređaja nije novog datuma. Prvu mehaničku mašinu za sabiranje konstruisao je 1642. godine francuski matematičar B. Paskal. Godine 1672. nemački matematičar G.V. Lajbnic usavršio je sličan mehanički računar, koji je, ponavljanjem operacija sabiranja i oduzimanja mogao da obavlja množenje i deljenje brojeva. Ideja o digitalnoj računskoj mašini s memorijom potiče od engleskog matematičara Č. Bebidža. Mašina je trebalo da bude potpuno mehanička. Preokret u razvoju digitalnih uređaja nastaje početkom primene Bulove binarne algebre (Dž. Bul) u tehnici, tj. u analizi logičko-prekidačkih mreža (Šenon, Nakasima, Šestakov 1938. godine). Prva moderna digitalna elektronska mašina konstruisana je u SAD tokom Drugog svetskog rata. Konstruisali su je elektroinženjer P. Ekert i fizičar Dž. Mokli. Taj računar se zvao ENIAC (engl. Electronic Numerical Integrator and Computer). Bio je težak oko 30 tona, sastojao se od 19 000 elektronskih cevi i trošio oko 200 kW električne snage. Program rada je zadavan izvođenjem žičanih veza na programskoj ploči, pa je operator morao da obavi mnogobrojna prespajanja kad bi postavljao nov program. Međutim, tek pojavom moderne tehnologije poluprovodnika, tj. tranzistora, a zatim i integrisanih kola, nastaje prava ekspanzija digitalnih uređaja. Tehnika integrisanih kola je prirodni nastavak razvoja poluprovodničke tehnologije i po tome ona nije neko fundamentalno naučno-tehničko otkriće, kao što je to bio tranzistor. Ipak, primena integrisanih kola je izazvala pravu revoluciju na svim nivoima elektronske industrije. Razdoblje između 1957. i 1959. godine može se smatrati početkom epohe integrisanih kola. Prvo kolo je patentirao 1959. godine Dž. Kilbi iz američke firme *Texas Instruments*. Samo dve godine posle pronalaska prvog integrisanog kola, firma Texas Instruments je napravila prvi digitalni uređaj za obradu podataka s integrisanim kolima. Posle toga masovno počinju da se pojavljuju najraznovrsniji digitalni uređaji.

Danas se služimo mnoštvom ručnih i stonih kalkulatora, malih, srednjih i velikih digitalnih računara za obradu raznih vrsta podataka. Razvijen je, dakle, veliki broj digitalnih računskih mašina koje se međusobno znatno razlikuju, kako po tehnologiji primenjenih komponenata, tako i po složenosti, programskim mogućnostima i vrstama primene. U tom smislu se često govori o različitim generacijama računara, od kojih svaka odgovara određenom vremenskom razdoblju. Pojava elektronskih računara je označila početak ere automatizacije i onih intelektualnih procesa u kojima je čovek bio nezamenljiv. Do tada se težilo da mašina zameni čovekov fizički rad. Računari su označili kvalitativan korak napred. Mašina počinje da "misli" zamenjujući čoveka i u intelektualnim poslovima. Do pojave računara nijedna mašina nije imala tako velike i raznovrsne mogućnosti, niti je njena primena izazivala tako krupne posledice u gotovo svim oblastima ljudske delatnosti. Danas je već postalo nezamislivo da ljudi i dalje rade hiljade knjigovodstvenih kalkulacija, obavljaju obračune plata, statistička istraživanja, evidenciju distribucije eletrične energije i mnoge druge slične rutinske poslove. Računari obavljaju i znatno složenije poslove, kao što su upravljanje tehnološkim proizvodnim procesima, upravljanje telekomunikacionim sistemima, obrada slika, zatim planiranje i optimizacija proizvodnje, ispitivanje tržišta i planiranje zaliha robe i skladišnog prostora, planiranje velikih gradova. Problemi saobraćaja, obrada raznih kartoteka, evidencije građana, razni informacioni sistemi i mnogi drugi poslovi danas su u nadležnosti računsko-informacionih centara. Ekonomska moć jedne zemlje meri se brojem i nivoom primene elektronskih računara.

Iako se računske mašine konstruišu da bi automatizovale jedan deo čovekovog intelektualnog rada, ipak se ne mogu praviti neka značajnija upoređivanja na relaciji čovek–mašina. Računar izvodi dugačak i mukotrpan rutinski kalkulatorski posao mnogo brže i preciznije od čoveka, striktno sledeći pri tome program koji mu je čovek zadao. Računar ne poseduje nikakvu inteligenciju. Kreativne ljudske sposobnosti, kao što su prilagođavanje, analiza stečenih iskustava, stvaralačka mašta i intuicija, ne mogu se za sada realizovati pomoću računara, mada se i to pokušava (adaptivni računari i drugi). Međutim, trebalo bi očekivati da svi napori uloženi u razvoj nove tehnike oslobode čoveka od prinudnog rada, stvore materijalno blagostanje, ljudske slobode i slobodno vreme u kojem će čovek moći još više da razvija svoje spontane, kreativne aktivnosti.

1.1. POJAM INFORMACIJA I NJIHOVO KODOVANJE

Reč informacija u svakodnevnom životu uglavnom znači isto što i obaveštenje. Ako se procenjuje količina primljenih obaveštenja - informacija, nije bitan broj reči od kojih se obaveštenje sastoji, već sadržaj obaveštenja, odnosno koliko novosti to obaveštenje ima. Svako obaveštenje o događaju koji se sa sigurnošću očekuje, tj. čija je verovatnoća ostvarivanja vrlo velika, sadrži malo informacija. I obrnuto, saopštenje o neočekivanom događaju, čija je verovatnoća ostvarivanja vrlo mala, sadrži znatno više informacija. Pojam informacije u elektronici je prvo korišćen u prenosu informacija, tj. vesti telekomunikacionim sistemima. Danas je taj pojam dobio šire značenje i odnosi se na obradu informacija, tj. podataka.

Predstavljanje informacija pomoću simbola koji su elementi nekog skupa, takozvane kodne azbuke, naziva se kodovanje informacija. Jednostavan ali važan tip informacija su binarno kodovane informacije, sastavljene od skupa samo dva različita simbola. Ako se za ova dva simbola usvoje brojne oznake 1 i 0 (uzete iz decimalnog brojnog sistema), onda je takva informacija izražena - kodovana - u binarnom brojnom sistemu. U teoriji informacija je definisano da ukoliko se u binarnom skupu, simboli 1 i 0 pojavljuju s istom verovatnoćom $p_0 = p_1 = 1/2$, onda pojava svakog simbola iz tog skupa nosi sa sobom informaciju

$$I = -\log_2 p = -\log_2 1/2 = 1 \text{ bit}$$

Na taj način je uvedena jedinica za merenje količine informacija 1 BIT (od engleskog termina BInary uniT). Dakle, jedan bit je jedinica mere za količinu informacije koja može imati samo jednu od dve moguće vrednosti -1 ili 0. Međutim, u žargonu digitalne tehnike pod terminom bit odomaćio se naziv za jednu cifru binarnog broja 1 ili 0 (BInary digiT), odnosno za impuls kojim je ta cifra predstavljena, bez obzira na verovatnoću 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. Razlog tome je najlakša mogućnost tehničke realizacije elektronskih elemenata sa dva različita stabilna stanja pomoću kojih se mogu predstaviti binarne cifre 0 i 1. Tako je *bistabilni elemenat* postao najvažnija komponenta u savremenim digitalnim računarima, a *binarni brojni sistem* najvažniji brojni sistem u svetu elektronike.

Kod digitalnih uređaja se razlikuju tri vrste informacija: *naredbe* (*komande*), *podaci* i *adrese*. Podaci su informacije koje je računar primio od spoljašnjeg sveta, ili ono što je dobijeno kao rezultat neke obrade u samom računaru. Podaci se uvode u digitalni računar u kome se obrađuju i na osnovu njih se dobijaju nove informacije. Međutim, pre uvođenja u digitalni računar, podaci se moraju prevesti u binarni oblik da bi računar mogao da ih obrađuje. Podaci su, dakle, skup brojeva sa kojima treba obaviti određene računske operacije. S druge strane, instrukcije (naredbe ili komande) nalažu računaru šta treba da radi sa podacima. Prema tome, niz instrukcija koji čini logičku celinu predstavlja program po kome računar radi. I instrukcije i podaci su binarno kodovane informacije, tj. binarni brojevi, pa ih računar, zahvaljujući internoj organizaciji i programu, razlikuje jedne od drugih.

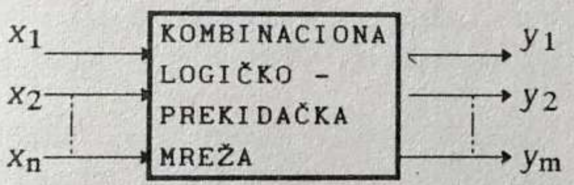
1.3. DIGITALNA KOLA I MREŽE

Prema funkciji koju obavljaju, svi elementi, ili elementarne komponente u digitalnim uređajima, mogu se svrstati u tri grupe: a) elementi koji generišu i uobličavaju električne impulse, b) logički elementi ili elementi kombinacione logike i c) momorijski elementi.

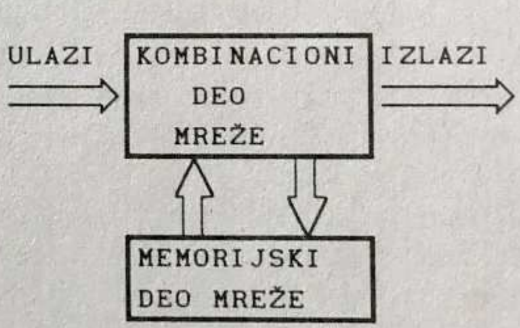
a) *Elementi* koji služe za *generisanje* i *uobličavanje* impulsa čine standardna kola impulsne tehnike, kao što su razne vrste multivibratora, komparatori, kola za diferenciranje i integriranje i druga.

b) *Logički elementi* ili *logička kola* služe za realizovanje logičkih stanja, odnosno njihovih kombinacija bez mogućnosti pamćenja. Stanje na izlazu logičkog kola zadržava se samo dotle dok postoje signali na ulazu koji su ga i prouzrokavali. Njihova uloga se uglavnom svodi na odlučivanje šta, kada i kako treba uraditi sa binarnim informacijama u procesu obrade, što odgovara stavovima formalno-logičkog mišljenja *tačno* ili *pogrešno*. Električne mreže realizovane na ovaj način nazivaju se *kombinacione logičko-prekidačke mreže*. Njihov opšti oblik je prikazan na slici 1.1. Mreža ima n ulaza na koje se dovode binarne nezavisno promenljive $x_1, x_2, ..., x_n$. Na izlazu se dobija m zavisno promenljivih binarnih veličina ili logičko-prekidačkih funkcija $y_1, y_2, ..., y_m$.

c) *Memorijski* elementi služe za pamćenje - memorisanje - prethodnih stanja i uopšte za čuvanje - skladištenje - informacija. Električne mreže realizovane s ovakvim elementima nazivaju se *sekvencijalne logičko-prekidačke mreže*. Na slici 1.2 prikazana je uopštena strukturna šema takve mreže u kojoj se ceo memorijski deo može izdvojiti u jedan zaseban blok, a kombinacioni deo mreže u drugi blok. U posmatranom trenutku na ulaz mreže dolaze spoljašnji signali koji odgovaraju tom trenutku i koji se mogu nazvati sadašnje ulazne promenljive. Iz mreže se uzimaju sadašnji izlazni signali, dok se iz memorijskog dela interpretiraju sadašnja stabilna stanja, a iz kombinacionog u memorijski deo ulaze signali koji će promeniti postojeća - sadašnja stabilna stanja memorijskih elemenata i prouzrokovati nova koja će se koristiti u narednim vremenskim intervalima.



Sl. 1.1. Kombinaciona logičko-prekidačka mreža



Sl. 1.2. Sekvencijalna logičko-prekidačka mreža

Činjenica da se na ulazima i izlazima svih digitalnih kola, i kombinacionih i memorijskih, mogu pojaviti signali koji imaju samo jedno do dva moguća logička stanja 0 ili 1, tj. samo dve vrednosti napona, omogućava da se izlaz jednog digitalnog kola može priključiti direktno na ulaz jednog ili više ostalih digitalnih kola, čime se stvaraju uslovi za jednostavno povezivanje elementarnih digitalnih kola.

Impulsi u jednom digitalnom uređaju mogu da budu:

- *takt-impulsi*, koji se dobijaju iz posebnog takt-generatora i daju osnovni ritam rada celom uređaju. Ovi impulsi se najčešće javljaju u celom sistemu;
- *informacioni impulsi*, pomoću kojih se izražavaju informacije u vidu binarno kodovanih podataka, kao što su brojevi, slova i simboli;
- *kontrolni impulsi*, koji upravljaju redosledom izvršavanja pojedinih operacija u uređaju.

Zadatak analize jedne logičko-prekidačke mreže jeste da se, za zadate ulazne komande i podatke, odrede vrednosti izlaznih funkcija koje mreža realizuje. Postupak analize izvodi se tako što se postupno određuju veličine na izlazima svih logičkih kola i memorijskih elemenata u posmatranom trenutku, krećući se od ulaza prema izlazima mreže. Za to je poželjno numerisati izlaze iz pojedinih elemenata i podskupova mreže.

Pri sintezi - projektovanju logičko-prekidačke mreže, tj. digitalnog uređaja, treba najpre definisati blok-šemu uređaja, zadatke pojedinih blokova, kapacitete memorija pojedinih blokova i izvršiti izbor logičkih i memorijskih elemenata za realizaciju uređaja.

Digitalnim uređajima se obrađuju diskretne informacije. Zadatak koji digitalni uređaj treba da uzvrši mora biti izražen u obliku redosledno-sekvencijalnog niza aritmetičkih operacija koji se naziva *algoritam*. To znači da se svaki problem čije je rešenje moguće izraziti u obliku algoritma može realizovati pomoću digitalnog uređaja. Redosledni niz elementarnih operacija, instrukcija ili naredbi koje uređaj treba da izvrši da bi tehnički realizovao neki algoritam naziva se *program*.

U terminologiji digitalnih računara često se sreću termini *hardver* i *softver*, za koje ne postoje adekvatni nazivi u našem jeziku. Pod rečju *hardver* (engl. Hardware), podrazumeva se sve materijalno od čega je računar napravljen. To su sve mehaničke, elektromehaničke i elektronske komponente, delovi i sklopovi od kojih se računar sastoji, sve ono što je čvrsto i trajno izvedeno i što se ne može menjati. Pod rečju *softver* (engl. Software) podrazumeva se skup svih programa nekog računara. Programi i podaci mogu biti ugrađeni u računar, a mogu se nalaziti i van njega, na magnetnoj traci ili disku.

Pojava integrisanih kola izmenila je značenje pojma elementarne komponente, kao tranzistor, dioda, otpornik, logičko kolo, memorijski element i slično. U tehnici integrisanih kola veliki broj elementarnih komponenata se povezuje u složenija kola i stavlja na veoma male pločice i kapsulira u jedinstvene celine - *čipove*.

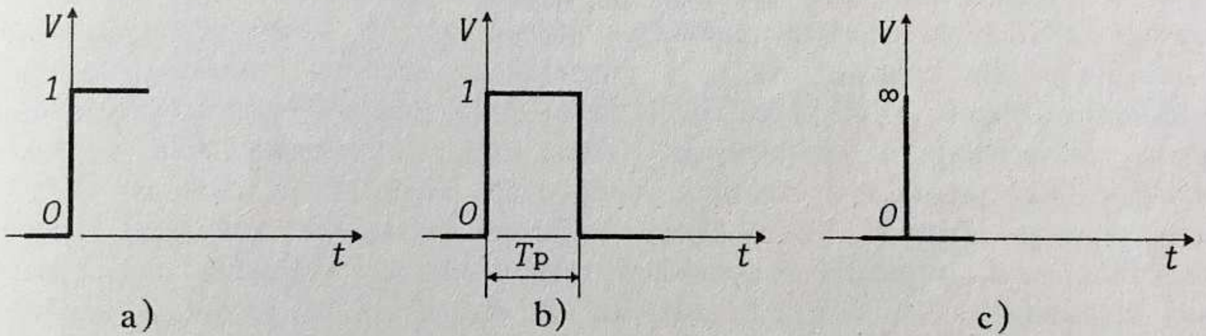
Danas se u tehnici integrisanih kola realizuju čitave kombinacione i sekvencijalne mreže, memorijski blokovi, aritmetičke jedinice, takt-generatori, kontroleri, A/D i D/A konvertori, kao i kola za specijalne namene. Iako ovakva integrisana pakovanja sadrže na hiljade elementarnih komponenata, danas se tretiraju kao "elementarne" komponente ili, bolje rečeno, funkcionalni blokovi, koji se koriste za gradnju digitalnih uređaja i sistema. Svedoci smo masovne primene mikroprocesora koji su postali univerzalni elektronski funkcionalni blokovi i koji su omogućili bitno poboljšanje nivoa automatizacije i ugradnju "inteligencije" u raznovrsne digitalne uređaje. Tako su mikroprocesori danas praktično nezamenljivi u gradnji telekomunikacionih uređaja i sistema, u automatizovanim uređajima za upravljanje procesima, biomedicinskim uređajima, u upravljačkim uređajima u avionima, brodovima i automobilima, u automatskim mernim uređajima, u uređajima široke potrošnje itd. Fizičku ugradnju mikroprocesora kataloški je propisao proizvođač komponenata, a različitost njihove primene se određuje načinom programiranja.

Od pojave prvog tranzistora 1948. godine, do pojave prvog mikroprocesora *Intel* 4004 1971. godine protekao je relativno kratak period. Dakle, skupom pomenutih komercijalno raspoloživih funkcionalnih blokova može se dobiti i čitav uređaj za obradu numeričkih podataka na površini jedne štampane ploče veličine manje od $20 \times 20\ cm^2$. Takav uređaj se naziva mikroračunar.

1.3.1. TIPIČNI OBLICI IMPULSA

Reč "impuls" je latinskog porekla i označava udarac, podsticaj, a u elektrotehnici kratak udar energije, odnosno naglu promenu amplitude napona ili struje. Pojam impulsa je u elektrotehniku uveo S. Morze 1837. godine svojim patentom o telegrafskom sistemu za prenos informacija.

Celokupni sistem digitalne obrade podataka zasniva se na primeni impulsa. Njima se, pre svega, predstavljaju brojne vrednosti podataka u nekom pogodnom brojnom sistemu. Sa tako predstavljenim podacima-brojevima, obavljaju se potrebne računske operacije. Ovi podaci, kao i dobijeni rezultati, čuvaju se duže vreme takođe u impulsnom obliku. Sve instrukcije u vezi s obradom, kontrolom i sinhronizacijom procesa pri obradi podataka izvode se pomoću impulsa. Prema tome, i digitalna kola su u krajnjoj liniji impulsna kola sa specijalnom namenom.

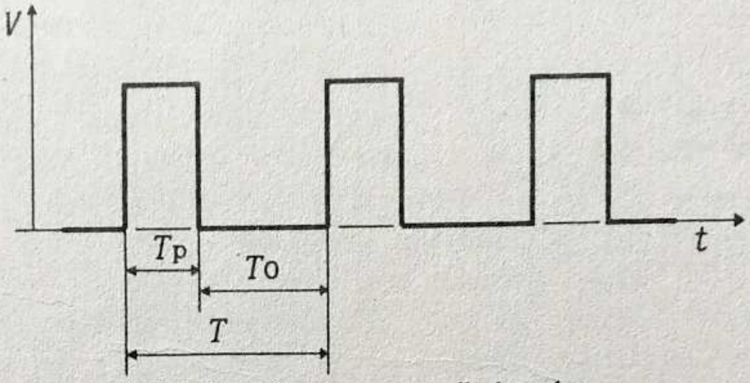


Sl. 1.3. Idealizovani oblici tipičnih impulsa

Svaki oblik struje ili napona koji nije sinusoidnog karaktera može se na određeni način svrstati u impulsne pojave. Jedna od osnovnih karakteristika impulsa jeste njegov oblik. U analizi elektičnih kola i prenosnih sistema najčešće se koriste teorijski idealizovani oblici impulsa. Impuls na slici 1.3.a) ima trenutnu promenu amplitude i poznat je kao Hevisajdova funkcija, odskočna funkcija ili step-impuls. Ako se amplituda step-impulsa označi sa 1, ovakav se impuls naziva i jedinični.

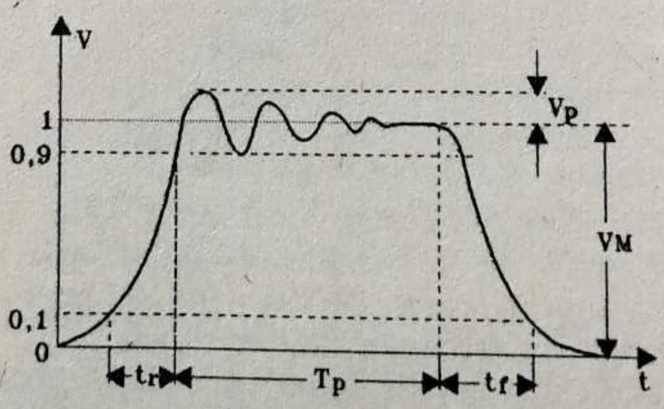
Kada se odskočnoj funciji posle vremena T_p doda još jedna takva funkcija s negativnim predznakom, dobija se idealan pravougaoni impuls kao na slici 1.3.b). Promena amplitude takvog impulsa je trenutna, pa je širina pravougaonog impulsa T_p precizno određena. Specijalni oblik idealnog pravougaonog impulsa prikazan je na slici 1.3.c). To je potpuno teorijski impuls, koji bi postojao samo u jednom trenutku, i to s beskonačno velikom amplitudom. Ovakav impuls je poznat kao Dirakova funkcija ili delta-impuls. Dobija se od idealnog pravougaonog impulsa tako što se njegovo trajanje sve više smanjuje, a površina impulsa ostaje nepromenjena.

Ako se pojedinačni pravougaoni impulsi periodično ponavljaju, onda nastaje povorka impulsa kao na slici 1.4. Učestanost ponavljanja ovakvog impulsa je $f = \frac{1}{T} = \frac{1}{T_p + T_o}$, gde je T_p trajanje impulsa, T_o pauza između impulsa, a T perioda ponavljanja impulsa. U slučaju kada je $T_p = T_o$, pravougaoni impulsi postaju kvadratni, jer je kod njih trajanje impulsa i pauze podjednako.



Sl. 1.4. Povorka pravougaonih impulsa

U realnim uslovima idealni oblik pravougaonog impulsa dat na slici 1.3.b) praktično je neostvarljiv. Pre svega, idealni impuls pretpostavlja mogućnost trenutne promene napona, što znači da se bočne ivice impulsa formiraju u vremenskom intervalu čije je trajanje jednako nuli. U stvarnosti ovakve diskontinualne promene napona ne postoje. Naime, svaka promena napona, ma kako bila brza, dešava se u vremenskom intervalu konačnog trajanja, jer aktivni elementi u elektronskim kolima imaju konačno vreme prekidanja, odnosno uključivanja i isključivanja. Osim toga, elektronska kola u većoj ili manjoj meri poseduju i reaktivne komponente koje takođe utiču na formiranje stvarnog impulsa. Podrobnijom analizom vremenske zavisnosti bočnih ivica uviđa se da promena napona kod realnih impulsa odgovara eksponencijalnoj zakonitosti. Na slici 1.5 prikazan je stvarni oblik pravougaonog impulsa. Za iznalaženje trajanja bočnih ivica impulsa trebalo bi odrediti trenutke kada napon dostiže nultu, odnosno konačnu vrednost. Zbog velike zakrivljenosti napona u tim ekstremnim tačkama praktično je nemoguće odrediti pomenute trenutke s pouzdanom tačnošću. Zbog toga se vreme uspona prednje ivice impulsa t_r definiše kao vreme potrebno da se napon impulsa promeni od 10 - 90 % njegove amplitude. Analogno tome, definiše se i vreme opadanja impulsa t_f tokom koga napon opadne sa 90 na 10 % amplitude impulsa.



Sl. 1.5. Realni oblik pravougaonog impulsa

Navedeni parametri t_t i t_f karakterišu prelazna stanja impulsa. Kod stvarnih impulsa parametar T_p, tj. trajanje impulsa, definiše se na razne načine. Kod pravougaonog impulsa T_p se meri ili pri 50 % ili pri 90 % amplitude impulsa. Gornja ivica nije uvek ravna kao što bi to trebalo da bude u idealnom slučaju. Pri prelazu sa prednje na gornju ivicu impulsa ispoljava se tendencija oscilovanja. Ova pojava se naziva premašenje i u nekim primenama je čak i poželjna. Premašenje se izražava odnosom amplitude prve oscilacije V_p i amplitude impulsa V_m, tj. $p = \frac{V_p}{V_m}\,100(\%)$.

PITANJA I ZADACI

1. Šta su digitalni uređaji i odakle potiče njihov naziv?
2. Šta je greška diskretizacije?
3. Kako se definiše jedinica za merenje količine informacija?
4. Koje se vrste informacija razlikuju kod digitalnih računara?
5. Šta je algoritam, a šta program?
6. Šta je hardver, a šta softver?
7. Nacrtati i objasniti stvarni oblik pravougaonog impulsa.

BROJNI SISTEMI

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. U opštem slučaju, za osnovu brojnog sistema može da se uzme bilo koji broj veći od 1. Tako su, pored decimalnog brojnog sistema s osnovom 10, koji predstavlja "prirodni" brojni sistem za čoveka, poznati i trinarni sistem s osnovom 3, oktalni s osnovom 8, heksadecimalni s osnovom 16 i drugi. U uvodu je rečeno da je u digitalnoj tehnici najpogodniji za primenu binarni brojni sistem s osnovom 2 koji predstavlja "prirodni" jezik računara.

U bilo kom brojnom sistemu s osnovom B može se od n cifara napisati B^n brojeva. Npr., u decimalnom sistemu sa četiri cifre može se napisati $10^4 = 10\,000$ različitih brojeva, dok ih je u binarnom sistemu mnogo manje, tj. $2^4 = 16$. Očigledno je, dakle, da binarni sistem za isti broj zahteva znatno više cifarskih mesta od decimalnog broja. Bez obzira na ovaj nedostatak, binarni brojni sistem je najpovoljniji za korišćenje u digitalnim uređajima zbog jednostavnosti tehničke realizacije i pouzdanosti, jer je dovoljno razlikovati samo dva stanja koja se predstavljaju naponskim ili strujnim nivoom. Ako bi se koristio decimalni brojni sistem, elektronska kola bi morala precizno da razlikuju deset nivoa, što se može postići brižljivim projektovanjem, ali bi uređaji bili veoma skupi i složeni, a mogućnost grešaka vrlo velika. Može se reći da nijedan brojni sistem nema apsolutne prednosti nad ostalim sistemima. Zbog toga su i razvijeni različiti brojni sistemi. Svaki od njih se koristi u specifičnim uslovima, kada je u izvesnoj prednosti u odnosu na druge sisteme.

2.1. BINARNI BROJNI SISTEM

Decimalni brojni sistem ima deset različitih cifara 0,1,2,...,9. Poznato je da u nekom višecifrenom decimalnom broju prva cifra zdesna ima vrednost ili "težinu" jedinica. To se može napisati kao $a \cdot 10^0$, gde je a decimalna cifra koja označava broj jedinica. Druga cifra ima "težinu" desetica ili $b \cdot 10^1$, treća ima težinu stotina ili $c \cdot 10^2$ itd. Na primer, kod broja 1989, cifra 9 znači da ovaj broj ima devet jedinica ili $9 \cdot 10^0$. Sledeća cifra označava osam desetica ili $8 \cdot 10^1$, itd., tako da se pomenuti broj može napisati u sledećem obliku:

$$1989 = 1 \cdot 10^3 + 9 \cdot 10^2 + 8 \cdot 10^1 + 9 \cdot 10^0 = 1000 + 900 + 80 + 9$$

Ako se ovaj postupak uopšti, onda se bilo koji ceo broj u decimalnom brojnom sistemu može napisti na sledeći način:

$$A = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 \qquad (2\text{-}1)$$

Koeficijenti $a_0, a_1, \dots, a_n$ uzimaju vrednosti iz skupa $0, 1, 2, \dots, 9$.

Na isti način za razlomljeni decimalni broj 19,74 važi:

$$19{,}74 = 1 \cdot 10^1 + 9 \cdot 10^0 + 7 \cdot 10^{-1} + 4 \cdot 10^{-2} = 10 + 9 + 7/10 + 4/100$$

U opštem slučaju za neki broj sa n celih i m razlomljnih mesta važi:

$$A = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \dots a_1 b^1 + a_0 b^0 + a_{-1}b^{-1} + a_{-2}b^{-2} + \dots + a_{-m}b^{-m} = \sum_{i=-m}^{i=n-1} a_1 \cdot b_1 \qquad (2\text{-}2)$$

gde a_i označava brojnu vrednost cifre na nekoj poziciji, a b_i predstavlja osnovu ili bazu brojnog sistema. Stepenovanjem broja b u svakom sabirku dobija se poziciona vrednost, 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 vrednost određena mestom te cifre u odnosu na pozicioni zarez.

Binarni brojni sistem, kao i decimalni, spada u težinske pozicione brojne sisteme. On ima osnovu 2 i samo dva cifarska simbola 0 i 1. Po analogiji sa (2-1), za binarni brojni sistem važi:

$$A = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0 \qquad (2\text{-}3)$$

Svaki član u ovom nizu ima težinu dvostruku od prethodnog člana. Koeficijenti $a_0, a_1 \dots, a_n$ imaju vrednosti 0 ili 1 i predstavljaju cifre binarnog broja.

Na primer, broj 14 u binarnom sistemu glasi:

$$(14)_{10} = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = (1110)_2$$

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

$$(1{,}8125)_{10} = 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 1{,}1101$$

2.2. OKTALNI I HEKSADECIMALNI BROJNI SISTEM

Za binarnu predstavu brojeva potrebni su dugački nizovi nula i jedinica, nepreglednih za korisnika i teških za manipulaciju uz veliku mogućnost pogrešne interpretacije. Čoveku je najprirodnije da radi sa decimalnim brojevima, ali je pretvaranje binarnog broja u decimalni i obrnuto mukotrpan postupak i najčešće se ne može brzo izvesti. Međutim, ako se binarne cifre grupišu po tri, četiri (ili više), i svaka grupa izrazi u nekom sistemu sa višom osnovom brojanja, onda se dobija predstava brojeva koji su kraći za pisanje i jednostavniji za memorisanje. Na primer, oktalni i heksadecimalni brojni sistemi koriste se za prikazivanje sadržaja računara na kontrolnim konzolama, kao i za sažeto zapisivanje binarnih brojeva radi lakšeg pamćenja i pisanja programa, tj. kao neka vrsta stenografije za binarne brojeve. To je njihova najvažnija primena. Naravno, pre unošenja u računar, podatke treba vratiti u binarne brojeve.

Princip stvaranja oktalnog i heksadecimalnog brojnog sistema je potpuno jednak decimalnom i binarnom; zapravo, i oni spadaju u težinske pozicione kodove.

Pošto digitalni računari ne rade s otkalnim i heksadecimalnim brojevima računske operacije sa njima neće biti prikazane.

2.2.1. OKTALNI BROJNI SISTEM

Osnova ili baza oktalnog brojnog sistema je 8, a upotrebljavaju se cifre 0,1,2,3,4,5,6, i 7, uzete iz decimalnog brojnog sistema. Težine određene položajem levo od pozicionog zareza su $...,8^3,8^2,8^1,8^0$, a za brojeve manje od 1, tj. desno od pozicionog zareza su $8^{-1},8^{-2},8^{-3},...$ Na primer:

$$24057_{(8)} = 2\cdot 8^4 + 4\cdot 8^3 + 0\cdot 8^2 + 5\cdot 8^1 + 7\cdot 8^0 = 2\cdot 4096 + 4\cdot 512 + 0\cdot 64 + 5\cdot 8 + 7\cdot 1$$

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

2.2.2 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. Težinske vrednosti su određene pozicijom cifre i iznose $...16^3,16^2,16^1,16^0,16^{-1},16^{-2},16^{-3},...$ U ovom sistemu se upotrebljava 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 vrednosti:

$$A = 10,\quad B = 11,\quad C = 12,\quad D = 13,\quad E = 14,\quad F = 15$$

Ova slova se uvode da bi se izbegle dvocifarske oznake koje otežavaju pisanje. Sledeći primer pokazuje kako se decimalni broj 8090 predstavlja u heksadecimalnom sistemu:

$$1F9A = 1\cdot16^3 + F\cdot16^2 + 9\cdot16^1 + A\cdot16^0 = 1\cdot16^3 + 15\cdot16^2 + 9\cdot16^1 + 10\cdot16^0 = 8090_{(10)}$$

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 sistem kada on treba da obavlja potpuno istu funkciju kao i oktalni. Razlog tome je pogodnost primene heksadecimalnog sistema u radu sa binarnim brojevima. Jedan heksadecimalni broj se može prikazati s četiri bita, dva broja se mogu prikazati s osam bita, itd. Mikroprocesori rade sa binarnim brojevima upravo takvog formata, pa se heksadecimalni brojni sistem najviše upotrebljava u radu s mikroprocesorima. U tabeli 2.1 prikazani su ekvivalenti decimalnih brojeva do 16 u binarnom, oktalnom i heksadecimalnom brojnom sistemu.

DECIM.	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

2.3. KONVERZIJA BROJEVA IZ JEDNOG U DRUGI BROJNI SISTEM

Binarno predstavljanje brojeva je teško zamislivo kao deo 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.

2.3.1. KONVERZIJA DECIMALNIH BROJEVA U BINARNE I OBRNUTO

Prevođenje celih decimalnih brojeva u binarne vrši se metodom sukcesivnog deljenja. Metoda se sastoji u tome što se broj koji se konvertuje deli s bazom sistema, a ostatak deljenja posebno zapisuje. Konvertovani broj se dobija kada se napišu cifre ostataka obrnutim redom od onog kako su dobijene. Na primer, za decimalni broj 120 postupak prevođenja u binarni sistem će biti:

$$
\begin{array}{l}
120 : 2 \\
60 \;\ldots\ldots\; 0 \\
30 \;\ldots\ldots\; 0 \\
15 \;\ldots\ldots\; 0 \\
\;\;7 \;\ldots\ldots\; 1 \\
\;\;3 \;\ldots\ldots\; 1 \\
\;\;1 \;\ldots\ldots\; 1 \\
\;\;0 \;\ldots\ldots\; 1 \\
(120)_{10} = (1111000)_2
\end{array}
$$

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

Na primer:

$$
\begin{array}{lcl}
 & & \text{CELOBROJNI DEO} \\
0,375 \cdot 2 = 0,75 & & 0 \\
0,75 \;\; \cdot 2 = 1,5 & & 1 \\
0,5 \;\;\; \cdot 2 = 1,0 & & 1 \\
\hline
(0,375)_1 = (0,011)_2 & &
\end{array}
$$

Konverzija mešovitih decimalnih brojeva vrši se tako što se posebno konvertuje celobrojni deo, a posebno deo sa razlomljenim vrednostima, pa se dobijeni rezultati sabiraju. Ovde treba naglasiti da se celi brojevi mogu izraziti precizno u binarnoj formi, dok se konverzija brojeva koji nisu celi ne može uvek izvesti do kraja, što podrazumeva primenu aproksimacija. Ovo nikako ne znači da je binarni sistem manje tačan od decimalnog, već da on samo zahteva veći broj cifara za izražavanje određene veličine željenom preciznošću.

Za obrnuti postupak konverzije binarnih brojeva u decimalne može se koristiti direktno sumiranje članova prema jednačini (2-1), kao na primer:

$$
\begin{aligned}
(1101011,01)_{(2)} &= 1\cdot 2^6 + 1\cdot 2^5 + 0\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 \,,\; 0\cdot 2^{-1} + 1\cdot 2^{-2} = \\
&= 64 + 32 + 0 + 8 + 0 + 2 + 1,0 + 1/4 = (107,25)_{(10)}
\end{aligned}
$$

2.3.2. KONVERZIJA BINARNIH BROJEVA U OKTALNE I OBRNUTO

Pošto je $8 = 2^3$, to znači da za jedan jednocifreni oktalni broj treba tri bita. Prema tome, binarni brojevi se mogu podeliti u grupe po tri bita, počevši od pozicionog zareza. Svakoj takvoj grupi može se pripisati jedan oktalni broj.

Na primer: $110100101011_{(2)} = \underset{6}{110}\ \underset{4}{100}\ \underset{5}{101}\ \underset{3}{011}_{(2)} = 6453_{(8)}$

I oktalni broj se vrlo jednostavno pretvara u binarni.

Na primer: $70152_{(8)} = \underset{7}{111}\ \underset{0}{000}\ \underset{1}{001}\ \underset{5}{101}\ \underset{2}{010}_{(2)} = 111000001101010_{(2)}$

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

2.3.3. KONVERZIJA OKTALNIH BROJEVA U DECIMALNE I OBRNUTO

Kada je to potrebno, programeru je jednostavnije da oktalne brojeve pretvori u decimalne, izvrši željenu operaciju u decimalnom sistemu i rezultat ponovo pretvori u oktalni broj.

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

$$\begin{aligned} 1267_{(8)} = 7 \cdot 8^0 &= 7 \\ +6 \cdot 8^1 &= 48 \\ +2 \cdot 8^2 &= 128 \\ +1 \cdot 8^3 &= 512 \\ \hline & 695_{(10)} \end{aligned}$$

Dakle, $1267_{(8)} = 695_{(10)}$

Da se ova množenja ne bi uvek iznova radila, može se napraviti pomoćna tabela sa prikazanim proizvodima težinskih vrednosti 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 2.2

Obrnuti postupak se može izvesti metodom oduzimanja stepena 8, koji je najbliži zadatom decimalnom broju. Za to može poslužiti tabela 2.2. To se može ilustrovati na primeru pretvaranja decimalnih broja 643 u oktalni broj. U tabeli 2.2. najpre treba potražiti najbliži broj, ali manji od decimalnog. To je broj 512 koji se nalazi u koloni 3, što znači da će oktalni ekvivalent imati četiri cifre. Broj 512 se nalazi u vrsti 1, a to znači da je 1 odgovarajuća oktalna cifra na četvrtoj poziciji, odnosno 1, itd.

Ceo postupak se može prikazati na sledeći način:

$643_{(10)}$		(oktalno)	$1\ 2\ 0\ 3_{(8)}$
$-\ 512$			
131	=	$1\ 8^3$	
$-\ 128$			
3	=	$2\ 8^2$	
$-\ 0$			
3	=	$0\ 8^1$	
$-\ 3$			
0	=	$3\ 8^0$	

Dakle, $643_{(10)} = 1203_{(8)}$.

Decimalno-oktalno pretvaranje se može izvesti i metodom sukcesivnog deljenja decimalnog broja osnovom 8.

2.3.4. KONVERZIJA BINARNIH BROJEVA U HEKSADECIMALNE I OBRNUTO

Pošto je osnova heksadecimalnog sistema $16 = 2^4$, to se svaki binarni broj koji treba pretvoriti u heksadecimalni deli u grupe po četiri bita i svakoj grupi posebno dodeljuje odgovarajući heksadecimalni ekvivalent.

Na primer:

$$1001101000011111_{(2)} = \underset{9}{1001}\ \underset{A}{1010}\ \underset{1}{0001}\ \underset{F}{1111} = 9A1F_{(16)}$$

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

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

Na primer:

$$E6A2_{(16)} = \underset{1110}{E}\ \ \underset{0110}{6}\ \ \underset{1010}{A}\ \ \underset{0010}{2} = 1110011010100010_{(2)}$$

2.3.5. KONVERZIJA HEKSADECIMALNIH BROJEVA U DECIMALNE I OBRNUTO

Kao i u slučaju oktalnih brojeva, i ovde se može napraviti tabela s proizvodima odgovarajućih vrednosti heksadecimalnih cifara.

POZICIJA CIFRE				HEKSADECIMALNA CIFRA
3	2	1	0	
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 2.3

Kao ilustracija može poslužiti sledeći primer:

$1\ E\ 9\ B_{(16)}$

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 primeniti tabela, kao i za konverziju u oktalni sistem. Međutim, ovom prilikom se metoda sukcesivnog deljenja može ilustrovati na primeru pretvaranja decimalnog broja $3921_{(10)}$ u odgovarajući heksadecimalni broj:

CELOBROJNI DEO	OSTATAK
$3921_{(10)} : 16 = 245$	1
$245 : 16 = 15$	5
$15 : 16 = 0$	15

$= F\ 5\ 1_{(16)}$

2.4. OSNOVNE ARITMETIČKE OPERACIJE U BINARNOM SISTEMU

Aritmetičke operacije u binarnom brojnom sistemu obavljaju se po pravilima sličnim kao u decimalnom sistemu. Pošto binarni sistem ima osnovu 2, množenje sa dva se izvodi pomeranjem zareza udesno, dok se pri deljenju sa dva zarez pomera ulevo. Ako se posmatra binarni broj $1001,00 = 9_{10}$, i ako se zarez pomeri za jedno mesto udesno, biće:

$$10010,0 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16 + 0 + 0 + 2 + 0 = 18_{10},$$

što znači da je zaista izvršeno množenje sa dva. U istom primeru pomeranje zareza za jedno mesto ulevo daje:

$$100,1 = 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} = 4 + 0 + 0 + 1/2 = 4,5_{10}.$$

Ovim je dokazano da je izvršeno deljenje sa dva.

Osnovne aritmetičke operacije u binarnom brojnom sistemu obavljaju se prema pravilima za sabiranje, oduzimanje, množenje i deljenje, koja su navedena u sledećim tablicama:

TABLICA ZA SABIRANJE

$A + B = O$	
$0 + 0 = 0$	
$0 + 1 = 1$	
$1 + 0 = 1$	
$1 + 1 = 1$	i 1 za prenos

TABLICA ZA ODUZIMANJE

$A - B = D$	
$0 - 0 = 0$	
$0 - 1 = 1$	i 1 pozajmljeno
$1 - 0 = 1$	
$1 - 1 = 0$	

TABLICA ZA MNOŽENJE

$A \cdot B = P$
$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

TABLICA ZA DELJENJE

$A:B = Q$
$0:0 = ?$
$0:1 = 0$
$1:0 = ?$
$1:1 = 1$

Sledeći primeri će najbolje objasniti ove operacije:

$$\begin{array}{r} 11101,01 \\ +\quad 1001,101 \\ \hline 100110,111 \end{array} = \begin{array}{r} 29,25 \\ 9,625 \\ \hline 38,875 \end{array}$$

Oduzimanje se takođe obavlja po sličnom postupku kao u decimalnom brojnom sistemu:

$$\begin{array}{r} 01100 \\ -\ 00111 \\ \hline 00101 \end{array} = \begin{array}{r} 12 \\ -\ 7 \\ \hline 5 \end{array}$$

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

- pošto se u najnižem (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 vrednost $2^2 = 2^1 + 2^1$. Jedna cifra 1 sa težinom 2^1 se zadržava u drugom razredu, a preostalih $2^1 = 10$ se prenosi u prvi razred. Najzad, binarna razlika u prvom razredu 10-1 daje vrednost 1;
- sada u drugom razredu već postoji pozajmljena cifra 1 s 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 oduzimaju se na isti način:

$$\begin{array}{r} 1010,1 \\ 111,1 \\ \hline 011,0 \end{array}$$

Množenje binarnih brojeva prikazano je sledećim primerom:

$$\begin{array}{r} 101,1 \cdot 10,01 \\ \hline 1011 \\ 0000 \\ 0000 \\ 1011 \\ \hline 1100,011 \end{array}$$

Binarno deljenje je slično decimalnom, s tim što se i međuoperacije množenja i oduzimanja izvode, naravno, u binarnom sistemu, na primer:

$$(1010,1:111,1)_2 \Leftrightarrow (10,5:7,5)_{10}$$

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

$$\begin{array}{l} 10101 : 1111 = 1,011; \qquad 21_{10}:15_{10} = 1,4_{10} \\ \underline{-1111} \\ 0011000 \\ \quad \underline{-1111} \\ \quad 010010 \\ \qquad \underline{-1111} \\ \qquad 0011 \end{array}$$

Iz ovoga primera se vidi da je rezultat deljenja 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.

PITANJA I ZADACI

1. Konvertovati razlomljeni decimalni broj 72,125 u binarni.

Rešenje:

Konverzija decimalnih brojeva vrši se tako što se posebno konvertuje celobrojni deo, a posebno deo s razlomljenim vrednostima, pa se dobijeni rezultati saberu:

$$\begin{array}{rl} 72 & : 2 \\ 36 & \ldots\ldots 0 \\ 18 & \ldots\ldots 0 \\ 9 & \ldots\ldots 0 \\ 4 & \ldots\ldots 1 \\ 2 & \ldots\ldots 0 \\ 1 & \ldots\ldots 0 \\ 0 & \ldots\ldots 1 \\ \hline \multicolumn{2}{l}{(72)_{10}=(1001000)_2} \end{array} \qquad \begin{array}{l} 0,125 \cdot 2 = 0,25 \ \ldots\ldots 0 \\ 0,25 \ \cdot 2 = 0,5 \ \ \ldots\ldots 0 \\ 0,5 \ \ \cdot 2 = 1 \qquad \ldots\ldots 1 \\ \hline (0,125)_{10} = (0,001)_2 \end{array}$$

Konačno je : $(72{,}125)_{10} = (1001000{,}001)_2$

2. Izvršiti oduzimanje sledećih binarnih brojeva: 100010 i 1010.

Rešenje:

$$\begin{array}{r} 10010 \\ -\ 1010 \\ \hline 1000 \end{array} \qquad \begin{array}{r} 18 \\ -\ 10 \\ \hline 8 \end{array}$$

3. Ako se paran decimalni broj konvertuje u binarni brojni sistem, kakva mora biti cifra najnižeg razreda?

Rešenje:

Težina cifre najnižeg razreda u binarnom brojnom sistemu je 1 (neparan broj), dok su težine svih ostalih cifara parne, jednačina (2-3). Stoga, ako se u binarnom sistemu nalazi cifra najnižeg razreda tj. 1, decimalna vrednost tog broja je uvek neparna, u protivnom je parna.

4. Koliko je cifara potrebno da se decimalni broj 18 izrazi u binarnom brojnom sistemu?

5. Koju vrednost ima cifra najveće težine (razreda) u binarnim brojevima od pet cifara koji odgovaraju decimalnim brojevima između 16 i 31?

6. Oktalni broj $(37501)_8$ 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.

256:2 = 128 0
128:2 = 64 0
64:2 = 32 0
32:2 = 16 0

GLAVA III

KODOVI

Kodovanje je način predstavljanja informacija pomoću simbola koji su elementi skupa. Razni nacionalni jezici čine sredstvo za kodovanje informacija u ljudskom društvu. Pisani tekst je jedan od načina kodovanja i memorisanja reči i pojmova. Reči su razložene na glasove, pa je svakom glasu pripisan odgovarajući azbučni simbol. Tako dobijeni skup simbola (azbuka) jeste kôd za pisani način izražavanja informacija. Grupa simbola kojom se predstavlja - koduje neki pojam naziva se *kodna reč*, a broj simbola u kodnoj reči je dužina te reči. Ako su sve reči jednog koda iste dužine, onda se radi o tzv. ravnomernom kodu. Ukoliko neki kôd sadrži simbole koji ne označavaju nikakvu, ili nikakvu novu informaciju, onda se za njega kaže da je redundantan. Primer za ovo je označavanje glasova sa dva znaka (lj, nj, dž).

Postupkom kodovanja može se bilo kakva diskretna informacija izraziti, tj. kodovati u obliku brojeva u bilo kom brojnom sistemu. Međutim, kodne kombinacije unutar jednog brojnog sistema mogu se praviti na razne načine, tj. mogu se formirati različiti kodovi.

3.1. BINARNI KODOVI

Binarni kodovi se mogu podeliti na *težinske i redosledne*. Kod prve vrste, kombinovanje binarnih simbola je izvršeno tako što se svakom od njih pripisuje određena težina. Primer težinskog koda je već opisani binarni brojni sistem. Međutim, kodne kombinacije se mogu praviti na različite načine, prema drugim težinskim zakonitostima. Tako je razvijeno mnoštvo binarnih kodova za specijalne namene. Između binarne kodne reči u težinskom kodu i decimalnog broja postoji sledeća veza:

$$A_{10} = \sum_{i=-m}^{i=n-1} a_i \cdot t_i$$

gde je a_i koeficijent koji može biti 0 ili 1, a t_1 je težinska vrednost binarnog mesta u kodnoj reči. Pri konstrukciji težinskih kodova mora se voditi računa da težinska vrednost nekog mesta ne bude za više od jedan veća od zbira težinskih vrednosti svih prethodnih binarnih mesta. U protivnom se svi brojevi neće moći predstaviti u tom kodu. Na primer, u prirodnom binarnom kodu težinska vrednost četvrtog mesta $a_4 = 8$ je za jedan veća od zbira težinskih vrednosti prethodna tri mesta. Ako bi a_4 bilo 9, onda broj 8 ne bi mogao biti predstavljen.

Svi kodovi koji nisu težinski spadaju u grupu redoslednih kodova. Ovde binarne cifre nemaju odgovarajuću težinu, već je bitan samo njihov redosled u kodnoj reči. Kod ovakvih kodova veza između decimalnog broja, odnosno cifre i binarne kodne reči data je ili preko složenih matematičkih izraza ili, najčešće, preko kodnih tabela.

3.2. BCD KODOVI SA TETRADAMA

Da bi se ukomponovale dobre osobine binarnog i decimalnog sistema i izašlo u susret ljudskoj navici da se misli decimalno, razvijeni su *binarno kodovani decimalni* ili, skraćeno, BCD kodovi.

BINARNA NOTACIJA				PRIRODNI BINARNI KOD	8-4-2-1 KOD	AJKENOV KOD	KOD VIŠE 3	4-2-2-1 KOD	5-4-2-1 KOD
4	3	2	1						
0	0	0	0	0	0	0		0	0
0	0	0	1	1	1	1		1	1
0	0	1	0	2	2	2		2	2
0	0	1	1	3	3	3	0	3	3
0	1	0	0	4	4	4	1		4
0	1	0	1	5	5		2		
0	1	1	0	6	6		3	4	
0	1	1	1	7	7		4	5	
1	0	0	0	8	8		5		5
1	0	0	1	9	9		6		6
1	0	1	0	10			7		7
1	0	1	1	11		5	8		8
1	1	0	0	12		6	9	6	9
1	1	0	1	13		7		7	
1	1	1	0	14		8		8	
1	1	1	1	15		9		9	

Tabela 3.1

Kod ovih sistema se odvojeno koduje svaka cifra decimalnog broja koristeći grupu od 4 bita za jednu decimalnu cifru. Raspored binarnih cifara u ovim grupama se kreće prema nekom utvrđenom zakonu. Da bi se moglo kodovati svih 10 decimalnih cifarskih simbola, moraju se upotrebiti najmanje 4 bi-

Navedene skraćenice predstavljaju komandne oznake za teleprinterske pisaće mašine. Na primer, STX (engl. Start Of Text) označava početak teksta, EOT (engl. End Of Transmission) označava kraj prenosa itd. Ove komandne oznake se, normalno, ne upisuju u tekst. Na primer, simbolu # odgovara kodna reč:

$$b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1 = 0100011$$

Očigledno je da je ovo BCD kôd sa 7 bita po kodnoj reči.

B I T I	0	0	0	0	1	1	1	1	b_7
	0	0	1	1	0	0	1	1	b_6
b_4 b_3 b_2 b_1	0	1	0	1	0	1	0	1	b_5
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	\	1			
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 3.5.

3.7. KONTROLA KODOVANJA

Pri prenosu binarno kodovanih informacija preko ma kojeg prenosnog kanala može se sa sigurnošću računati na izvesne smetnje. Otuda uvek postoji verovatnoća da će se određeni broj bita kodne grupe registrovati sa greškom. Uzročnici smetnji na prenosnom putu mogu biti veoma različiti (atmosferske smetnje, termički šumovi, preslušavanje itd.) i na njih se uglavnom ne može uticati. Stoga se pri projektovanju digitalnih sistema posvećuje velika pažnja kontroli kodovanih podataka, tj. zaštiti od smetnji, kao i detekciji i korekciji grešaka koje one prouzrokuju.

Ako se greške posmatraju statistički, onda se one mogu podeliti na *slučajne*, kada se pojavljuju usamljeno, bilo gde u grupi kodnih reči i *grupne*, kada je grupa grešaka zgusnuta u određenom segmentu. Verovatnoća pojavljivanja greške p zavisi od nivoa signala, vrste prenosnog kanala, nivoa smetnji, korišćene frekvencije itd. U nekim primenama, kao što je prenos faksimila, informacija je sama po sebi toliko redundantna da ove greške uopšte nisu kritične. Međutim, kod većine informacija ovo nije slučaj.

Detekcija greške na prijemnoj strani nikad nije sama sebi cilj, već je ona uvek povezana s nekim od načina njenog korigovanja. Najjednostavniji metod za detekciju grešaka sastoji se u ponavljanju pojedinih reči, određenih blokova reči ili čitavih informacija. Direktna i ponovljena informacija se na prijemnoj strani upoređuju i, na osnovu njihove eventualne razlike, utvrđuje se da li je pri prenosu nastala greška. Ovaj postupak je jednostavan i jeftin, ali se brzina prenosa smanjuje na polovinu prvobitne vrednosti. Stoga se mnogo češće koristi metoda detekcije grešaka pomoću visokoredundantnih kodova.

Što se tiče korekcije grešaka, u osnovi se mogu razlikovati dve metode. Prva od njih, koja se obično zove ARQ (engl. Automatic Repeat Request) zasniva se na ponavljanju pogrešno primljenih informacija, kada se to zatraži od otpremne strane posebnim, povratnim kanalom. Druga metoda koristi visokoredundantne kodove pomoću kojih se određuje lokacija jednog ili više pogrešno primljenih bita. Dekoder zatim automatski koriguje greške. Ovde ne postoji povratni kanal, a koder i dekoder su mnogo kompleksniji nego u prethodnom slučaju. Kod ovog sistema koji se zove FEC (engl. Forward Error Control), dekoder prosleđuje nepromenjene, dakle pogrešne podatke korisniku kada nije u stanju da locira grešku.

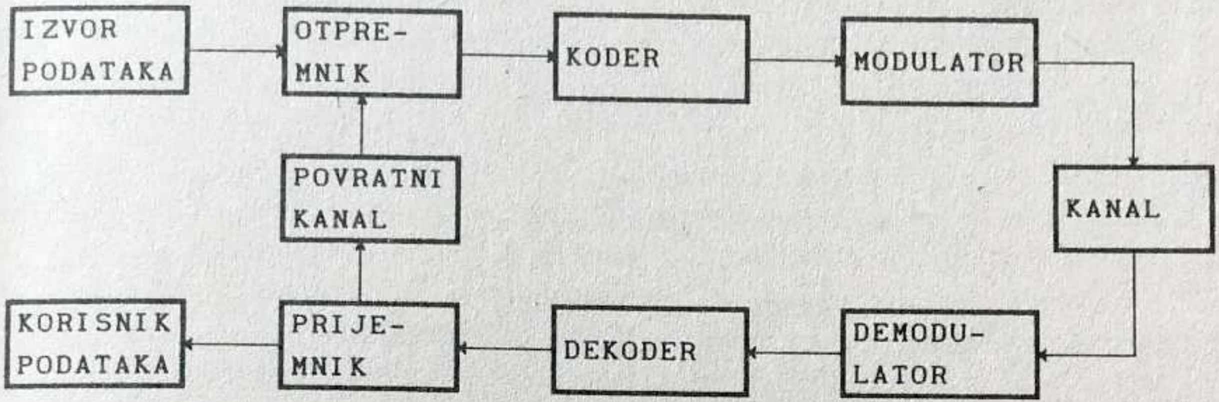
Nažalost, i kod jednog i kod drugog sistema smanjuje se brzina prenosa, utoliko više, ukoliko je mogućnost korekcije grešaka, odnosno stepen pouzdanosti veći. Jasno je da uzrok leži u tome što se jedan deo vremena raspoloživog za prenos mora rezervisati bilo za ponavljanje pogrešno primljenih informacija, bilo za redundantni (zaštitni) deo koda.

3.7.1 KOREKCIJA GREŠAKA PONAVLJANJEM INFORMACIJA

U ovom sistemu se informacije na otpremnoj strani podele u grupe blokova kodnih reči. Svakom bloku se zatim dodaju kontrolni biti pomoću kojih se greška na prijemnoj strani može detektovati. Ako je blok kodnih reči primljen korektno, prijemnik to konstatuje i signalizira predajniku preko povratnog kanala. Međutim, ukoliko je pri prenosu nastala greška, prijemnik detektuje njeno postojanje i preko povratnog kanala se traži ponavljanje celog bloka. Ovo, naravno, podrazumeva da se blok kodnih reči na otpremnoj strani memoriše sve dok se ne dobije obaveštenje da li treba njegovu otpremu ponoviti. Na slici 3.1 data je uopštena blok–šema ARQ sistema.

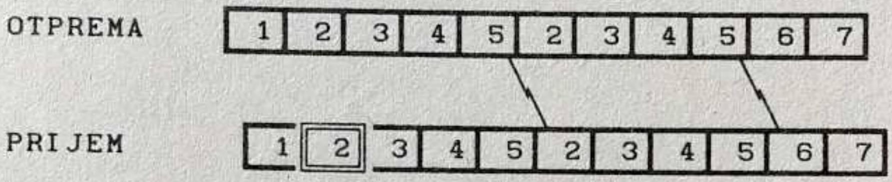
ARQ sistemi se mogu podeliti na dve grupe. Prva grupa je jednostavnija i češće korišćena. Kod nje posle otpreme bloka kodnih reči otpremnik čeka na pozitivno ili negativno obaveštenje od prijemnog kraja pre nego što otpremi sledeći blok ili ga ponovi (engl. Stop and Wait ARQ).

Druga grupa se naziva kontinualni ARQ sistem. Ovde otpremna strana posle emitovanja bloka ne čeka na obaveštenje 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 posle njega. Slika 3.2 ilustruje ovaj princip. U bloku 2 je detektovana greška.



Sl. 3.1. Uopštena blok-šema ARQ sistema

Međutim, dok je otpremna strana primila informaciju o ovome, emitovala je u međuvremenu 3,4 i 5. Zbog toga se ponavlja prenos blokova 2-5.



Sl. 3.2. Ilustacija rada kontinualnog ARQ sistema

Prednost ARQ postupka je u njegovoj jednostavnosti, a nedostatak je potreba za povratnim kanalom.

Razvijeni su takođe i hibridni sistemi u kojima se koristi kombinacija ARQ postupaka i kodova za korekciju grešaka s ciljem da se smanji učestanost ponavljanja blokova kodnih reči.

3.7.2. KODOVI ZA DETEKCIJU I KOREKCIJU GREŠAKA

U svakom realnom digitalnom sistemu postoji verovatnoća da će se određeni broj bita kodne reči registrovati s greškom. Zbog toga su razvijeni kodovi čijom se primenom može detektovati greška na prijemnoj strani, ili, čak, detektovati i korigovati. To su tzv. *zaštitni kodovi*.

Za kodovanje 8 različitih vrednosti nekog signala u prirodnom binarnom kodu potrebna su najmanje 3 bita ($2^3 = 8$). Ako se bilo koji bit jedne kodne grupe ovog koda registruje pogrešno, za uređaje će to biti ispravan rezultat, jer i tačna i pogrešna kodna grupa postoje kao mogući digitalni ekvivalenti pomenutog signala.

Kodne grupe ili kodne reči u prirodnom binarnom kodu se međusobno razlikuju u najmanje jednoj bit poziciji. Ova minimalna razlika se zove *distanca* kodnih reči. Na primer, kodne reči 0001 i 0011 se međusobno razlikuju samo u bitu druge pozicione vrednosti. Kada je, kao u prethodnom primeru, distanca kodnih reči jedinica, greška ne može biti detektovana, jer su sve kombinacije koje se mogu napraviti sa 4 bita iskorišćene za kodovanje. Da bi se omogućilo detektovanje greške nastale usled pogrešno registrovanog digita, minimalna distanca između kodnih reči treba da bude bar dva. To pak, zahteva povećanje broja bita u kodnoj grupi, iako je broj tih grupa ostao isti. Ako se za kodovanje 8 različitih vrednosti iz prethodnog primera upotrebe 4 digita, onda se minimalna distanca između kodnih reči može povećati na dva, jer polovina mogućih kombinacija ($2^4 = 16$) nije iskorišćena. Tabela 3.6 ilustruje ovaj primer. Iz nje se može videti da se bilo koje dve kodne reči razlikuju međusobno u najmanje dve bit pozicije.

	a_3	a_2	a_1	a_0
0	0	0	0	0
1	1	0	1	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 3.6

Dođe li do pogrešnog prijema, odnosno registrovanja jednog digita u nekoj kodnoj grupi, dobiće se kodna kombinacija (kodna reč) koja nije upotrebljena za kodovanje, što uređaj može da registruje kao grešku. Na primer, ako je kodna kombinacija 1010 koja odgovara vrednosti signala 2 (videti tabelu), primljena nekorektno kao 1011, prijemnik će detektovati grešku, jer reč 1011 nije korišćena za kodovanje. Ovde se pošlo od pretpostavke da je samo jedan bit pogrešno registrovan. Ukoliko to nije slučaj, već je broj pogrešno primljenih bita paran (dva ili četiri), tada greška ne može biti detektovana, jer je novonastala nekorektna kodna kombinacija korišćena u kodovanju.

Da bi se omogućilo da sistem detektuje i koriguje grešku, distanca između kodnih grupa mora se povećati na najmanje tri. Tada digitalni uređaj odabira kao korektnu onu kodnu reč koja je najbliža pogrešno primljenoj.

Osnovni nedostatak ovakvog načina kodovanja je da se brzina rada sistema, odnosno brzina prenosa, smanjuje. Ovo je sasvim razumljivo jer je jedan deo vremena rezervisan za dodatni bit, koji u suštini ne nosi informaciju, već služi za povećanje distance, čime se redukuje vreme raspoloživo za korisni signal.

Ovde treba napomenuti da je K. Šenon (Claude Shannon) svojom drugom teoremom o kodovanju dokazao da teorijski uvek postoji optimalan kôd koji omogućava prenos informacija uz neznatno smanjenje brzine prenosa, a da pri tome prosečna verovatnoća pogrešno primljenih informacija bude proizvoljno mala. Nažalost, takav konkretni optimalan zaštitni kôd do danas još nije pronađen.

3.7.2.1. KÔD SA KONTROLOM PARNOSTI

Najčešće korišćen kôd za povećanje distance između kodnih reči je kôd sa kontrolom parnosti. Za binarnu kodnu reč se kaže da je *parna* ako u sebi sadrži paran broj binarnih jedinica (1), i obrnuto, ako je broj binarnih jedinica neparan, onda je u pitanju *neparna* kodna grupa. Na primer, binarna reč 100110110 ima 5 bita čija je vrednost jedan, pa se za nju kaže da je neparna.

mogućih različitih binarnih reči iznosi 2^k. Jedna od njih mora biti rezervisana za pokazivanje postojanje greške, odnosno za saopštavanje da li je kodna reč primljena korektno. Ostalih 2^k-1 reči može se iskoristiti za definisanje pozicije pogrešnog bita. Odatle proizlazi da se mora zadovoljiti nejednakost:

$$2^k - 1 \geq n + k$$

PITANJA I ZADACI

1. Konvertovati decimalni broj 145 u BCD 8421 kod.
   Rešenje:
   Svaka cifra decimalnog broja prevodi se pojedinačno u četvorocifreni broj prema tabeli 3.2:

1	4	5
0001	0100	0101

2. Kako se definiše kodna reč?
3. Objasniti težinske i redosledne kodove.
4. Mogu li se pravila binarnog sabiranja uvek primeniti na BCD 8421 kôd?
5. Decimalne brojeve 0 i 11 napisati u "više 3" kodu.
6. Koju osobinu ima Grejov kôd?
7. Koje su najpoznatije metode za detekciju i korekciju grešaka?