# Zaštitno kodiranje signala

## Laboratorijska vježba 5

SP

# Kodiranje, dekodiranje
## (otkrivanje i ispravak pogrešaka)
# za linearne blok-kodove

Student:

| | | |
|---|---|---|
| prezime | Ime | mat. broj |
| | Laboratorij | Datum |

# *Sadržaj:*

# 5. 5. KODIRANJE, DEKODIRANJE (OTKRIVANJE I ISPRAVAK POGREŠAKA) ZA LINEARNE BLOK-KODOVE

Koristiti:

1. "Upute za rad simulacijskim alatom Logisim 2.7.1." (Moodle)
2. "Simulacijski primjeri" (Moodle)

Priprema za vježbu

Iz skripte "*Zaštitno kodiranje signala*" pročitati poglavlja:

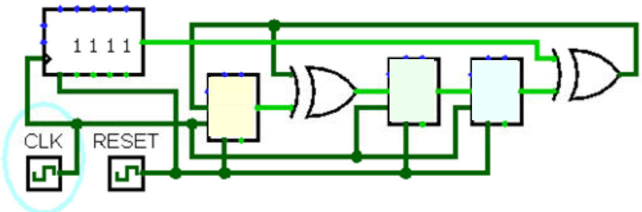
Sazdano od:

Napomena: Cjelovit opis vježbe nalazi se na: MOODLE

## *5.1. 5.1. VJEŽBA: LFSR i (7, 4) kod*

### 5.1.1. 5.1.1. ZADATAK ZA (7, 4) KOD

Nacrtajte prikazani LFSR u simulacijskome alatu LogiSim.




Napomena: U svakoj tablici popunjeni su prvi i zadnji stupac (kao uzorci).

#### *5.1.1.1. 5.1.1.1. Popunite priložene tablice*

| | | | Popunite sljedeću tablicu: Ulazni vektori = | 0000 | 0110 | 1010 | 0010 | 1100 | 0100 | 1000 | 1110 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| br. | R1 | R2 | R3 | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* |
| 1. | $d$ | $d$ | $0$ | 000 | | | | | | | 000 |
| 2. | $c$ | $d+c$ | $d$ | 000 | | | | | | | 110 |
| 3. | $b+d$ | $b+d+c$ | $d+c$ | 000 | | | 011 | | | | 101 |
| 4. | $a+d+c$ | $a+d+c+b+d$ | $b+d+c$ | 000 | | | | | | | 010 |
| | $p_0$ | $p_1$ | $p_2$ | ⟵ sadržaji LFSR registara su paritetni bitovi | | | | | | | |

| | | | Popunite sljedeću tablicu: Ulazni vektori = | 1111 | 0111 | 1011 | 0011 | 1101 | 0101 | 1001 | 0001 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| br. | R1 | R2 | R3 | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* | *abcd* |
| 1. | $d$ | $d$ | $0$ | 110 | | | | | | | 110 |
| 2. | $c$ | $d+c$ | $d$ | 011 | | | | | | | 101 |
| 3. | $b+d$ | $b+d+c$ | $d+c$ | 111 | | | | | | | 010 |
| 4. | $a+d+c$ | $a+d+c+b+d$ | $b+d+c$ | 101 | | | | | | | 111 |
| | $p_0$ | $p_1$ | $p_2$ | ⟵ sadržaji LFSR registara su paritetni bitovi | | | | | | | |

#### *5.1.1.2. 5.1.1.2. Pitanje:*

U kojoj matrici su izravno prikazane paritetne jednadžbe gornjih tablica i kako?

**G** ......................................................................................................................................

**H** ......................................................................................................................................

$\mathbf{H}^T$ ......................................................................................................................................

### 5.1.1.3. 5.1.1.3. Napišite tu matricu!

### 5.1.1.4. 5.1.1.4. Zadatak:

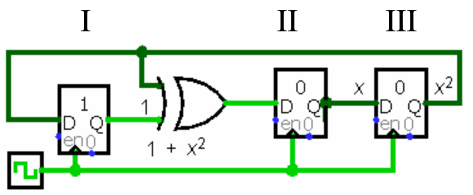
Preradite gornji sklop tako da se prikazana matrica dobije simulacijski bez ulaznoga vektora!

## 5.2. 5.2. VJEŽBA: LFSR i (6, 3) kod

### 5.2.1. 5.2.1. ZADATAK ZA (6, 3) KOD

Za zadanu matricu pariteta $\mathbf{H}^T$ odredite matrice **G**, **H**, koder i dekoder (6, 3) koda!

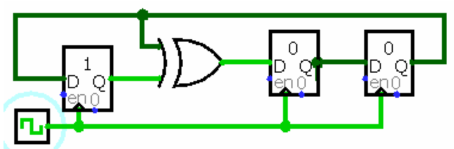
Polazište je:




| | stanja registra R1 | I | II | III |
|---|---|---|---|---|
| | početno stanje | 1 | 0 | 0 |
| posmik | 1. | 0 | 1 | 0 |
| | 2. | 0 | 0 | 1 |
| | 3. | 1 | 1 | 0 |
| | 4. | 0 | 1 | 1 |
| | ~~5.~~ | ~~1~~ | ~~1~~ | ~~1~~ |
| | 6. | 1 | 0 | 1 |

Rezultat je matrica $\mathbf{H}^T$ prikazana dolje:

Pitanje [vezano za (6, 3) kod]?



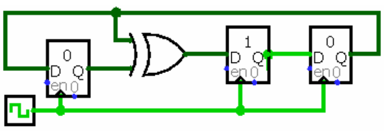
$$\mathbf{H}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

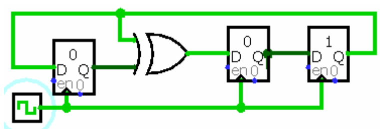
Odgovor [vezano za (6, 3) kod]?!

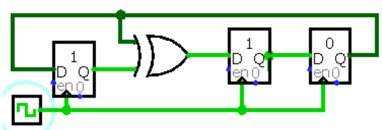
Kako preskočiti ovaj redak pri automatskome generiranju redaka za (6, 3) kod

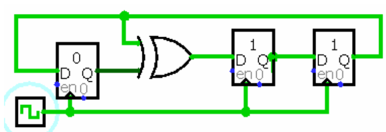
Kako preskočiti ovaj redak pri automatskome generiranju redaka za (6, 3) kod

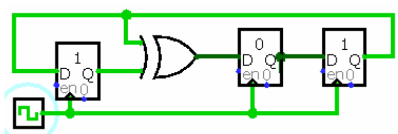
To se postiže sklopom na slici 3.1 u 7 koraka uključujući kao prvi, gornji korak, plus koraci a-f (bez e):


a)


b)


c)


d)


f)

Sljedeći posmik vraća sustav u početno stanje 100 prikazano na slici 3.1.

### 5.2.2. 5.2.2. ZADATAK

Nacrtajte simulacijski sklop u Logisimu za stvaranje redaka paritetne matrice $\mathbf{H}^T$ (6, 3) koda?

### 5.2.3. 5.2.3. PRIMJER LINEARNOGA (6, 3) BLOK-KODA

#### *5.2.3.1. 5.2.3.1. Zadatak 1*

Zadana je generator-matrica linearnoga (6, 3) blok-koda.

1. Nacrtajte pripadni koder
2. Odredite sve kodne riječi (6, 3) koda i upišite ih u priloženu tablicu.
3. Zadana je shema kodera linearnoga (6, 3) blok-koda. Odredite sve kodne riječi i upišite ih u tablicu

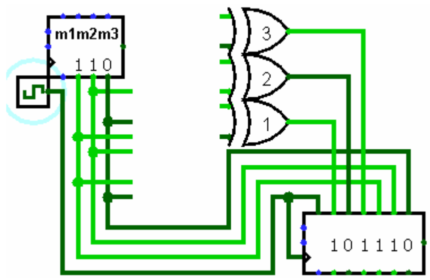
$$\mathbf{G} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Odnos vektora (i pripadnih paritetnih bitova u $\mathbf{G}$:

$\mathbf{v}_1 \Rightarrow p_1 = 1. + 3.$

$\mathbf{V}_2 \Rightarrow p_1 = 1. + 2.$

$\mathbf{V}_3 \Rightarrow p_1 = 2. + 3.$




| Vektor poruke | Kodna riječ |
|---|---|
| $m_1m_2m_3$ | $[\mathbf{P} \mid \mathbf{I}_k]$ |
| 0 0 0 | |
| 1 0 0 | |
| 0 1 0 | |
| 1 1 0 | |
| 0 0 1 | |
| 1 0 1 | |
| 0 1 1 | |
| 1 1 1 | |

4. Može li se izostaviti bilo koji od redaka paritetne matrice $\mathbf{H}^T$ (osim redaka s 1 jedinicom)?

........................................................................................................................................................................

........................................................................................................................................................................

........................................................................................................................................................................

........................................................................................................................................................................

5. Kako se to odražava na same kodne riječi?

........................................................................................................................................................................

........................................................................................................................................................................

........................................................................................................................................................................

........................................................................................................................................................................

6. Usporedite kodirane riječi (7, 4) koda i (6, 3) koda. U čemu se razlikuju?

........................................................................................................................................................................

........................................................................................................................................................................

........................................................................................................................................................................

........................................................................................................................................................................

## 5.3. 5.3. Položaji jedinične matrice

Vektor poruke ($m_1m_2m_3$ = 3 bita), za $\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k]$ (jedinična matrica postavljena je *DESNO* u **G**), preslikava se s *desne* strane u kodiranoj riječi!

$$\boxed{\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k] \Rightarrow \mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T] \Rightarrow \mathbf{H}^T = \begin{bmatrix} \mathbf{I}_{n-k} \\ \cdots\cdots \\ \mathbf{P} \end{bmatrix}}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \Rightarrow \mathbf{H}^T = \begin{array}{r} \\ \mathbf{I}_{n-k}\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right. \\ \mathbf{P}\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right. \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{array}$$

Pitanje:

7. Daje li **G** matrica iste kodirane informacije u obliku $\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k]$ i u obliku $[\mathbf{I}_k \mid \mathbf{P}]$?

.....................................................................................................................................................

.....................................................................................................................................................

.....................................................................................................................................................

Ako je za iste vektore $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, koda (6, 3), jedinična matrica postavljena (desno/lijevo) u generator-matrici, nacrtajte koder, pronađite sve kodne riječi te ih upišite u tablicu.

*Tablica 5.1. Pridruživanje kodnih riječi porukama (ako je jedinična matrica postavljena DESNO odnosno LIJEVO u* **G***)*

| Kodne riječi **u** | Vektor poruke | Kodna riječ | | Kodna riječ | |
|---|---|---|---|---|---|
| | **m** | $[\mathbf{P} \mid \mathbf{I}_k]$ | | $[\mathbf{I}_k \mid \mathbf{P}]$ | |
| 1 | 000 | 101 | | | 101 |
| 2 | 100 | 011 | | | 011 |
| 3 | 010 | 110 | | | 110 |
| 4 | 110 | 000 | | | 000 |
| 5 | 001 | 000 | | | 000 |
| 6 | 101 | 110 | | | 110 |
| 7 | 011 | 011 | | | 011 |
| 8 | 111 | 101 | | | 101 |

Prirodno pitanje koje treba postaviti je: "Za linearan kod, kako odabrati kodnu riječ koja je izvan prostora od $2^8$ 8-morki? "Ne postoji jedinstveno rješenje, ali postoje ograničenja u načinu kako izabrati. Ovdje su elementi koji pogađaju središte rješenja toga problema. Slijedi primjer!

1. Broj kodnih riječi je $2^k = 2^3 = 8$.

Pitanje

Što je to svojstvo *zatvorenosti*.

2. Mora se primijeniti svojstvo *zatvorenosti*. Ovo svojstvo određuje da zbroj bilo kojih dviju kodnih riječi u tome prostoru mora dati valjanu kodna riječ iz toga prostora. (Ovo znači minimizaciju broja vektora u generator matrici - za generator matricu **G** (6, 3) koda, dovoljna su 3 vektora za napraviti 6 različitih, ali međusobno povezanih generator-vektora)

Pitanje

Je li vektor "sve 0" rezultat svojstva *zatvorenosti* i zašto?

3. Vektor sa samim nulama mora biti jedna od kodnih riječi. Ovo svojstvo, rezultat je svojstva *zatvorenosti*, jer bilo kodna riječ koja se zbraja (modulo-2) sama sobom, daje vektor sve-nule.
4. Svaka kodna riječ ima 6 bitova.
5. Budući da je $d_{\min} = 3$, težina svake kodne riječi (osim kodne riječi "sve 0") također mora biti najmanje 3 (na temelju svojstva *zatvorenosti*). Težina vektora definira se kao broj komponenata vektora različitih od nule (jednakih 1).
6. Pretpostavimo da je kod sustavan pa 3 bita *sasvim desno* u svakoj kodnoj riječi predstavljaju odgovarajuću poruku.

Sljedeći zadatak je kandidate kodnih riječi pridružiti porukama koje ispunjavaju sve prethodno navedene uvjete.

| *poruke* | *kodne riječi* |
|---|---|
| 000 | 000000 |
| 100 | 110100 |
| 010 | 011010 |
| 110 | 101110 |
| 001 | 101001 |
| 101 | 011101 |
| 011 | 110011 |
| 111 | 000111 |

*Oblikovanje skupa kodnih riječi* može početi na veoma *proizvoljan način*, jedino je potrebno pridržavati se svojstava težine i sustavnoga oblika koda. Izbor prvih nekoliko kodnih riječi često je jednostavan. Međutim, kako se proces nastavlja, postupak izbora postaje sve teže, a izbori postaju ograničeni zbog potrebe da se združe prema *svojstvu zatvorenosti*.

Za jediničnu matricu **I** postavljenu *DESNO* u **G** matrici, dobije se kodirana riječ kao:

$$\mathbf{u}_4 = [1\ 1\ 0]\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \mathbf{v}_3\end{bmatrix} = 1\cdot\mathbf{v}_1 + 1\cdot\mathbf{v}_2 + 0\cdot\mathbf{v}_3 = 1\,1\,0\,1\,0\,0 + 0\,1\,1\,0\,1\,0 + 0\,0\,0\,0\,0\,0 = 1\,0\,1\,1\,1\,0$$

(kodna-riječ za vektor poruke (informacijski vektor što se kodira u kodnu riječ) 1 1 0).

Za jediničnu matricu **I** postavljenu LIJEVO u **G** matrici, dobije se kodirana riječ kao:

$$\mathbf{u}_4 = [1\ 1\ 0]\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \mathbf{v}_3\end{bmatrix} = 1\cdot\mathbf{v}_1 + 1\cdot\mathbf{v}_2 + 0\cdot\mathbf{v}_3 = 1\,0\,0\,1\,1\,0 + 0\,1\,0\,0\,1\,1 + 0\,0\,0\,0\,0\,0 = 1\,1\,0\,1\,0\,1$$

(kodna-riječ za vektor poruke (informacijski vektor što se kodira u kodnu riječ) 1 1 0).

Za ovaj (6, 3) kod i riječ $\mathbf{u}_4$ prednja i zadnja 3 bita (simetrija) samo zamijene svoja mjesta!

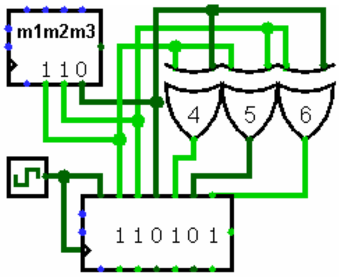
U nastavku za obrađene primjere, koristim ovu generator matricu **G**, čija je jedinična matrica $\mathbf{I}_k$ postavljena LIJEVO. Simulacijski primjer dan je kao nastavak (2. dio) tablice 5.1.

*Tablica 5.1. (2. dio) Pridruživanje kodnih riječi porukama (ako je jedinična matrica postavljena* LIJEVO *u* **G***)*

Simulacija: *Koder za linearan (6, 3) blok kod (ako je jedinična matrica postavljena* LIJEVO *u* **G***).*

Nacrtajte simulacijski sklop za (6, 3) kod i popunite priloženu tablicu:

$$\mathbf{G} = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \mathbf{v_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$




| Vektor poruke | Kodna riječ |
|---|---|
| | $[\mathbf{I}_k \mid \mathbf{P}]$ |
| 000 | |
| 100 | |
| 010 | |
| 110 | |
| 001 | |
| 101 | |
| 011 | |
| 111 | |

Vektor poruke (3 bita), proširuje kodiranu riječ s desne strane (vrijedi za $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$)!

**Pitanje**

Što je to sustavno svojstvo?

.............................................................................................................................................

.............................................................................................................................................

.............................................................................................................................................

Pitanje:

Čemu služi generator matrica?

Odgovor:

Kodni vektor što odgovara vektoru poruke, linearna je kombinacija redaka generator-matrice **G**.

$$\mathbf{G} = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \mathbf{v_3} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}_{\text{DESNO}} \tag{5.1}$$

Budući da kod potpuno definira matrica **G**, koder treba samo pohraniti $k$ redova (3 retka) **G matrice** umjesto ukupno $2^k$ vektora koda $\mathbf{H}^T$ matrice. Za ovaj primjer, primijetimo da generatorsko polje dimenzije$(3 \times 6)$ u jednadžbi (5.1) zamjenjuje niz izvornih kodnih riječi dimenzije $(8 \times 6)$ u tablici 5.1, što smanjuje složenost sustava.

## 5.4. 5.4. Sustavni linearni blok-kodovi

Sustavan $(n, k)$ linearan blok-kod, $k$-dimenzijski vektor poruke preslikava u $n$-dimenzijsku kodnu riječ na takav način da se dio generiranoga niza *podudara* s $k$ znamenaka poruke. Preostalih $(n–k)$ znamenaka su paritetne znamenke. Sustavan linearan blok-kod ima generator-matricu oblika:

$$\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k] = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1(n-k)} & 1 & 0 & \cdots & 0 \\ p_{12} & p_{22} & \cdots & p_{2(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{k(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5.2}$$

gdje je **P** paritetan dio niza generator-matrice $p_{i,j}$ = (0 ili 1), a $\mathbf{I}_k$ je $k \times k$ jedinična matrica (1 na glavnoj dijagonali, a 0 na ostalim mjestima). Primijetimo da se složenost kodiranja dodatno smanjuje sustavnim generatorom, budući da nije potrebno pohraniti dio polja jedinične matrice. Napomenimo da se jedinična matrica može postaviti lijevo ili desno u odnosu na paritetnu matricu, čime se dobivaju dva različita skupa kodiranih riječi. Kombiniranjem jednadžbi (5.1) i (5.2), svaka kodna riječ izražava se kao:

$$u_1, u_2, \ldots, u_n = [m_1, m_2, \ldots, m_k] \times \left[\begin{array}{cccc|cccc} p_{11} & p_{12} & \cdots & p_{1(n-k)} & 1 & 0 & \cdots & 0 \\ p_{12} & p_{22} & \cdots & p_{2(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{k(n-k)} & 0 & 0 & \cdots & 1 \end{array}\right]$$

gdje je

$$u_i = m_1 p_{1i} + m_2 p_{2i} + \ldots + m_k p_{ki} \qquad \text{za } i = 1, \ldots, (n-k)$$

$$= m_{i-n+k} \qquad \text{za } i = (n-k+1), \ldots, n$$

Zadana *k-torka* poruke je

$$\mathbf{m} = m_1, m_2, \ldots, m_k$$

i općenito $n$-torka kodnoga vektora je

$$\mathbf{u} = u_1, u_2, \ldots, u_k$$

*sustavni kodni vektor* može se izraziti kao

$$\mathbf{u} = \underbrace{p_1, p_2, \ldots, p_{n-k}}_{\text{paritetni bitovi}}, \underbrace{m_1, m_2, \ldots, m_k}_{\text{bitovi poruke}}, \tag{5.3}$$

Pitanja:

Što je to sustavan kodni vektor?

Od koja dva skupa bitova se sastoji kodirana ulazna poruka?

gdje je

$$\begin{aligned} p_1 &= m_1 p_{11} + m_2 p_{21} + \cdots + m_k p_{k1} \\ p_2 &= m_1 p_{12} + m_2 p_{22} + \cdots + m_k p_{k2} \\ p_{n-k} &= m_1 p_{1(n-k)} + m_2 p_{2(n-k)} + \cdots + m_k p_{k(n-k)} \end{aligned} \tag{5.4}$$

Sustavne kodne riječi nekada se pišu tako da bitovi poruke zauzimaju lijevi dio kodne riječi, a paritetni bitovi zauzimaju desni dio. Ovo preuređenje nema nikakvoga utjecaja na svojstava koda pri otkrivanju ili ispravci pogrešaka pa ćemo ih prema potrebi dalje razmatrati.

Za primjer (6, 3) koda iz poglavlja 6.4.3, kodne riječi su dobivene na sljedeći način:

$$\mathbf{u} = [m_1, m_2, m_3] \cdot \begin{array}{r} 1. \\ 2. \\ 3. \end{array} \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \tag{5.5}$$

$$\qquad\qquad \underbrace{\phantom{1\ 1\ 0}}_{\mathbf{P}} \quad \underbrace{\phantom{1\ 0\ 0}}_{\mathbf{I}_3}$$

$$= \underbrace{m_1 + m_3}_{u_1}, \underbrace{m_1 + m_2}_{u_2}, \underbrace{m_2 + m_3}_{u_3}, \underbrace{m_1}_{u_4}, \underbrace{m_2}_{u_5}, \underbrace{m_3}_{u_6} \tag{5.6}$$

Jednadžba (5.6) pruža uvid u strukturu linearnih blok-kodova. Vidimo da se zalihosne znamenke proizvode na različite načine. Prvi paritetan bit je zbroj prvoga i trećega bita poruke. Drugi paritetan bit je zbroj prvoga i drugoga bita poruke. Treći paritetan bit je zbroj drugoga i trećega bita poruke. Intuicija nam kazuje da takva struktura, u usporedbi s jednostrukom provjerom pariteta ili jednostavnim postupcima postavke znamenaka, može osigurati veću sposobnost otkrivanja i ispravke pogrešaka.

### 5.4.1. 5.4.1. MATRICA PROVJERE PARITETA

Definirajmo matricu **H**, nazvanu *matrica provjere pariteta*, koja će nam omogućiti *dekodiranje* primljenoga vektora. Za svaku ($k$×$n$) generator-matricu **G**, postoji [($n$–$k$)×$n$] matrica **H**, takva da su retci matrice **G** *okomiti* na retke matrice **H**; tj., $\mathbf{G}\cdot\mathbf{H}^T = \mathbf{0}$, gdje je $\mathbf{H}^T$ transponirana matrica **H**, a **0** je [$k$×($n$–$k$)] matrica "sve 0".

Pitanje/Zadatak:

$\mathbf{G}\cdot\mathbf{H}^T = \mathbf{0}$ … napraviti zadatak za provjeru u laboratorijskim vježbama!

$$\boxed{\mathbf{u} = \mathbf{m}\cdot\mathbf{G} \Rightarrow \mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k] \Rightarrow \mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T] \Rightarrow \mathbf{H}^T = \begin{bmatrix} \mathbf{I}_{n-k} \\ \hline \mathbf{P} \end{bmatrix} \Rightarrow \mathbf{u}\cdot\mathbf{H}^T = \mathbf{0}}$$

$$\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k]$$

$\mathbf{H}^T$ je [$n$×($n$–$k$)] matrica čiji *stupci* su *retci* matrice **H** i čiji *retci* su *stupci* matrice **H**. Da bi se ispunili zahtjevi okomitosti za sustavan kod, komponente **H** matrice napisat ćemo kao

$$\mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T] \tag{5.7}$$

Na temelju prethodnih objašnjenja, matrica $\mathbf{H}^T$, pomoću $\mathbf{I}_{n-k}$ i $\mathbf{P}^T$ piše se (okomito na matricu **H**) kao

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{I}_{n-k} \\ \hline \mathbf{P} \end{bmatrix} = \tag{5.8}$$

ili općenito:

$$= \begin{bmatrix} 1 & 0 & \cdots & \\ 0 & 1 & \cdots & \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \\ p_{11} & p_{12} & \cdots & p_{1,(n-k)} \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} \\ \vdots & \vdots & & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} \end{bmatrix} \tag{5.9}$$

Dakle, prema jednadžbi (5.7), **H** matrica sastoji od dvije pod-matrice: jedinične $\mathbf{I}_{n-k}$ i matrice pariteta $\mathbf{P}^T$). Položaj ovih dviju matrica međusobno je zamjenjiv (gore-dolje) prema već danome objašnjenju.

$$\mathbf{I}_{n-k} = \mathbf{I}_{6-3} = \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{P}^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

odnosno

$$\text{Za: } \mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k] \quad \Rightarrow \mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \Rightarrow \mathbf{H}^T = \begin{bmatrix} \mathbf{I}_{n-k} \\ \hline \mathbf{P} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Za: $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$ $\Rightarrow \mathbf{H} = [\mathbf{P}^T \mid \mathbf{I}_{n-k}] = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ $\Rightarrow \mathbf{H}^T = \begin{bmatrix} \mathbf{P} \\ \hdashline \mathbf{I}_{n-k} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Dakle, ovakav raspored pod-matrica $\mathbf{P}^T$ i $\mathbf{I}_{n-k}$, omogućuje generiranje 2 različita koda iste vrste (6, 3) ovisno o njihovome međusobnome položaju.

---

Za naš primjer matrice **G** jednadžba (5.1) (jedinična pod-matrica *DESNO* u **G**):

$$\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k] \Rightarrow \mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T] \Rightarrow \mathbf{H}^T = \begin{bmatrix} \mathbf{I}_{n-k} \\ \hdashline \mathbf{P} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

i prethodne matrice **H**:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \Rightarrow \mathbf{H}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Kao što imamo različite kodirane informacijske vektore ovisno o položaju jedinične pod-matrice $\mathbf{I}_k$ unutar generator-matrice **G**, , tako i umnošci generator-matrice **G** i transponirane **H** matrice ($\mathbf{H}^T$), daju različite rezultate. Stoga se mora strogo paziti na položaj $\mathbf{I}_k$ unutar **G** za provjeru okomitosti sustavnoga koda. Rezultat mora biti nul-matrica **0**.

Ako se ne poštuje ovakav raspored, rezultat provjere umnoška $\mathbf{G}\cdot\mathbf{H}^T$ ***nije*** **0**-matrica! Razlika u redoslijedu pod-matrica $\mathbf{P}^T$ i $\mathbf{I}_{n-k}$, je u rasporedu paritetnih bitova kodiranih ulaznih informacija. Pisana u sustavnome obliku, **G** matrica generira kodne riječi tako da se ulazna informacija smješta:

- *LIJEVO* u kodiranoj riječi za LIJEVI položaj jedinične matrice u **G**, odnosno

$$\mathbf{G}\cdot\mathbf{H}^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}_{\text{LIJEVO}} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\text{DOLJE}} = \mathbf{0}$$

| Vektor poruke | Kodna riječ $\mathbf{G}=[\mathbf{I}_k \mid \mathbf{P}]$ |
|---|---|
| 000 | 000 000 |
| 100 | 100 110 |
| 010 | 010 011 |
| 110 | 110 101 |
| 001 | 001 101 |
| 101 | 101 011 |
| 011 | 011 110 |
| 111 | 111 000 |

- *DESNO* u kodiranoj riječi za DESNI položaj jedinične matrice u **G**.

$$\mathbf{G}\cdot\mathbf{H}^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}_{\text{DESNO}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}_{\text{GORE}} = \mathbf{0}$$

| Vektor poruke | Kodna riječ |
|---|---|
| | $\mathbf{G}=[\mathbf{P}\vert\mathbf{I}_k]$ |
| 000 | 000 000 |
| 100 | 110 100 |
| 010 | 011 010 |
| 110 | 101 110 |
| 001 | 101 001 |
| 101 | 011 101 |
| 011 | 110 011 |
| 111 | 000 111 |

Lako je provjeriti da je umnožak $\mathbf{u}\cdot\mathbf{H}^T$ svake kodne riječi $\mathbf{u}$, generirane matricama $\mathbf{G}$ i $\mathbf{H}^T$, jednak

$$\mathbf{u}\cdot\mathbf{H}^T = p_1 + p_1, p_2 + p_2, \ldots, p_{n-k} + p_{n-k} = \mathbf{0},$$

gdje su paritetni bitovi $p_1$, $p_2$, …, $p_{n-k}$ definirani u jednadžbi (5.4). Dakle, nakon što se konstruira matrica provjere pariteta $\mathbf{H}$ za ispuniti zahtjeve navedene okomitosti, možemo ju koristiti za ispitati je li primljeni vektor ispravan član skupa kodnih riječi. Riječ $\mathbf{u}$, što ju generira matrica $\mathbf{G}$, je kodna riječ onda i samo onda ako je

$$\mathbf{u}\cdot\mathbf{H}^T = \mathbf{0}.$$

### 5.4.2. 5.4.2. PROVJERA SINDROMA

Neka je $\mathbf{r} = [r_1, r_2, \ldots, r_n]$ primljen vektor (jedan od $2^n$ $n$-torki), što rezultira iz prijenosa $\mathbf{u} = [u_1, u_2, \ldots, u_k]$ (jedna od $2^k$ $n$-torki). Dakle, može opisati $\mathbf{r}$ kao

$$\mathbf{r} = \mathbf{u} + \mathbf{e} \tag{5.10}$$

gdje je $\mathbf{e} = [e_1, e_2, \ldots, e_n]$ vektor pogreške ili uzorak pogreške što ga je uveo kanal. Ima ukupno $2^n-1$ potencijalno različitih ne nultih uzoraka pogrešaka u prostoru od $2^n$ $n$-torki. Sindrom $\mathbf{r}$ definira se kao

$$\mathbf{s} = \mathbf{r}\cdot\mathbf{H}^T \tag{5.11}$$

Sindrom je posljedica provjere pariteta obavljene na $\mathbf{r}$ da bi se utvrdilo je li $\mathbf{r}$ punopravan član skupa kodnih riječi. U stvari, ako je $\mathbf{r}$ član skupa kodnih riječi, sindrom $\mathbf{s}$ ima vrijednost $\mathbf{0}$. Ako $\mathbf{r}$ sadrži mjerljive pogreške, sindrom ima neke ne nulte vrijednosti. Ako $\mathbf{r}$ sadrži ispravljive pogreške, sindrom (kao simptom bolesti) ima ne nulte vrijednosti koje trebaju označiti određen uzorak pogreške. Dekoder, ovisno o tome je li pripremljen za obavljanje FEC ili ARQ, poduzet će radnju za pronalazak pogreške i njen ispravak (FEC), u protivnome će zatražiti ponavljanje prijenosa (ARQ). Kombinirajući jednadžbe (5.10) i (5.11), sindrom $\mathbf{r}$ se vidi kao

$$\mathbf{s} = (\mathbf{u} + \mathbf{e})\mathbf{H}^T = \mathbf{u}\cdot\mathbf{H}^T + \mathbf{e}\cdot\mathbf{H}^T \tag{5.12}$$

Međutim, $\mathbf{u}\cdot\mathbf{H}^T = \mathbf{0}$ za sve članove skupa kodnih riječi. Stoga je

$$\mathbf{s} = \mathbf{e}\cdot\mathbf{H}^T \tag{5.13}$$

Naveden razvoj, što započinje jednadžbom (5.10) i završava jednadžbom (5.13), dokaz je da provjera sindroma, bilo da se izvodi na oštećenome kodnome vektoru ili na uzorku pogreške što ju je prouzročio, daje isti sindrom. Važno svojstvo linearnih blok-kodova, što se temelji na procesu dekodiranja, jest da je preslik između ispravljivih uzoraka pogrešaka i sindroma jedan na jedan.

Zanimljivo je napomenuti sljedeće dvije zahtijevane osobine matrice provjere pariteta.

1. Ni jedan stupac $\mathbf{H}$ ne može biti "sve 0", ili pogreška na odgovarajućemu položaju u kodnoj riječi neće utjecati na sindrom ili se neće otkriti.

2. Svi stupci $\mathbf{H}$ moraju biti jedinstveni. Ako su dva stupca matrice $\mathbf{H}$ jednaka, pogreške na odgovarajućim položajima u ovim dvjema kodnim riječima neće se primijetiti.

[1] za vježbu

### 5.4.3. 5.4.3. PRIMJER 5.3. PROVJERA SINDROMA

Pretpostavimo da se prenosi kodna riječ **u** = **101110** iz primjera u poglavlju 6.4.3. (*DESNO*), a primi se vektor **r** = 001110; to jest, pogrešno se primi bit sasvim lijevo. Pronađite vrijednost vektora sindroma **s** = $\mathbf{r}\cdot\mathbf{H}^T$ i uvjerite se da je jednaka $\mathbf{e}\cdot\mathbf{H}^T$. Jedinična matrica postavljena je *DESNO* u **G**, *GORE* u $\mathbf{H}^T$.

***Rješenje***

$$\mathbf{s} = \mathbf{r}\cdot\mathbf{H}^T = [0\ 0\ 1\ 1\ 1\ 0]\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\\1 & 1 & 0\\0 & 1 & 1\\1 & 0 & 1\end{bmatrix} = [1,\ 1{+}1,\ 1{+}1] = [\boxed{1}\ 0\ 0]$$ (sindrom oštećenoga kodnoga vektora)

Simulacija: Dekoder 6, 3 DESNO.circ

$$\mathbf{G}\cdot\mathbf{H}^T = \begin{bmatrix}1 & 1 & 0 & 1 & 0 & 0\\0 & 1 & 1 & 0 & 1 & 0\\1 & 0 & 1 & 0 & 0 & 1\end{bmatrix}_{\text{DESNO}}\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\\1 & 1 & 0\\0 & 1 & 1\\1 & 0 & 1\end{bmatrix}_{\text{GORE}} = \mathbf{0}$$

Zatim smo potvrdili da je sindrom oštećenoga kodnoga vektora isti kao i sindrom *uzorka* pogreške ($\boxed{1}$00) što je uzrokovala pogrešku:

$$\mathbf{s} = \mathbf{e}\cdot\mathbf{H}^T = [1\ 0\ 0\ 0\ 0\ 0]\cdot\mathbf{H}^T = [\boxed{1}00] \text{ (sindroma uzorka pogreške)}$$

Ispravljen vektor je: 101110.

### 5.4.4. 5.4.4. ISPRAVAK POGREŠAKA

Otkrili smo jednu pogrešku te smo pokazali da provjera sindroma izvedena na oštećenoj kodnoj riječi, ili na uzorku pogreške koji ju je prouzročio, daje isti sindrom. To bi trebao biti znak, ne samo da se može otkriti pogreška, već, budući da je preslik između ispravljivih uzoraka pogrešaka i sindroma jedan na jedan, možemo ispraviti takve uzorke pogrešaka.

Presložimo $2^n$ $n$-torki koje predstavljaju moguće primljene vektore u polju, što se zove *standardno polje*, tako da prvi redak sadrži sve kodne riječi, počevši kodnom riječi sa svim nulama, a prvi stupac sadrži sve ispravljive uzorke pogrešaka. Podsjetimo se iz osnovnih svojstava linearnih kodova da vektor "sve 0" mora biti član skupa kodnih riječi. Svaki redak, nazvan koskup, sastoji se od jednoga uzorka pogreške u prvome stupcu, a zove se čelnik koskupa, iza čega slijede kodne riječi ometane uzorkom pogreške. Standardan format za ($n$, $k$) kod je kako slijedi:

$$\begin{array}{cccccc}
\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_i & \cdots & \mathbf{u}_{2^k}\\
\mathbf{e}_2 & \mathbf{u}_2+\mathbf{e}_2 & \cdots & \mathbf{u}_i+\mathbf{e}_2 & \cdots & \mathbf{u}_{2^k}+\mathbf{e}_2\\
\mathbf{e}_3 & \mathbf{u}_2+\mathbf{e}_3 & \cdots & \mathbf{u}_i+\mathbf{e}_3 & \cdots & \mathbf{u}_{2^k}+\mathbf{e}_3\\
\vdots & \vdots & \vdots & \vdots & \cdots & \vdots\\
\mathbf{e}_j & \mathbf{u}_2+\mathbf{e}_j & & \mathbf{u}_i+\mathbf{e}_j & \cdots & \mathbf{u}_{2^k}+\mathbf{e}_l\\
\vdots & \vdots & \vdots & \vdots & \cdots & \\
\mathbf{e}_{2^{n-k}} & \mathbf{u}_2+\mathbf{e}_{2^{n-k}} & \cdots & \mathbf{u}_i+\mathbf{e}_{2^{n-k}} & \cdots & \mathbf{u}_{2^k}+\mathbf{e}_{2^{n-k}}
\end{array} \qquad (5.14)$$

Imajte na umu da kodna riječ $\mathbf{u}_1$, "sve 0", igra dvostruku ulogu. Ona je jedna od kodnih riječi i također se može razmatrati kao uzorak pogreške $\mathbf{e}_1$ - uzorak koji ne predstavlja nikakvu pogrešku, tako da je **r** = **u**. Polje sadrži svih $2^n$ $n$-torki u prostoru $V_n$. Svaka $n$-torka pojavljuje se samo na jednome mjestu ništa ne nedostaje i ništa se ne ponavlja. Svaki koskup sastoji se od $2^k$ $n$-torki. Stoga, postoji $(2^n/2^k) = 2^{n-k}$ koskupova.

Poziva se algoritam dekodiranja da oštećen vektor (a to je bilo koja $n$-torka osim onih u prvome retku), zamijeni važećom kodnom riječi s vrha stupca što sadrži oštećen vektor. Pretpostavimo da je preko kanala sa šumom prenesena kodna riječi $\mathbf{u}_i$ ($i = 1, \ldots, 2^k$) što rezultira primljenim (oštećenim) vektorom $\mathbf{u}_i + \mathbf{e}_j$. Ako je uzorak pogreške $\mathbf{e}_j$ uzrokovan kanalom, čelnik koskupa, gdje je indeks $j = 1, \ldots, 2^{n-k}$, dobiven vektor će se pravilno dekodirati u poslanu kodnu riječ $\mathbf{u}_i$. Ako uzorak pogreške nije čelnik koskupa, onda je rezultat, *pogreška dekodiranja*.

### 5.4.5. 5.4.5. SINDROM KOSKUPA

Ako je $\mathbf{E}_j$ čelnik koskupa ili uzorak pogreške $j^{th}$ koskup, onda je $\mathbf{u}_i+\mathbf{e}_j$ $n$-torka u ovome koskupu. Sindrom te $n$-torke može se pisati

$$\mathbf{s} = (\mathbf{u}_i + \mathbf{e}_j)\mathbf{H}^T = \mathbf{u}\cdot\mathbf{H}^T + \mathbf{e}_j\cdot\mathbf{H}^T$$

Budući da je $\mathbf{u}_i$ kodni vektor, $\mathbf{u}_i\mathbf{H}^T = \mathbf{0}$, možemo pisati, kao u jednadžbi (5.13),

$$\mathbf{s} = (\mathbf{u}_i + \mathbf{e}_j)\mathbf{H}^T = \mathbf{e}_j\cdot\mathbf{H}^T \tag{5.15}$$

Naziv koskup je kratica za "skup brojeva koje imaju zajedničku značajku." Što je zajedničko članovima svakoga retka (koskupa)? Iz jednadžbe (5.15) jasno je da svaki član koskupa ima isti sindrom. Sindrom za svaki koskup je različit od bilo kojega drugoga koskup u kodu. To je sindrom koji se koristi za procjenu uzorka pogreške.

## 5.5. 5.5. Dekodiranje i ispravak pogrešaka

Postupak dekodiranja uz ispravak pogrešaka je sljedeći:

1. Izračunajte sindrom $\mathbf{r}$ korištenjem $\mathbf{s} = \mathbf{r}\cdot\mathbf{H}^T$.
2. Pronađite čelnika koskupa (uzorak pogreške) $\mathbf{e}_j$, čiji je sindrom jednak $\mathbf{r}\cdot\mathbf{H}^T$.
3. Za ovakav uzorak pogreške pretpostavlja se da je uzrok oštećenju, kanal.
4. Ispravljen primljen vektor, ili kodna riječ, prepoznaje se kao $\mathbf{u} = \mathbf{r}+\mathbf{e}_j$. Može se kazati da smo primili valjanu kodnu riječ oduzimanjem prepoznate pogreške. U aritmetici modulo-2, operacija oduzimanje identična je operaciji zbrajanja.

#### 5.5.1.1. 5.5.1.1. *Pronalazak uzorka pogreške*

$$\boxed{\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k] \Rightarrow \mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T] \Rightarrow \mathbf{H}^T = \begin{bmatrix} \mathbf{I}_{n-k} \\ \hline \mathbf{P} \end{bmatrix}}$$

Vratimo li se primjeru , u standardnome polju možemo organizirati $2^6$ = šezdeset četiri 6-torke kao što prikazuje (tablica) slika 5.11. (Jedinična pod-matrica $\mathbf{I}_k$ smještena je *DESNO* unutar generator-matrice $\mathbf{G}$.)

| Informacije što se kodiraju: | | 100 | 010 | 110 | 001 | 101 | 011 | 111 |
|---|---|---|---|---|---|---|---|---|
| Ispravne kodne riječi | 000000 | 110100 | 011010 | 101110 | 101001 | 011101 | 110011 | 000111 |
| | 000001 | 110101 | 011011 | 101111 | 101000 | 011100 | 110010 | 000110 |
| | 000010 | 110110 | 011000 | 101100 | 101011 | 011111 | 110001 | 000101 |
| Čelnici koskupova svih ispravljivih | 000100 | 110000 | 011110 | 101010 | 101101 | 011001 | 110111 | 000011 |
| uzoraka jednostrukih pogrešaka | 001000 | 111100 | 010010 | 100110 | 100001 | 010101 | 111011 | 001111 |
| | 010000 | 100100 | 001010 | 111110 | 111001 | 001101 | 100011 | 010111 |
| | 100000 | 010100 | 111010 | 001110 | 001001 | 111101 | 010011 | 100111 |
| Čelnik koskupa dvostrukih pogrešaka | 010001 | 100101 | 001011 | 111111 | 111000 | 001100 | 100010 | 010110 |

*Slika 5.1. Primjer standardnoga polja za (6, 3) kod i jediničnu matricu smještenu u* **G**=*DESNO*.

Ispravne kodne riječi su osam vektora u prvome retku (zelena pozadina), a ispravljivi uzorci pogrešaka su sedam ne-nultih čelnika koskupova u prvome stupcu. Uočite da su ispravljivi svi uzorci pogrešaka s jednim bitom. Također uočite da nakon iscrpljivanja uzoraka pogrešaka s jednim bitom, još preostaje nekakva sposobnost ispravke pogrešnih bitova, jer još nismo pretražili sve šezdeset četiri 6-torke. Preostaje jedan nepridružen čelnik koskupa. Stoga, preostaje mogućnost ispravaka pogrešaka za jedan dodatan uzorak (ukupno 2 jedinice u uzorku). Detalji izbora čelnika koskupova dvostrukih pogrešaka opisani su u: Matrice koda (6, 3).doc.

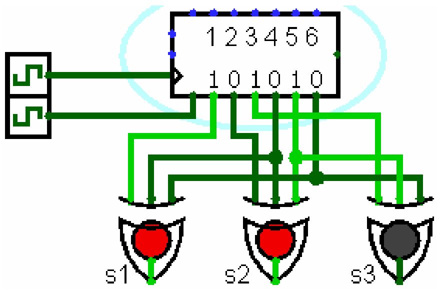
Imamo rastezljivost u odabiru ovoga uzorka pogreške da bude bilo koja od *n*-torki u preostalome koskupu. ~~Na slici 5.1 odabire se taj završni popravljiv uzorak pogreške, ponekad proizvoljno, kao uzorak pogreške 010001 od 2 bita.~~ Dekodiranje će biti točno onda i samo onda ako je uzorak pogreške što ga uzrokuje kanal, jedan od čelnika koskupa. Sada određujemo sindrom koji odgovara svakome od nizova ispravljivih pogrešaka računanjem $\mathbf{e}_j \cdot \mathbf{H}^T$ za svakoga čelnika koskupa, kako slijedi:

$$\mathbf{G}_{DESNO} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{s} = \mathbf{e}_j \cdot \mathbf{H}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Rezultati su navedeni u Tablici 6.2.

*Tablica 6.2 Pregledna tablica sindroma (DESNO)*

| Uzorak pogreške | sindrom |
|---|---|
| 000000 | 000 |
| 000001 | 101 |
| 000010 | 011 |
| 000100 | 110 |
| 001000 | 001 |
| 010000 | 010 |
| 100000 | 100 |
| 010001 | 111 |



*(Verified)*

$s_1 = r_1 + r_4 + r_6$

$s_2 = r_2 + r_4 + r_5$

$s_3 = r_3 + r_5 + r_6$

Budući da je svaki sindrom u tablici jedinstven, dekoder može prepoznati pripadni uzorak pogreške **e**.

#### 5.5.1.2. 5.5.1.2. Primjer ispravaka pogrešaka

Primamo vektor **r** i računamo njegov sindrom pomoću $\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T$. Zatim koristimo preglednu tablicu sindroma (Tablica 6.2), razvijena u prethodnome odlomku, za pronaći odgovarajući uzorak pogreške. Ovaj uzorak pogreške je procjena pogreške i označimo ga $\hat{\mathbf{e}}$. Dekoder zatim $\hat{\mathbf{e}}$ dodaje u **r** za pribaviti procjenu poslane kodne riječi $\hat{\mathbf{u}}$:

$$\hat{\mathbf{u}} = \mathbf{r} + \hat{\mathbf{e}} = (\mathbf{u} + \mathbf{e}) + \hat{\mathbf{e}} = \mathbf{u} + (\mathbf{e} + \hat{\mathbf{e}}) \qquad (5.16)$$

Ako je procijenjen uzorak pogreške isti kao i stvarni uzorak pogreške, to jest, ako je $\mathbf{e}+\hat{\mathbf{e}}$, onda je procjena $\hat{\mathbf{u}}$ jednaka poslanoj kodnoj riječi **u**. S druge strane, ako je procjena pogreške netočna, dekoder će procijeniti kodnu riječi koja se nije prenosila pa ćemo imati *neotkrivenu pogrešku dekodiranja.*

## 5.6. 5.6. Primjer 5.4 Ispravak pogrešaka za $I_k$ u G (desno)

Pretpostavimo da se prenosila kodna riječi **u** = 101110, a dobije se vektor **r** = 001110. Prikažimo kako dekoder, koristeći tablicu 5.2. tj. preglednu tablicu sindroma, može ispraviti pogrešku.

***Rješenje*** (Dekoder 6,3 DESNO.circ)

$$\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k] \Rightarrow \mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T] \Rightarrow \mathbf{H}^T = \begin{bmatrix} \mathbf{I}_{n-k} \\ \hline \mathbf{P} \end{bmatrix}$$

$$\mathbf{G}_{DESNO} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Sindrom **r** računa se:

$$\mathbf{s} = [0\ 0\ 1\ 1\ 1\ 0] \cdot \mathbf{H}^T = [1\ 0\ 0]$$

Koristeći tablicu 6.2, uzorak pogreške što odgovara gornjemu sindromu, procjenjuje se da je:

$$\hat{\mathbf{e}} = 1\ 0\ 0\ 0\ 0\ 0$$

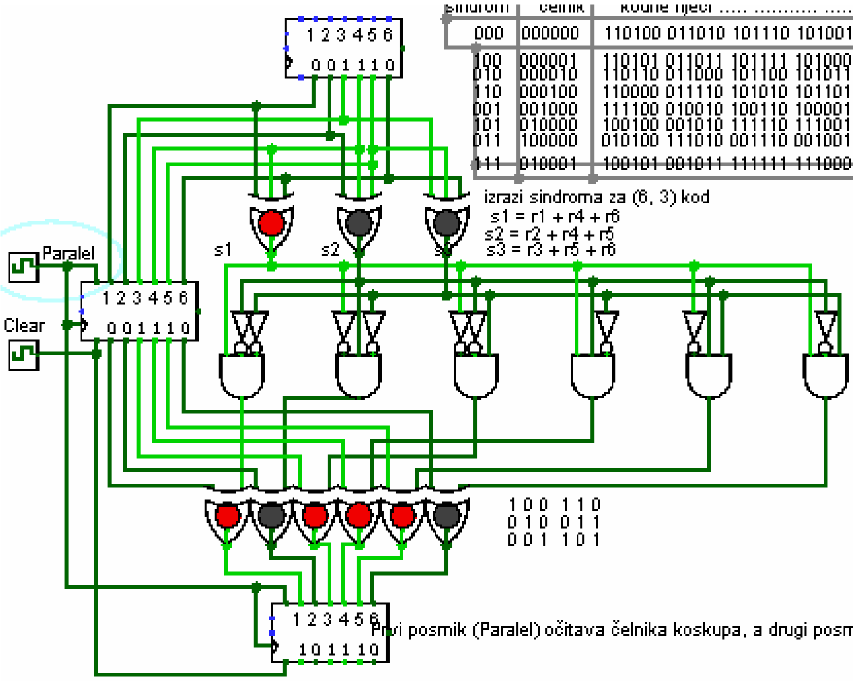
Korigiran vektor se zatim procjenjuje pomoću:

$$\hat{\mathrm{U}} = \mathbf{r} + \hat{\mathbf{e}} = 0\ 0\ 1\ 1\ 1\ 0 + 1\ 0\ 0\ 0\ 0\ 0 = 101110$$

$$\mathbf{s} = \mathbf{r}{\cdot}\mathbf{H}^T = [0\ 0\ 1\ 1\ 1\ 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = [1,\ 1{+}1,\ 1{+}1] = [1\ 0\ 0]$$

(sindrom oštećenoga kodnoga vektora)

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$




$$s_1 = r_1 + r_4 + r_6$$
$$s_2 = r_2 + r_4 + r_5$$
$$s_3 = r_3 + r_5 + r_6$$

$$\mathbf{H}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

*Simulacija: Dekoder (6, 3) DESNO.circ*

Budući da je procijenjen uzorak pogreške zapravo i stvaran uzorak pogreške u ovome primjeru, postupak ispravaka pogreška vodi do $\hat{\mathbf{u}} = \mathbf{u}$.

Uočite da se proces dekodiranja oštećene kodne riječi prvo otkrivanjem, a zatim ispravkom pogreške može usporediti s poznatom medicinskom analogijom. Bolesnik (potencijalno oštećena kodna riječ) ulazi u medicinsku ustanovu (*dekoder*). Liječnik ispituje fizičko stanje pacijenta testiranjem (množenje $\mathbf{H}^T$), kako bi pronašao simptom (*sindrom*). Zamislite da liječnik utvrdi karakteristične točke na pacijentu *x*-zrakama. Iskusan liječnik ne bi odmah prepoznao povezanost između simptoma i bolesti (uzorak pogreške), recimo tuberkuloza.

Liječnik-početnik možda bi morao koristiti priručnika za simptom pridružiti bolesti, tj., sindrom u odnosu na uzorak pogreške oblikovan AND vratima kao na slici  za (8, 2) kod, ili kao što je nacrtano za (6, 3) kod.

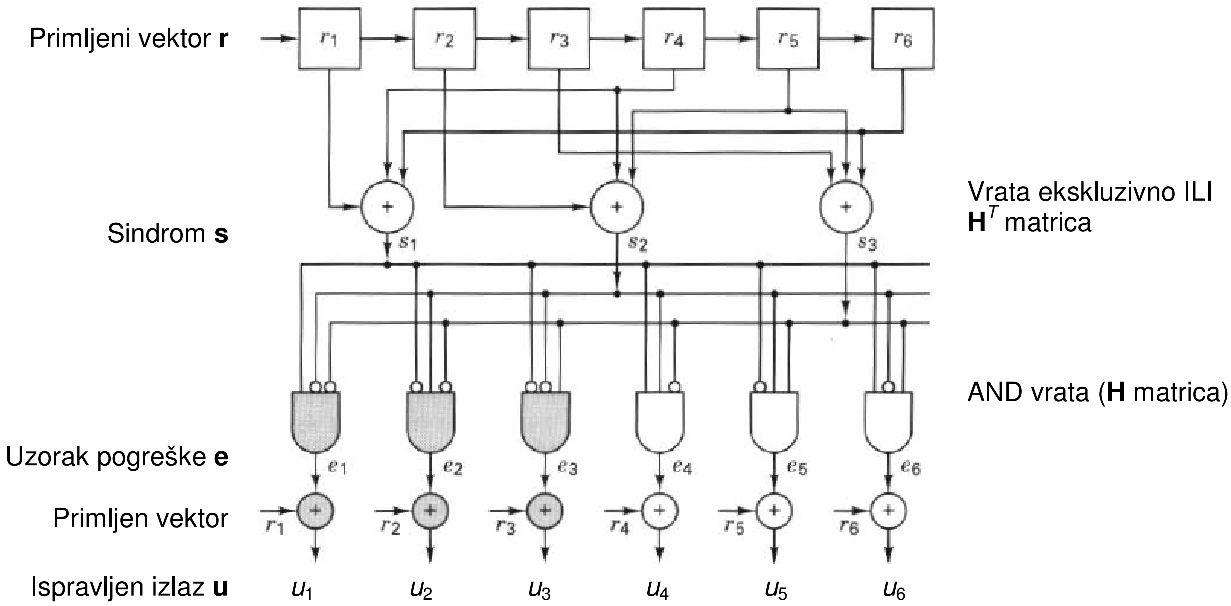
Završni korak osigurava odgovarajući lijek pacijentu, čime se izbjegava bolest (drugim riječima, metodom modulo-2 dodaje (zbraja) se uzorak pogreške oštećenoj kodnoj riječi, čime se ispravlja manjkava kodna riječ). U kontekstu binarnih kodova, ovdje se koristi neobična vrsta medicine (homeopatija). Pacijent se izliječi ponavljanjem primjene izvorne bolesti (Homeopatija liječi slično sličnim.)

## 5.7. 5.7. Primjena dekodera

Kada je kod kratak, kao u slučaju (6, 3) koda opisanoga u prethodnim poglavljima, dekoder se može primijeniti jednostavnim sklopovima. Razmotrite korake koje dekoder mora poduzeti:

1. izračunati sindrom,
2. pronaći uzorak pogreške i
3. uzorak pogreške i primljen vektor zbrojiti modulo-2 (a to uklanja pogrešku).

U primjeru 6.4, započeli smo oštećenim vektorom i vidjeli kako ovi koraci daju ispravljene kodne riječi. Sada, razmotrite krug na slici 5.2, što se sastoji od ekskluzivno ILI-vrata i I-vrata pa se može postići isti rezultat za bilo koji *uzorak* **jednostruke** *pogreške* u (6, 3) kodu.



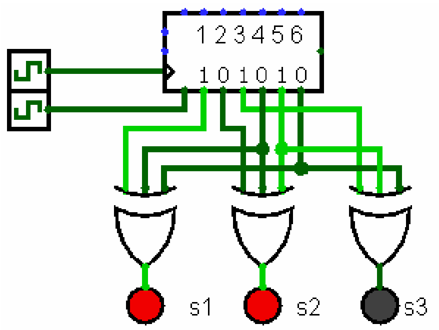

Simulacijski primjer: Dekoder 6,3 DESNO 146 245 356.circ

*Slika 5.2 Primjena (6, 3) dekodera .*

| *DESNO* | sindrom | čelnik | Kodne riječi … za jediničnu matricu – *DESNO*.......... ……… | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ispravne kodne riječi | 000 | 000000 | 110100 | 011010 | 101110 | 101001 | 011101 | 110011 | 000111 |
| Čelnici koskupova svih ispravljivih uzoraka jednostrukih pogrešaka | 101 | 000001 | 110101 | 011011 | 101111 | 101000 | 011100 | 110010 | 000110 |
| | 011 | 000010 | 110110 | 011000 | 101100 | 101011 | 011111 | 110001 | 000101 |
| | 110 | 000100 | 110000 | 011110 | 101010 | 101101 | 011001 | 110111 | 000011 |
| | 001 | 001000 | 111100 | 010010 | 100110 | 100001 | 010101 | 111011 | 001111 |
| | 010 | 010000 | 100100 | 001010 | 111110 | 111001 | 001101 | 100011 | 010111 |
| | 100 | 100000 | 010100 | 111010 | 001110 | 001001 | 111101 | 010011 | 100111 |
| Čelnik koskupa dvostrukih pogrešaka | ~~111~~ | ~~010001~~ | ~~100101~~ | ~~001011~~ | ~~111111~~ | ~~111000~~ | ~~001100~~ | ~~100010~~ | ~~010110~~ |

| LIJEVO | sindrom | čelnik | Kodne riječi … za jediničnu matricu – LIJEVO...................... | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ispravne kodne riječi | 000 | 000000 | 100110 | 010011 | 110101 | 001101 | 101011 | 011110 | 111000 |
| Čelnici koskupova svih ispravljivih uzoraka jednostrukih pogrešaka | 100 | 000001 | 100111 | 010010 | 110100 | 001100 | 101010 | 011111 | 111001 |
| | 010 | 000010 | 100101 | 010001 | 110111 | 001111 | 101001 | 011100 | 111010 |
| | 110 | 000100 | 100010 | 010111 | 110001 | 001001 | 101111 | 011010 | 111100 |
| | 001 | 001000 | 101110 | 011011 | 111101 | 000101 | 100011 | 010110 | 110000 |
| | 101 | 010000 | 110110 | 000011 | 100101 | 011101 | 111011 | 001110 | 101000 |
| | 011 | 100000 | 000110 | 110011 | 010101 | 101101 | 001011 | 111110 | 011000 |
| Čelnik koskupa dvostrukih pogrešaka | ~~111~~ | ~~010001~~ | ~~110111~~ | ~~000010~~ | ~~100100~~ | ~~011100~~ | ~~111010~~ | ~~001111~~ | ~~101001~~ |

Iz tablice 5.2 i jednadžbe 5.15, možemo napisati izraz za svaku od znamenki sindroma u smislu primljene znamenke kodne riječi kao




$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T$$

$$\mathbf{s} = [r_1, r_2, r_3, r_4, r_5, r_6] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{s} = [110101] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = 101$$

| | **000000** | **110100** | | izračunati sindromi | $\mathbf{r} \cdot \mathbf{H}^T$ | 000 | **000000** |
|---|---|---|---|---|---|---|---|
| | 000001 | [110101] | | (1+1+1), (1+1), (1) | 101 | 100 | 000001 |
| | 000010 | [110110] | | (1+1), (1+1+1), (1) | 011 | 010 | 000010 |
| | 000100 | [110000] | | (1), (1), (0) | 110 | 110 | 000100 |
| $\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T =$ | 001000 | [111100] | $\times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} =$ | (1+1), (1+1), (1) | 001 | 001 | 001000 |
| | 010000 | [100100] | | (1+1), (1), (0) | 010 | 101 | 010000 |
| | 100000 | [010100] | | (1), (1+1), (0) | 100 | 011 | 100000 |
| dvostruka | 010001 | [100101] | | (1), (1+1), (1) | 101 | 111 | 010001 |

| Množenje vektora matricom | | |
|---|---|---|
| za neispravan vektor [110101]: | | |
| 100 | | 1. redak matrice $\mathbf{H}^T$ |
| 010 | | 2. redak matrice $\mathbf{H}^T$ |
| 110 | | 4. redak matrice $\mathbf{H}^T$ |
| 101 | + | 6. redak matrice $\mathbf{H}^T$ |
| 101 | | konačan rezultat množenja |
| | | |

$$\begin{array}{r} \text{1. redak} \\ \text{2. redak} \\ \text{3. redak} \\ \text{4. redak} \\ \text{5. redak} \\ \text{6. redak} \end{array} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \mathbf{H}^T$$

i

$$s_1 = r_1 + r_4 + r_6,\ s_2 = r_2 + r_4 + r_5,\ s_3 = r_3 + r_5 + r_6$$

Koristimo ove izraze sindroma za ožičenje kruga na slici 5.2. Vrata ekskluzivno ILI rade istu operaciju kao aritmetika modulo-2 pa se stoga koristi isti simbol. Mala kružnica na završetku bilo kojega dovoda što ulazi u I vrata, ukazuje na logički komplement signala.

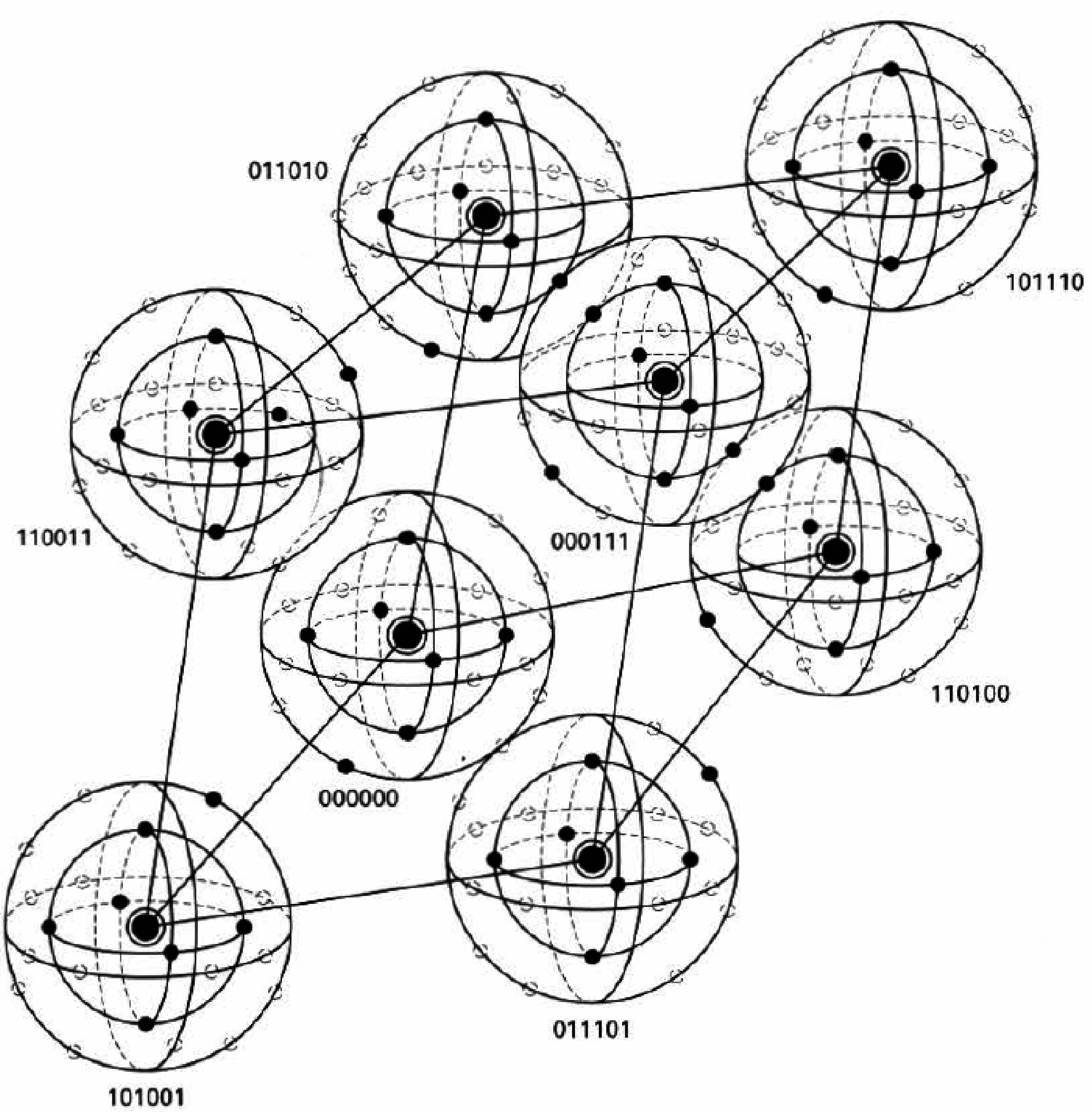
Oštećen signal ulazi istovremeno u dekoder na dva mjesta. Gornji dio sklopa, računa sindrom, a u donjemu dijelu, sindrom se pretvara u odgovarajući uzorak pogreške. Pogreška se uklanja tako da se zbroji primljenim vektorom, a rezultat je ispravljena kodna riječ.

Imajte na umu da, zbog razloga udžbenika, na slici 5.2 naglašeno su nacrtani koraci algebarskoga dekodiranja izračuna sindroma, uzorka pogreške i ispravljenoga izlaza. U stvarnome svijetu, $(n, k)$ kod obično se konfigurira u sustavnome obliku.

U tome slučaju, dekoder ne mora dostaviti cjelovitu kodnu riječ. Njegov izlaz sastojat će se od podataka samih bitova. Dakle, sklop na slici 5.2 postaje jednostavniji uklanjanjem vrata koja su prikazana osjenčano. Za duže kodove, takva provedba vrlo je složena pa povlaštene tehnike dekodiranja štede sklopove korištenjem serijskoga pristupa umjesto ove paralelne metode [4]. Također je važno naglasiti da je slika 5.2 konfigurirana samo za otkriti i ispraviti *uzorke jednostrukih pogrešaka* za (6, 3) kod. Kontrola pogreška za dvostruke uzorka pogreške zahtijeva dodatne krugove.

### 5.8.1. 5.8.1 PROSTOR VIZUALIZACIJE 6-TORKI

Slika 5.3 je vizualizacija osam kodnih riječi iz primjera.




*Slika 5.3 Primjer osam kodnih riječi u 6-strukome prostoru.*

Kodna riječ generira se iz linearne kombinacije tri nezavisne 6-torke . Kodna riječi oblikuje trodimenzijski podprostor. Slika 3 prikazuje takav podprostor potpuno zauzet s osam kodnih riječi (velike crne kružnice). Koordinate podprostora namjerno su nacrtane za naglasiti nepostojanje njihove okomitosti.

Slika 5.3 je pokušaj prikaza cijeloga prostora, što sadrži 64 6-torke, iako ne postoji precizan način crtanja ili izgradnje takvoga modela. Kuglasti slojevi ili školjke prikazane su oko svake kodne riječi. Svaki od unutarnjih slojeva koji se ne križaju predstavlja Hammingovu udaljenost od 1 s pripadajućom kodnom riječi, a svaki vanjski sloj prikazuje Hammingovu udaljenost od 2 kodne riječi. Veće udaljenosti nisu korisne u ovome primjeru. Za svaku kodnu riječi, dva prikazana sloja zauzeta su ometanim kodnim riječima. Ima 6 takvih točaka u svakoj unutarnjoj sferi (ukupno 48 točaka), što predstavlja šest mogućih vektora ometanih pogreškom od 1 bita povezane sa svakom kodnom riječi. Te ometane kodne riječi od 1 bita razlikuju se u smislu da ih se najbolje može povezati samo s jednom kodnom riječi pa se stoga mogu ispraviti. Kao što se vidi iz standardnoga polja tu je i jedan uzorak pogreške od 2 bita što se može ispraviti.

Uz svaku kodnu riječ može se pridodati ukupno $\binom{6}{2} = 15$ različitih uzoraka pogreške od 2 bita, ali samo jedan od njih, u našemu primjeru [0 1 0 0 0 1] uzorak pogreška, može se ispraviti. Ostalih 14 vektora uzoraka pogreški od 2 bita ne mogu se jedinstveno pridružiti samo jednoj kodnoj riječi. Ovi uzorci pogrešaka što se ne mogu popraviti vode vektorima koji su ekvivalent za vektore pogrešnih uzoraka od dvije ili više kodnih riječi. Na slici, svi ispravljivi (pedeset šest) uzorci kodnih riječi oštećeni s 1 ili 2 bita, prikazani su malim crnim kružnicama. Kodne riječi sa smetnjama što se ne mogu ispraviti, prikazane su malim bijelim kružnicama.

Slika 5.3 korisna je za vizualizaciju svojstva razreda kodova poznatih kao *savršeni kodovi*. Kod s $t$ ispravljivih pogrešaka naziva se savršen kod, ako njegova standardna polja imaju sve uzorke pogrešaka manje ili jednake $t$, a nemaju druge čelnike koskupova (nema sposobnost ispravke preostalih pogrešaka). U pogledu slike 5.3, za savršen kod što ispravlja $t$ pogrešaka prikladan je onaj kod što može, s maksimalnom mogućnošću dekodiranja, ispraviti sve ometane kodne riječi što zauzimaju sferu na Hammingovoj udaljenosti manjoj ili jednakoj $t$ od svoje izvorne kodne riječi, a ne može ispraviti bilo koji ometan vektor što zauzima sferu na udaljenostima većoj od $t$.

Slika 5.3 također je korisna za razumijevanje osnovnoga cilja traženja dobrih kodova. Za prostor želimo da je ispunjen sa što je moguće više kodnih riječi (učinkovito korištenje dodane zalihosti), a također bi željeli da ove kodne riječi budu što udaljenije jedna od druge. Očito da su ovi ciljevi proturječni.