OSNOVE DIGITALNE ELEKTRONIKE

Kodovi za otkrivanje i ispravljanje pogrešaka

Zdravko Kunić
zdravko.kunic@algebra.hr

Kodovi za otkrivanje i ispravljanje pogrešaka

Ishod 2 Definirati metode za otkrivanje i ispravljanje pogrešaka u prijenosu podataka. Otkriti i ispraviti pogreške u prijenosu podataka.

- The sleep-deprived university student
- Last-minute research paper
- 3 cups of coffee and a deadline
- And a…

… COMPUTER CRASH 😱

Fortunately, it does not happen very often because of:

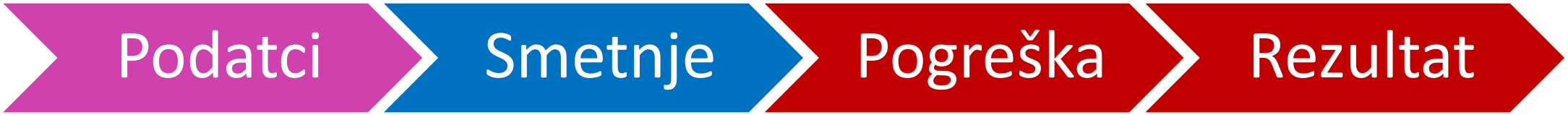
Error Detection and Correction Codes

Heroes that save your data when your coffee fails to save you ☕💥

Sadržaj predavanja

- Principi otkrivanja i ispravljanja pogrešaka
- Distanca
- Zalihost
- Paritet
- Hammingov kôd

Utjecaj smetnji na prijenos podataka



Pogreška = neželjena promjena bitova u kodnoj riječi

- Jednostruka pogreška
  - promjena vrijednosti jednog bita ($0 \rightarrow 1$ ili $1 \rightarrow 0$)
- Višestruka pogreška
  - promjena više bitova

Rezultat:

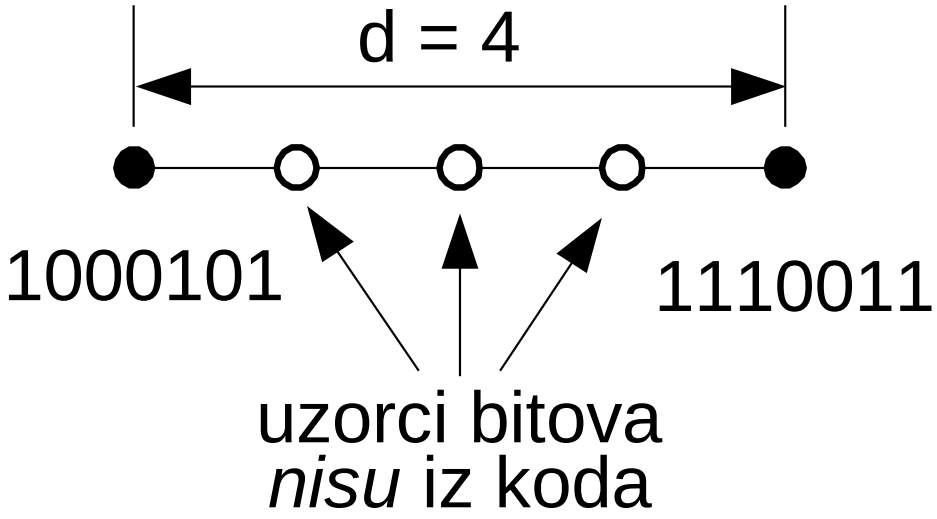
- Neupotrijebljena (zabranjena) kombinacija
  - Može se detektirati
- Neka druga (ispravna) kodna riječ
  - Nije moguće otkriti da je došlo do pogreške!

Osnova otkrivanja (i ispravljanja) pogrešaka

- Udaljenost između kodnih riječi (distanca)
- Paritetni bit(ovi)

Distanca

- Broj bitova koje treba promijeniti da se jedna kodna riječ pretvori u drugu
- Ako je broj promijenjenih bitova jednak distanci, tada pogreška ostaje neotkrivena !!!



Distanca

- Distanca ne mora biti jednaka za sve parove kodnih riječi
- Za “snagu” koda (mogućnost otkrivanja i korekcije pogrešaka), bitna je najmanja(minimalna) distanca d_{min}

Minimalna distanca d_{min}	Mogućnost	
	otkrivanja	ispravljanja
1	0	0
2	1	0
3	2	1
4	3	1
5	4	2

Minimalna distanca koda d_{min}

- Najmanji razmak između **bilo koje** dvije kodne riječi:
  - kôd 8421: $d_{min} = 1$
  - bikvinarni kôd: $d_{min} = 2$
  - Grayev kôd: $d_{min} = d = 1$
- Kôd pruža zaštitu od t pogrešaka

$$t = d_{min} - 1 \quad \rightarrow \quad d_{min} \geq (t+1)$$

- Primjer:
  - Kôd s $\boldsymbol{d_{min} = 2}$ omogućuje otkrivanje jednostruke pogreške

	2^3	2^2	2^1	2^0
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
	1	0	1	0
	1	0	1	1
	1	1	0	0
	1	1	0	1
	1	1	1	0
	1	1	1	1

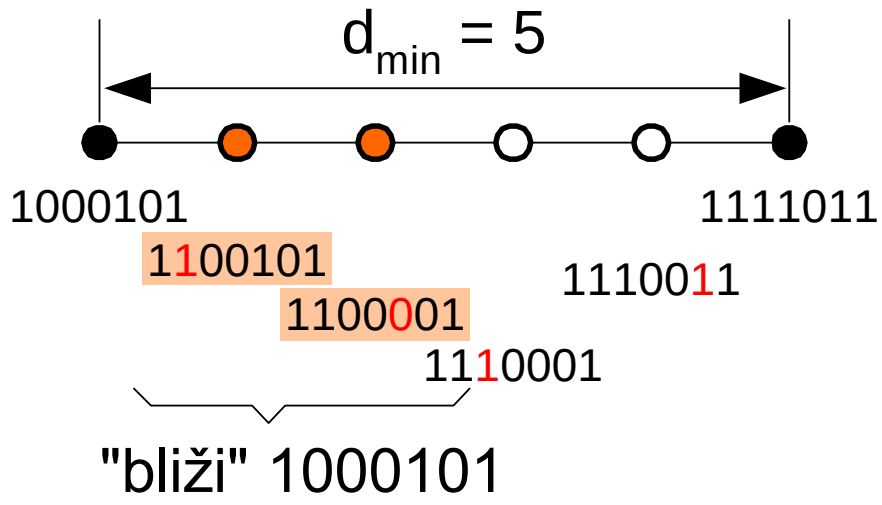
Zalihost (redundancija)

Veći broj bitova od minimalno potrebnih za prikaz informacije

- npr. bikvinarni kod koristi samo 10 od 128 mogućih kombinacija

Dvije skupine zaštitnih kodova

- Kodovi s mogućnošću **otkrivanja** pogrešaka, EDC (engl. Error Detecting Codes)
  - $d_{min} \geq t + 1$ za otkrivanje t pogrešaka
- Kodovi s mogućnošću **ispravljanja** pogrešaka ECC (engl. Error Correcting Codes):
  - $d_{min} \geq 2 \cdot t + 1$ za ispravljanje t pogrešaka



Paritet

- Najjednostavniji način zaštite
- Binarnoj riječi se dodaje paritetni bit

$$p\ b_6b_5b_4\ b_3b_2b_1b_0$$

  - nova kodna riječ ima **paran ili neparan broj jedinica**:
    - **paran/neparan paritet**
  - jedan paritetni bit nije dovoljan za ispravljanje pogreške već samo za detekciju da je do pogreške došlo

  - dodavanje paritetnog bita povećava distancu kôda

Primjer

- Paritetni bit određujemo tako da ukupan broj jedinica (uključujući i paritetni bit) bude:
  - **paran** (parni paritet)
  - **neparan** (neparni paritet)

Znak		Paritet	
		parni	neparni
A	100 0001	0 100 0001	1 100 0001
a	110 0001	1 110 0001	0 110 0001
*	010 1010	1 010 1010	0 010 1010

Višestruko ispitivanje pariteta

- Povećava snagu zaštite!
  - podržava veći broj paritetnih ispitivanja
  - koristi veći broj zaštitnih bitova (veća zalihost)
- Više mogućnosti implementacije:
  - dvodimenzijski kod
  - Hammingov kod

Dvodimenzijski kod

- 2D matrica informacijskih bitova ("pravokutni" kod)
- Uzdužna i poprečna paritetna zaštita:
  - kodna riječ ← paritetni bit retka (p_r)
  - stupac ← uzdužna paritetna zaštita (p_s)
    - Longitudinal Redundancy Check, LRC
  - cijeli blok kodnih riječi ← paritetna riječ
    - Block Check Character, BCC
- Može otkriti i ispraviti jednostruku pogrešku
  - Pogrešan paritet u retku i stupcu (koordinate za lociranje pogreške)
- Može samo otkriti (ne i ispraviti) dvostruku pogrešku
  - Pogrešan paritet samo u retku ili samo u stupcu (jedna koordinata nedostaje za točno lociranje pogrešnog bita)

Primjer

Paritetni bitovi redaka

p_r

Kod bez zaštite

0	1	0	0	0	0	0	1
1	1	1	0	0	0	0	1
1	0	1	0	1	0	1	0
	0	0	0	1	0	1	0

p_s

Paritetni bitovi stupaca

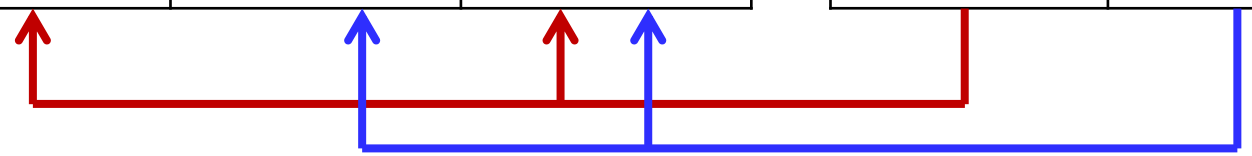
Kodovi za ispravljanje pogrešaka

- Omogućuju točno određivanje mjesta pogreške u zaštićenoj kodnoj riječi

- **kodna riječ** = informacijski bitovi + ispitni (zaštitni) bitovi
- Mjesto pogreške se otkriva **višestrukim** paritetnim ispitivanjem **različitih** kombinacija informacijskih i ispitnih bitova

Princip višestrukog ispitivanja pariteta*

Informacijski bitovi			Paritetni bitovi	
i_1	i_2	i_3	p_1	p_2
1	0	1	1	0
0	1	1	0	1



* Vrijednost 1 u tablici predstavlja pripadnost grupi provjere pariteta

- Nedostatak ovog principa: pogreška u paritetnim bitovima ostaje neotkrivena!

Princip višestrukog ispitivanja pariteta*

Informacijski bitovi			Paritetni bitovi		Dodatni paritetni bit
i_1	i_2	i_3	p_1	p_2	p_3
1	0	1	1	0	0
0	1	1	0	1	0
1	1	0	0	0	1

* Vrijednost 1 u tablici predstavlja pripadnost grupi provjere pariteta

Pogreška u jednom **informacijskom** bitu rezultira **dvjema** pogreškama pariteta

Pogreška u **paritetnom** bitu rezultira **jednom** pogreškom pariteta

Princip višestrukog ispitivanja pariteta*

	i_1	i_2	i_3	p_1	p_2	p_3
G1	0	1	1	1	0	0
G2	1	0	1	0	1	0
G3	1	1	0	0	0	1

Pogrešan i_1 – neispravni pariteti grupa G2, G3
Pogrešan i_2 – neispravni pariteti grupa G1, G3
Pogrešan i_3 – neispravni pariteti grupa G1, G2
Pogrešan p_1 – neispravan paritet grupe G1
Pogrešan p_2 – neispravan paritet grupe G2
Pogrešan p_3 – neispravan paritet grupe G3

* Vrijednost 1 u tablici predstavlja pripadnost grupi provjere pariteta

Hammingov kôd

Sustavni mehanizam za izgradnju niza kodova za ispravljanje pogrešaka (R.W. Hamming, 1950.)

- Princip:
  višestruko (nezavisno) paritetno ispitivanje
- Bolja efikasnost kodiranja, manja zalihost
- Naročito popularan za ispravljanje jednostruke pogreške
  - tipična primjena: memorijski sklopovi

Hammingov kôd

- Nezavisna paritetna ispitivanja
  - Ne mogu se dobiti kombinacijom preostalih
- Izgrađuje kodnu riječ tako da:
  - svaki zaštitni bit "pokriva" (= štiti) drugi podskup bitova podatka
  - svaki bit podatka zaštićen je s više zaštitnih bitova
- Pri sastavljanju zaštitne kodne riječi Hamingov kod miješa zaštitne bitove s podatkovnima
- Minimalna distanca ovog koda je 3 (razlika u vrijednostima bitova između bilo koje dvije različite kodne riječi)
  - Omogućava otkrivanje najviše dvije, a ispravljanje najviše jedne pogreške

Hammingov kôd

- Kod Hammingovog (7,4) kôda na 4 informacijska bita dolaze 3 zaštitna (paritetna) bita
- Paritetni bitovi u kodnim riječima Hammingovog koda dolaze na mjesta koja su potencije broja 2 (1,2,4,8,…)
- u 7-bitnoj kodnoj riječi kod Hammingovog (7,4) koda:
  - **prvi**, **drugi** i **četvrti** bit su **zaštitni** bitovi (`P1,P2,P4`)
  - preostali bitovi su **informacijski** (`I3,I5,I6,I7`)

P1 P2 I3 P4 I5 I6 I7

Hammingov kôd

- Paritetni bitovi mogu se dodavati počevši od lijeve ili desne strane, ovisno o poziciji bita najmanje važnosti (LSB - Least Significant Bit).

P1 P2 I3 P4 I5 I6 I7

LSB on the left

I7 I6 I5 P4 I3 P2 P1

LSB on the right

Svi primjeri u ovom predavanju imaju LSB na lijevoj strani.

Hammingov kod

- Paritetni bitovi (7,4) kôda određuju se na sljedeći način:
  - P_1 = b3 ⊕ b5 ⊕ b7
  - P_2 = b3 ⊕ b6 ⊕ b7
  - P_3 = b5 ⊕ b6 ⊕ b7
- operacija ⊕ predstavlja logičku operaciju XOR
  - zbrajanje po modulu 2 (isključivo ILI)

$$Y = X \oplus N$$

X	⊕	N	=	Y
0	⊕	0	=	0
0	⊕	1	=	1
1	⊕	0	=	1
1	⊕	1	=	0

Hammingov kôd - primjer

Zaštita riječi **1010** Hammingovim kodom (7,4) s parnim paritetom:

b1	b2	b3	b4	b5	b6	b7
P1	P2	I1	P3	I2	I3	I4
		1		**0**	**1**	**0**

- $P_1 = b3 \oplus b5 \oplus b7 = \mathbf{1} \oplus \mathbf{0} \oplus \mathbf{0} = \mathbf{1}$
- $P_2 = b3 \oplus b6 \oplus b7 = \mathbf{1} \oplus \mathbf{1} \oplus \mathbf{0} = \mathbf{0}$
- $P_3 = b5 \oplus b6 \oplus b7 = \mathbf{0} \oplus \mathbf{1} \oplus \mathbf{0} = \mathbf{1}$

b1	b2	b3	b4	b5	b6	b7
P1	P2	I1	P3	I2	I3	I4
1	**0**	**1**	**1**	**0**	**1**	**0**

Hammingov kôd - algoritam

- Odrediti broj potrebnih zaštitnih bitova na temelju broja podatkovnih bitova
- Rezervirati pozicije zaštitnih bitova x_i u kodnoj riječi ($x_i \rightarrow 2^i$, gdje je i element skupa $[0,n]$)
- Izračunati zaštitne bitove

Ako je u komunikacijskom kanalu između dva sustava došlo do pogreške na nekom od bitova, moguće je otkriti i ispraviti pogrešan bit, čak i ako se pogreška pojavila na samom zaštitnom bitu.

Računanje zaštitnih bitova

- Svaki zaštitni bit se računa na temelju točno određenih podatkovnih bitova, a najlakše ih je odrediti pomoću indeksa bita kojeg računamo
- Algoritam je moguće opisati s "od n, po n, alternirano n".
  - Npr. počevši od zaštitnog bita na indeksu 4 uzimamo 4 bita, zatim 4 preskačemo pa opet uzimamo 4 i tako do zadnjeg raspoloživog bita.

Indeks	**1**	**2**	3	**4**	5	6	7	**8**	9	10	11
	p1	**p2**	i1	**p3**	i2	i3	i4	**p4**	i5	i6	i7
G4	0	0	0	0	0	0	0	1	1	1	1
G3	0	0	0	1	1	1	1	0	0	0	0
G2	0	1	1	0	0	1	1	0	0	1	1
G1	1	0	1	0	1	0	1	0	1	0	1

Računanje vrijednosti zaštitnih bitova

- Odrediti bitove koji pripadaju pojedinom zaštitnom bitu
- Izračunati **modulo 2** sumu (XOR) za svaki zaštitni bit
  - Ako je rezultat **paran**, **direktno** ga zapisujemo na mjesto zaštitnog bita
  - Ako je rezultat **neparan**, na mjesto zaštitnog bita zapisujemo **invertirani** rezultat

Implementacija

- Paritetni bitovi su na mjestima (1, 2, 4, 8,..)
- Ostali bitovi su informacijski
- Primjer: kod (11,7)
  - ukupno **11** bitova, od čega **7** nose podatke
  - korisno za zaštitu ASCII-znakova

1	**2**	3	**4**	5	6	7	**8**	9	10	11
p1	**p2**	i1	**p3**	i2	i3	i4	**p4**	i5	i6	i7

Primjer - zaštita dekadskih znamenki

- **Informacijski bitovi** služe za binarni prikaz dekadskih znamenki (4 bita)
- **Ispitni (zaštitni) bitovi** su dodatni bitovi za paritet (3 bita)
- Ispitivanje pariteta izvodi se za svaki zaštitni bit (3x)
- Ako je ispitivanje **uspješno**, rezultat se označava s **0**
- Ako je ispitivanje **neuspješno**, rezultat se označava s **1**
- 3-bitni broj dobiven ispitivanjem označava mjesto pogreške
- Ako su sva tri ispitna bita nula, nema pogreške

Konstrukcija koda

- Na mjesto informacijskih bitova dolaze bitovi kodnih riječi
- Primjer: nezaštićena (originalna) kodna riječ je: **1001**
  - u Hammingovom kodu riječ 1001 ima sljedeću strukturu:

1	2	3	4	5	6	7
p1	**p2**	i1	**p3**	i2	i3	i4
		1		**0**	**0**	**1**

	p1	p2	i1	p3	i2	i3	i4	p4	i5	i6	i7
G_4	0	0	0	0	0	0	0	1	1	1	1
G_3	0	0	0	1	1	1	1	0	0	0	0
G_2	0	1	1	0	0	1	1	0	0	1	1
G_1	1	0	1	0	1	0	1	0	1	0	1

1	2	3	4	5	6	7
p1	p2	i1	p3	i2	i3	i4
0	0	1	1	0	0	1

Sindrom

- Uzorak zaštitnih bitova koji ukazuje na mjesto pojave pogreške
- Obično ga i pišemo u obliku rednog vektora $s = G_4G_3G_2G_1$, gdje je G modulo 2 suma pripadajućih binarnih znamenaka

	p1	**p2**	i1	**p3**	i2	i3	i4	**p4**	i5	i6	i7
G_4	0	0	0	0	0	0	0	1	1	1	1
G_3	0	0	0	1	1	1	1	0	0	0	0
G_2	0	1	1	0	0	1	1	0	0	1	1
G_1	1	0	1	0	1	0	1	0	1	0	1

Primjer otkrivanja pogreške

Originalna riječ: **0011001**

Riječ s pogreškom: **1011001**

b1	b2	b3	b4	b5	b6	b7
1	**0**	**1**	**1**	**0**	**0**	**1**

- Ispitivanje (parnog) pariteta svake grupe:
  - G_1 = b1 ⊕ b3 ⊕ b5 ⊕ b7 = **1 + 1 + 0 + 1 = 1** → neispravan paritet
  - G_2 = b2 ⊕ b3 ⊕ b6 ⊕ b7 = **0 + 1 + 0 + 1 = 0** → ispravan paritet
  - G_3 = b4 ⊕ b5 ⊕ b6 ⊕ b7 = **1 + 0 + 0 + 1 = 0** → ispravan paritet
- Sindrom $G_3G_2G_1$ = **001_2 = 1_{10}** → pogrešan je 1. bit (p1)

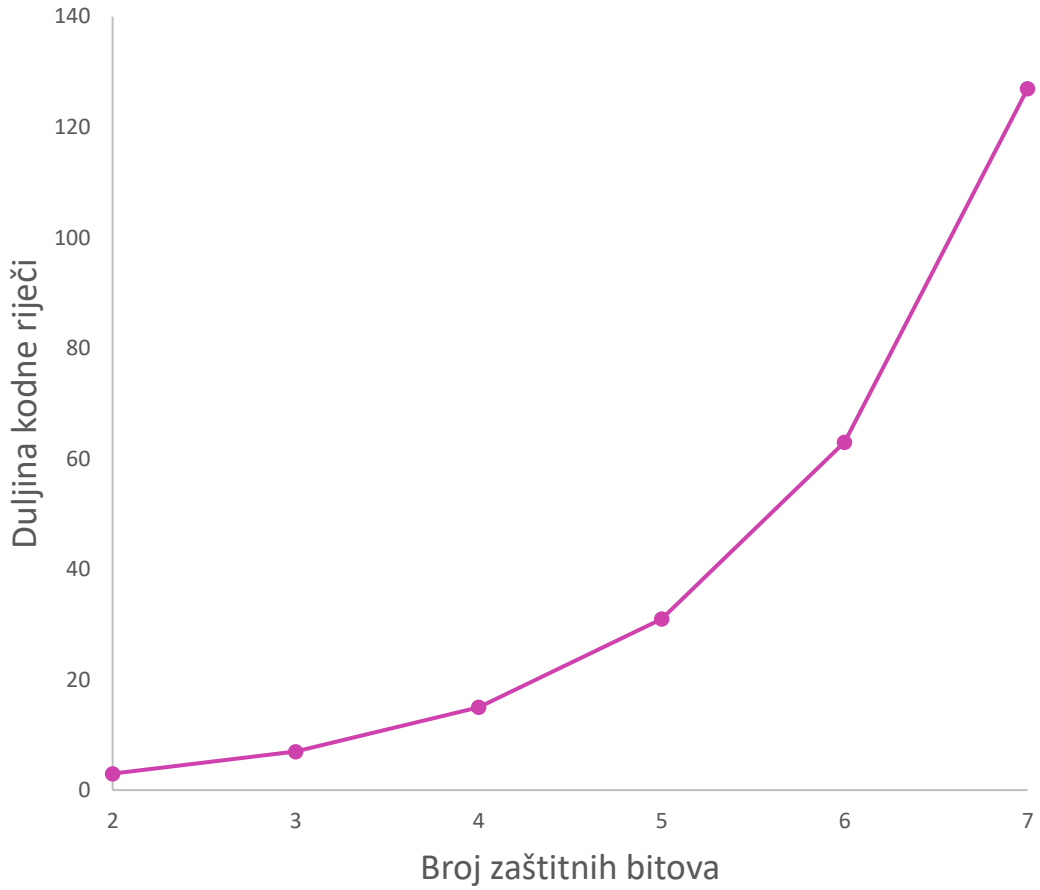
Odnos zaštitnih i informacijskih bitova

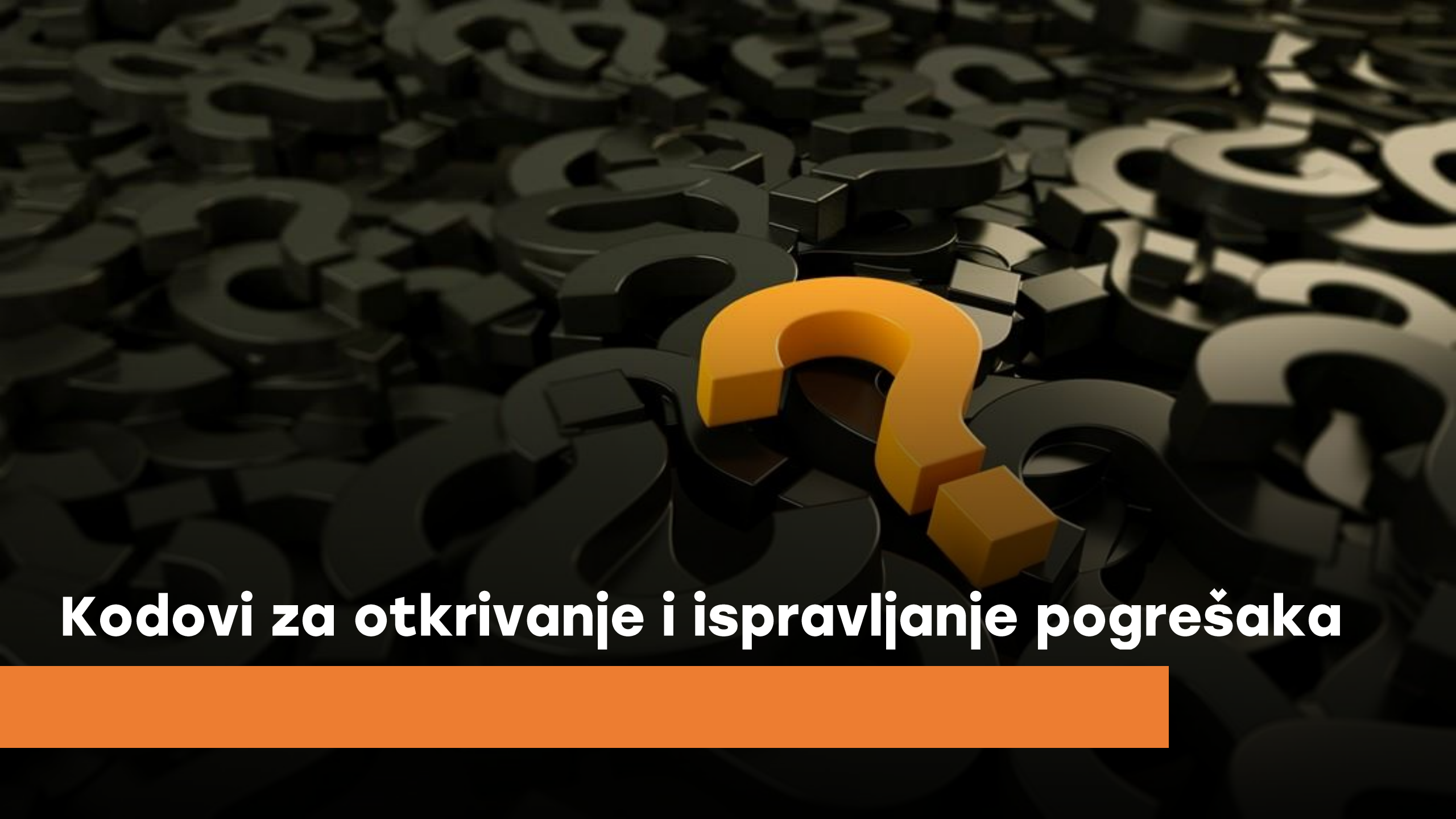
- Zaštićena kodna riječ se sastoji od informacijskih i zaštitnih bitova
- Duljinu zaštićene kodne riječi određuju broj zaštitnih bitova p i broj informacijskih bitova i, odnosno $n = p + i$
- Ispitni broj s p paritetnih bitova može prikazati pogrešku na ukupno 2^p-1 pozicija jer u ispitnoj tablici nema nule.
- Za uspješno otkrivanje jednostruke pogreške vrijedi:

$$2^p - 1 \geq p + i \quad \rightarrow \quad 2^p \geq p + i + 1$$

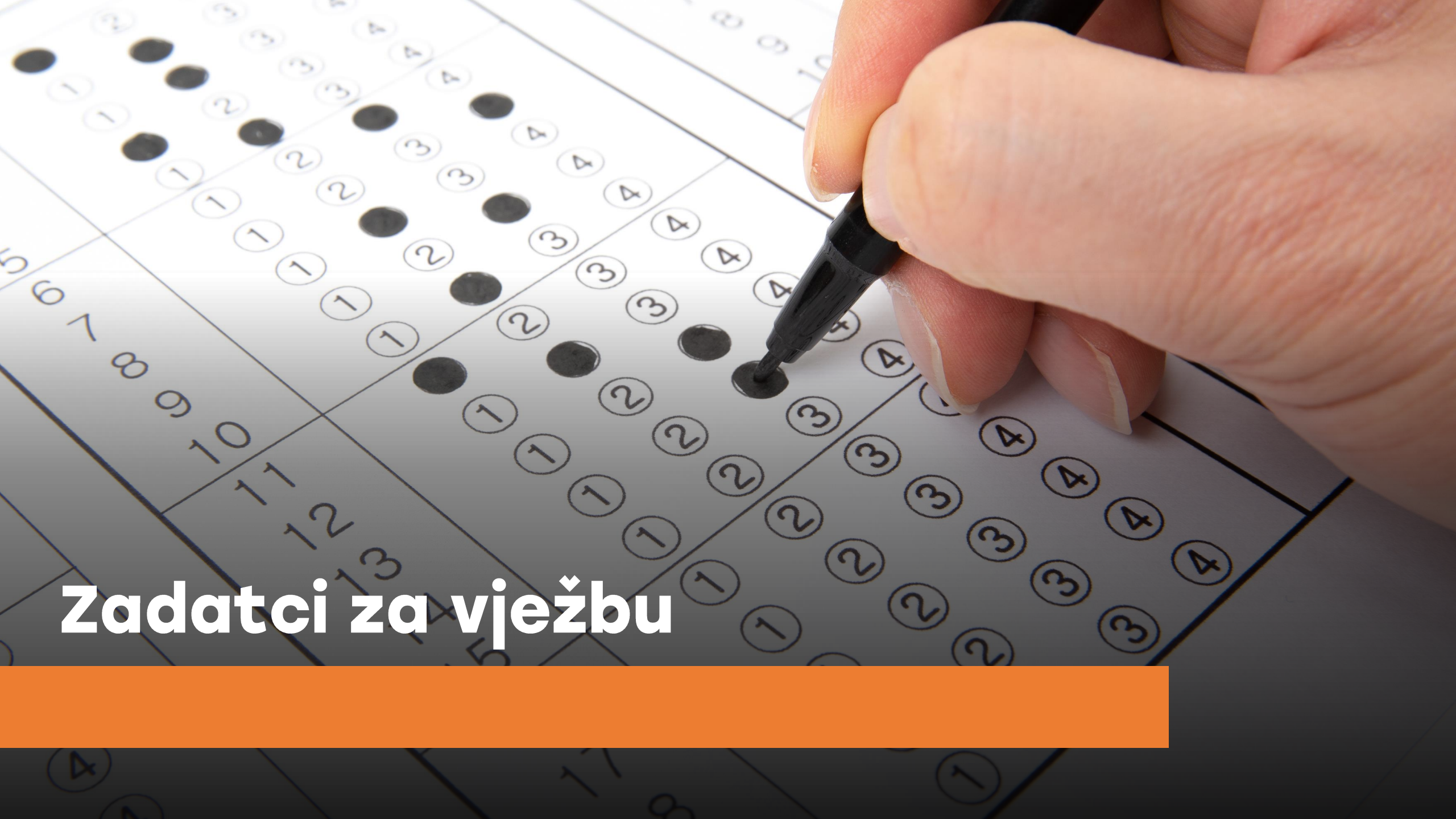
Odnos zaštitnih i informacijskih bitova

Broj informacijskih bitova	Broj zaštitnih bitova	Duljina kodne riječi
1	2	3
2 - 4	3	7
5 - 11	4	15
12 - 26	5	31
27 - 57	6	63
58 - 120	7	127





Kodovi za otkrivanje i ispravljanje pogrešaka



Zadatci za vježbu

Zadatak

- Zaštitite riječ **1001** Hammingovim kodom
  - Primijenite parni paritet
- Koliko ispitnih bitova ste dodali u kodnu riječ?

Rješenje (1001)

b1	b2	b3	b4	b5	b6	b7
P1	P2	I1	P3	I2	I3	I4
		1		0	0	1

- $P_1 = b1 \oplus b3 \oplus b5 \oplus b7 =$
  $? \oplus 1 \oplus 0 \oplus 1 = 0$
- $P_2 = b2 \oplus b3 \oplus b6 \oplus b7 =$
  $? \oplus 1 \oplus 0 \oplus 1 = 0$
- $P_3 = b4 \oplus b5 \oplus b6 \oplus b7 =$
  $? \oplus 0 \oplus 0 \oplus 1 = 1$

b1	b2	b3	b4	b5	b6	b7
P1	P2	I1	P3	I2	I3	I4
0	0	1	1	0	0	1

Zadatak

- Zaštitite riječ **1100 0011** Hammingovim kodom
  - Primijenite parni paritet
- Koliko ispitnih bitova ste dodali u kodnu riječ?

Rješenje

b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12
P1	P2	I1	P3	I2	I3	I4	P4	I5	I6	I7	I8
		1		**1**	**0**	**0**		**0**	**0**	**1**	**1**

- P_1 = b1 ⊕ b3 ⊕ b5 ⊕ b7 ⊕ b9 ⊕ b11 =
  ? ⊕ **1** ⊕ **1** ⊕ **0** ⊕ **0** ⊕ **1** = **1**
- P_2 = b2 ⊕ b3 ⊕ b6 ⊕ b7 ⊕ b10 ⊕ b11 =
  ? ⊕ **1** ⊕ **0** ⊕ **0** ⊕ **0** ⊕ **1** = **0**
- P_3 = b4 ⊕ b5 ⊕ b6 ⊕ b7 ⊕ b12 =
  ? ⊕ **1** ⊕ **0** ⊕ **0** ⊕ **1** = **0**
- C_4 = b8 ⊕ b9 ⊕ b10 ⊕ b11 ⊕ b12 =
  ? ⊕ **0** ⊕ **0** ⊕ **1** ⊕ **1** = **0**

b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12
P1	P2	I1	P3	I2	I3	I4	P4	I5	I6	I7	I8
1	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**1**

Zadatak

- Na ulaz digitalnog sustava primljena je riječ zapisana u Hammingovom kodu **0001 1010 1010**
  - Primijenjen je parni paritet
- Koliko ispitnih bitova sadrži kodna riječ?
- Utvrdite eventualnu pogrešku

Rješenje

b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12
P1	P2	I1	P3	I2	I3	I4	P4	I5	I6	I7	I8
0	0	0	1	1	0	1	0	1	0	1	0

- $C_1 = b1 \oplus b3 \oplus b5 \oplus b7 \oplus b9 \oplus b11 =$
  $0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0$
- $C_2 = b2 \oplus b3 \oplus b6 \oplus b7 \oplus b10 \oplus b11 =$
  $0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$
- $C_3 = b4 \oplus b5 \oplus b6 \oplus b7 \oplus b12 =$
  $1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$
- $C_4 = b8 \oplus b9 \oplus b10 \oplus b11 \oplus b12 =$
  $0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$

$C_4C_3C_2C_1 = 0100_2 \rightarrow$ **pogrešan je 4. bit**

Primjeri zadataka s prethodnih ispita*

Ishod učenja 2 – 9 bodova - 25 min

1. [I2_M, 1 bod] Metodom neparnog pariteta osigurati pravilan prijenos podataka:
a) _0101111, b) _1101000; (0,5 bodova za svaki točan odgovor)

2. [I2_M, 2 boda] Niz dekadskih znamenki: 0,3,7,9 napisati u kodu XS-3 te zatim cijeli blok zaštititi uzdužnim i poprečnim parnim paritetom. (1 bod za točno napisane znamenke u zadanom kodu; 1 bod za točan paritet)

3. [I2_M, 3 boda] U zadane informacije ubaciti bitove provjere - zaštititi Hammingovim kodom (7,4) (0,5 bodova za korektno izračunate zaštitne bitove; 0,5 bodova za svaku korektno napisanu zaštićenu kodnu riječ):

a) 1110101 b) 1011010 c) 1010110

4. [I2_Ž, 3 boda] Informacija **0011 1010 0011** zaštićena je Hammingovim kodom. Treba otkriti eventualnu pogrešku u prijenosu informacije i ispraviti je. (1,5 bodova za korektan postupak i otkrivenu pogrešku; 1 bod za korektno napisanu ispravljenu informaciju zaštićenu Hammingovim kôdom; 0,5 bodova za korektno napisanu informaciju bez bitova provjere)

* Primjer ispita je ilustrativan. Vrste zadataka na budućim brzim testovima i ispitima mogu biti drugačije.

LITERATURA:

- Uroš Peruško: Digitalni sustavi
  - Str. 68 - 76