Zaštitno kodiranje signala

Laboratorijska vježba 14

SP

Kodiranje, dekodiranje i ispravak pogrešaka difuzijskim pragom

Student:

prezime	Ime	mat. broj
	Laboratorij	Datum

Sadržaj:

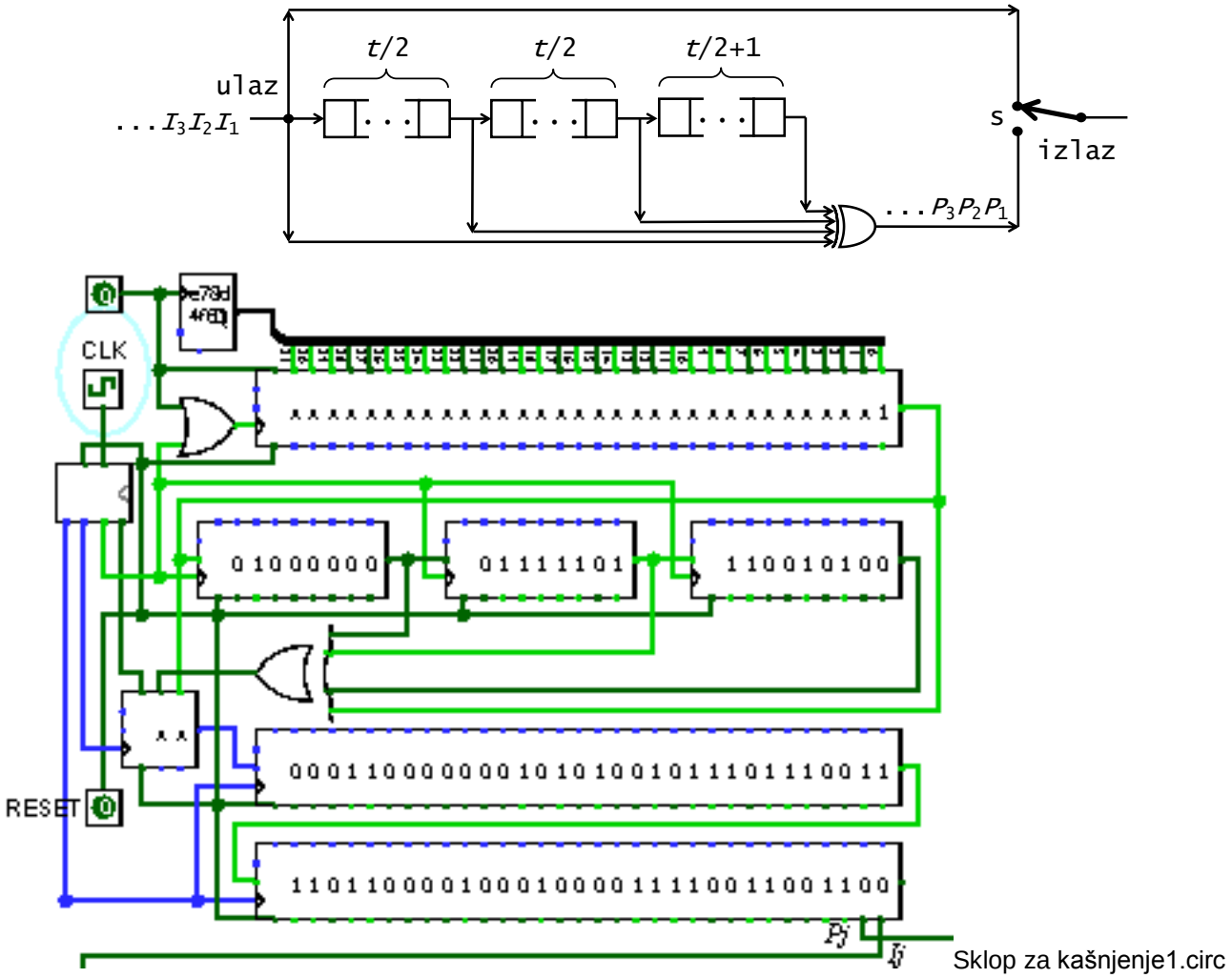
Sazdano od:

Napomena: Cjelovit opis vježbe nalazi se na: MOODLE

14. LABORATORIJSKA VJEŽBA 14: KODIRANJE KONVOLUCIJSKIH KÔDOVA IMPULSNIM ODZIVOM

14.1. Postupak kodiranja

Koder kôda namijenjen dekodiranju difuzijskim pragom, prikazuje slika 6.2. zajedno s njegovom Logisim™ inačicom.



Slika 6.2 Kôder konvolucijskih kodova difuzijskim pragom dekodiranja[2]

Koder se sastoji od posmičnih registara, podijeljenih u tri dijela. Prva dva dijela su duljine $t/2$, a treći je duljine $(t/2)+1$, gdje t označava duljinu praska što ga treba ispraviti. Imajte na umu da:

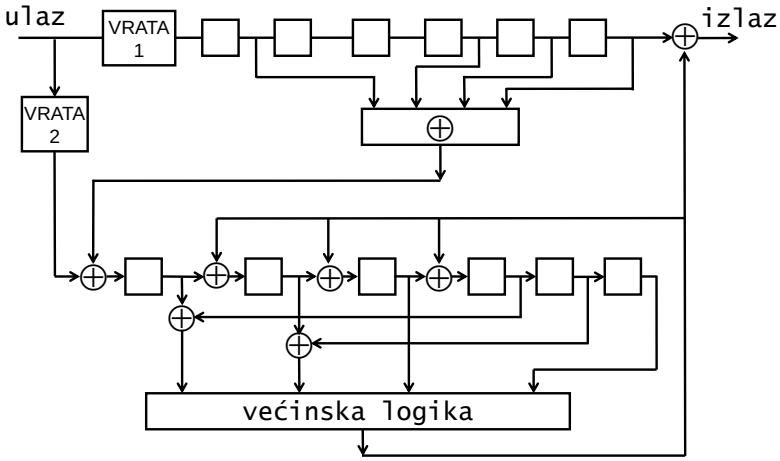
- t treba biti parna veličina, a
- $t/2$ treba biti cio broj.

Postoje XOR vrata čija su četiri ulaza jasno naznačena na slici. Uzastopni izlazi XOR vrata su paritetni bitovi. U koder serijski ulaze informacijski bitovi … I_3 I_2 I_1 počevši s I_1. Ne zaboravite da računamo s desna pa je I_1 prvi bît što ulazi u krug, zatim slijedi I_2, itd. Informacijski bitovi na ulazu, razilaze se u tri smjera. Oni paralelno ulaze u *posmični registar*, u *XOR vrata* i izravno *na izlaz*.

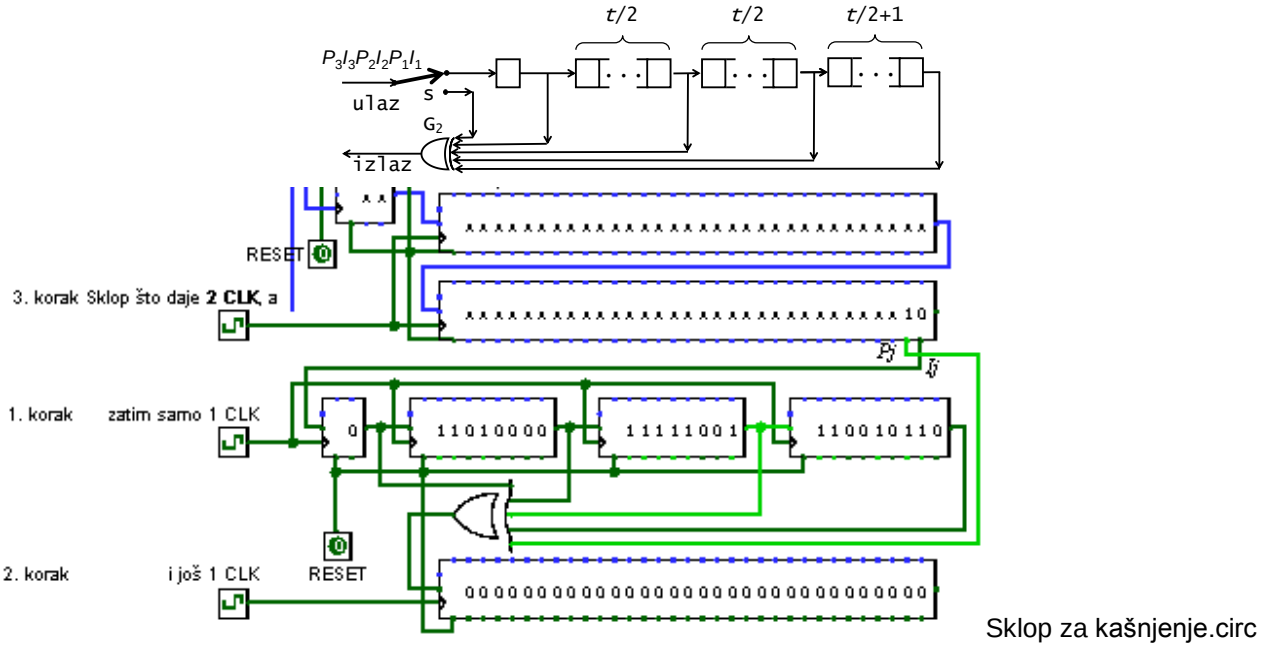
Sklopka S naizmjence spaja gornji i donji kanal tj., prekidač se prebacuje dvostrukom brzinom od brzine kojom se unose informacijski bitovi. Dakle, za svaki učitan informacijski bît, kroz prekidač izlaze *dva* bita i prenose se dalje. Znači da je brzina prijenosa jednaka dvostrukoj brzini unosa. Ova dva bita su: *informacijski bît* što se unosi gornjim kanalom i *paritetan bît*, što predstavlja izlaz XOR vrata, a puni se preko donjega kanala. Kako su … I_3 I_2 I_1 uzastopne vrijednosti ulaznih bitova, a … P_3 P_2 P_1, su uzastopni paritetni bitovi što ih stvaraju XOR vrata, izlaz iz odašiljača je … P_3 I_3 P_2 I_2 P_1 I_1.

[2] Napraviti ovaj koder kao vježbu!!!

[2] Napraviti ovaj koder kao vježbu!!!



Slika 6.3 prikazuje dekoder koji je "obrnuta slika" kodera na slici 6.2.



Slika 6.3 Otkrivanje pogrešaka. Obrnuta (lijevo-desno) slika kodera[3] [4]

Ovaj niz … P_3 I_3 P_2 I_2 P_1 I_1 što ga stvara kôder, izravno se dovodi (pogreške još ne djeluju) na ulaz sklopke S koja razdvaja informacijske bitove (desno) … I_3 I_2 I_1 i paritetne bitove … P_3 P_2 P_1 (dolje) u zasebne tokove. Informacijski bitovi ulaze u gornji kanal i posmiču se u 4 serijski spojena registra. Paritetni bitovi ulaze u donji kanal i obrađuju ju ih XOR vrata ali u sprezi s informacijskim bitovima na izlazima iz pojedinih stupnjeva gornjih registara. U razdoblju nekoga takta #*j* (gdje je #1 referentna točka u vremenu) pet ulaza u XOR vrata su P_j, I_j, $I_{(j-t/2)}$, $I_{(j-t)}$, $I_{(j-3t/2-1)}$. (Ulaz $I_{(j-t/2)}$ označava (j–t/2)-ti informacijski bīt.)

---

[3] Napomena: U trenutku #*j* s ulaza se uzimaju 2 bita (… P_j, I_j). Zato što se bitovi mogu oštetiti u prijenosnome kanalu, pišemo ih kao (… p_j, i_j). Njihove vrijednosti preslikavaju se na ulaz u prvi registar odnosno na ulaz XOR vrata. Nakon prvoga posmika u gornji dio dekodera, informacijski bit na ulazu u registar, ulazi u sam registar (1. posmik). Donji dio dekodera, posmiče se NAKON posmika bita u gornji dio registra (2. posmik), jer se jedino tako mogu usporediti informacijski i paritetan bit kada se istovremeno pojave na ulazima XOR vrata. Vrijednost informacijskoga bita u 1. registru, preslikava se na izlaz registra, što znači da se pridružuje paritetnome bitu na jednome od 5 ulaza u XOR vrata pa sada oba bita ravnopravno sudjeluju u oblikovanju izlaza XOR vrata. Izlaz iz XOR vrata jednak je 1, ako je SAMO jedan od ulaznih bitova jednak 1. Konačan rezultat dekodiranja je stanje "sve 0" donjega registra, što ukazuje na *nepostojanje* praskovitih pogrešaka.

[4] Napraviti ovaj (koder i) dekoder kao vježbu! U rezultat kodiranja (donja dva registra u koderu) unijeti pogrešku i ispratiti kako se ona otkriva i ispravlja.

Promatrajući kôder na slici 6.2, napominjemo da je paritetan bit $P_j = I_j \oplus I_{(j-t/2)} \oplus I_{(j-t)} \oplus I_{(j-3t/2-1)}$. To znači da je izlaz iz XOR vrata na slici 6.3 stalno 0, dok je ulaz P_j uvijek jednak zbroju ostala četiri ulaza u XOR vrata. Tok … $P_3\ I_3\ P_2\ I_2\ P_1\ I_1$ što ga je stvorio kôder, prenosi se preko kanala, ali može doživjeti pogreške. Označimo njegovu primljenu inačicu kao … $p_3\ i_3\ p_2\ i_2\ p_1\ i_1$, …. Kada se ovi primljeni bitovi unesu u krug na način prikazan na slici 6.3, izlazi iz XOR vrata čitavo vrijeme su 0, ali samo u slučaju ispravno primljenoga niza informacijskih i paritetnih bitova. Neka je:

(1) $p_j = P_j \oplus Y_j$

(2) $i_j = I_j \oplus Z_j$

Bitovi Y_j i Z_j "ukazuju na pogrešne bitove". Ako je $Y_j = 1$, onda je paritetan bit p_j pogrešan (ali on se ne mora ispravljati). Ako je $Z_j = 1$, onda je i_j pogrešan (informacijski bit se ispravlja). Za … $p_3\ i_3\ p_2\ i_2\ p_1\ i_1$ kao ulaz u krug na slici 6.3, izlaz iz XOR vrata, kada se primi neki p_j je:

$$p_j = i_j \oplus i_{(j-(t/2))} \oplus i_{(j-t)} \oplus i_{(j-(3t/2)-1)}$$

Na temelju jednadžbi (1) i (2) ovaj izlaz je:

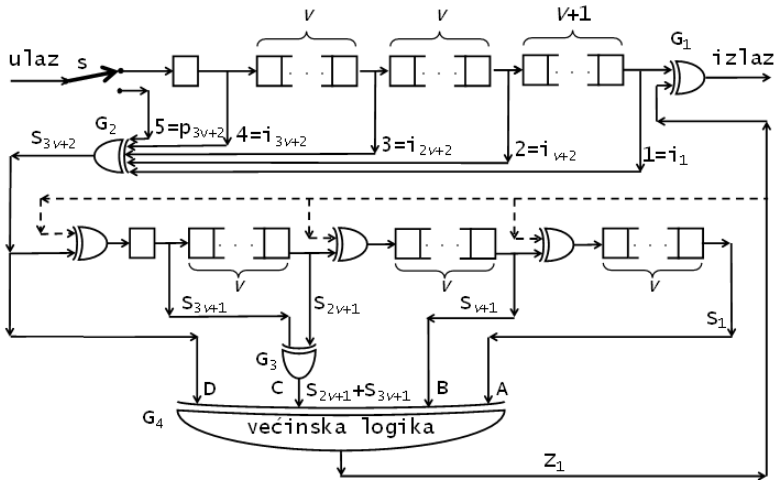
$$\{P_j \oplus Y_j\} \oplus \{I_j \oplus Z_j\} \oplus \{I_{j-(t/2)} \oplus Z_{j-(t/2)}\} \oplus \{I_{(j-t)} \oplus Z_{(j-t)}\} \oplus \{I_{(j-(3t/2)-1)} \oplus Z_{(j-(3t/2)-1)}\} =$$
$$= \{P_j \oplus I_j \oplus I_{j-(t/2)} \oplus I_{(j-(3t/2)-1)}\} \oplus \{Y_j \oplus Z_j \oplus Z_{j-(t/2)} \oplus Z_{(j-t)} \oplus Z_{(j-(3t/2)-1)}\}$$

Imajte na umu da je sada sadržaj {~~prvoga para zagrada~~} jednak 0, kao što se već prije opisalo. *Izlaz iz XOR vrata tada je sadržaj* {*drugoga para zagrada*}, *što predstavlja zbroj odgovarajućih bitova pokazatelja pogreške*. Kada radimo *blok-kodovima*, pokazalo se da se sindrom pogreške dobije ***množenjem*** *primljene poruke* i *matrice pariteta*, a ***jednak*** je rezultatu dobivenome samo ***množenjem*** *uzorka pogreške* i ove *matrice*. *Prethodno razmatranje svojstveno je konvolucijskim kodovima.*

Od sada pa nadalje, izlazi XOR vrata na slici 6.3 izrazit će se samo pojmovima bitova naznake pogreške Y_j i Z_j. Dekoder kodova dekodiranja difuzijskim pragom temelji se prije svega na krugu što ga prikazuje slika 6.3. Kao što će se uskoro pokazati, ovome osnovnome sklopu dodaje se ***neki*** sklop zbog otkrivanja je li određeni Z_j (*pokazatelj pogreške informacijskoga bita*) jednak 1. U tome slučaju, odgovarajući informacijski bīt I_j prepoznao se kao pogrešan pa ga se ispravlja.[5]

Svrha dekodera je samo ispravak informacijskih bitova. To ne znači da paritetni bitovi ne smiju biti pogrešni, jednostavno, nije ih potrebno ispravljati. Iako *moramo* temeljiti shemu ispravke na činjenici da i oni mogu biti pogrešni. Ovdje vidimo jasnu razliku između *kôda što se dekodira difuzijskim pragom* i *kôdova kojima smo se do sada bavili*. U tim kodovima, *jednako* smo se odnosili prema *paritetnim* i *informacijskim* bitovima, tako da shema dekodiranja ispravlja *bilo koji* primljen bīt na isti način. Dekoder kôda dekodiranja difuzijskim pragom prikazuje slika 6.4.

Ima još jedan slučaj, a to je neispravnost i informacijskoga i paritetnoga bita?

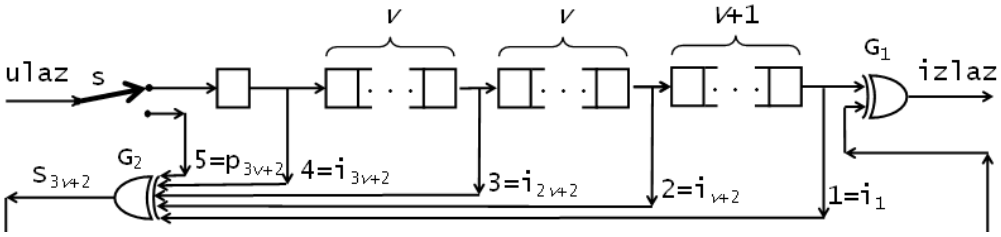


Slika 6.4 Ispravak otkrivenih pogrešaka. Dekoder difuzijskim pragom - dekodiranje konvolucijskih kodova većinskom logikom[7]

[5] Laboratorijska vježba!

[7] Pri otkrivanju i ispravci pogrešaka, **dekodiranje većinskom logikom** je način dekodiranja ponavljajućih kodova, a temelji se na pretpostavci da se upravo najčešće pojavljuje prenošen simbol.

Zbog lakšega označavanja, ovdje se vrijednost $t/2$ označava kao v. Razne vrijednosti, napisane na različitim mjestima crteža su one, dobivene kada se primi bît $p_{(3v+2)}$ (on upravo ulazi u "`ulaz`" ulaz u gornjem lijevom kutu). Oznaka i_1 ovdje označava prvi primljen bît. To je primljena inačica I_1, što je bio bît kojime se započeo prijenos. Nema bitova što se prenose prije bîta I_1. Gornji dio crteža jednak je crtežu na slici 6.3, gdje v označava $t/2$.



Bît i_1 upravo izlazi na desnu stranu i ulazi na ulaz #1 XOR vrata G_2. Ulaz #2 od G_2 je informacijski bît što dođe v+1 ($=\frac{t}{2}+1$) mjesta nakon i_1, jer, kao što se vidi na slici, postoji kašnjenje od v+1 bita između ulaza #1 i #2. Vrijednost na ulazu #2 je, dakle, $i_{(v+2)}$. Ulaz #3 kasni v bitova u odnosu na ulaz #2. Vrijednost što ondje postoji, je dakle $i_{(2v+2)}$. Iz istoga razloga ulaz #4 trenutno ima vrijednost $i_{(3v+2)}$. Ulaz #5 ima vrijednost $p_{(3v+2)}$ i predstavlja primljen paritetan bît. $S_{(3v+2)}$ je trenutan izlaz iz G_2. (Općenito, ako je p_j bît primljen u određenome trenutku, za neki j, onda je S_j izlaz iz G_2 u tome trenutku). Kao što je navedeno, izlaz iz G_2 jednak je zbroju bitova naznake pogreške. Onda imamo:

(a) $S_{(3v+2)} = Y_{(3v+2)} \oplus Z_{(3v+2)} \oplus Z_{(2v+2)} \oplus Z_{(v+2)} \oplus Z_1$

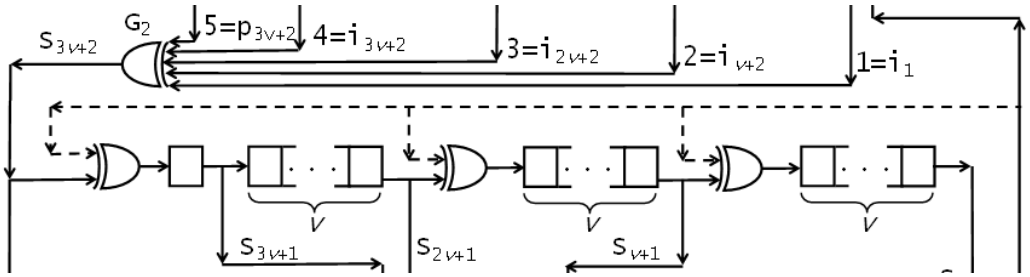
Napomene:

$S_{(indeks)}$ … označava izlaze iz XOR sklopa G_2 u različitim trenutcima,

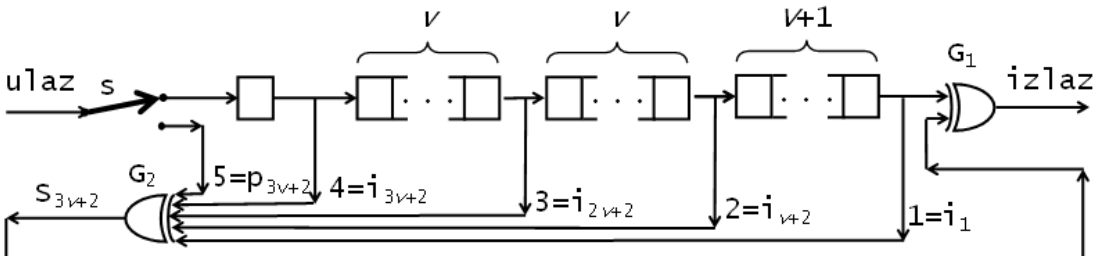
$Y_{(indeks)}$ … označava paritetne bitove u različitim trenutcima,

$Z_{(indeks)}$ … označava informacijske bitove u različitim trenutcima, a

Z_1 … označava prvi informacijski bit.



Izlaz iz G_2 ulazi u registar koji se sastoji od četiri dijela. Prvi dio ima jedno stanje, a ostala tri dijela su dugačka v stanja. XOR vrata između ovih dionica obradit će se poslije. Također, zanemarimo za trenutak nacrtane isprekidane crte. Ako je izlaz sadašnjega ciklusa iz G_2 jednak $S_{(3v+2)}$, izlazi sljedećih odjeljaka donjega registra su $S_{(3v+1)}$, $S_{(2v+1)}$, $S_{(v+1)}$ i S_1, kao što prikazuje crtež. Bît vrijednosti S_1, što sada izlazi iz krajnjega desnoga stanja registra, bio je izlazna vrijednost G_2, kada se primio p_1. (Pazi! Ovo je uputa za sinkronizaciju rada čitavoga sklopa!)



Provjerimo sada svih pet ulaza u G_2 u vrijeme nakon toga prvoga takta (i prihvata, najprije i_1, a iza njega i p_1). Ulaz #5 imao je vrijednost p_1, a ulaz #4 imao je vrijednost i_1. Budući da je i_1 prvi primljen

---

[7] Pri otkrivanju i ispravci pogrešaka, **dekodiranje većinskom logikom** je način dekodiranja ponavljajućih kodova, a temelji se na pretpostavci da se upravo najčešće pojavljuje prenošen simbol.

bĩt, gornji niz ostalih registara još uvijek je ispunjen nulama pa ulazi #3, #2, #1 očitavaju samo 0. Dakle na ulazima XOR vrata G_2, imamo zbroj SAMO dvaju bitova što mogu biti različiti od nule, i_1 (na izlazu prvoga registra) i p_1 (na ulazu "**ulaz**" u dekoder), a SVI ostali bitovi još uvijek SIGURNO imaju vrijednosti 0:

(b) $S_1 = Y_1 \oplus Z_1$.

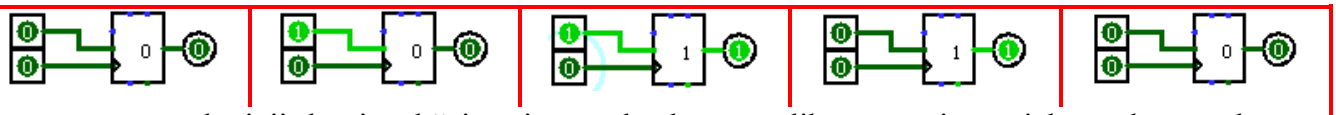
Kada se primio $p_{(v+1)}$ na izlazu G_2 bila je vrijednost $S_{(v+1)}$. U ovome trenutku, vrijednost ulaza #3 je i_1. Istovremeno, ulaz #4 ima vrijednost $i_{(v+1)}$ a ulaz #5 ima $p_{(v+1)}$. Izlazi #1 i #2 još uvijek su 0. Onda imamo:

(c) $S_{(v+1)} = Y_{(v+1)} \oplus Z_{(v+1)} \oplus Z_1$.

Kada se primio $p_{(2v+1)}$ onda se generirao $S_{(2v+1)}$. Tada je i_1 bila vrijednost na ulazu #2 u G_2. Ulazi #3, #4, odnosno #5 imaju vrijednosti $i_{(v+1)}$, $i_{(2v+1)}$, odnosno $p_{(2v+1)}$. Onda imamo:

(d) $S_{(2v+1)} = Y_{(2v+1)} \oplus Z_{(2v+1)} \oplus Z_{(v+1)} \oplus Z_1$.

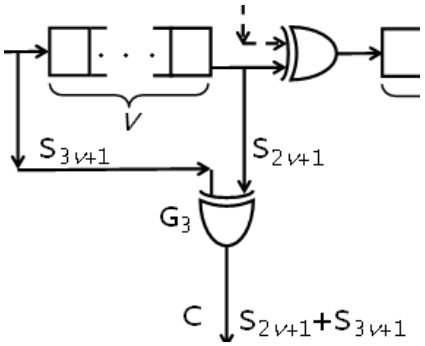
Kada se primio $p_{(3v+1)}$ onda se generirao $S_{(3v+1)}$. Ovo je posljednji put da ulaz #1 u G_2 i dalje ima vrijednost 0 prema definiciji. Sada i_1 boravi u predzadnjemu desnome stanju gornjega registra (registar je dužine v+1) te će se pojaviti na ulazu #1 tek sljedećim posmikom.



U LogisimTM, krajnji desni sadržaj registra, odmah se preslikava na njegov izlaz, a bĩt na ulazu u registar čeka okidni impuls i onda ulazi u registar. Istovremeno, taj okidni impuls prazni trenutno stanje registra pa se vrijednost novo unesenoga bita s ulaza, sada preslikava na njegov izlaz.

Ulazi #2, #3, #4 odnosno #5 imaju vrijednosti $i_{(v+1)}$, $i_{(2v+1)}$, $i_{(3v+1)}$ odnosno $p_{(3v+1)}$. Onda imamo:

(e) $S_{(3v+1)} = Y_{(3v+1)} \oplus Z_{(3v+1)} \oplus Z_{(2v+1)} \oplus Z_{(v+1)}$

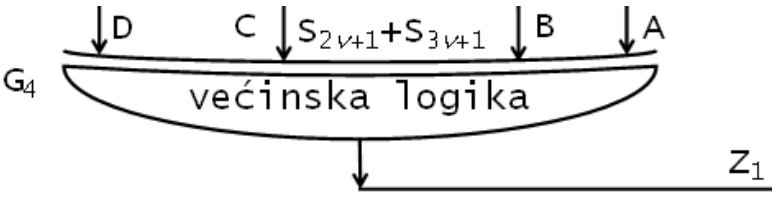


Izlaz iz vrata G_3 je:

(f)

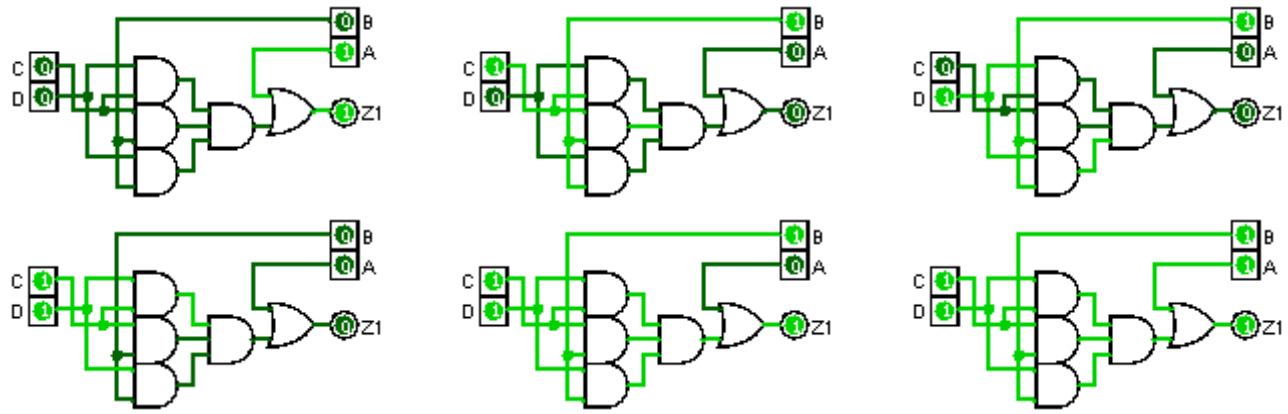
$$S_{(2v+1)} \oplus S_{(3v+1)} = \{Y_{(2v+1)} \oplus \cancel{Z_{(2v+1)}} \oplus \cancel{Z_{(v+1)}} \oplus Z_1\} \oplus \{Y_{(3v+1)} \oplus Z_{(3v+1)} \oplus \cancel{Z_{(2v+1)}} \oplus \cancel{Z_{(v+1)}}\}$$
$$= Y_{(3v+1)} \oplus Y_{(2v+1)} \oplus Z_{(3v+1)} \oplus Z_1$$

Imajte na umu da se $Z_{(2v+1)}$ i $Z_{(v+1)}$ poništavaju kao parovi.

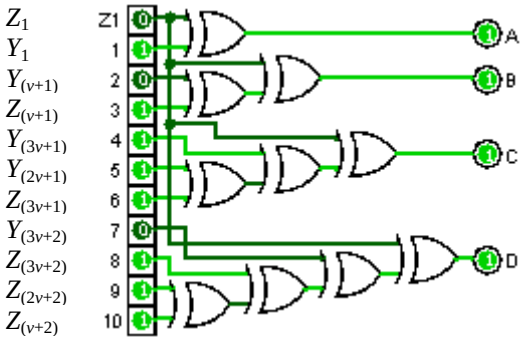


Vrata G_4, pod nazivom ***većinska logika***, logička su vrata čiji je izlaz jednak 1 onda i samo onda ako je većina njegovih ulaza (A, B, C i D) jednaka 1 (odnosno, ako barem 3, ili sva 4 ulaza imaju vrijednost 1). Inače, izlaz je 0. Izlaz Z_1 je rezultirajući informacijski bit što se u prvome trenutku označio kao i_1, a u toku prethodne rasprave nazvao se Z_1. On je konačan rezultat većinske logike i višestrukih

paritetnih provjera u gornjemu sklopi dekodera. Sklop većinske logike za ovaj dekoder izgleda kao na slici (Slika6.4 Koder.circ):



Na temelju jednadžbi (b), (c), (f) i (a), vrijednosti četiriju ulaza vrata većinske logike su (slova se odnose na one označene na slici) (Slika6.4 VRLO DOBAR.circ):



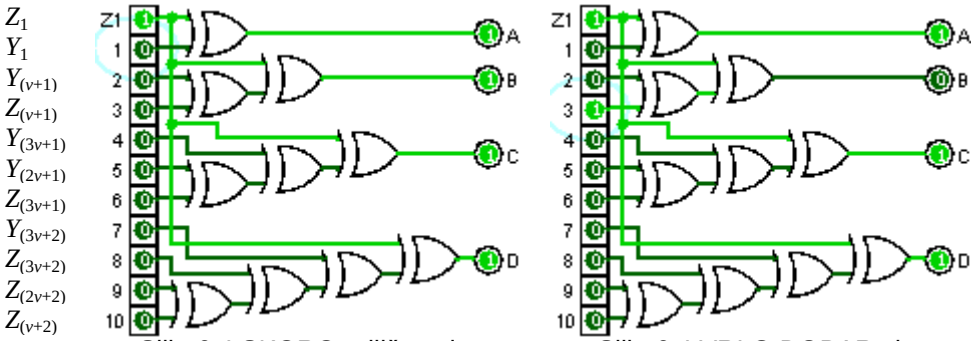
$A = Y_1 \oplus Z_1$

$B = Y_{(v+1)} \oplus Z_{(v+1)} \oplus Z_1$

$C = Y_{(3v+1)} \oplus Y_{(2v+1)} \oplus Z_{(3v+1)} \oplus Z_1$

$D = Y_{(3v+2)} \oplus Z_{(3v+2)} \oplus Z_{(2v+2)} \oplus Z_{(v+2)} \oplus Z_1$

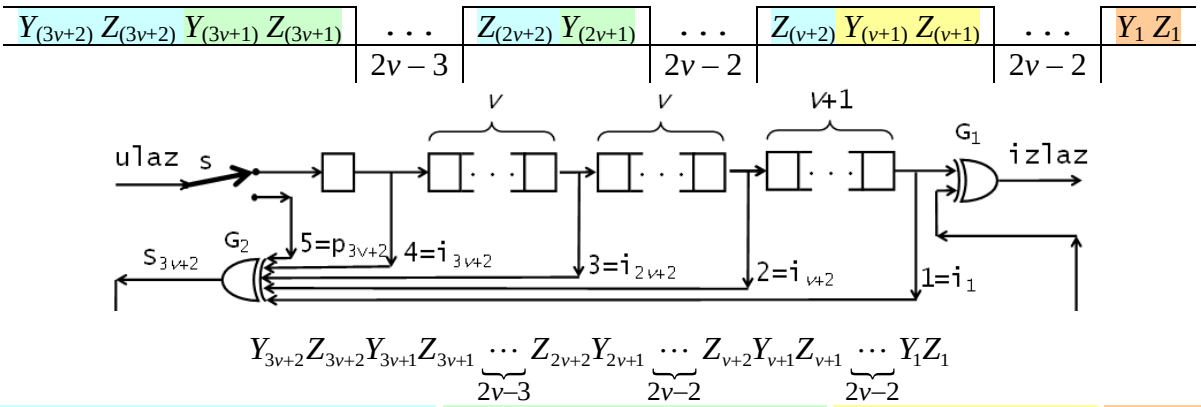
Napomenimo sada osnovna svojstva jednadžbi za A, B, C i D. Postoji ukupno jedanaest različitih *bitova indikatora pogrešaka* u četiri jednadžbe. Član Z_1 pojavljuje se u *svakome izrazu*. Svaki od deset preostalih različitih bitova-indikatora pogrešaka, pojavljuje se samo jedanput. Dakle, osim Z_1, niti jedan od preostalih 10 bitova-indikatora ne ponavlja se u dvije ili više jednadžbi. To znači, ako je **samo** $Z_1 = 1$ (a ostali bitovi-indikatori su 0), onda su: A = B = D = C = 1.



Slika6.4 SKORO odličan.circ Slika6.4 VRLO DOBAR.circ

Ako je osim Z_1, još **samo jedan** (bilo koji) bît od deset preostalih jednak 1, onda barem tri izraza od: A, B, C i D, imaju pojedinačnu, a time i ukupnu vrijednost, jednaku 1 (npr., izraz $B=Z_{(v+1)} \oplus Z_1=0$ ili $A=Y_1 \oplus Z_1 = 0$). Ako su 2 ili više bitova-indikatora jednaki 1 (neovisno o Z_1), onda, prema definiciji većinske logike, konačan izlaz jednak je 0 (jer su najviše 2 od A, B, C i D jednadžbi jednaki 1, što prema definiciji većinske logike nije dovoljno da bi izlaz iz sklopa većinske logike G_4, Z_1 bio jednak 1). Ako još jedan (bilo koji) bît ima vrijednost 1 u **svakoj** od jednadžbi: A, B, C i D, zbrojit će se (XOR) zajedno sa Z_1 pa će ukupan izraz za **svaku jednadžbu** biti jednak 0. Bit-indikator Z_1 ostaje jedini izraz što se pojavljuje u svakoj od četiriju jednadžbi (i ima vrijednost 1).

U nastavku, objašnjavamo pogreške praska duljine $t = 2v$, počevši sa Z_1, osiguravajući da barem 3 od indikatora A, B, C i D imaju vrijednost 1 (većinska logika). Slika 6.5 pokazuje raspored jedanaest bitova indikatora pogrešaka u vremenu, što se preslikavaju u A, B, C i D.



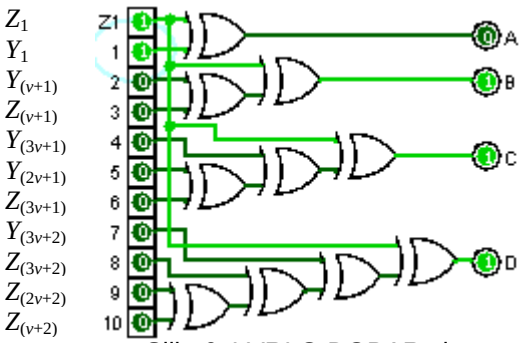
$$Y_{3v+2}Z_{3v+2}Y_{3v+1}Z_{3v+1}\underbrace{\cdots}_{2v-3}Z_{2v+2}Y_{2v+1}\underbrace{\cdots}_{2v-2}Z_{v+2}Y_{v+1}Z_{v+1}\underbrace{\cdots}_{2v-2}Y_1Z_1$$

$D = S_{(3v+2)} \oplus Z_{(3v+2)} \oplus Z_{(2v+2)} \oplus Z_{(v+2)} \oplus Z_1$ $C = S_{(3v+1)} \oplus S_{(2v+1)} \oplus Z_{(3v+1)} \oplus Z_1$ $B = S_{(v+1)} \oplus Z_{(v+1)} \oplus Z_1$ $A = S_1 \oplus Z_1$

Slika 6.5 Razmještaj bitova naznake pogreške u A, B, C i D

Slika pokazuje koji bitovi se pojavljuju jedan za drugim i veličinu praznina između bitova koji nisu jedan iza drugoga. Pretpostavimo da se u primljenome skupu bitova dogodila praskovita pogreška duljine $t = 2v$, počevši od prvoga primljenoga bita i_1. Nadalje pretpostavimo da iza ove praskovite pogreške slijedi područje oporavka što se sastoji od najmanje $4v+4$ bita. Ako praskovita pogreška počinje s i_1, to znači da je $Z_1 = 1$. Promatrajući sliku 6.5, vidimo da od jedanaest bitova indikatora pogrešaka, jedini bit što pripada prasku osim Z_1, je Y_1.

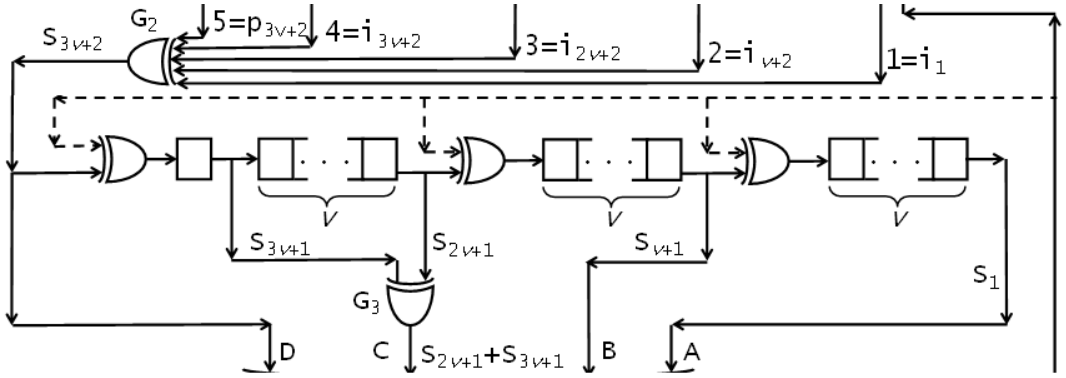
Ostatak bitova upada u području oporavka te su prema definiciji jednaki 0. Kako bi ti bitovi trebali biti 0, dovoljna je veličina područja oporavka $4v+4$. Učinak praskovite pogreške na vrijednosti A, B, C, D sada je jasan. Ako je samo $Z_1 = 1$, tada su A = B = C = D = 1. Jedino u slučaju kada je $Z_1 = Y_1 = 1$, onda su A = 0, a B = C = D = 1.



Slika6.4 VRLO DOBAR.circ

Zaključak 6.1 Ako se u primljenim bitovima dogodi prasak pogrešaka duljine $t = 2v$ na samome početku, a slijedi ga područje oporavka veličine $4v+4$, onda je izlaz G_4 jednak 1. Ako informacijski bit i_1 nije pogrešan onda je izlaz G_4 jednak 0. Drugim riječima *izlaz* iz G_4 je Z_1.

Kao što se na slici 6.4 vidi, izlaz iz G_4 vodi se na XOR vrata i zbraja se s i_1, a prije konačnoga stvaranja izlaza. Budući da je ovo izlaz Z_1, slijedi da je završni izlaz iz dekodera, izveden iz G_1, jednak $i_1+Z_1 = I_1$. Izvorno prenesen i oštećen informacijski bit i_1 tako se oporavlja. U prethodnoj raspravi, i_1 je primljena inačica prvoga *prenesenoga* bita. Cijela rasprava, međutim, još uvijek vrijedi ako se bit razmatra kao prvi *pogrešan* informacijski bit što se primio bilo gdje unutar niza bitova primljenih preko kanala ometanoga šumom.

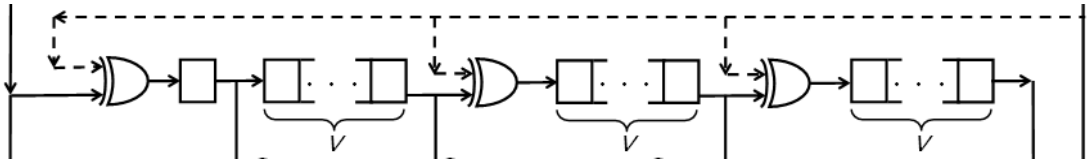


To je zbog činjenice da tako dugo dok nema pogreške, sadržaj donjega registra (sadrži S_j) su "sve 0" i sva promatranja u prijašnjem obrazloženju ostaju nepromijenjena. Prepušta se studentu, da kao vježbu, iznađe što se događa, ako prvi pogrešno primljen bīt izvorno prenosi parnost.[8] Upravo smo pokazali kako se ispravlja prvi bīt izvan pogreške praska duljine $t = 2v$. Nismo pokazali zašto se cijeli prasak ispravlja. Objašnjenje je jednostavno. Jednadžbe (a), (b), (c), (d), (e) i (f) su još uvijek na snazi, ako dodamo 1 svim indeksima.

Zaključak iza činjenice da je Z_1 izlaz G_1 (za vrijednosti prikazane na slici 6.4), vrijedi za Z_2, Z_3, …, Z_v. Stoga prvih v izlaza iz G_1, predstavlja v izvornih informacijskih bitova. Budući da između bilo koja dva informacijska bita postoji paritetan bīt, ispravak v uzastopnih informacijskih bitova jednak je ispravku praskovite pogreške duljine $2v$ što se javlja u prenesenim bitovima. Ovakvo razmišljanje također objašnjava potrebu posjedovanja područja oporavka dužine $6v+2$, jer dodavanje $v–1$ svakome indeksu na slici 6.5 (dovodi da prvi bīt ima indeks v), posljednji bit je $Y_{(4v+1)}$. Posljednji bīt ispravljenoga praska pogrešaka je Y_v. Postoji udaljenost od $6v+2$ prenesenih bitova između Y_v i $Y_{(4v+1)}$ i svi ti bitovi unutar toga raspona moraju biti bez pogrešaka, kako bi shema ispravno radila.

Također smo prethodno objasnili zašto svi bitovi između $Z_{(v+1)}$ i $Y_{(3v+2)}$ moraju biti bez pogrešaka, kako bi G_4 proizveo Z_1. Sada smo dodali $v–1$ svim tim indeksima. Prethodan opis objašnjava kako se ispravlja praskovita pogreška duljine $2v$, a temelji se na postojanju područja oporavka čija dužina iznosi *barem* $6v+2$. Riječi "*barem*" temelji se na činjenici da će kraće područje oporavka, onesposobiti shemu. Pitamo se, je li veličina područja oporavka od $6v+2$ *dovoljna* za dodatan ispravan rad sklopa. Drugim riječima, ako imamo još jedan pogrešan bīt neposredno nakon $6v+2$ bitova bez pogrešaka, možemo li ga ispraviti jednostavnim posmikom u sklop na slici 6.4? Ovo napraviti kao laboratorijsku vježbu i seminarske radove!

Kako bi ispravili i_j trebamo donji registar (što sadrži S_j') vratiti u početno stanje nakon unosa. Ako je ovaj uvjet zadovoljen, onda se sve praktične pojave mogu obraditi na isti način kao što se prije uradilo. To omogućuju isprekidane crte i XOR vrata duž donjega registra, što se prikazalo na slici 6.4.



Svrha ovoga dijela sklopa je *ovaj* registar dovesti u početno stanje, ako nakon praska pogrešaka slijedi $6v+2$ bita bez pogrešaka. Resetiranje se objašnjava promatranjem jednadžbi (a), (c) i (d).

Svaki $S_{(3v+2)}$, $S_{(2v+1)}$ i $S_{(v+1)}$, pohranjen u donjemu registru, sadrži Z_1, kao jedan od svojih pribrojnika. Imajte na umu da se Z_1 puni preko isprekidane crte i nakon toga zbraja XOR sa $S_{(3v+2)}$, $S_{(2v+1)}$ i $S_{(v+1)}$. Pribrojnici Z_1 zatim se krate u jednadžbama (a), (c) i (d). Po istome principu ostali pribrojnici u tim jednadžbama i sami se krate, nakon što ih generira G_4. Registru se onda jamči da će se pravovremeno vratiti u početno stanje, što omogućuje daljnjih ispravak praskovitih pogrešaka.

[8] Vježba!