

# Izrada sklopa za kodiranje 4-bitnog podatka u Hammingov kod, umetanje i ispravljanje greške.

---

**Smeraldo, Denis**

**Undergraduate thesis / Završni rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:295540>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2025-03-14**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**IZRADA SKLOPA ZA KODIRANJE 4-BITNOG PODATKA U  
HAMMINGOV KOD, UMETANJE I ISPRAVLJANJE GREŠKE**

Rijeka, rujan 2022.

Denis Smeraldo

0069084616

SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**IZRADA SKLOPA ZA KODIRANJE 4-BITNOG PODATKA U  
HAMMINGOV KOD, UMETANJE I ISPRAVLJANJE GREŠKE**

Mentor: doc. dr. sc. Ivan Volarić

Rijeka, rujan 2022.

Denis Smeraldo

0069084616

**SVEUČILIŠTE U RIJECI**  
**TEHNIČKI FAKULTET**  
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za automatiku i elektroniku**  
Predmet: **Digitalna elektronika**  
Grana: **2.03.03 elektronika**

## ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Denis Smeraldo (0069084616)**  
Studij: **Preddiplomski sveučilišni studij elektrotehnike**

Zadatak: **Sklop za kodiranje 4-bitnog podatka u Hammingov kod, umetanje i ispravljanje greške / Digital circuit for coding a 4-bit input to Hamming code with insertion and correction of 1-bit error**

**Opis zadatka:**

U sklopu završnog rada potrebno je izraditi sklop koji se sastoji od tri djela. Ulaz u prvi dio sklopa je 4-bitni podatak koji se unosi preko 4-bitnog DIP prekidača spojenog u pull-down konfiguraciju. Koristeći XILI vrata potrebno je izračunati tri paritetna bita Hammingovog koda, te je tako generirani Hammingov kod izlaz iz prvog djela sklopa i ulaz u drugi. Drugi dio sklopa zadužen je za unos pogreške u Hammingov kod. Selekcija u kojim bitovima će se dogoditi greška se vrši preko DIP prekidača spojenog u pull-down konfiguraciju. Unos greške izvodi se preko XILI vrata (iskorištava se njihova mogućnost selektivnog negiranja). Hammingov kod s unesenom greškom je izlaz iz drugog djela sklopa i ulaz u treći dio sklopa. U trećem djelu sklopa potrebno je provjeriti točnost paritetnih bitova (pomoću XILI vrata). Novo dobiveni paritetni bitovi pokazuju na kojem se bitu dogodila greška, te ih je potrebno dovesti na ulaz 3/8 dekodera, a izlaze dekodera potrebno je spojiti na XILI vrata koja selektivno negiraju onaj bit u kojem se dogodila pogreška. Ispravljeni Hammingov kod je izlaz iz trećeg djela sklopa. Izlaze iz svakog djela sklopa potrebno je prikazati pomoću 7 LED dioda (4 za podatkovne i 3 za paritetne bitove).

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.



Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:



---

Doc. dr. sc. Ivan Volarić

Predsjednik povjerenstva za  
završni ispit:



---

Prof. dr. sc. Viktor Sučić

## IZJAVA

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku preddiplomskih sveučilišnih studija/stručnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 1. veljače 2020., izjavljujem da sam samostalno izradio završni rad prema zadatku preuzetom dana 23. ožujka, 2022.

Rijeka 15. rujna, 2022.



Denis Smeraldo



# Sadržaj

<b>1. Uvod</b> .....	1
<b>2. Kodovi za otkrivanje i ispravljanje pogrešaka</b> .....	2
<b>2.1 Paritetni bit</b> .....	2
<b>2.2 Hamming-ov kod</b> .....	2
2.2.1. Kodna distanca (eng. Hamming distance) .....	2
2.2.2. Richard Hamming.....	4
2.2.3. Princip rada Hamming-ovog koda.....	5
<b>3. Komponente korištene u sklopu</b> .....	7
<b>3.1 Eksperimentalna pločica (eng. Breadboard)</b> .....	7
<b>3.2 Pasivni i nelinearno pasivni elementi</b> .....	8
3.2.1 Otpornici .....	8
3.2.2 LED Dioda (eng. light-emitting diode).....	9
3.3.3 DIP prekidač (eng. DIP switch) .....	10
<b>3.3 Integrirani krugovi (eng. IC – Integrated Circuit)</b> .....	11
3.3.1. XILI vrata.....	11
3.3.3. 3/8 Dekoder.....	13
<b>4. Izrada sklopa</b> .....	16
<b>4.1 Prvi dio: generiranje Hamming-ovog koda</b> .....	16
<b>4.2 Drugi Dio: ručno ubacivanje pogreške</b> .....	20
<b>4.3. Treći dio: Detekcija i ispravak pogreške</b> .....	23
4.3.1. Detektiranje pogreške .....	23
4.3.2. 3/8 dekodeer i ispravljanje pogreške .....	25
<b>5. Zaključak</b> .....	29
<b>6. Literatura</b> .....	30
<b>Sažetak i ključne riječi</b> .....	31
<b>Summary and key words</b> .....	32

# 1. Uvod

U sklopu ovoga završnog rada izrađen je sklop koji kodira 4-bitni podatak u Hamming-ov kod, unosi pogrešku, te ju detektira i ispravlja. Prema tome, sklop možemo podijeliti u tri dijela, koja su ukratko opisana u nastavku.

Ulaz u prvi dio sklopa je 4-bitni podatak koji se unosi preko 4-bitnog DIP prekidača spojenog u *push-up* konfiguraciju. Koristeći integrirani krug koji se sastoji od XILI vrata generira se Hamming-ov kod (7,4) koji se sastoji od 3 paritetna bita i 4 podatkovna bita.

Drugi dio sklopa služi za ručno ubacivanje pogreške u Hamming-ov kod. To je napravljeno pomoću sedam XILI vrata spojenih u konfiguraciju selektivnih NE vrata. Jedan ulaz XILI vrata je spojen na prethodno generirani Hamming-ov kod, dok je drugi ulaz spojen na 7-bitni DIP prekidač spojen u *push-up* konfiguraciju.

Treći dio sklopa služi za detektiranje i automatsko ispravljanje pogreške u kodu. Ulaz je Hamming-ov kod s umetnutom pogreškom te ponovnim korištenjem integriranih krugova sa XILI vratima vrši se detekcija u kojem je bitu nastala pogreška. Zatim, ti su izlazi spojeni na ulaze 3/8 dekodera čiji su izlazi spojeni na XILI vrata koja selektivno negiraju bit u kojem se dogodila pogreška. Time je greška u kodu ispravljena, te je izlaz iz trećeg djela sklopa jednak izlazu iz prvog dijela sklopa, neovisno o ručno umetnutoj pogrešci u drugom djelu sklopa.



## 2. Kodovi za otkrivanje i ispravljanje pogrešaka

Ukoliko se govori o digitalnim sustavima, prilikom slanja ili obrade informacija u bilo kojem formatu, može doći do pogreške. Šum ili kvar dovodi do pogreške koja dolazi u prijenosu između odašiljača i prijemnika. Ako dođe do pogreške koju ne detektiramo niti ju ispravimo, možemo imati ozbiljne posljedice u prijenosu podataka koje npr. mogu rezultirati kvarove uređaja u nekom radnom mehanizmu. U slučaju da se ona nalazi u binarnom obliku, pogrešku predstavlja promjena jednog ili više bitova u kodnoj riječi. Način na koji se otkrivaju pogreške sastoji se dodavanjem ili jednog ili više bita originalnim podacima, pomoću kojeg primatelj poruke može detektirati da li je došlo do pogreške, te je u nekim slučajevima samostalno ispraviti bez potrebe za ponovnim slanjem podataka [1].

### 2.1 Paritetni bit

Paritetni bit je bit koji se dodaje u nizu podatkovnih bitova kako bi se detektirala eventualna pogreška prilikom prijenosa podataka. Postoji parni i neparni paritet. U parnom paritetu dodaje se paritetni bit u podatkovni niz tako da ukupni broj jedinica u kodnoj riječi bude paran, dok za neparni paritet vrijedi obrnuto. Na primjer, 8-bitni ASCII kod za „A“ glasi 01000001, a dodavanjem paritetnog bita na početak koda s parnim paritetom, nova kodna riječ postaje 001000001. Ukoliko je kodna riječ kodirana s neparnim paritetom, tada će paritetni bit glasiti "1", a kodna riječ će glasiti: 101000001. Neparni bit je komplement parnog bita. U praksi, najčešće se susrećemo sa korištenjem parnog pariteta.

Provjera paritetnog bita ima svoje mane. S ovom metodom moguće je detektirati jednu, tri, pet, itd. grešaka, ali nije moguće detektirati dvije, četiri, šest, itd. grešaka. Naime, kada se dogodi paran broj grešaka, paritet poruke će ostati nepromijenjen, te će paritetni bit indicirati na točnost poruke. Također, s jednim paritetnim bitom, moguća je samo detekcija pogreške, ali nije moguće ispravljanje pogreške.

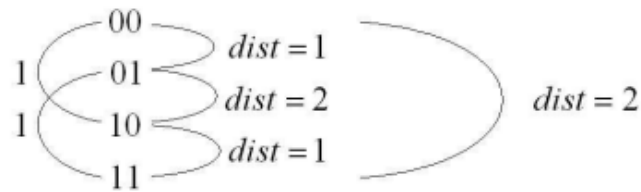
### 2.2 Hamming-ov kod

Ovo je jedna od metoda detektiranja i ispravljanja pogrešaka te je ovo i metoda koja je korištena u implementaciji ovoga sklopa. Hamming-ov kod definira koliko će se i na koja mjesta unutar koda dodati paritetni bitovi s ciljem ne samo detekcije greške, nego i njezinog ispravka.

#### 2.2.1. Kodna distanca (eng. Hamming distance)

Hamming-ova distanca predstavlja broj bitova koji se razlikuju između dvije kodne riječi. Važno svojstvo koda jest minimalna Hamming-ova distanca, tj. najmanji broj bitova koji se trebaju promijeniti, da bi se jedna kodna riječ pretvorila u drugu [2].

Na slici 2.1 je prikazan primjer računanja Hamming-ove distance. U primjeru je prikazan 2-bitni kod sa četiri kodne riječi, te Hamming-ovim distancama između svih kodnih riječi, iz čega je vidljivo da je minimalna Hamming-ova distanca 1.



Slika 2.1. Hamming-ova distanca [2].

Minimalna Hamming-ova distanca služi kao mjera kvalitete koda po pitanju detekcije i ispravka pogrešaka. S minimalnom Hamming-ovom distancom od 1, greška u kodnoj riječi može rezultirati drugom validnom kodnom riječi, te ne možemo ni detektirati niti ispraviti pogrešku. Što je već minimalna Hamming-ova distanca, to je moguće više grešaka ili detektirati ili ispraviti, po sljedećem izrazu:

$$D = A + B + 1, B \geq A, \quad (2.1)$$

gdje je  $D$  minimalna Hamming-ova distanca koda,  $A$  broj pogrešaka koje je moguće otkriti, dok je  $B$  broj pogrešaka koje je moguće ispraviti.

U praksi, veoma često dolazi do pogreške u samo jednom bitu, što znači da kod mora imati minimalnu Hamming-ovu distancu  $D = 3$  da bi bio sposoban ispraviti nastalu pogrešku.

$$D = 3, \quad (2.2)$$

$$B = 1, \quad (2.3)$$

$$D = A + B + 1, \quad (2.4)$$

$$3 = A + 1 + 1, \quad (2.5)$$

$$A = 1. \quad (2.6)$$

U slučaju da je minimalna Hamming-ova distanca koda  $D = 1$ , jedino moguće rješenje izraza (2.1) jest  $A = B = 0$ , što znači da nije moguće niti pronaći niti ispraviti pogreške. Dok u slučaju  $D = 1$ , jedino moguće rješenje jest  $A = 1, B = 0$ , što znači da je moguće detektirati jednu pogrešku.

Da bi kod imao minimalnu Hamming-ovu distancu od 3, na originalne podatke potrebno je dodati paritetne bitove. Broj potrebnih paritetnih bitova definiran je sljedećim izrazom:

$$2^k \geq m + k + 1, \quad (2.7)$$

$$n = m + k, \quad (2.8)$$

gdje je  $m$  broj podatkovnih bitova, dok je  $k$  broj bitova provjere.

U sklopu ovog završnog rada, baviti ćemo se samo  $m = 4$  bitnim podacima, stoga minimalni  $k$  za koji je izraz (2.7.) zadovoljen jest  $k=3$ :

$$2^3 \geq 4 + 3 + 1. \quad (2.9)$$

Primjena Hamming-ovoga koda se nalazi u satelitima, memoriji za računala, modemima, ugrađenim sustavima i slično.

### 2.2.2. Richard Hamming

Američki matematičar Richard Wesley Hamming prikazan na slici 2.1., rođen je 11. veljače 1915. u Chicago-u. Dobio je doktorat iz područja matematika na sveučilištu Illinois. Tijekom 2. svjetskog rata bio je glavni matematičar na projektu Manhattan Project koji je 1945. proizveo prvo nuklearno naoružanje. Nakon završetka rata, bavio se telekomunikacijama i računalnim inženjerstvom.

Radio je u Bell Labs-u na računalu „Model V“ koje je bilo elektromehaničko računalo opće namjene. Stroj je radio sa probušenim papirnatim karticama koje su se sastojale od šest rupa po redu. Tijekom radnog ciklusa, stroj bi detektirao pogreške u releju te bi se zaustavio i dao do znanja operaterima da je došlo do pogreške. Problematika stroja koji sam detektira pogrešku, no ne i njenu lokaciju je bila originalna motivacija Richarda Hamming-a koja ga je dovela do rješavanja problema kreirajući nove algoritme. Sljedećih nekoliko godina proveo je u istraživanju tih algoritama, koje je na kraju objavio 1950. godine pod nazivom „Hamming-ov kod“.

Također, Hamming-ov kod je jedna od njegovih brojnih kreacija. Njegova poznata dijela su Hamming-ov prozor, Hamming-ovi brojevi, Hamming-ova udaljenost. Tijekom svojega životnoga vijeka osvojio je brojne nagrade poput Turnig-ove nagrade, Harold Pender nagrade te IEEE medalje.

1997. godine postao je Profesor emeritus te je samo nekoliko tjedana kasnije preminuo od srčanog udara [3,4].



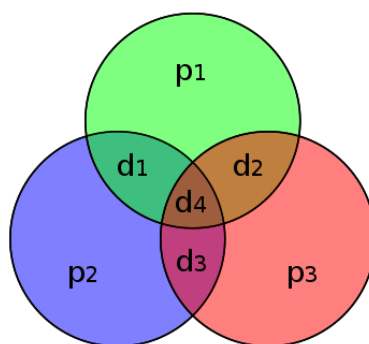
Slika 2.1. Richard Hamming [4].

### 2.2.3. Princip rada Hamming-ovog koda

Ideja Hamming-ovog koda je kodirati riječ tako da primatelj može sam provjeriti i detektirati ako se pogreška dogodila, te u slučaju neispravne kodne riječi, može ju ispraviti, uz unaprijed dogovorene uvijete. Može se kodirati riječ uz korištenje parnog ili neparnog pariteta [2].

U ovome završnom radu korišten je Hamming-ov kod (7,4) koji pretvara četiri podatkovna bita u 7-bitnu kodnu riječ koja se sastoji od 3 paritetna bita. Također, korišten je parni paritet za kodiranje kodne riječi.

Cilj Hamming-ovog koda je stvoriti skup paritetnih bitova koji se preklapaju kako bi se detektirala jednobitna pogreška u podatkovnom ili paritetnom bitu koja se može detektirati i ispraviti. Opća metoda prikazana je na slici 2.2.



Slika 2.2. Skup paritetnih bitova [5].

Svaki paritetni bit pokriva određene prenesene bitove u kodnoj riječi. U tablici 2.1. prikazano je koje bitove provjerava svaki pojedini paritetni bit.

Tablica 2.1. Provjera ispravnosti.

D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	P <sub>3</sub>	D <sub>1</sub>	P <sub>2</sub>	P <sub>1</sub>
X		X		X		X
X	X			X	X	
X	X	X	X			

Paritetni bitovi P<sub>1</sub>, P<sub>2</sub> te P<sub>3</sub> održavaju parnost u podatkovnim bitovima D<sub>4</sub>, D<sub>3</sub>, D<sub>2</sub> te D<sub>1</sub> u tri različite kombinacije, te taj algoritam omogućava detektiranje i ispravljanje jednobitne pogreške. Na primjer, paritetni bit P<sub>1</sub> osigurava parnost podatkovnih bitova D<sub>4</sub>, D<sub>2</sub>, D<sub>1</sub> na način da prikaže 1 ili 0 ovisno o potrebi kako bi se osigurala parnost. Ostale dvije kombinacije paritetnih bitova bazirane su na istoj metodi za različite podatkovne bitove.

Na primjer, koristimo li ulazni podatak 0110 kojeg kodiramo Hamming-ovim kodom koristeći parni paritet on će iznositi 0110011. Nakon toga vrši se provjera ispravnosti kodiranog podatka koristeći tri različita paritetna bita iz algoritma danog pomoću tablice 2.1..

Za prvu kombinaciju, bitovi D<sub>4</sub> i D<sub>1</sub> predstavljaju logičku nulu, dok bitovi D<sub>2</sub> i P<sub>1</sub> predstavljaju logičku jedinicu, čime paritetni bit P<sub>1</sub> osigurava parni broj jedinica i uvjet se ispunjava.

Za drugu kombinaciju, bitovi D<sub>4</sub> i D<sub>1</sub> predstavljaju logičku nulu, dok bitovi D<sub>3</sub> i P<sub>2</sub> predstavljaju logičku jedinicu, čime paritetni bit P<sub>2</sub> osigurava parni broj jedinica i uvjet se ispunjava.

Za treću kombinaciju, bitovi D<sub>4</sub> i P<sub>3</sub> predstavljaju logičku nulu, dok bitovi D<sub>3</sub> i D<sub>2</sub> predstavljaju logičku jedinicu, čime paritetni bit P<sub>3</sub> osigurava parni broj jedinica i uvjet se ispunjava.

Nakon sistematske provjere, utvrđena je ispravnost kodiranog podatka korištenjem Hamming-ove metode. Nakon ove provjere, podatak se odašilje prema prijemniku koji koristeći istu ovu metodu može utvrditi postoji li pogreška u kodiranom podatku, to jest, je li je zaprimljeni podatak ispravan.

U slučaju da je došlo do jednobitne pogreške, npr. u trećem bitu D<sub>2</sub> te on umjesto 1 on sada iznosi 0, primatelj može detektirati te ispraviti pogrešku u primljenom kodiranom podatku. Ponovo se vrši sistemska provjera.

Za prvu kombinaciju, bitovi D<sub>4</sub>, D<sub>2</sub>, D<sub>1</sub> predstavljaju logičku nulu, dok bit P<sub>1</sub> predstavlja logičku jedinicu. Ovo rezultira u neparnom broju jedinica, te se odmah može iz prve provjere detektirati pogreška, no ne i locirati.

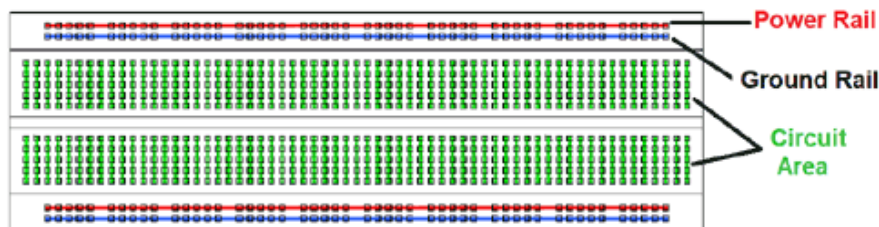
Za drugu kombinaciju, bitovi D<sub>4</sub> i D<sub>1</sub> predstavljaju logičku nulu, dok bitovi D<sub>3</sub> i P<sub>2</sub> predstavljaju logičku jedinicu, čime paritetni bit P<sub>2</sub> osigurava parni broj jedinica i uvjet se ispunjava.

Za treću kombinaciju, bitovi D<sub>4</sub>, D<sub>3</sub>, D<sub>2</sub> predstavljaju logičku nulu, dok bit P<sub>3</sub> predstavlja logičku jedinicu. Ovo ponovo rezultira u neparnom broju jedinica čime se uvjet ne ispunjava.

Sa ovim informacijama, primatelj je siguran da postoji pogreška u kodiranom podatku. Bitovi D<sub>4</sub>, D<sub>3</sub>, D<sub>1</sub> te P<sub>2</sub> su ispunili uvjet, stoga su sigurno ispravni, što znači da se greška nalazi u jednom od preostalih bitova D<sub>2</sub>, P<sub>1</sub> te P<sub>3</sub>. Jedini bit koji se preklapa usporedivši prvu i treću kombinaciju jest bit D<sub>2</sub> te on predstavlja bit u kojem se dogodila pogreška. Koristeći ovu metodu, primatelj uspješno može detektirati i ispraviti pogrešku u zaprimljenom kodu u slučaju pogreške jednoga bita.

### 3. Komponente korištene u sklopu

#### 3.1 Eksperimentalna pločica (eng. Breadboard)

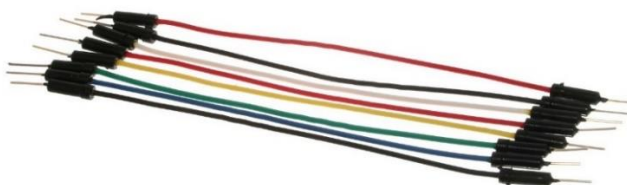


Slika 3.1. Shema pločice [6].

Eksperimentalna pločica se koristi kao podloga na koju se spajaju komponente i spojni vodiči. Na slici 3.1. prikazan je najčešći oblik, no postoje različite izvedbe, na primjer, sa duplim brojem vodljivih žica (eng. *Circuit area*) i odjeljaka za ulazni napon i uzemljenje.

Vodoravne linije predstavljaju mjesto gdje se najčešće dovodi napajanje sklopa. One nisu direktno povezane sa zelenim vodljivim žicama, niti sa drugim odjeljkom ulaza/izlaza. Potrebno ih povezati, npr. koristeći spojne žice. Primjer spojnih žica prikazan je na slici 3.2.

Okomite linije prikazane zelenom bojom prikazuju dio gdje se najčešće spajaju elementi (eng. *Circuit area*).

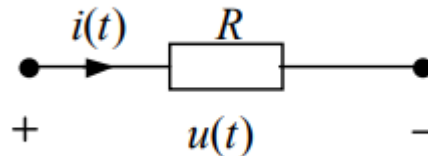


Slika 3.2. Primjer spojne žice [7].

## 3.2 Pasivni i nelinearno pasivni elementi

### 3.2.1 Otpornici

Otpornici su pasivni dvopolni elementi koji pružaju otpor struji. U elektroničkim krugovima, najčešće se koriste za ograničenje jakosti struje, kao dijelilo napona te kao terminatori komunikacijskih vodiča, itd.



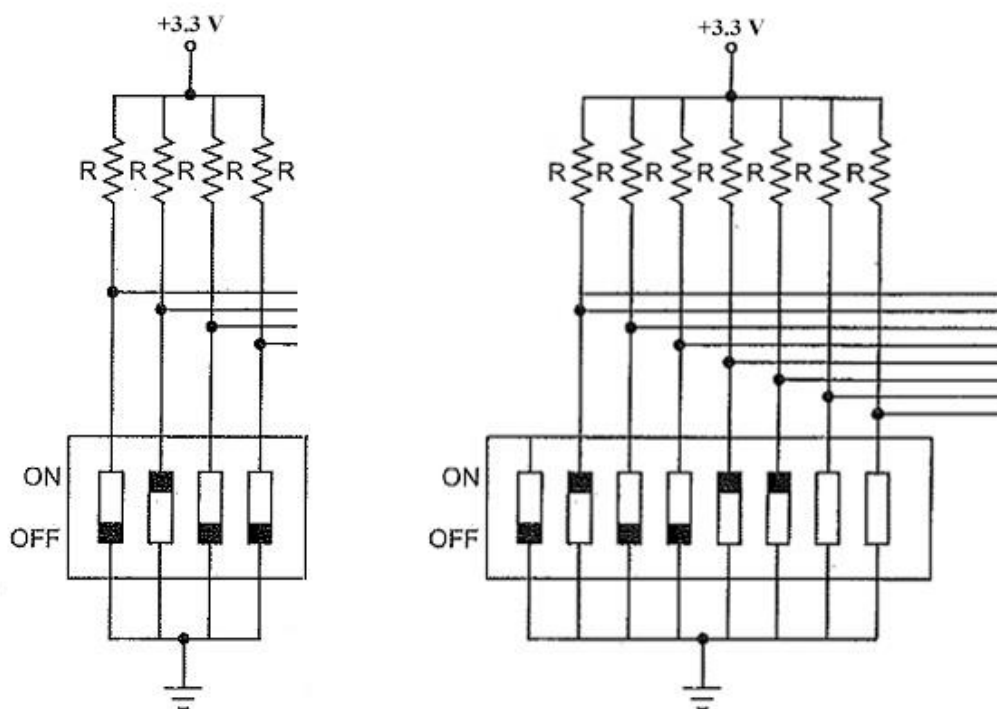
Slika 3.3. Prikaz otpornika u shemi [8].

U ovom radu korišteni su otpornici od 500 Oma prikazani na slici 3.4. u dvije svrhe. U prvoj za ograničenje struje kroz LED diode ispod preporučenih 20 mA. U drugoj u *push-up* konfiguraciji preko kojeg korisnik unosi logičku jedinicu kao ulazni podatak ili prilikom ručnog ubacivanja pogreške.

*Push-up* konfiguracija je prikazana je na slici 3.5. s kojom su generirani ulazni signali D1-D4, te G1-G7 na budućim shemama. Kada je sklopka zatvorena, izlaz je spojen direktno na masu, koja predstavlja logičku nulu. Kada je sklopka otvorena, izlaz je preko otpornika spojen na napon napajanja, što predstavlja logičku jedinicu.



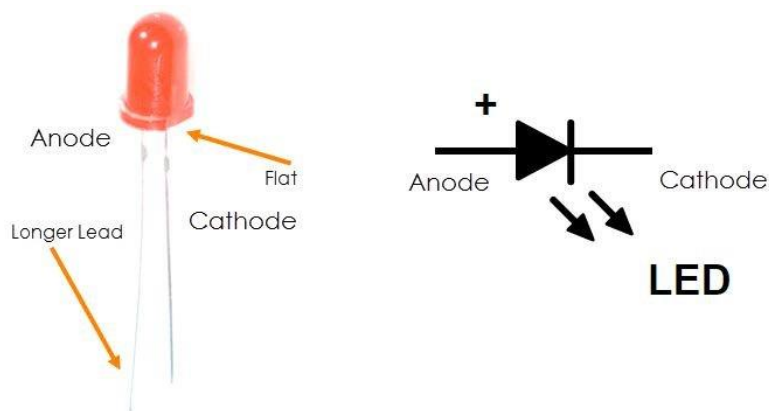
Slika 3.4. Otpornik od 500 Om-a [9].



Slika 3.5. Shema 4-bitne i 7-bitne push-up konfiguracije [10].

### 3.2.2 LED Dioda (eng. light-emitting diode)

LED Dioda (eng. light-emitting diode) je nelinearni pasivni poluvodički element koji pretvara električnu energiju u svjetlosnu.



Slika 3.6. Primjer LED Diode [11].

Sastoji se od dva izvoda: anode i katode. Propusno je polarizirana kada je napon na anodi veći od napona na katodi, a svijetliti će kada je taj napon veći od napona praga, koji za moderne LED diode iznosi 2-3V, ovisno o boji.



### 3.3.3 DIP prekidač (eng. *DIP switch*)

DIP (eng. *dual in-line package*) prekidač je mali preklopnik koji se koristi kao sklopka. U ovome završnom radu korišten je 4-bitni DIP prekidač za unos podatka koji će se kasnije kodirati, te 8-bitni prekidač za ručni unos pogreške.



Slika 3.7. 3 4-bitni DIP prekidač [12].



Slika 3.8. 8-bitni DIP prekidač [13].

Općenita primjena DIP prekidača nije toliko popularna kao što je bila u prošlosti, no i dalje ih koriste mnogi uređaji za provjeru postavki uređaja bez uključivanja. Primjer korištenja su garažna vrata, stropni ventilatori, sustavi za kućnu automatizaciju i slično [14].

### 3.3 Integrirani krugovi (eng. IC – Integrated Circuit)

Ključni elementi korišteni u radu koji ostvaruju implementacije logike u hardver su XILI vrata ili isključivo ILI vrata (eng. *XOR gate*) te dekodler 3/8.

#### 3.3.1. XILI vrata

XILI vrata su digitalna logična vrata koja daju na izlazu 1 (HIGH) u slučaju kada su ulazi različiti, tj. jedan ulaz iznosi 1, a drugi 0. U slučaju da su ulazi jednaki, neovisno jesu li oboje 1 ili 0, na izlazu će biti 0 (LOW). Ta logika omogućava detekciju i ispravljanje pogrešaka u kodu, te je prikazana pomoću tablice 3.1. Česti naziv za ova vrata još glasi vrata ekvivalencije ili vrata koincidencije [2].

Tablica 3.1. Tablica XILI vrata [15].

Ulaz		Izlaz
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Izraz za XILI vrata glasi :

$$A \oplus B = A\bar{B} + \bar{A}B. \quad (3.1)$$

Ako u gornji izraz uvrstimo sve moguće kombinacije ulaza A i B, dobivamo tablicu istine za XILI vrata.

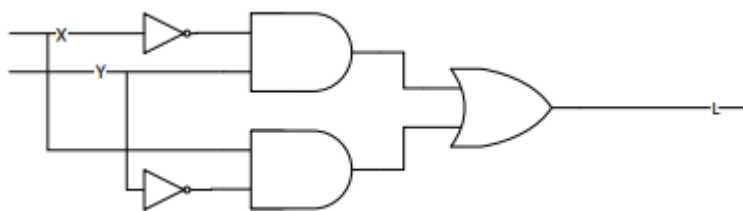
$$A \oplus B = 0 \oplus 0 = A\bar{B} + \bar{A}B = 0\bar{0} + \bar{0}0 = 0, \quad (3.2)$$

$$A \oplus B = 0 \oplus 1 = A\bar{B} + \bar{A}B = 0\bar{1} + \bar{0}1 = 1, \quad (3.3)$$

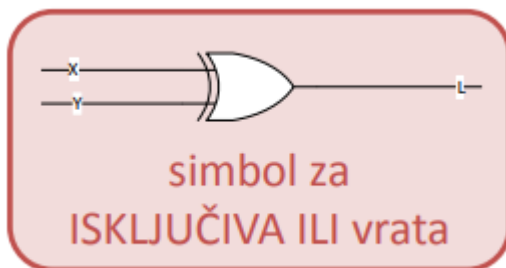
$$A \oplus B = 1 \oplus 0 = A\bar{B} + \bar{A}B = 1\bar{0} + \bar{1}0 = 1, \quad (3.4)$$

$$A \oplus B = 1 \oplus 1 = A\bar{B} + \bar{A}B = 1\bar{1} + \bar{1}1 = 0. \quad (3.5)$$

Integrirani krug se realizira pomoću međusobne kombinacije logičkih sklopova I, ILI i NE prikazane na slici 3.9., no za jednostavniju realizaciju prilikom projektiranja, koristi se simbol prikazan na slici 3.10.



Slika 3.9. Realizacija XILI vrata [2].



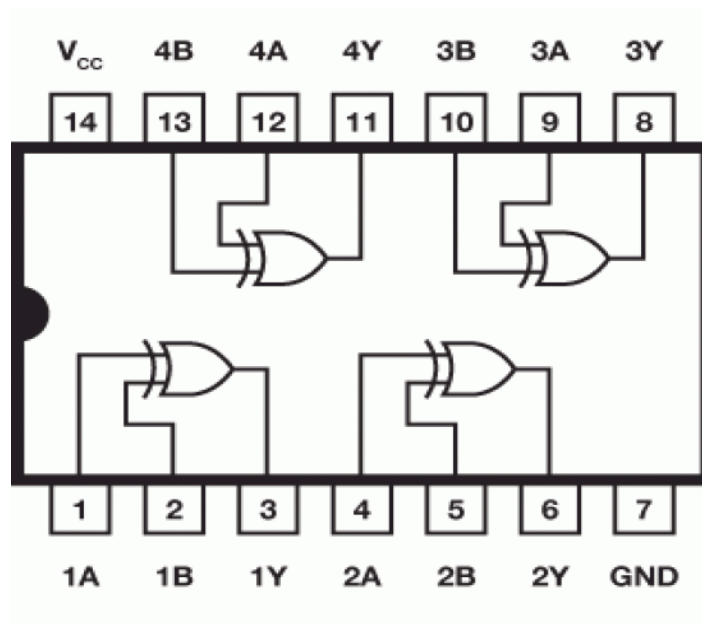
Slika 3.10. Simbol za XILI vrata [2].

U praksi način na koji se implementira XILI vrata jest preko integriranih krugova. U ovome završnom radu korišten je integrirani krug SN74HC86.



Slika 3.11. IC SN74HC86 [16].

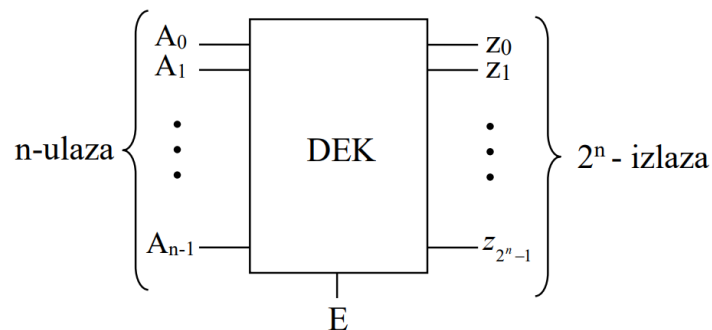
Sastoji se od 14 pinova, 7 sa svake strane. Prvi pin integriranog kruga je obično naznačen utorom ili kružićem, a enumeracija preostalih pinova slijedi slovo U. Nadalje, za ispravno korištenje integriranog kruga potrebna je podatkovna tablica (*eng. data sheet*) od proizvođača, u kojoj su dane sve tehničke karakteristike integriranog kruga. Najvažnije za izradu ovog završnog rada jest raspored pinova, prikazan na slici 3.12.. Na slici se vidi da se integrirani krug sastoji od 4 XILI vrata, te dva pina (7 i 14) za napajanje.



Slika 3.12. Raspored pinova za IC SN74HC86 [17].

### 3.3.3. 3/8 Dekoder

Generalno dekoderi su kombinacijski logički sklopovi koji za svaku binarnu kodnu riječ na ulazu imaju poseban izlaz koji je u stanju „1“ samo ako je na ulazu tražena riječ. Za  $n$ -ulaznih bitova takvih izlaza može biti najviše  $2^n$ . Također, dekoder može imati i dodatni ulaz (E - *enable*) koji služi za aktiviranje ili deaktiviranje sklopa [2]. Simbol dekodera koji se koristi prilikom projektiranja je prikazan na slici 3.13.



Slika 3.13. Simbol dekodera [2].

Ovisno o upotrebi, postoji više izvedba dekodera poput 2/4 dekodera, 3/8 dekodera, 4/16 dekodera, 8/256 dekodera i slično. Brojna im je primjena u praksi, te dekoderi poput 2/4 koriste se za optimizaciju sustava sa memorijama visokih performansi koje zahtijevaju niska vremena kašnjenja (*eng. delay*) prilikom prijenosa podataka. Također, dekoderi se mogu koristiti kao vremenski ili sekvencijski signali za uključivanje ili isključivanje uređaja u određeno vrijeme, jer kada u ulaze dekodera dolaze podaci iz brojača koji kontinuirano pulsira, izlazi dekodera aktivirat će se sekvencijalno.

Za ispravljanje detektirane pogreške u zadnjem djelu sklopa potreban je 3/8 dekodeer u kombinaciji sa 7 XILI vrata. Njegova izvedba prikazana je na slici 3.13..

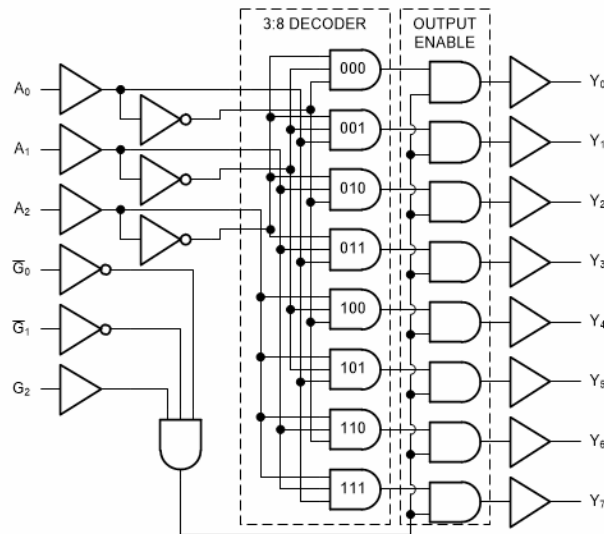


Slika 3.13. IC CD74HC238 [18].

Integrirani krug se sastoji od različitih kombinacija logičkih sklopova I, ILI i NE. Sve moguće kombinacije sklopa opisane su tablicom istinitosti 3.2. Također, vizualna reprezentacija, tj. unutarnja shema sklopa je opisana slikom 3.14. Sa ovim informacijama danima iz tablice je moguće automatski ispraviti pogrešku ako se na ulaz dovede točna informacija o pogrešnom bitu u kodu.

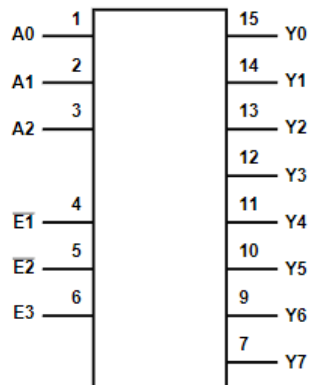
Tablica 3.2. Tablica istinitosti 3/8 dekodera.

A <sub>0</sub>	A <sub>1</sub>	A <sub>3</sub>	Y <sub>7</sub>	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



Slika 3.14. Blok dijagram 3/8 dekodera[16].

U sklopu ovog završnog rada korišten je integrirani krug CD74HC238. Sastoji se od 16 pinova, 8 sa svake strane. Iz podatkovne tablice, raspored pinova korištenog integriranog kruga prikazan je na slici 3.15..



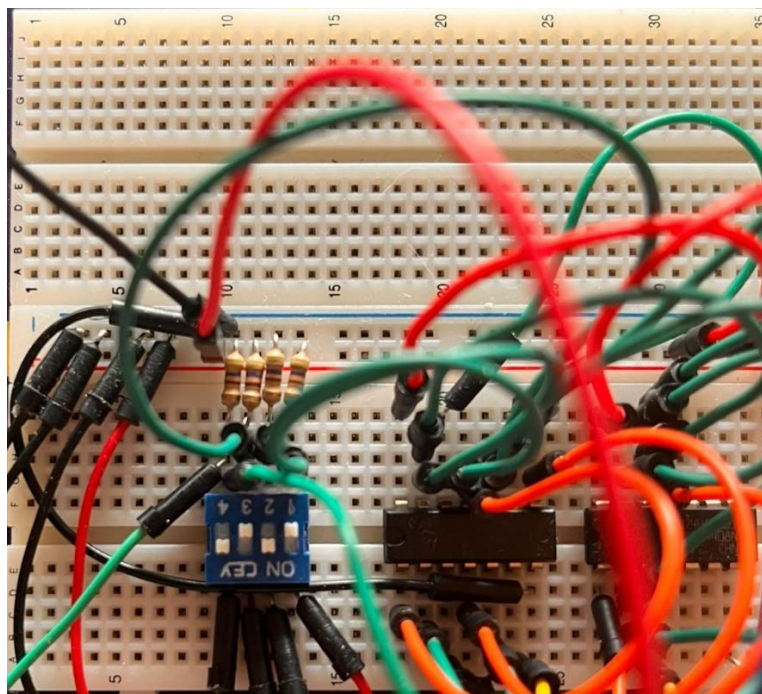
Slika 3.15 Raspored pinova za IC CD74HC238 [16].

Na slici 3.15. vidljivo je tri različite vrste pinova. Pinovi s oznakama  $A_0$ - $A_2$  predstavljaju tri adresna ulaza ( $A_2$  je MSB) u dekodera, pinovi s oznakama  $Y_0$ - $Y_7$  predstavljaju podatkovne izlaze iz dekodera, dok pinovi  $E_1$ - $E_3$  su takozvani *enable* pinovi, koji su interno spojeni na I vrata. Ako je rezultat I operacije logička jedinica, dekodera će normalno raditi, kao što je prije objašnjeno, dok u slučaju logičke nule, svi izlazi dekodera će biti u neaktivnom stanju, tj. u stanju logičke nule. U sklopu ovog završnog rada, funkcionalnost uključivanja/isključivanja rada dekodera nije potrebna, tj. samo je potrebno omogućiti kontinuirani rad dekodera, tako da na  $E_1$  i  $E_2$  spojimo logičku nulu, dok na  $E_3$  logičku jedinicu.

## 4. Izrada sklopa

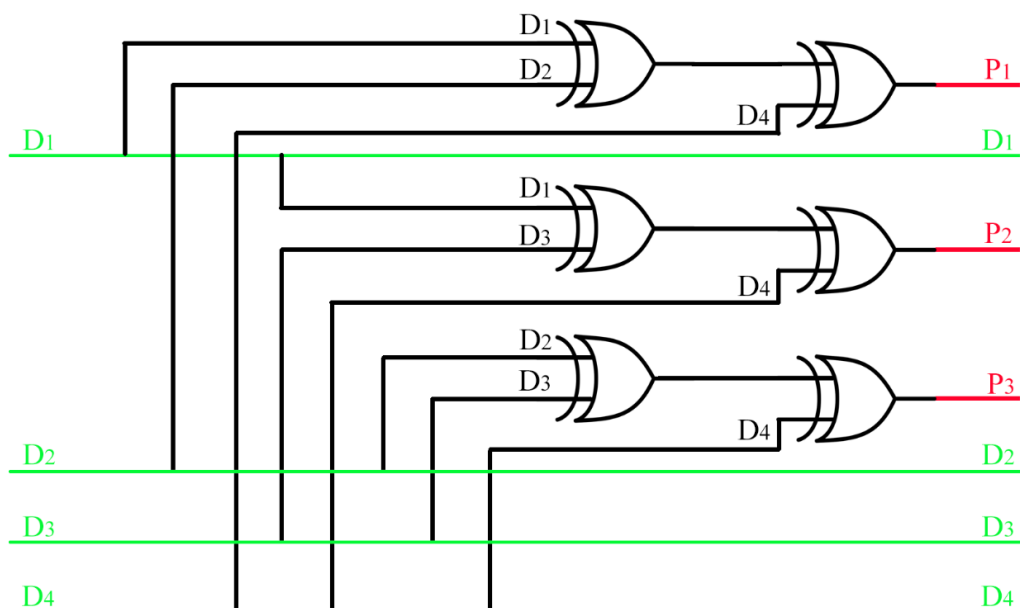
### 4.1 Prvi dio: generiranje Hamming-ovog koda

Za početak, potrebno je unijeti 4-bitni podatak pomoću *push-up* konfiguracije. Taj način omogućava da se samostalno biraju različite kombinacije ulaza. Realizacija je prilično jednostavna.



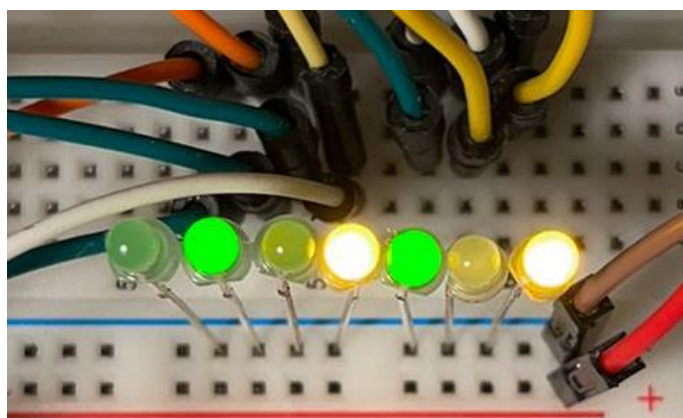
Slika 4.1. *Push-up* konfiguracija.

Nadalje, potrebno je generirati Hamming-ov kod. Shema implementiranog sklopa prikazana je na slici 4.2.. Kao što je objašnjeno u poglavlju 2.2.3., paritetni bit  $P_1$  provjerava paritet na podatkovnim bitovima  $D_4$ ,  $D_2$  i  $D_1$ . Iz tog razloga, kao što je vidljivo iz sheme, ulazi X1L1 vrata koji odgovaraju paritetnom bitu  $P_1$  su upravo spojeni na te podatkovne bitove. Analogno tome,  $P_2$  provjerava paritet na podatkovnim bitovima  $D_1$ ,  $D_3$  i  $D_4$ , dok paritetni bit  $P_3$  na bitovima  $D_2$ ,  $D_3$  i  $D_4$ .



Slika 4.2. Shema spoja.

Izlazi XILI vrata su ujedno izlazi iz prvog dijela sklopa, te su prikazani na 7 LED dioda. Zelene LED diode prikazuju podatkovne bitove, dok žute paritetne. Ispravnost ovog djela sklopa testirana je na dva ulazna podatka: 0101 i 0001. Na slici 4.4. prikazan je izlaz sklopa (0101101) za prvi ulazni podatak (0101).



Slika 4.4. Izlaz kombinacije 0101.



Prvi paritetni bit  $P_1$ , osigurava parni broj jedinica na bitovima  $D_4, D_2, D_1$ , te  $P_1$ . Na slici 4.4. sa lijeva na desno, gledamo diode pod rednim brojem 1., 3., 5., 7.. Iz slike je vidljivo da su prvi i treći bit u stanju logičke nule, dok su peti i sedmi u stanju logičke jedinice. To znači da sa dvije logičke jedinice, možemo zaključiti da je paritetni bit ispravno generiran jer imamo parni broj jedinica.

$$C1 = D4 \oplus D2 \oplus D1 \oplus P1 = 0 \oplus 0 \oplus 1 \oplus 1 = 0. \quad (4.1)$$

Drugi paritetni bit  $P_2$ , osigurava parni broj jedinica na bitovima  $D_4, D_3, D_1$ , te  $P_2$ . Na slici 4.4. sa lijeva na desno, gledamo diode pod rednim brojem 1., 2., 5., 6.. Iz slike je vidljivo da su prvi i šesti bit u stanju logičke nule, dok su drugi i peti u stanju logičke jedinice. To znači da sa dvije logičke jedinice, možemo zaključiti da je paritetni bit ispravno generiran jer imamo parni broj jedinica.

$$C2 = D4 \oplus D3 \oplus D1 \oplus P2 = 0 \oplus 1 \oplus 1 \oplus 0 = 0. \quad (4.2)$$

Treći paritetni bit  $P_3$ , osigurava parni broj jedinica na bitovima  $D_4, D_3, D_2$ , te  $P_3$ . Na slici 4.4. sa lijeva na desno, gledamo diode pod rednim brojem 1., 2., 3., 4.. Iz slike je vidljivo da su prvi i treći bit u stanju logičke nule, dok su drugi i četvrti u stanju logičke jedinice. To znači da sa dvije logičke jedinice, možemo zaključiti da je paritetni bit ispravno generiran jer imamo parni broj jedinica.

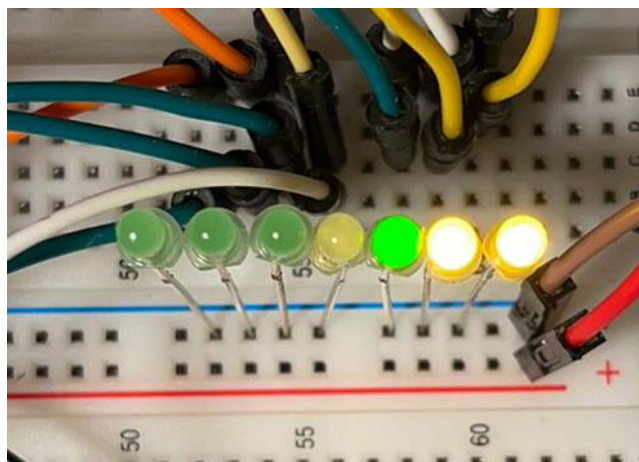
$$C3 = D4 \oplus D3 \oplus D2 \oplus P3 = 0 \oplus 1 \oplus 0 \oplus 1 = 0. \quad (4.3)$$

U drugome primjeru, kada su ulazni podatci 0001, izlaz iz prvog djela sklopa prikazan je na slici 4.5. Istom logikom, kao i u prethodnom primjeru, dolazimo do zaključka da je Hamming-ov kod ispravno generiran:

$$C1 = D4 \oplus D2 \oplus D1 \oplus P1 = 0 \oplus 0 \oplus 1 \oplus 1 = 0, \quad (4.4)$$

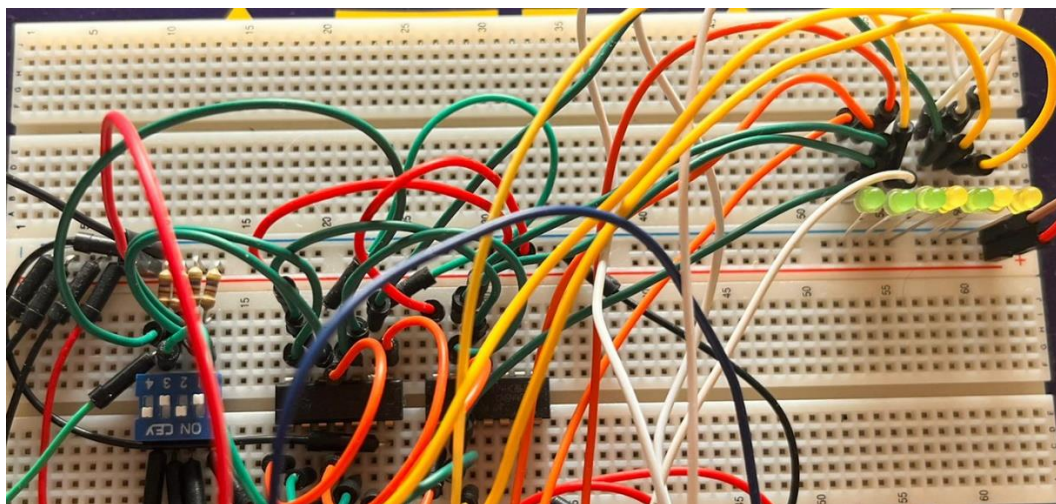
$$C2 = D4 \oplus D3 \oplus D1 \oplus P2 = 0 \oplus 0 \oplus 1 \oplus 1 = 0, \quad (4.5)$$

$$C3 = D4 \oplus D3 \oplus D2 \oplus P3 = 0 \oplus 0 \oplus 0 \oplus 0 = 0. \quad (4.6)$$



Slika 4.5. Izlaz kombinacije 0001.

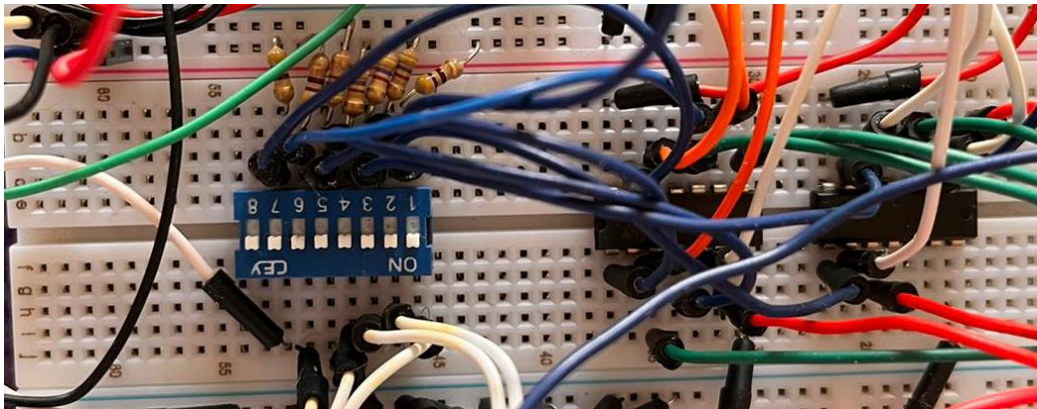
Prvi dio sklopa implementiran na eksperimentalnoj pločici prikazan je na slici 4.6.



*Slika 4.6. Prvi dio sklopa implementiran na eksperimentalnoj pločici.*

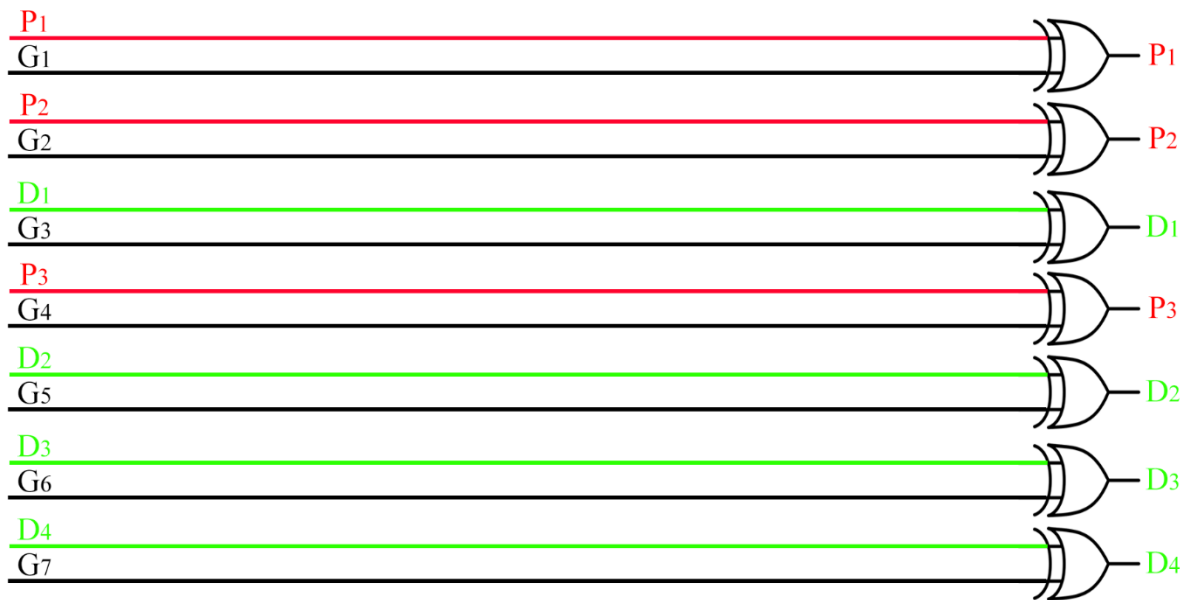
## 4.2 Drugi Dio: ručno ubacivanje pogreške

U drugome dijelu sklopa potrebno je unijeti pogrešku 7-bitnog podatka pomoću *push-up* konfiguracije. Taj način omogućava da se ubaci pogreška na bilo kojem bitu bez utjecaja na ulazni dio sklopa. Realizacija sklopa je proširena verzija 4-bitne *push-up* konfiguracije iz prvoga dijela sklopa. Kao što je vidljivo na slici 4.7., proces je isti kao za 4-bitni ulaz, razlika je što koristimo sedam ulaza, umjesto četiri.



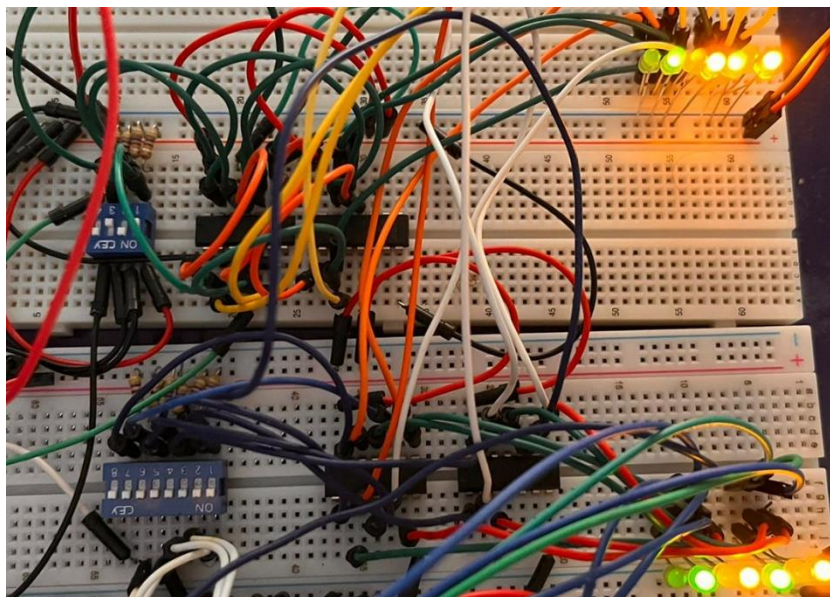
Slika 4.7. 8-bitna *push-up* konfiguracija.

Na slici 4.8 prikazana je shema sklopa koji unosi pogrešku. Na prvi. Ulaz XILI vrata spojen je izlaz iz prvog djela sklopa, tj. ispravni Hamming-ov kod, dok je na drugi ulaz spojen signal dobiven iz *push-up* konfiguracije. U slučaju da je DIP prekidač u stanju logičke nule, na XILI vratima imamo:  $P_1 \oplus 0 = P_1$ , tj. pojedinačna XILI vrata samo prosljeđuju prethodno generirani bit Hamming-ova koda. No, kada je DIP prekidač u stanju logičke jedinice, na XILI vratima imamo:  $P_1 \oplus 1 = \bar{P}_1$ , što znači da komplementiramo odgovarajući bit Hammingova koda, tj. unosimo grešku. Važno je napomenuti da je Hammingov kod sposoban detektirati (i ispraviti) samo jednu pogrešku, pa ako unesemo više od jedne pogreške, treći dio sklopa neće ispravno raditi.



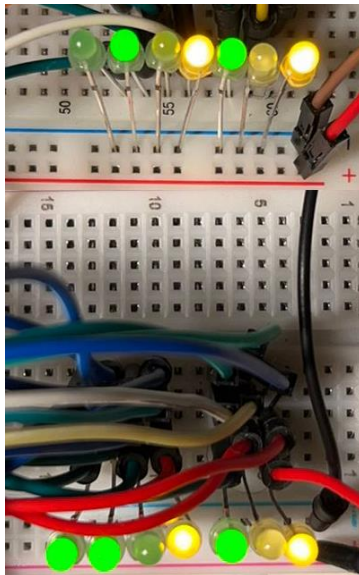
*Slika 4.8. Shema spoja.*

Na slici 4.9. prikazana je kombinacija ulaznog signala 0101, koji je ujedno ista kombinacija kao iz prvog dijela sklopa sa slike 4.6. U ovome slučaju, 8-bitni DIP prekidač ima sve sklopke na poziciji logičke nule 0 (LOW), što znači da greška nije unesena te je izlaz iz drugog djela sklopa jednak izlazu iz prvog. U slučaju da želimo inducirati pogrešku, jednostavno jednu od sklopki sa 8-bitnog DIP prekidača prebacimo na 1 (HIGH).



*Slika. 4.9. Kombinacija 0101 bez pogreške.*

U slučaju sa slike 4.10. sklopka na DIP switchu rednog broja 7. je u stanju logičke jedinice 1 (HIGH), čime se uzrokuje promjena na izlazu iz XILI vrata, kao što je vidljivo na slici, donje i gornje LED diode se razlikuju u samo sedmom bitu. Radi dokaza ispravnosti sklopa, koritit će se također i kombinacija sa slike 4.5. te će se uzrokovati pogreška na upaljenoj LED diodi.



Slika 4.10. Prikaz pogreške.

Na slici 4.11. uzrokovana je pogreška u paritetnom bitu  $P_1$  prebacivši sklopku na DIP switchu na 1 (HIGH) rednog broja 1. Rezultat te radnje jest ugašena LED dioda jer je sada jedan ulaz u XILI vrata 1 (HIGH).



Slika 4.11. Prikaz pogreške.

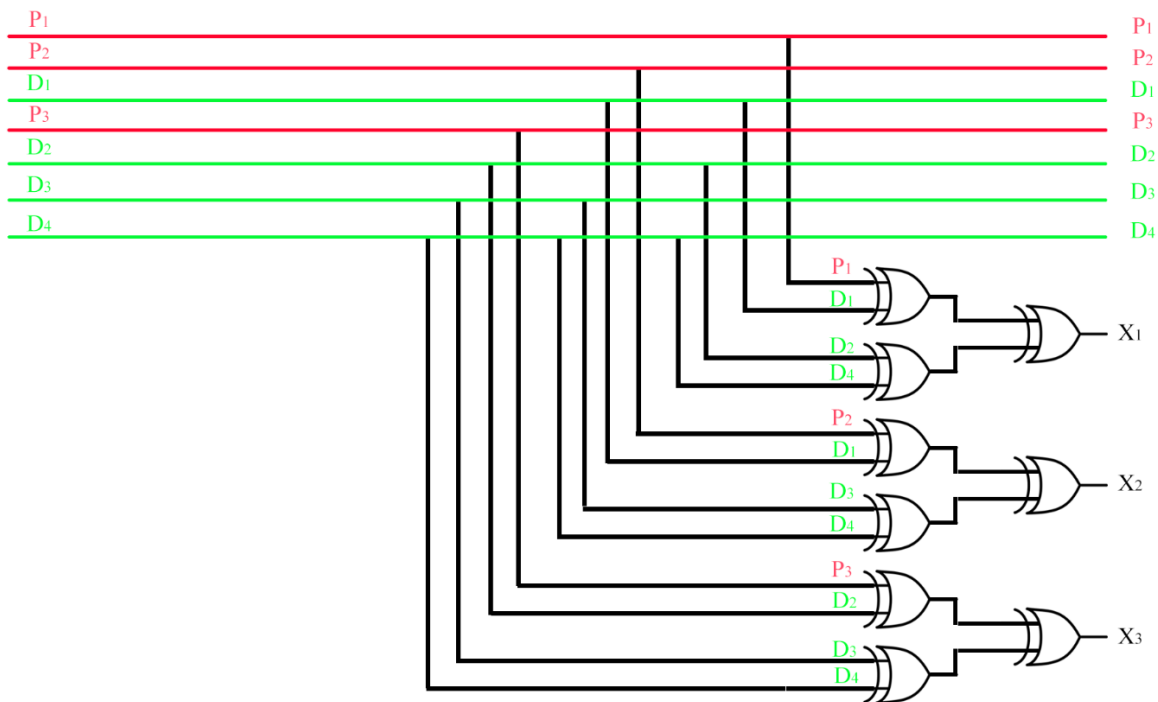
### 4.3. Treći dio: Detekcija i ispravlak pogreške

U ovome dijelu sklopa potrebno je detektirati u kojem bitu se dogodila pogreška, te koristeći tu informaciju, potrebno je automatski ispraviti pogrešku, tj. komplementirati onaj bit Hamming-ovog koda u kojem se dogodila pogreška.

#### 4.3.1. Detektiranje pogreške

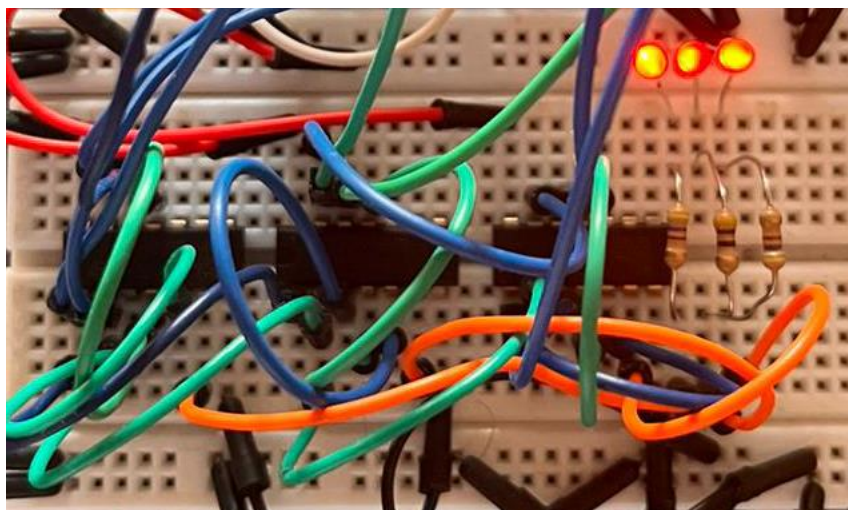
Prije nego se ispravi pogreška, potrebno je odrediti u kojem se bitu dogodila. Izlazni signal iz drugoga dijela sklopa koji sadrži pogrešku unosimo u ulaz trećeg sklopa.

Način na koji se detektira pogreška jest dosta sličan principu u prvome dijelu sklopa. Ponovo se koriste XILI vrata prema shemi sa slike 4.12. Detekcija greške se vrši provjerom pariteta. Kao što je prije rečeno, paritetni bit  $P_1$  osigurava paran paritet na bitovima  $P_1$ ,  $D_1$ ,  $D_2$  i  $D_4$ . Ako te bitove spojimo u jedna XILI vrata, izlaz iz XILI vrata  $X_1$  će biti logička nula ukoliko je na tim bitovima prisutan paran broj jedinica, što znači da se na tim bitovima nije dogodila pogreška. Ukoliko se dogodila greška na jednom od tih bitova, na ulazu XILI vrata bit će neparan broj jedinica, te će izlaz  $X_1$  biti u stanju logičke jedinice. Na sličan način potrebno je provjeriti i paritet preostale dvije kombinacije vezane za paritetne bitove  $P_2$  i  $P_3$ . Ako  $X_3X_2X_1$  interpretiramo kao binarni broj, taj broj indicira na kojem se bitu dogodila.



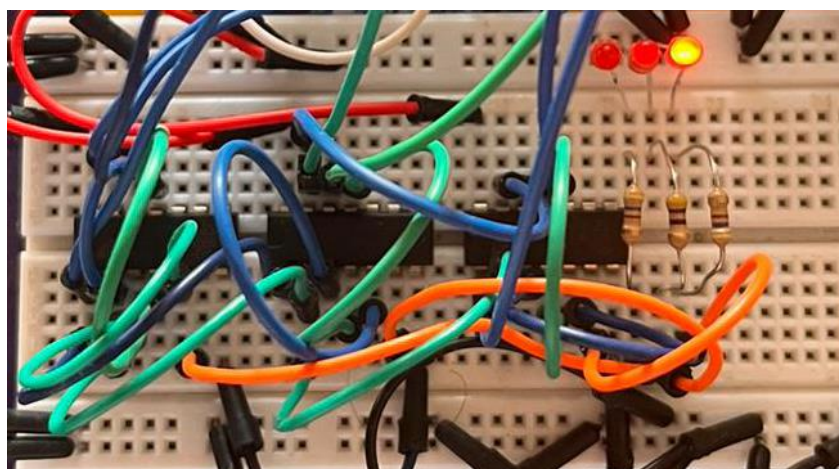
Slika 4.12. Shema sklopa.

Na slici 4.13. prikazana je točna detekcije pogreške u sedmom bitu iz primjera iz prošlog poglavlja. Sve tri LED diode su upaljene, što znači da je  $X_3X_2X_1 = 111_2 = 7_{10}$ .



*Slika 4.13. Detektiranje pogreške sedmog bita.*

Na slici 4.14. prikazana je točna detekcije pogreške u prvom bitu iz primjera iz prošlog poglavlja. LED diode indiciraju da je  $X_3X_2X_1 = 001_2 = 1_{10}$ .



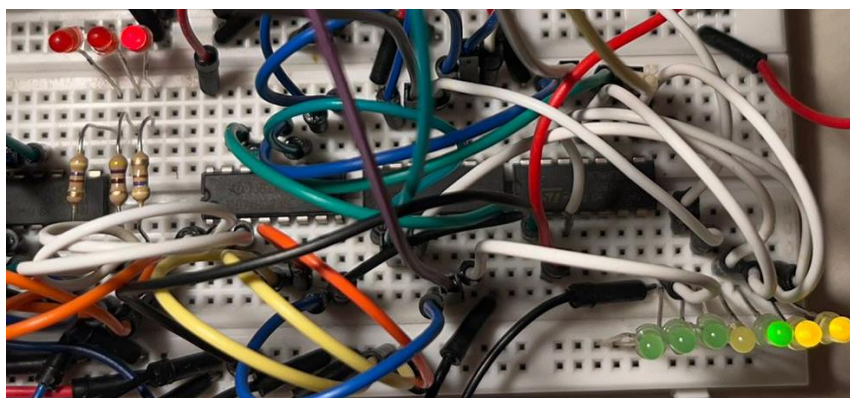
*Slika 4.14. Detektiranje pogreške prvoga bita.*

No, ovaj sklop će točno detektirati pogrešku samo u slučaju pogreške jednoga bita, u slučaju dvije ili više pogreške, sklop više ne obavlja svoj zadatak ispravno, te informacija koju daje nije ispravna.

Zanimljiva činjenica vezana za sklopove koji se bave detekcijom pogrešaka jest, ako osoba koja radi jedan takav sklop napravi pogrešku prilikom spajanja sklopa, sklop automatski navodi osobu u kojem dijelu sklopa se dogodila pogreška tijekom spajanja.

#### 4.3.2. 3/8 dekodera i ispravljanje pogreške

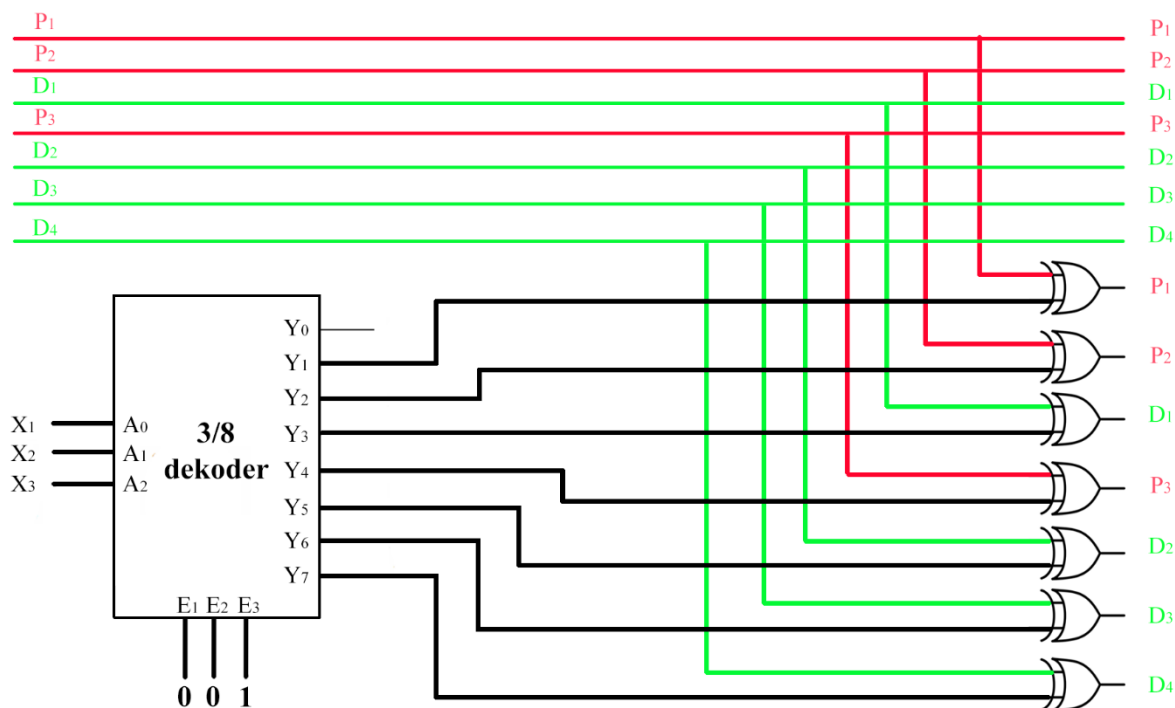
Signali  $X_3-X_1$  koji indiciraju na kojem se bitu dogodila greška se zatim dovode na adresne ulazne 3/8 dekodera. To znači, da će jedan, adresirani izlaz dekodera biti u stanju 1, a preostali 0. Npr. ako je ulaz u dekodera  $X_3X_2X_1 = 111$ , što indicira da se dogodila greška u 7. bitu, izlazi  $Y_0-Y_6$  će biti u stanju logičke nule, dok će  $Y_7$  biti u stanju logičke jedinice. Tu informaciju zatim možemo iskoristiti za selektivno negiranje Hamming-ova koda sa unesenom pogreškom, na način sličan kao i u drugome dijelu sklopa, i na izlazu ćemo dobiti ispravljeni Hamming-ov kod. Detektiranje i ispravljanje prikazano je na slici 4.15.



*Slika 4.15. Detektiranje i ispravljanje pogreške.*



Shema sklopa za ispravljanje pogreške prikazana je na slici 4.15. Izlaz dekodera  $Y_0$  nije spojen, jer u slučaju  $X_3X_2X_1 = 000_2$ , što indicira da se greška nije dogodila, nije potrebno negirati niti jedan bit Hamming-ova koda.



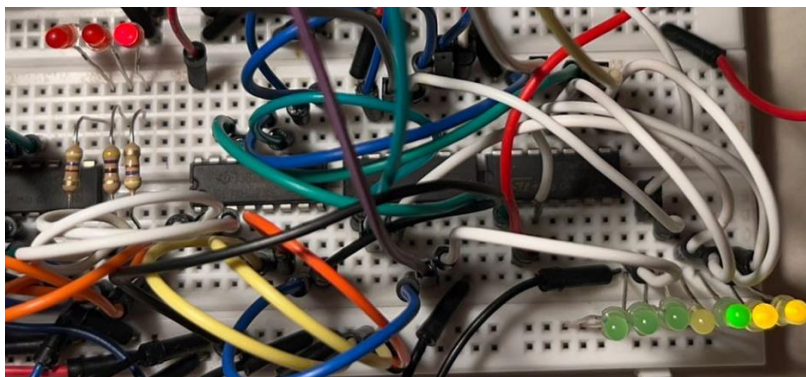
Slika 4.15. Shema sklopa za ispravljanje pogrešaka.

Ponovo koristeći XILI vrata može se ispraviti pogreška. U ovome slučaju, ako izlaz iz triju LED dioda prikazuje  $111_2$ , na zadnjem izlazu dekodera, koji predstavlja sedmi bit, iznosit će 1 (HIGH), te pripadajuća XILI vrata će negirati sedmi bit Hamming-ovog koda i time ispraviti pogrešku.



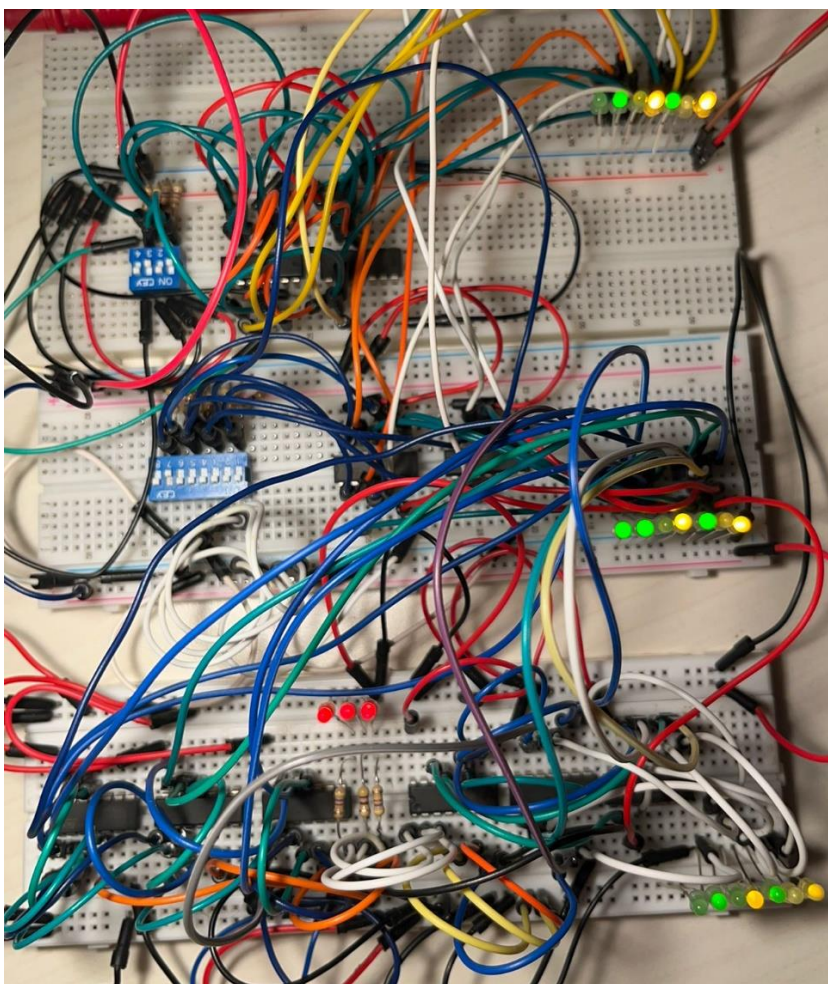
Slika 4.16. Ispravljeni izlaz kombinacije 0101.

Za slučaj 0001 sa slike 4.11., prikazivati će se bit  $001_2$ , koji je prvi izlaz iz dekodera i onda se pogreška ispravlja na tom bitu.



*Slika 4.17. Ispravljeni izlaz kombinacije 0001.*

Sa slika 4.16. i 4.17. vidljivo je stanje na izlazu sklopa, te je on jednak ulaznom signalu iz prvoga dijela sklopa sa slike 4.4. i 4.5., čime je uspješno detektirana te automatski ispravljena pogreška u signalu. Prikaz cijelog sklopa prikazan je na slici 4.18.



*Slika 4.18. Prikaz cijeloga sklopa.*

## 5. Zaključak

Prilikom pohranjivanja podataka, te njihovog prijenosa iz jednog oblika u drugi, neovisno o načinu na koji se proces odvija, uvijek postoji šansa da će se dogoditi pogreška. Bez metoda ispravljanja pogrešaka koje su matematičari izmislili još kasnih 1940-tih, podaci koje šaljemo bi zauvijek izgubili svoje originalno značenje.

Hamming-ov kod (7,4) je jedna od brojnih metoda koja se koristi u detektiranju i ispravljanju pogrešaka, te bez takvih kodova za ispravljanje pogrešaka, današnji život koji se temelji od svakidašnjih razmjena informacija poput slanja poruka i fotografija pomoću mobitela, e-maila, društvenih mreža, itd., bio bi nezamisliv.

## 6. Literatura

- [1] Maini, Digital Electronics Principles, Devices and Applications
- [2] Volarić, I.: predavanja iz kolegija „Digitalna elektronika“, Tehnički fakultet u Rijeci, 2019.
- [3] s interneta, <https://www.britannica.com/biography/Richard-Wesley-Hamming>, 8. rujna 2022.
- [4] s interneta, [https://hr.upwiki.one/wiki/Richard\\_Hamming#Bibliography](https://hr.upwiki.one/wiki/Richard_Hamming#Bibliography), 8. rujna. 2022.
- [5] s interneta, [https://en.wikipedia.org/wiki/Hamming\(7,4\)](https://en.wikipedia.org/wiki/Hamming(7,4)), 14. rujna. 2022.
- [6] s interneta, <https://components101.com/misc/breadboard-connections-uses-guide> 24. kolovoza 2022.
- [7] s interneta, [https://en.wikipedia.org/wiki/Jump\\_wire](https://en.wikipedia.org/wiki/Jump_wire), 24. kolovoza 2022.
- [8] Stojković, N.: predavanja iz kolegija „Električne mreže“, Tehnički fakultet u Rijeci, 2019.
- [9] s interneta, <https://www.buildingcontrolsgroup.com/500ohmresistor.html>, 24. kolovoza 2022.
- [10] s interneta, <https://www.eeeguide.com/interfacing-dip-switch/>, 24. kolovoza 2022.
- [11] s interneta, [https://www.researchgate.net/figure/LED-Light-Emitting-Diode\\_fig2\\_338229502](https://www.researchgate.net/figure/LED-Light-Emitting-Diode_fig2_338229502), 24. kolovoza 2022.
- [12] s interneta, <https://kronos.hr/prekidac-dip-i-bcd/7422-dil-4-schalter.html>, 24. kolovoza 2022.
- [13] s interneta, <https://kronos.hr/prekidac-dip-i-bcd/7194-dil-8-schalter.html>, 24. kolovoza 2022.
- [14] s interneta, <https://hr.eyewated.com/sto-je-dip-prekidac/>, 24. kolovoza. 2022.
- [15] s interneta, <https://riverglennapts.com/hr/logic-gate/541-x-or-gate-and-x-nor-gate.html>, 24. kolovoza 2022.
- [16] s interneta, <https://kronos.hr/ic-ttl/8202-sn-74hc86.html>, 24. kolovoza 2022.
- [17] s interneta, [https://www.ti.com/lit/ds/symlink/sn54hc86.pdf?ts=1661440689192&ref\\_url=https%253A%252F%252Fuge-one.com%252F](https://www.ti.com/lit/ds/symlink/sn54hc86.pdf?ts=1661440689192&ref_url=https%253A%252F%252Fuge-one.com%252F), 24. kolovoza 2022.
- [18] s interneta, <https://www.ti.com/product/CD74HC238>, 7. rujna 2022.

## Sažetak i ključne riječi

Hamming-ov kod je metoda ispravljanja pogrešaka prilikom slanja i primanja podataka između odašiljača i prijemnika. U ovome radu njegova funkcija detektiranja te ispravljanja pogrešaka je realizirana u hardverskom obliku korištenjem osnovnih digitalnih kombinacijskih sklopova: XILI vrata i 3/8 dekodera.. Sklop se sastoji od tri dijela, generiranje Hamming-ovoga koda, umetanja pogreške, detektiranje i automatsko ispravljanje te iste pogreške, te sve troje povezano u cjelinu čini sklop za detektiranje i ispravljanje pogrešaka.

**Ključne riječi:** Hamming-ov kod, kodovi za ispravljanje pogrešaka.

## Summary and key words

Hamming code is a method for error correction which occurred while sending and receiving data between a transmitter and a receiver. In this bachelor's thesis, the method for error detection and correction is implemented in hardware form, utilizing basic combinational digital modules, namely XOR gates and 3/8 decoder. The circuit consists of three parts: Hamming code generation, error insertion, and detection with automatic correction of the same error. All three are connected as one and they form the circuit for error detection and correction.

**Keywords:** Hamming code, error correcting codes.