**UNIVERZITET U BEOGRADU**

**SAOBRAĆAJNI FAKULTET**

**HEMINGOVO KODOVANJE**

**LINEARNI BLOK KODOVI**

Zadatak blok kodera je da prihvati izvestan broj (*k*) bita i da ih predstavi odgovarajućom kodnom reči dužine *n* bita. Pošto otkrivanje i eventualno ispravljanje grešaka zahtevaju unošenje redundanse, mora biti ispunjeno *n > k*.



Kodna reč se sastoji od:

* *k* informacionih bita, tipično označenih sa *i1*, *i2*, …, *ik*;
* *n-k* kontrolnih (zaštitnih) bita, tipično označenih sa *z1*, *z2*, …, *zn-k*.

Blok kod se označava sa (*n,k*), dok je veličina *R=k/n* u stvari *kodni količnik blok koda*. Na ulazu u blok kod je jedan od *2k* mogućih blokova od *k* informacionih simbola, a na ulazu je odgovarajuća kodna reč dužine *n*, koja je po nekom kriterijumu odabrana od *2k* mogućih “kandidata”.

Neka je dato konačno polje GF(*q*), tj. polje sa *q* simbola. Sekvence od po *n* elemenata polja čine vektorski prostor dimenzije *n* nad datim poljem. Linearni kod je podprostor vektorskog prostora nad GF(*q*).

Osobine LBK za *q*=2:

1. Kod mora da sadrži kombinacije “sve nule” i “sve jedinice” jer su to neutralni elementi u odnosu na sabiranje i množenje;
2. Zatvorenost u odnosu na sabiranje;
3. Zatvorenost u odnosu na množenje.

U nastavku će biti prikazan primer LBK kao i dekodera za Hemingove kodove (7,4) i (11,7). Biće prikazane i mogućnosti Hemingovih kodova u ispravljanju grešaka pri prenosu.

**Hemingov kod (7,4),** *n*=7, *k*=4

Posmatra se binarni zapis brojeva od 0 do 7:

|  |  |  |
| --- | --- | --- |
| 0 | 0 0 0 |  |
| 1 | 0 0 **1** | ***k1*** |
| 2 | 0 **1** 0 | ***k2*** |
| 3 | 0 1 1 | *I1* |
| 4 | **1** 0 0 | ***k3*** |
| 5 | 1 0 1 | *I2* |
| 6 | 1 1 0 | *I3* |
| 7 | 1 1 1 | *I4* |

gde su *kn*-kontrolni biti, a *In*-informacioni biti.

Prvi kontrolni bit *k1* se javlja na poziciji prve pojave jedinice u koloni sa trećim bitima svakog od brojeva, kontrolni bit *k2* javlja se na poziciji prve pojave jedinice u koloni sa drugim bitima svakog od brojeva, a kontrolni bit *k3* javlja se na poziciji prve pojave jedinice u koloni sa prvim bitima svakog od brojeva.

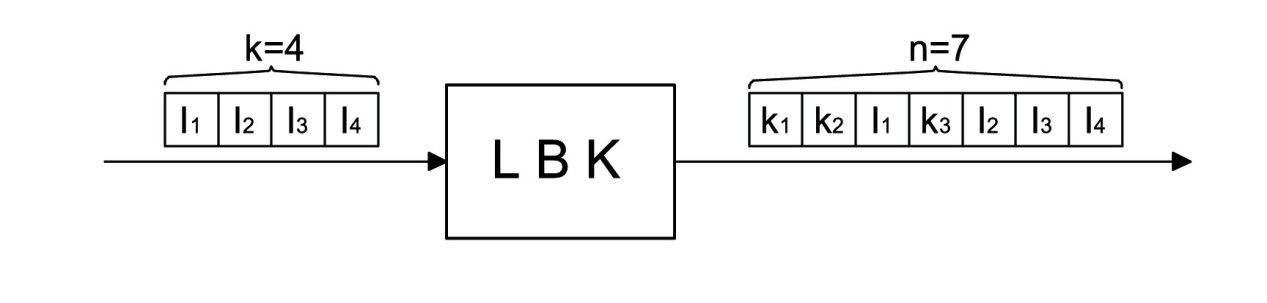
Informacioni biti *I1*, *I2*, *I3* i *I4* se javljaju redom na ostalim slobodnim bitima. Na osnovu ovakvog prikaza mogu se dobiti kontrolni biti preko informacionih tako što se za svaki kontrolni bit sabiraju po modulu 2 informacioni biti koji predstavljaju svaku sledeću pojavu jedinica u odgovarajućim kolonama svakog od brojeva.

*k1*= *I1**I2I4*

*k2*= *I1* *I3* *I4*

*k3*= *I2**I3* *I4*

Kodovanje Hemingovog koda (7,4) vrši se dovođenjem kodne reči, sastavljene od niza od četiri informaciona bita, na ulaz linearno-binarnog kodera, a na izlazu takvog kodera dobija se kodna reč od sedam bita koja sadrži i informacione i kontrolne bite. Primer takvog jednog kodera dat je na sledećoj slici:



Dakle, na izlazu iz kodera dobija se kodna reč oblika *c* = [ *k1* *k2**I1* *k3* *I2* *I3 I4*].

Ako se pretpostavi da je na ulaz u koder došla informaciona kodna reč tipa *I*=[1 1 0 0], gde je *I1*=1, *I2*=1, *I3*=0, *I4*=0 onda bi se na izlazu iz kodera dobio kod oblika *c* = [ *k1 k2* 1 *k3* 1 0 0]. Na osnovu datih informacionih bita, kao što je objašnjeno, mogu se odrediti kontrolni biti:

*k1*= *I1**I2I4* = 110= 0

*k2*= *I1* *I3* *I4*= 1 0 0= 1

*k3*= *I2**I3* *I4* = 10 0 = 1

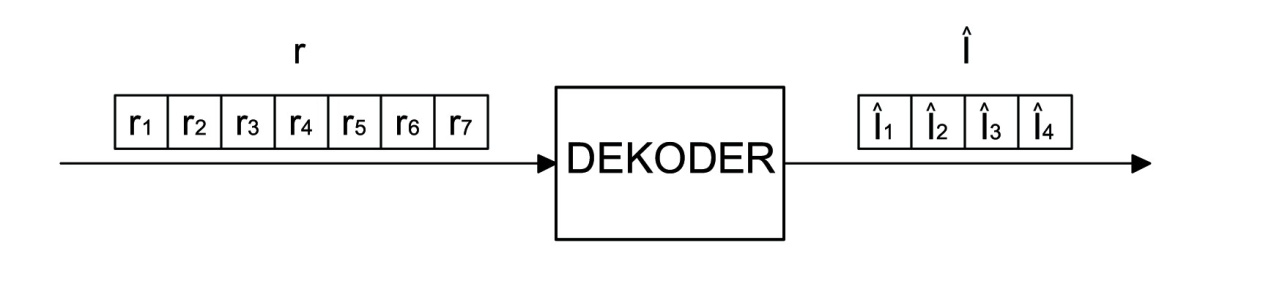
Na taj način dobija se kodna reč *c* = [ 0 11 1 1 0 0]. Simulacija ovakvog jednog kodera data je u prilogu.

Neka se sada pretpostavi da je pri prenosu koda došlo do greške na 6. bitu, odnosno u binarnom obliku bi se greška na šestom bitu prikazala kao *e* = [ 0 00 0 0 1 0]. Na mestu prijema bi smo onda dobili kodnu reč oblika:

*r* = *c* + *e* = [ 0 11 1 1 **1** 0]

pri čemu je *r1*=0, *r2*=1, *r3*=1, *r4*=1, *r5*=1, *r6*=1, *r7*=0.

Ovako, sa greškom, primljena reč ispravlja se u dekoderu. Dekodovanje Hemingovog koda (7,4) vrši se dovođenjem kodne reči od sedam bita sa greškom na ulaz u dekoder, a na izlazu iz dekodera dobija se kodna reč sastavljena od četiri informaciona bita.



Greška pri prenosu se ispravlja računanjem sindroma. Sindrom u binarnoj notaciji pokazuje poziciju pogrešnog bita, ako je došlo do jedne greške. Za Hemingov kod (7,4) sindromi se računaju na sledeći način:

*S1*= *r1 r3r5* *r7* = 0 11 0= 0

*S2*= *r2* *r3* *r6**r7*= 1 1 1 0= 1

*S3*= *r4* *r5* *r6**r7*= 1 1 1 0= 1

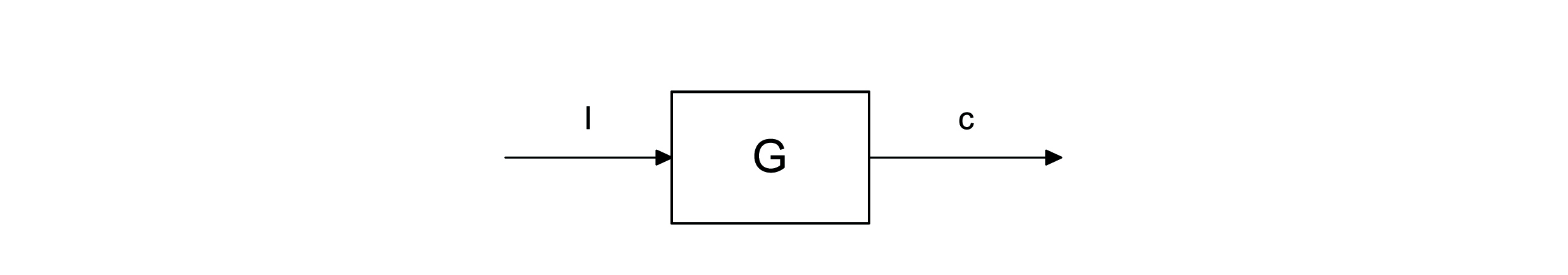
Sindrom *S1*se dobija tako što se saberu po modulu 2 svi oni biti iz kodne reči *r*, koji predstavljaju poziciju kontrolnog bita *k1* i svih bita koji predstavljaju poziciju informacionih bita od kojih se dobija kontrolni bit *k1*. Ovaj postupak se identično primenjuje i za sindrome *S2* i *S3*, s tim što se prate kontrolni biti *k2* i *k3*, respektivno. Pišući sada elemente sindroma *S*(*s3*,*s2*,*s1*) = *S*(1 1 0), što u binarnoj notaciji predstavlja broj 6, zaključuje se da se greška dogodila na šestoj poziciji i ispravlja se sabiranjem po modulu 2 pristiglog bita (*r6*) sa jedinicom, tj. njegovim komplementiranjem, tako da se sada dobija *r6*1 = 11 = 0. Na taj način se ispravila greška na šestom bitu i dobija se kodna reč *r* koja je jednaka kodnoj reči *c*:

*r* = *c* = [ 0 11 1 1 0 0]

Simulacija dekodera za Hemingov kod (7,4) data je u prilogu.

Poznato je da svaki prostor ima svoju bazu – skup linearno nezavisnih vektora čijim se linearnim kombinacijama (sabiranjem) dobijaju svi vektori prostora. Broj vektora u bazi je ravan dimenziji prostora. Za slučaj koda (*n,k*) dimenzija celog prostora je *n*, dok je dimenzija kodnog prostora *k*. Prema tome, jedan linearni kod može biti opisan svojom bazom – skupom linearno nezavisnih kodnih vektora koji “generišu” ceo kodni prostor. Ovi vektori se mogu složiti u obliku matrice dimenzija *kn*, te je njegova matrica dovoljna da opiše kod, dok bi u opštem slučaju – ako se kod zadaje tabelom – bilo potrebno napraviti (matricu) dimenzija *2kn*. Matrica se naziva ***generišuća matrica (generator matrix)*** i vrste su joj linearno nezavisne. Njen rang je upravio *k*. Poznato je da postoje linearne operacije nad vrstama pri kojima se rang matrice ne menja. Ovim operacijama se samo menja baza, ali kod ostaje isti. S druge strane, zamenom odgovarajućih kolona generišuće matrice dobija se generišuća matrica ekvivalentnog koda.

Kodna reč od sedam bita dobijena od četiri informaciona bita, može se dobiti i pomoću generišuće matrice G:



Odnosno:

*c* = *I* \* *G*

[ *k1 k2 I1 k3 I2 I3 I4*] = [ *I1 I2 I3 I4*]

*c* = [1 1 0 0] \* *G*

Generišuća matrica je dimenzija *k n*, jer su kodne reči *c* i *I* dimenzija *k* i *n*, respektivno. U prvoj koloni *G* matrice jedinice predstavljaju informacione bite od kojih je dobijen kontrolni bit *k1*. To važi i za kontrolne bite *k2* i *k3*, odnosno drugu i četvrtu kolonu, respektivno. U trećoj, petoj, šestoj i sedmoj koloni pojave jedinica predstavljaju pojavljivanje informacionih bita *I1*, *I2*, *I3* i *I4*, respektivno.

**HEMINGOVO RASTOJANJE, HEMINGOVA TEŽINA I SPEKTRI HEMINGOVIH KODOVA**

*Hemingovo rastojanje* je jednako broju pozicija gde se dve sekvence bita iste dužine razlikuju. Hemingovo rastojanje predstavlja broj jedinica u sekvenci grešaka koja jednu kodnu reč prevodi u drugu. *Minimalno Hemingovo rastojanje* je minimalan broj jedinica u sekvenci grešaka koji će dovesti do toga da greška ne bude otkrivena. Da bi kod mogao da detektuje *qe*grešaka, treba da bude:

***d*2*ec*+ 1,**

Da bi kod mogao da ispravi *ec* grešaka i detektuje *ed*grešaka treba da bude:

***dec*+ *ed*+ 1.**

Ako, na primer, uzmemo sve moguće sedmobitne kodne reči za Hemingov kod (7,4), simulacija za kodne reči (7,4) data je u prilogu:

***C1*= 0 0 0 0 0 0 0**

***C2* = 1 1 0 1 0 0 1**

***C3* = 0 1 0 1 0 1 0**

***C4* = 1 0 0 0 0 1 1**

***C5* = 1 0 0 1 1 0 0**

***C6* = 0 1 0 0 1 0 1**

***C7*= 1 1 0 0 1 1 0**

***C8* = 0 0 0 1 1 1 1**

***C9* = 1 1 1 0 0 0 0**

***C10* = 0 0 1 1 0 0 1**

***C11* = 1 0 1 1 0 1 0**

***C12* = 0 1 1 0 0 1 1**

***C13* = 0 1 1 1 1 0 0**

***C14* = 1 0 1 0 1 0 1**

***C15*= 0 0 1 0 1 1 0**

***C16* = 1 1 1 1 1 1 1**

ako uporedimo svaku reč sa bilo kojom drugom reči videćemo da se one razlikuju za 3 ili 4 bita, recimo kodna reč *c10* i *c11* se razlikuju za 3 bita , dok se kodne reči *c1* i *c2* razlikuju za 4 bita. Dakle, Hemingovo rastojanje za ove kodne reči su *d* = 3 i *d* = 4.

Broj jedinica u kodnoj reči naziva se *Hemingova težina*. Recimo, za kod *c2* Hemingova težina iznosi 4. Pošto je svaka kodna reč zbir dve druge kodne reči, rastojanja između pojedinih kodnih reči mogu se odrediti i preko Hemingovih težina.

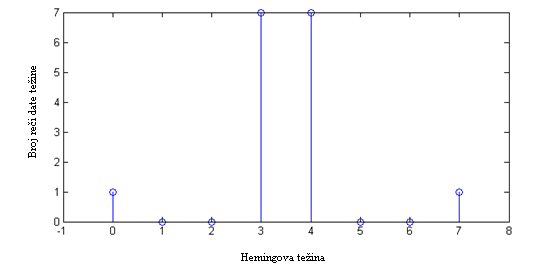
Broj kodnih reči sa određenim Hemingovim rastojanjima (težinama) predstavlja ***spektar kodnih rastojanja*** datog koda. Ako se sada vratimo na kodne reči Hemingovog koda (7,4) i pratimo kodne reči čije je *dmin*=3, odnosno minimalno rastojanje je 3. Ako ubacimo tu vrednost u prethodnu formulu, dobijamo:

***d*2*ec*+ 1**

**32*ec*+ 1**

***ec*= 1**

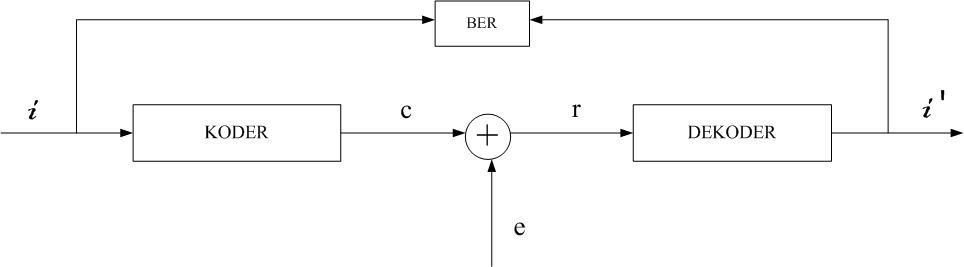
što nam govori, na osnovu predhodno objašnjenog, da ovaj kod može da detektuje i ispravi jednu grešku. To nam pokazuje i spektar koda (7,4), simulacija je data u prilogu.



*Spektar koda (7,4)*

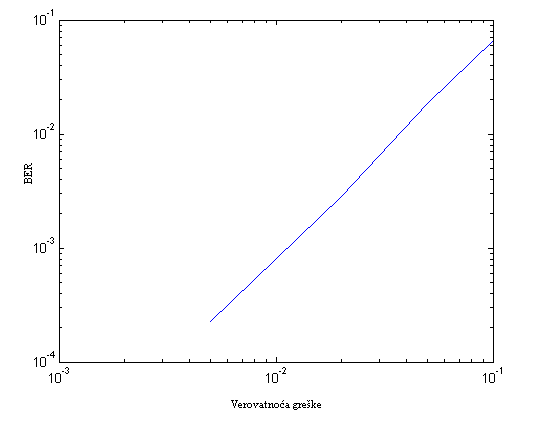
**PRENOS VEĆEG BROJA BITA KROZ CEO SISTEM; RAČUNANJE *BER*-a (*Bit Error Rate*)**

Kao krajnje razmatranje Hemingovih kodova rađena je simulacija koja predstavlja prenos kodnih reči sa velikim brojem bita kroz ceo sistem, odnosno prenos kroz koder, kanal gde dolazi do greške i dekoder. Svrha ovakve simulacije je računati *BER* (*Bit Error Rate* - verovatnoća greške po bitu). Data je blok šema koja pokazuje ovakav prenos:



Dakle, na ulaz kodera dolazi kodna reč *i* od *k* bita. Na izlazu iz kodera dobija se kodna reč *c* od *n* bita. Pri prenosu kodne reči *c* kroz kanal dolazi do greške, *e*, i dobija se kodna reč *r* sa greškom. Takva reč dovodi se na ulaz u dekoder gde se vrši ispravljanje grešaka. Na izlazu iz dekodera dobija se reč *i'* koja treba da bude identična sa kodnom reči *i* koja je bila na ulazu u koder. Na kraju se računa *BER*.

Kao primeri ovakve simulacije rađen je Hemingov kod (7,4). Za kod (7,4) uzeto je da je broj bita *n*=40000, a verovatnoće pojavljivanja greške u tih 40000 bita koje se upoređuju sa *BER*-om su *p* = 0.005; *p* = 0.02; *p* = 0.05; *p* = 0.1. Naravno, 40000 bita je podeljeno na delove od po 4 bita za ulaz u koder i od po 7 bita za ulaz u dekoder. Ovakva simulacija data je u prilogu. Slika pokazuje rezultate simulacije prikazane u logaritamskom koordinatnom sistemu:



***PRILOG***

**Koder (7,4)**

function C=koder7\_4(I)

G=[1 1 1 0 0 0 0;

1 0 0 1 1 0 0;

0 1 0 1 0 1 0;

1 1 0 1 0 0 1];

C=rem(I\*G,2);

**Dekoder (7,4)**

function Ir=dekoder7\_4(r)

R=r;

s1=rem(r(1)+r(3)+r(5)+r(7),2);

s2=rem(r(2)+r(3)+r(6)+r(7),2);

s3=rem(r(4)+r(5)+r(6)+r(7),2);

s=s1\*2^0+s2\*2^1+s3\*2^2;

if s~=0

R(s)=rem(r(s)+1,2);

end

Ir=[R(3) R(5) R(6) R(7)];

**Spektar (7,4)**

clear I;

w=zeros(1,8); % 8=n+1

for i=0:15; % 15=2^k-1

ib=dec2bin(i,4); % 4=k

for j=1:length(ib)

I(j)=str2num(ib(j));

end

C=koder7\_4(I); % odgovarajuci koder

q=sum(C);

w(q+1)=w(q+1)+1;

end

stem(0:7,w) % 7=n

w

**Verovatnoća greške**

function e=verovatnocagreske(N,p)

for n=1:N

x=rand(1,1);

if x<p

e(n)=1;

else

e(n)=0;

end

end

**Konačno (7,4)**

function BER=konacno7\_4(N,p)

% N - mora da bude deljivo sa 4 !!!

x=0; y=0; yr=0; xr=0;

x=randint(1,N);

for i=1:N/4

I=x(4\*i-3:4\*i);

C=koder7\_4(I);

y(7\*i-6:7\*i)=C;

end

e=verovatnocagreske(N/4\*7,p);

yr=rem(y+e,2);

for i=1:N/4

R=yr(7\*i-6:7\*i);

Ir=dekoder7\_4(R);

xr(4\*i-3:4\*i)=Ir;

end

BER=sum(rem(x+xr,2))/N;

**BER (7,4)**

BER=0;

N=40000;

p=[0.005 0.02 0.05 0.1];

for i=1:length(p)

BER(i)=konacno7\_4(N,p(i));

end

loglog(p,BER)

[p; BER]