

Zaštitno kodiranje signala

Laboratorijska vježba 2

SP

XOR vrata i paritet

Student:

prezime	Ime	mat. broj
	Laboratorij	Datum

2. 2. XOR VRATA I PARITET

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:

- 1.6. Utjecaj pogrešaka na digitalni komunikacijski sustav
- 1.7. Pojam pariteta
 - 1.7.1. Iteracijska provjera pariteta
 - 1.7.2. Paritet skupine bitova
- 1.8. Otkrivanje pogrešaka
- 1.9. Pojam blok-kodova
- 10.6. Vrste provjera zalihosti
- 1.10. Pregled ostalih uvedenih pojmova
 - 10.6.1. Uzdužna provjera zalihosti
 - 10.6.1.1. Otkrivanje pogreška
 - 10.6.2. Bit pariteta/okomita provjera zalihosti (VRC)
 - 10.6.3. Uzdužna provjera zalihosti (LRC)
 - 10.6.4. Ciklička provjera zalihosti (CRC)

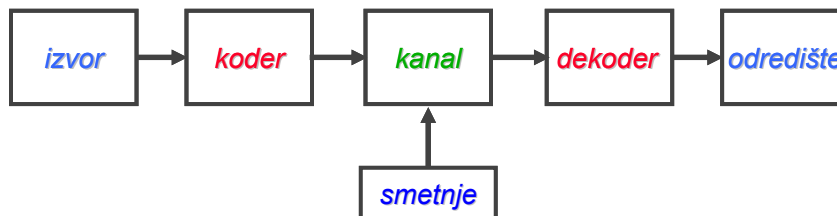
Sadržaj:

2.1.	2.1. Utjecaj pogrešaka na komunikacijski sustav	3
2.1.1.	2.1.1. LABORATORIJSKA VJEŽBA 2 (OD POGLAVLJA: 1.6. DO 1.9).....	3
2.1.1.1.	2.1.1.1. Primjer:.....	3
2.1.1.2.	2.1.1.2. Zadatak	3
2.1.1.3.	2.1.1.3. Zadatak	3
2.1.1.4.	2.1.1.4. Zadatak	4
2.1.1.5.	2.1.1.5. Zadatak	4
2.1.1.6.	2.1.1.6. Prijenos bitova kanalom	5
2.1.2.	2.2.2. POJAM PARITETA.....	5
2.1.3.	2.1.3. ITERACIJSKA PROVJERA PARITETA, PARAN I NEPARAN PARITET	6
2.1.3.1.	2.1.3.1. Zadatak	6
2.1.4.	2.1.4. LABORATORIJSKA VJEŽBA: PARITET	7
2.1.4.1.	2.1.4.1. Zadatak	8
2.1.4.2.	2.1.4.1. Zadatak	9
2.1.4.3.	Pitanja:	9
2.2.	2.2. Unesite svoja zapažanja o vježbi	10

2.1. 2.1. Utjecaj pogrešaka na komunikacijski sustav

2.1.1. 2.1.1. LABORATORIJSKA VJEŽBA 2 (OD POGLAVLJA: 1.6.DO 1.9 U SKRIPTI)

Binarne znamenke, nazvane *bitovi*, prenose se od izvora (*odašiljač*) prema odredištu (*prijemnik*). Na putu od odašiljača do prijemnika bitovi prolaze kroz medij (*kanal*). Prolazom kroz kanal, bitovi su podložni šumovima što uводи *pogreške* u smislu da neki primljeni bitovi, na mjestima podudarnim s položajima u poslanoj poruci, imaju suprotnu vrijednost. Bitove se prenose blokovima određene duljine, a ne umeću se i ne izostavljaju iz bloka tijekom prijenosa. Uvijek postoji sinkronizacija bloka (prijemnik zna gdje jedan blok završava i gdje drugi počinje).



Slika 1.2: Shannonov jednosmjerni komunikacijski kanal

2.1.1.1. 2.1.1.1. Primjer:

Šalje se blok 1101001, a primi se 1001101 blok. Primljen blok razlikuje od onoga poslanoga bloka u bitovima što se nalazi na drugome i petome mjestu, računajući s lijeva. (Ovi bitovi su 0 odnosno 1 u poslanome, a 1 odnosno 0 u primljenome bloku). Za nastavak, student treba naučiti logičku izričito ILI - EXOR (*exclusive OR*) operaciju, iz navedenoga dodatka (Logisim™™). Ona je dobro je poznate iz logike. Operacija EXOR u cijelome tekstu obilježava se znakom $\oplus$ (umjesto znakom +), jer znak $\oplus$ označava ILI operaciju.

2.1.1.2. 2.1.1.2. Zadatak

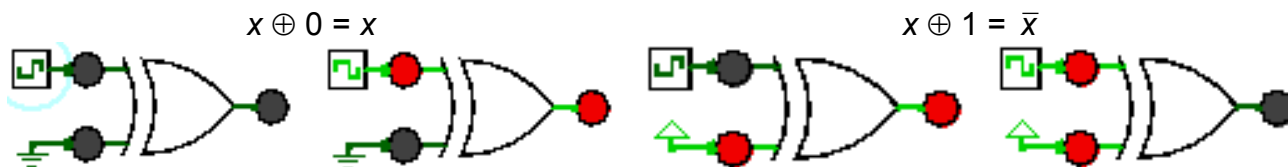
Sljedeće primjere napravite u simulacijskome programu LogiSim 2.7.1:



Operacija XOR može se sažeti tvrdnjom da operacija XOR bîta x s 0 daje bit vrijednosti x , a operacija XOR bîta x s 1 rezultira bîtom vrijednosti $\bar{x}$, gdje je $\bar{x}$ označava komplement od x . Sve slike u skripti predstavljaju ujedno i simulacijske primjere za laboratorijske vježbe ovoga kolegija.

2.1.1.3. 2.1.1.3. Zadatak

Sljedeće primjere napravite u simulacijskome programu LogiSim 2.7.1



Zbroj dvaju blokova bitova (iste duljine) predstavlja rezultat dobiven operacijom XOR na razini bitova dvaju blokova. Rezultat je također blok iste duljine kao i svaki pribrojnik. Njegov prvi element je rezultat dobiven operacijom XOR prvih elemenata dvaju blokova. Njegov drugi element je rezultat dobiven operacijom XOR drugoga elementa obaju blokova, itd. Rezultirajući blok onda ima vrijednost 1 na onim mjestima gdje se dva bloka razlikuju, a vrijednost 0 na mjestima gdje imaju istu vrijednost. Kao primjer, razmotrimo dva bloka 01001101 i 10111000. Ovi blokovi razlikuju se u 1., 2., 3., 4., 6. i 8. mjestu. Njihov zbroj je 11110101.

2.1.1.4. 2.1.1.4. Zadatak

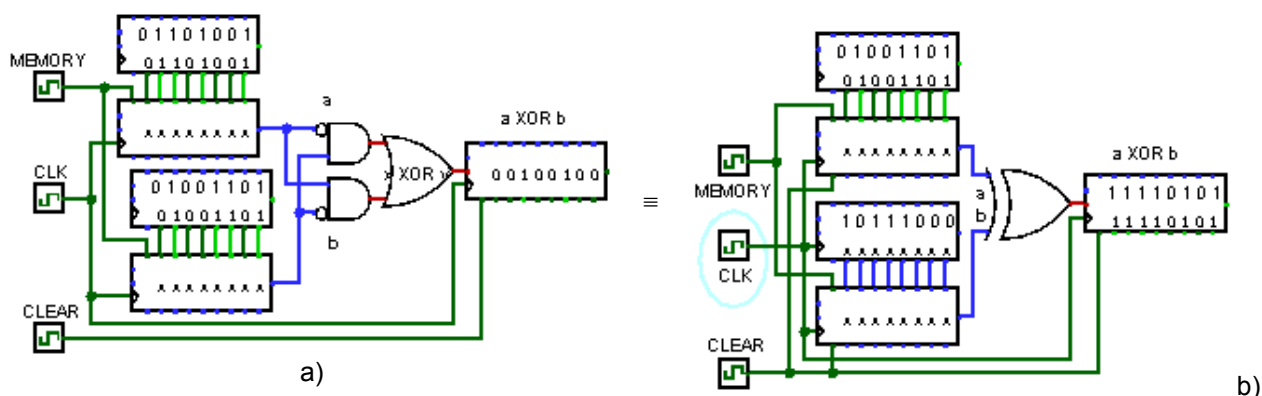
U simulacijskome alatu Logisim™ nacrtajte sljedeće sklopove i provjeriti navedene tvrdnje! Rezultate upišite u priloženu tablicu.

sklop	0 + 0	0 + 1	1 + 0	1 + 1	Zaključak
sklop	0 + 0	0 + 1	Zaključak		
sklop	1 + 0	1 + 1	Zaključak		

2.1.1.5. 2.1.1.5. Zadatak

Nacrtajte sklopove na donjoj slici i dokažite točnost sljedećega zbroja.

$$\begin{array}{r|l}
 \mathbf{a} = [01101001] & \\
 \mathbf{b} = [01001101] & \oplus \\
 \hline
 \mathbf{c} = [00100100] &
 \end{array}$$



Pitanja:

1. Od kojih osnovnih komponenti je sastavljen sklop a). Nabrojite ih?

.....

.....

.....

2. Od kojih osnovnih komponenti je sastavljen sklop b). Nabrojite ih?

.....

.....

3. Koju razliku uočavate između sklopova a) i b)?

.....

.....

4. Koja je sličnost sklopova a) i b)?

.....

.....

5. Koje je suštinsko svojstvo XOR operatora?

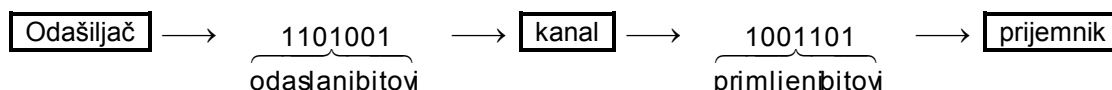
.....

.....

2.1.1.6. 2.1.1.6. Prijenos bitova kanalom

Operator $\oplus$ označava "ili **a** ili **b**, ali ne oba". Ako je rezultat $\mathbf{a} \oplus \mathbf{b} = \text{"sve 0"}$, znači da se blok bitova primio (prenio preko kanala) bez pogreške. Blokovi bitova **a** i **b** razlikuju se na onim mjestima gdje je pogreška što ju je unio šum kanala. Njihov zbroj, ima vrijednost 1 na mjestima gdje se blokovi razlikuju pa blok što sadrži **1-elemente**¹ ukazuje na mjesta pogrešaka. Imajte na umu da ako je $\mathbf{e} = \mathbf{a} \oplus \mathbf{b}$, onda je $\mathbf{b} = \mathbf{a} \oplus \mathbf{e}$, što znači da se ispravak primjene poruke dobije zbrojem bloka *uzorka pogreške* i primljenoga bloka (*kodna riječ*).

Kao primjer koji objašnjava navedene definicije, razmotrite poslane odnosno primljene blokove prikazane na slici.



Sinonimi za ovakav skup bitova su: "blok", "kodna riječ", "vektor", "slijed", "sekvenca" ili "niz", a ponekad i "bajt". Prethodno napravljenim simulacijskim sklopom, odredite njihov zbroj.

6. Pitanje: Koliki je zbroj dvaju binarnih brojeva: 1001101 i 0101000?

.....

.....

.....

Dva osnovna zadatka ovoga kolegija (*Zaštitno kodiranje signala*) su:

otkriti da postoje pogreške i

ispraviti otkrivene pogreške.

Za otkrivanje, u tu svrhu koristimo paritetnu provjeru.

2.1.2. 2.2.2. POJAM PARITETA

Parnost (paritet) cijeloga broja može biti parna (*even*) ili neparna (*odd*). Ona se može označiti brojanom vrijednosti. Paran paritet naziva se **parnost 0**. Neparan paritet naziva se **parnost 1**. Dakle, neparan broj brojeve ima paritet 1, a paran broj brojeva ima paritet 0. Također se može govoriti o **paritetu određene skupine bitova** što se odnosi na paritet broja elemenata vrijednosti 1 u skupini.

Razmotrite ova dva binarna broja:

0110 1010 0011 1010

0110 1010 0111 1010

Oni se razlikuju samo u jednome bitu (pogledajte skupinu 3#). Ako za prvi broj (1.) pretpostavljamo da je ono što bi trebalo biti smješteno na nekome memorijskome mjestu, a drugi broj (2.) je ono što se zapravo nalazi na tome mjestu, onda očito postoji problem.

Paritet (paritetan bit): Pribraja se svakome bajtu (8 bitova) podataka pa se taj dodatan bit postavlja u 1 ili 0 da bi ukupna suma bitova (jedinica) toga bajt sadržavala paran ili neparan broj jedinica. Na primjer, razmotrite ovaj binaran broj: 1101. Postoje tri jedinice u ovome broju, a ima ih neparan broj. Ako se u ovom krugu koristi "neparan" paritet, onda je paritetan bit 0 pa će i dalje ostati neparan broj jedinica u broju. Međutim, ako se u krugu koristi "paran" paritet, onda je paritetan bit postavljena na 1, kako bi se imalo 4 jedinice u broju, što je paran broj jedinica. Slijedi gornji broj s parnim i neparnim

¹ 1-elementi ... znači niz binarnih znamenaka (u bloku) jednakih 1.

paritetnim bitovima. Paritetni bitovi su na najmanje značajnome položaju (bitovi sasvim desno) te su odvojeni od izvornoga broja razmakom (prazninom) zbog jasnoće:

1101 0 neparan (*odd*)

1101 1 paran (*even*)

Sljedeća tablica pokazuje nekoliko primjera koji mogu pomoći u definiranju ovoga koncepta. U svakome slučaju, paritetan bit koristi se za napraviti podatkovni bajt parnim paritetom (praznine su ostavljene u podatkovnome bajtu zbog jasnoće).

Tablica: Prikaz paritetnoga bita

Podatkovni bajt	Paritetan bit
0000 0000	0
0000 0001	1
0000 0011	0
0000 0100	1
1111 1110	1
1111 1111	0

Paritet je jednostavan koncept i predstavlja temelj za jednu od najvažnijih osnovnih metoda provjere pogrešaka. Kao primjer, ako se prenosi neki bajt koristeći neparan paritet, a podaci se primaju parnim brojem bitova, tada se jedan od bitova promijenio tijekom prijenosa.

2.1.3. 2.1.3. ITERACIJSKA PROVJERA PARITETA, PARAN I NEPARAN PARITET

2.1.3.1. 2.1.3.1. Zadatak

Prenosi se 7 bajtova po 7 bitova [1011011], [1011001], [1101101], [0011001], [1101001], [0011100], [1111000]. Koristeći *iteracijsku metodu provjere parnoga pariteta* (vidi predavanja), popunite sljedeću tablicu. Svakome bajtu pridružite paritetan bit. Također oblikujte i paritetan bajt što sadrži paritetne bitove za svih prethodnih sedam bajtova (koristite *paran* paritet).

bajt	Podatkovni bitovi (<i>paran</i> paritet)							Paritetan bit
	1	2	3	4	5	6	7	
1								
2								
3								
4								
5								
6								
7								
P								

Isto napravite ali za *neparan* paritet:

bajt	Podatkovni bitovi (<i>neparan</i> paritet)							Paritetan bit
	1	2	3	4	5	6	7	
1								
2								
3								
4								
5								
6								
7								
P								

7. **Pitanje:** Na koje se sve načine može iskoristiti **polje** koje je smješteno dolje-desno u tablicama i uokvireno debljom crtom?

1.
2.
3.
4.

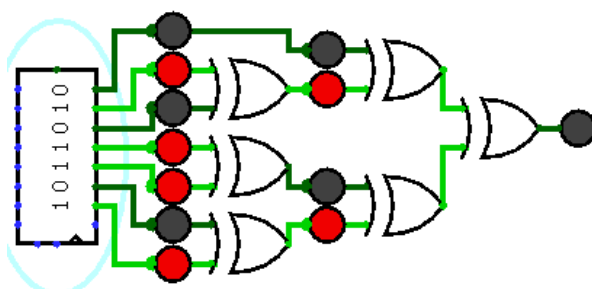
8. **Pitanje:** Ima li to polje istu vrijednost za sva tri smjera (vodoravno, okomito, koso)? Provjerite na jednome primjeru!

.....

.....

.....

2.1.4. 2.1.4. LABORATORIJSKA VJEŽBA: PARITET



Slika:

Prema predlošku na gore prikazanoj slici, u simulacijskome alatu LogisimTM nacrtajte odgovarajuću shemu (i precrtajte je u donji okvir lijevo) te izračunajte paritet prethodnih 7 skupina bitova pomoću standardnih logičkih XOR vrata s 2 ulaza.² Izlaz takvih vrata prikazuje paritet skupine bitova koji tvore njegove ulaze. Rezultate zapišite u priloženu tablicu.

bajt	Podatkovni bitovi (neparan paritet)							Paritetan bit
	1	2	3	4	5	6	7	
1								
2								
3								
4								
5								
6								
7								
P								

² Napomena: Mnogi autori(teti) tvrde da oblikovanje ponašanja XOR vrata treba odgovarati vratima s neparnim paritetom, ali ne postoji dogovor o tome pitanju. Zadano ponašanje za XOR vrata u LogisimTM simulacijskome alatu temelji se na standardu IEEE 91. Ponašanje je također u skladu s intuitivnim značenjem na kojemu počiva pojam isključivo ILI: Ako Vas konobar pita želite li prilog od pire krumpira, mrkve, graška ili kelja, on prihvaća samo jedan izbor, a ne tri, bez obzira što neki autori(teti) govorili. Ipak, moram priznati, da ovu izjavu nisam podvrgnuo strogim testovima. Možete konfigurirati XOR i XNOR vrata koristeći postavku atributa: **Multiple-Input Behavior** ⇒ **When an odd number are ON** u LogisimTM simulacijskome alatu.

bajt	Podatkovni bitovi (paran paritet)							Paritetan bit
	1	2	3	4	5	6	7	
1								
2								
3								
4								
5								
6								
7								
P								

Isto napravite koristeći XOR vrata s višestrukim ulazima na slici i odgovorite na sljedeća pitanja.

9. **Pitanje:** Ako se na XOR vrata sa 7 ulaza dovede niz bitova [0101101] što je izlaz?

.....

.....

10. **Pitanje:** Koji je preduvjet ispravnoga prethodnoga odgovora?

.....

.....

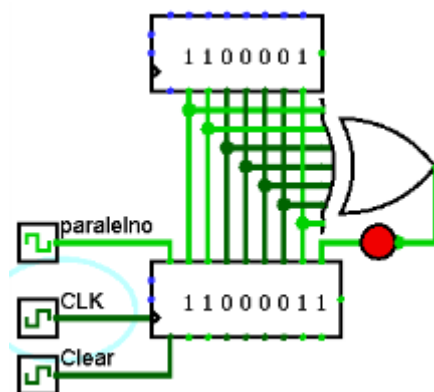
2.1.4.1. 2.1.4.1. Zadatak

Za prikazani primjer, odredite izlaz za zadani ulaz.



Slika:

U simulacijskome alatu Logisim™, nacrtajte simulacijski sklop (blok duljine k u blok duljine $k+1$ koji ima paritet 0) i provjerite izlaz za različite kombinacije ulaznih binarnih znamenaka.



Slika: Sklop za automatsko stvaranje pariteta

Isti sklop prilagodite za različite vrste pariteta (*even/odd*).

11. **Pitanje:** Čime se postiže takva prilagodba?

.....

.....

2.1.4.2. 2.1.4.1. Zadatak

Odredite paran i neparan paritet svih riječi u obje tablice. Također popunite podatak o pripadnome paritetu pojedinačnih bitova u svih 7 riječi.

bajt	Podatkovni bitovi (<i>paran</i> paritet)							Paritetan bit
	1	2	3	4	5	6	7	
1	1	0	1	1	0	0	1	
2	0	1	1	1	0	0	1	
3	0	0	1	1	0	0	0	
4	1	0	1	0	1	1	1	
5	1	1	1	0	0	1	0	
6	0	0	0	0	0	0	0	
7	1	1	1	1	1	1	1	
P								

bajt	Podatkovni bitovi (<i>neparan</i> paritet)							Paritetan bit
	1	2	3	4	5	6	7	
1	1	0	1	1	0	0	1	
2	0	1	1	1	0	0	1	
3	0	0	1	1	0	0	0	
4	1	0	1	0	1	1	1	
5	1	1	1	0	0	1	0	
6	0	0	0	0	0	0	0	
7	1	1	1	1	1	1	1	
P								

2.1.4.3. 2.1.4.3. Pitanja:

12. Ako se na XOR vrata sa 7 ulaza dovede niz bitova [0101101], što je izlaz?

.....

.....

13. Koju binarnu znamenku treba pridružiti skupu bitova [01101101] za dobiti ukupan paritet 0?

.....

.....

14. Da bi se skupu bitova pridružila paritetna znamenka, za odrediti ukupan paritet, koja se logička operacija koristi?

.....

.....

15. Gdje se pridružuje paritetan bit?

.....

.....

16. Kakav je odnos bloka bitova sa i bez izračuna parnosti?

.....

.....

17. Što je to vodoravna provjera zalihosti?

.....

.....

18. Što je to okomita provjera zalihosti?

.....

.....

19. Što je to uzdužna provjera zalihosti?

.....

.....

[illegible]