

UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA U
NOVOM SADU

**Dizajn i verifikacija MIPS aritmetičko logičke jedinice
otporne na otkaz**

projekat iz predmeta
SISTEMI OTPORNI NA OTKAZ

Profesor:
Rastislav Struharik

Student:
Tomislav Tumbas

Novi Sad 2018

Sadržaj

<u>1 Zadatak.....</u>	<u>3</u>
<u>2 Uvod.....</u>	<u>4</u>
2.1 Tolerancija kvarova.....	4
2.2 Tehnike hardverske tolerancije na kvarove.....	5
<u>3 "Triplex-Duplex" tehnika hibridne redundanse.....</u>	<u>6</u>
3.1 Glasač.....	8
<u>4 Implementacija.....</u>	<u>13</u>
4.1 Aritmetičko logičke jedinice.....	13
4.2 Glasač.....	14
<u>5 Verifikacija.....</u>	<u>16</u>
5.1 Verifikacija na nivou aritmetičko logičke jedinice.....	16
<u>6 Zaključak.....</u>	<u>18</u>

1 Zadatak

Postojeću aritmetičko logičku jedinicu mikroprocesora MIPS r2000 redizajnirati kao aritmetičko logičku jedinicu koja je otporna na otkaze koristeći toleranciju na hardverske greške pristupom *Triplex-Duplex*.

2 Uvod

Sistemi otporni na otkaz su sistemi koji imaju mogućnost da nastavu da izvršavaju funkciju za koju su namenjeni bez obzira na otkaz. Oni su neophodni jer skoro nemoguće napraviti idealan sistem.

Fundamentalni problem je to što se kompleksnost sistema povećava, dok se pouzdanost drastično pogoršava, osim ako se neke kompenzatorske mere ne uvedu (preduzmu).

Mere koje se mogu uvesti su:

1. tolerancija na kvarove,
2. prevencija kravora,
3. uklanjanje kvarova i
4. predviđanje kvarova.

Tolerancija kvarova je sposobnost sistema da radi ispravno u prisustvu greške. Prevencija kvarova cilja da spreči pojavu ili početak greške (kvara). Uklanjanje kvarova cilja da smanji broj kvarova koji su prisutni u sistemu. Predviđanje kvarova je metoda koja predviđa koliko kvarova ima u sistemu, moguće buduće pojave kvara i uticaj kvara na sistem.

U ovom radu korišćićemo tehniku tolerancije na kvarove.

2.1 Tolerancija kvarova

Tolerancija kvarova ima za cilj razvoj sistema, koji može da funkcioniše ispravno i u prisustvu kvara. Ovo je moguće korišćenjem neke vrste redundantnosti, tj. neka funkcionalna sposobnost koja ne bi bila potrena u sistemu bez mogućnosti kvara. Redundantnost omogućava da se kvar zamaskira, ili da se detektuje i nakon toga da se lokalizuje, ograniči njegov efekat i oporavi sistem.

2.2 Tehnike hardverske tolerancije na kvarove

Hardverska tolerancija na kvarove je dostignuta kroz hardversku redundancu, pružanjem dve ili više fizičke instance hardverske komponente. Dizajneri su u ranijim računarskim sistemima replicirali komponente na nivou gejta i na nivou flipflop-a i koristili upoređivanje ili glasanje za detekciju ili ispravljanje kvarova, a kasnije, sa poboljšavanjem hardvera, redundansa se koristila na veće komponente kao što su memorije, aritmetičke jedinice itd.

Postoje 3 osnovne forme hardverske redundance:

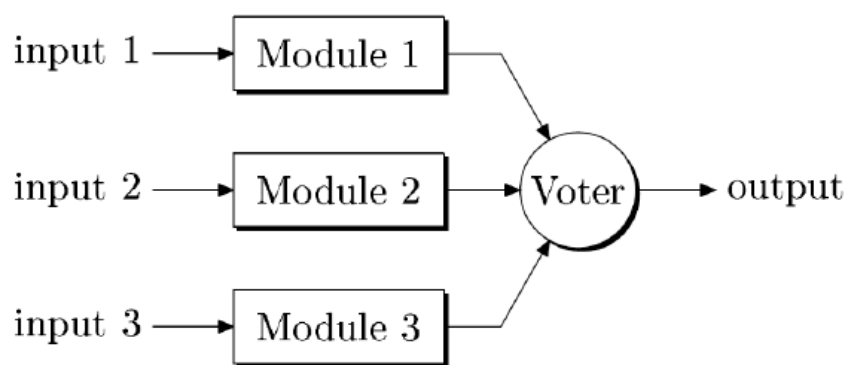
1. Pasivna – Postignuta je tolerancija kvara maskiranjem kvara koji se desio bez potrebe intervenisanja na tom delu sistema ili operatora.
2. Aktivna – Ztražuje da se kvar detektuje pre nego što se može tolerisati. Nakon detekcije kvara, akcije lokalizovanja, obuzdavanja i oporavka su izvođene da bi se otklonila pokvarena komponenta iz sistema.
3. Hibridna – Kombinuje aktivni i pasivni pristup. Maskiranje kvara se koristi za prevenciju generisanja pogrešnih rezultata. Detekcija kvara, lokaizacija i oporavak su korišćene da zamene pokvarenu komponentu sa rezervnom komponentom.

U ovom radu koristi se “Triplex-Duplex“ tehnika hibridne redundanse.

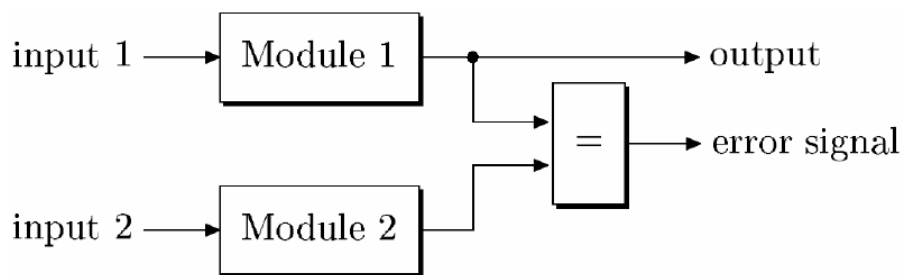
3 "Triplex-Duplex" tehnika hibridne redundanse

Triplex-Duplex redundansa (slika 3.3) kombinuje trostruku modularnu redundansu (slika 3.1) i redundansu sa poređenjem (slika 3.2).

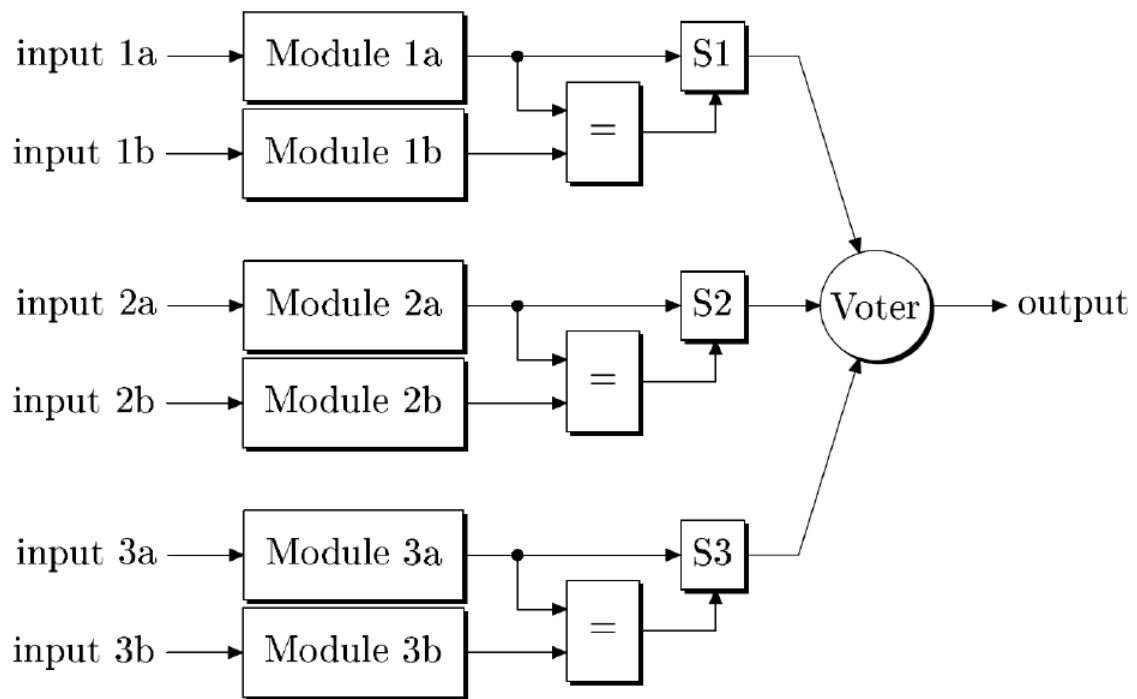
Modul se replicira šest puta i prave se tri grupe od po dva modula. Za svaku grupu od dva modula postoji komparator koji poredi izlaze iz modula. Ako se rezultati slažu, izlaz iz tog para učestvuje u glasanju. U slučaju ne slaganja rezultata, par modula se proglašava neispravnim i uklanja se iz sistema glasanja.



Slika 3.1: Trostruka modularna redundansa



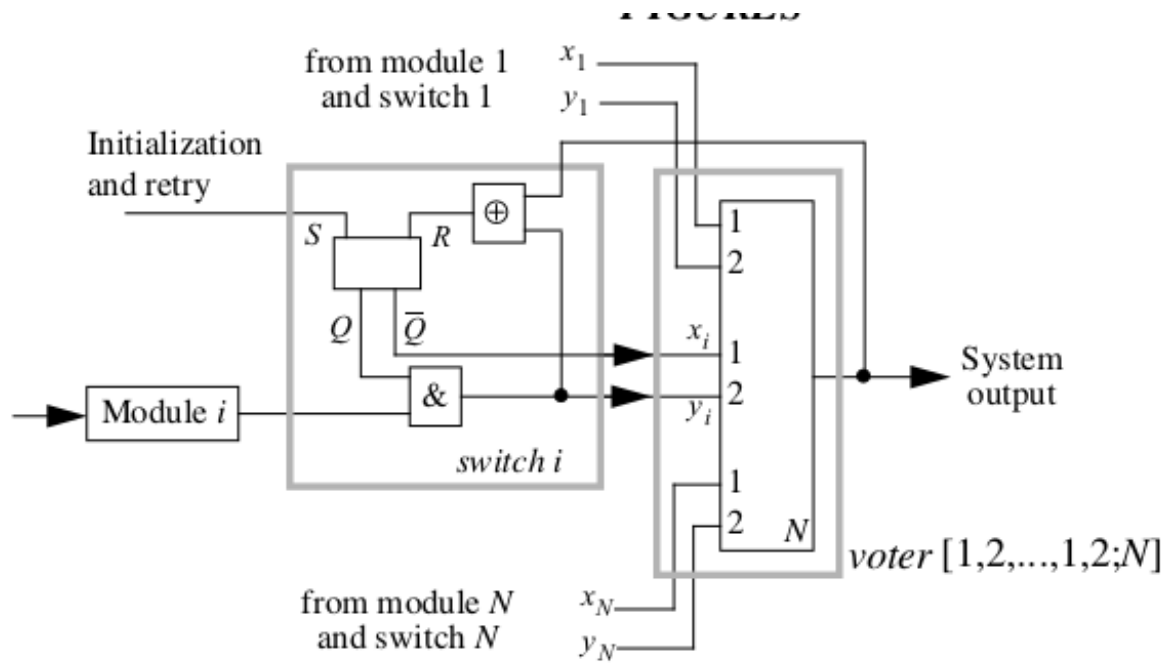
Slika 3.2: Redudansa sa poređenjem



Slika 3.3: "Triplex-Duplex" redundansa

3.1 Glasač

Glasač ima ulogu da odluči koji je tačan rezultat i prosledi ga na izlaz modula. U slučaju da je neki par modula prijavio grešku, glasač će zanemariti ulaz iz te grupe modula.



Slika 3.4: Self-Purging glasač sa logikom resetovanja modula iz tehnike N modularne redudanse sa rezervom

Ako su dve grupe modula neispravne kompletan sistem više nije u stanju da radi ispravno.

Tablica istinitosti glasača:

	A	B	C	D	E	F	OUT		A	B	C	D	E	F	OUT
0.	0	0	0	0	0	0	0	32.	1	0	0	0	0	0	0
1.	0	0	0	0	0	1	0	33.	1	0	0	0	0	1	x
2.	0	0	0	0	1	0	0	34.	1	0	0	0	1	0	1
3.	0	0	0	0	1	1	0	35.	1	0	0	0	1	1	x
4.	0	0	0	1	0	0	0	36.	1	0	0	1	0	0	x
5.	0	0	0	1	0	1	0	37.	1	0	0	1	0	1	1
6.	0	0	0	1	1	0	x	38.	1	0	0	1	1	0	1
7.	0	0	0	1	1	1	0	39.	1	0	0	1	1	1	1
8.	0	0	1	0	0	0	0	40.	1	0	1	0	0	0	1
9.	0	0	1	0	0	1	x	41.	1	0	1	0	0	1	1
10.	0	0	1	0	1	0	1	42.	1	0	1	0	1	0	1
11.	0	0	1	0	1	1	x	43.	1	0	1	0	1	1	1
12.	0	0	1	1	0	0	0	44.	1	0	1	1	0	0	x
13.	0	0	1	1	0	1	0	45.	1	0	1	1	0	1	1
14.	0	0	1	1	1	0	x	46.	1	0	1	1	1	0	1
15.	0	0	1	1	1	1	0	47.	1	0	1	1	1	1	1
16.	0	1	0	0	0	0	0	48.	1	1	0	0	0	0	0
17.	0	1	0	0	0	1	0	49.	1	1	0	0	0	1	0
18.	0	1	0	0	1	0	x	50.	1	1	0	0	1	0	x
19.	0	1	0	0	1	1	0	51.	1	1	0	0	1	1	0
20.	0	1	0	1	0	0	0	52.	1	1	0	1	0	0	0
21.	0	1	0	1	0	1	x	53.	1	1	0	1	0	1	x
22.	0	1	0	1	1	0	1	54.	1	1	0	1	1	0	1
23.	0	1	0	1	1	1	x	55.	1	1	0	1	1	1	x
24.	0	1	1	0	0	0	x	56.	1	1	1	0	0	0	x
25.	0	1	1	0	0	1	1	57.	1	1	1	0	0	1	1
26.	0	1	1	0	1	0	1	58.	1	1	1	0	1	0	1
27.	0	1	1	0	1	1	1	59.	1	1	1	0	1	1	1
28.	0	1	1	1	0	0	0	60.	1	1	1	1	0	0	0
29.	0	1	1	1	0	1	x	61.	1	1	1	1	0	1	x
30.	0	1	1	1	1	0	1	62.	1	1	1	1	1	0	1
31.	0	1	1	1	1	1	x	63.	1	1	1	1	1	1	x

gde su :

- A – ulaz podataka u glasač iz prvog para modula.
- B – ulaz iz komparatora koji proverava da li su moduli u prvoj grupi dali isti rezultat.

Logička jedinica označava pojavu neslaganja rezultata. Logička nula označava slaganje rezultata.

- C – ulaz podataka u glasač iz drugog para modula.
- B – ulaz iz komparatora koji proverava da li su moduli u drugoj grupi dali isti rezultat.

Logička jedinica označava pojavu neslaganja rezultata. Logička nula označava slaganje rezultata.

- C – ulaz podataka u glasač iz trećeg para modula.
- B – ulaz iz komparatora koji proverava da li su moduli u trećoj grupi dali isti rezultat.

Logička jedinica označava pojavu neslaganja rezultata. Logička nula označava slaganje rezultata.

U slučaju da:

1. dva para modula ne rade ispravno ili
2. jedna par modula ne radi ispravno, a druga dva para ne daju isti rezultat

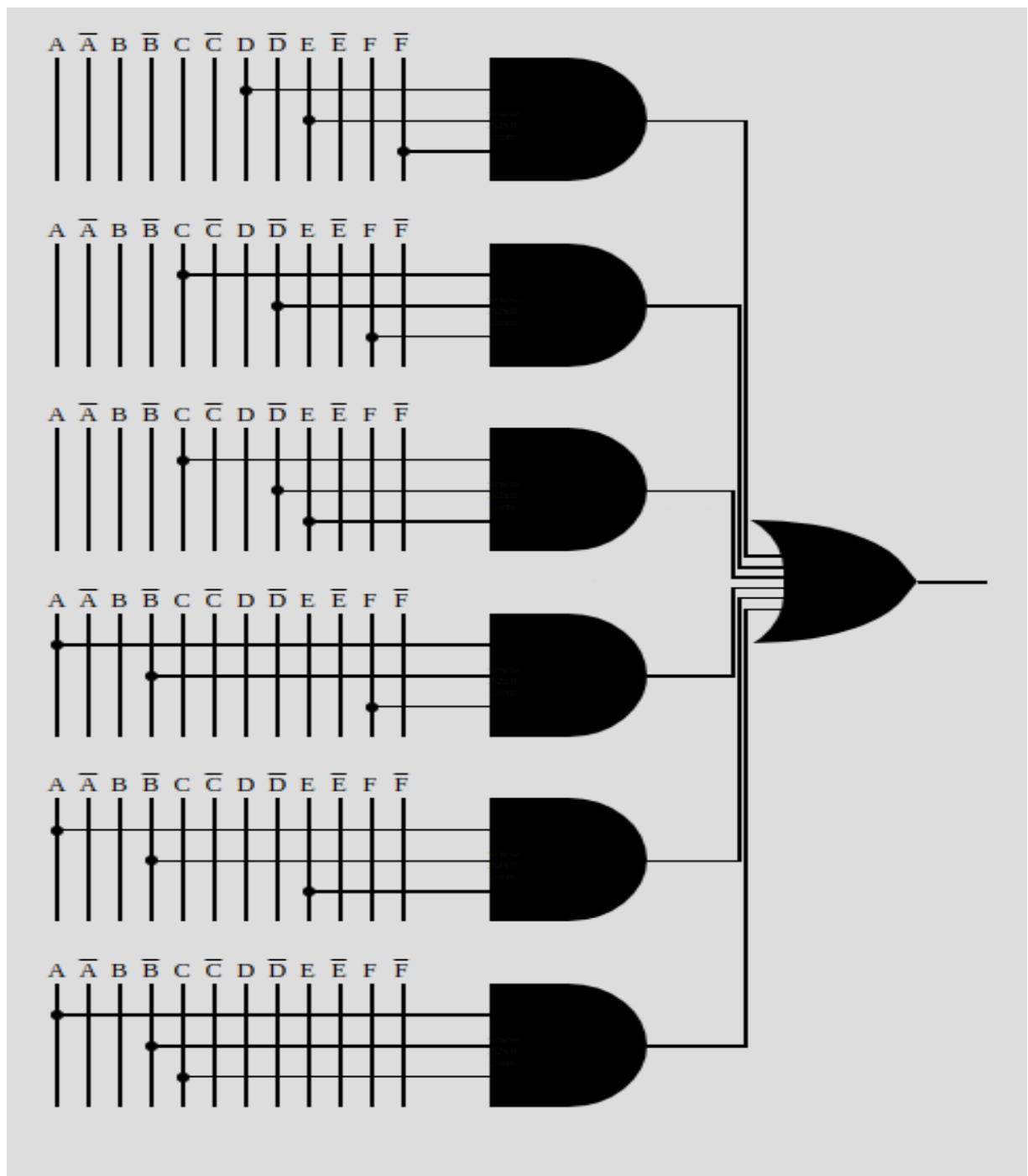
glasač ne može da odredi tačan izlaz. Za te kombinacije tablici istinitosti koristimo X koje ćemo kasnije moći iskoristiti u karnoovim mapama.

Karnoove mape:

	$\overline{D}\overline{E}\overline{D}$	$\overline{D}\overline{E}F$	$\overline{D}E\overline{E}$	$\overline{D}\overline{E}F$	$\overline{D}EF$	$D\overline{E}F$	DEF	$D\overline{E}F$
$\overline{A}\overline{B}\overline{C}$	0	0	0	0	0	0	0	x
$\overline{A}\overline{B}C$	0	x	x	1	0	0	0	x
$\overline{A}B\overline{C}$	X	1	1	1	0	X	X	1
$\overline{A}BC$	0	0	0	X	0	X	X	1
$A\overline{B}\overline{C}$	0	X	X	1	X	1	1	1
$A\overline{B}C$	1	1	1	1	X	1	1	1
$AB\overline{C}$	X	1	1	1	0	X	X	1
ABC	0	0	0	X	0	X	X	1

Izlazna funkcija:

$$y = D\overline{E}F' + C\overline{D}F + C\overline{D}\overline{E} + A\overline{B}F + A\overline{B}\overline{E} + A\overline{B}\overline{C}$$

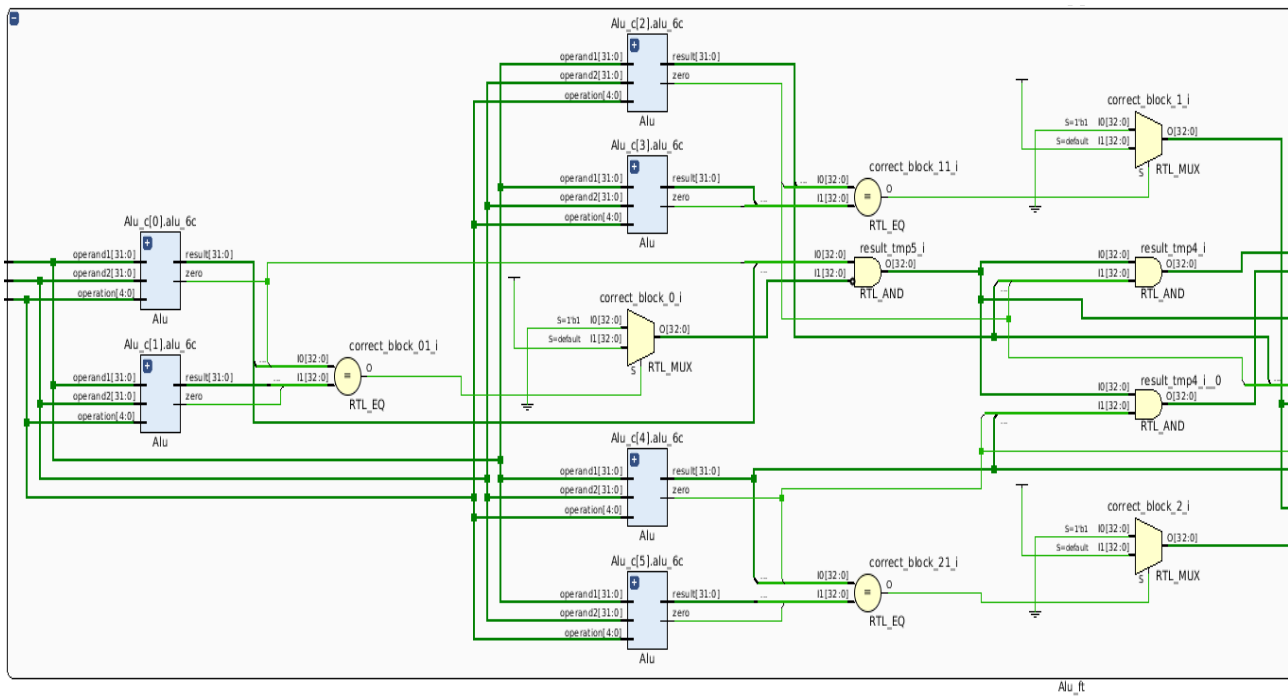


Slika 3.4: Dizajn glasača

4 Implementacija

4.1 Aritmetičko logičke jedinice

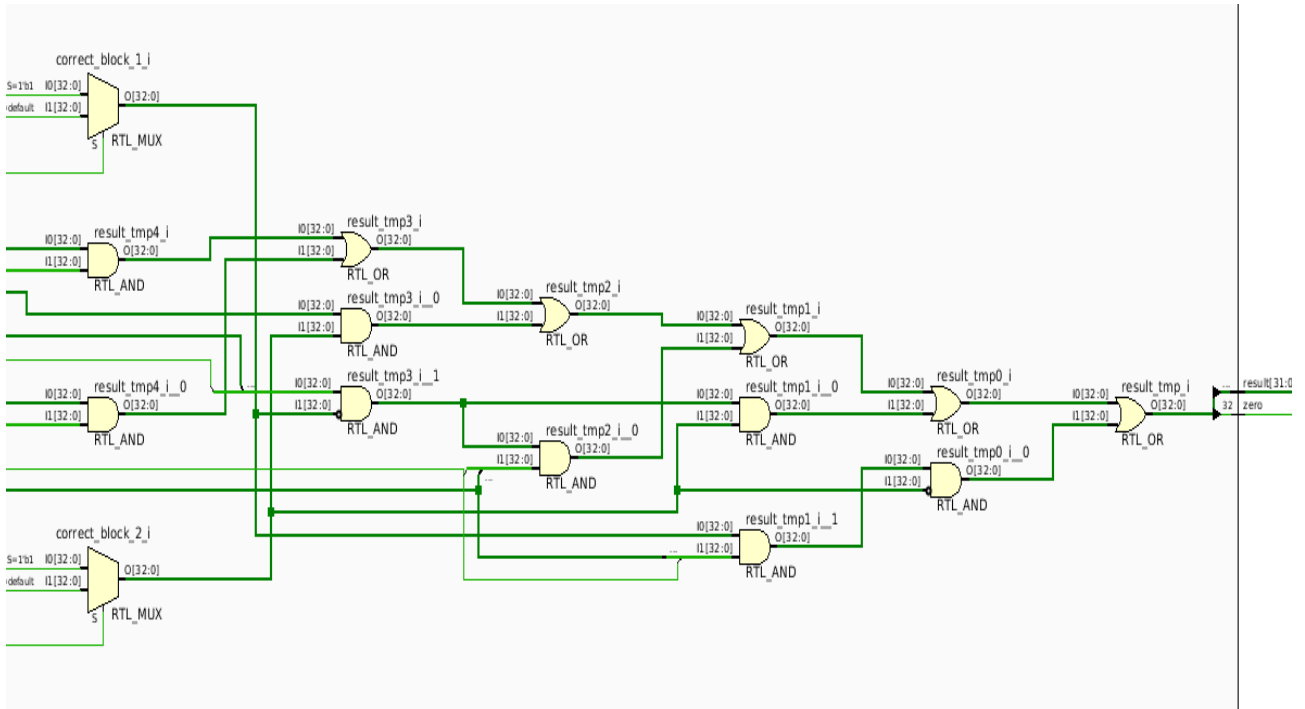
Aritmetičko logičke jedinice su replicirane 6 puta i grupisane u parove. Svakom paru aritmetičko logičkih jedinica pridružen je komparator, koji vrši proveru jednakosti izlaza iz modula. U slučaju poklapanja rezultata modul daje na izlaz vektor širine 32 bita logičke nule. U slučaju ne poklapanja rezultata na izlazu modul daje vektor širine 32 bita logičke jedinice.



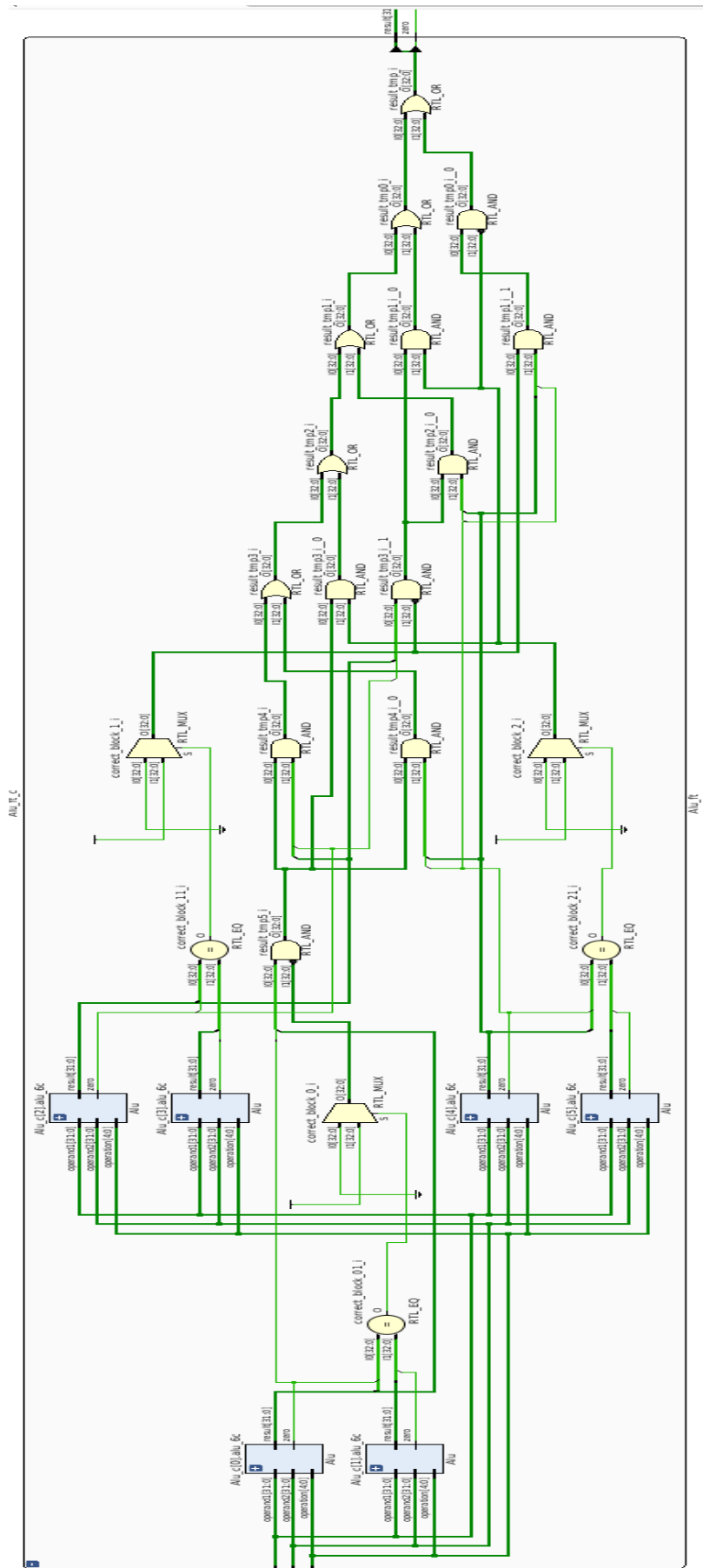
Slika 4.1: Aritmetičko logičke jedinice sa proverom jednakosti rezultata

4.2 Glasač

Glasač je implementiran kao kombinaciona mreža i ili logičkih kola.



Slika 4.2: Implementacija glasača



Slika 4.3: Aritmetičko logička jedinica otporna na otkaze

5 Verifikacija

Verifikacija ispravnosti rada aritmetičko logičke jedinice otporne na otkaze sastoji se iz dva dela:

1. standardnog verifikacionog okruženja, *test bench-a*, koji generiše pobudne signale i proverava tačnost izlaza.
2. modula koji simulira kvarove i forsira određenje signale na unapred zadate vrednosti.

Modul za simuliranje kvarova u svakom ciklusu forsira izlaze jedne od aritmetičko logičkih jedinica ili dve aritmetičko logičke jedinice iz istog para na netačne vrednosti. Na taj način simuliramo neispravnost aritmetičko logičke jedinice i proveravamo da li će kvar biti okrivljen i ispravljen.

Forsiranje signala na netačne vrednosti rađeno je pomoću programskog paketa *Modelsim*. Korištena je biblioteka *Modelsim.util.all* i komande *signal_force*.

5.1 Verifikacija na nivou aritmetičko logičke jedinice

Na slici 5.1 vidi se kako sve aritmetičko logičke jedinice sem jedne imaju isti rezultat. Aritmetičko logička jedinica koja nema isti rezultat je pod uticajem modula za forsiranje signala na izlazu i daje netačan izlaz.

Izlaz iz *Triplex-Duplex* bloka vidimo da se slaže većinom aritmetičko logičkih jedinica i daje tačan izlaz. Što nam potvrđuje i blok za proveru tačnosti izlaza unutar verifikacionog okruženja. Iz toga vidimo da blok za toleranciju na hardverske greške pristupom *Triplex-Duplex* radi ispravno.

Messages	Decimal	Hex	Binary
/alu_ft_tb/alu_ft_c/operand1	192061479	821077879	1452577857
/alu_ft_tb/alu_ft_c/operand2	1870418611	346101685	1038305055
/alu_ft_tb/alu_ft_c/operation	alu_xor	alu_add	alu_sub
/alu_ft_tb/alu_ft_c/result	1678701204	1167179564	414272802
/alu_ft_tb/alu_ft_c/zero	0		
/alu_ft_tb/alu_ft_c/alu_out_6	(001100100000011)	(00100010110010001101111100101100)	(00011001100110011001100110011001)
(5)	1678701204	1167179564	414272802
(4)	1678701204	858993459	3880694494
(3)	-2516266092	1167179564	414272802
(2)	1678701204	1167179564	414272802
(1)	1678701204	1167179564	414272802
(0)	1678701204	1167179564	414272802
/alu_ft_tb/alu_ft_c/correct...	0000000000000000		
/alu_ft_tb/alu_ft_c/correct...	1111111111111111		
/alu_ft_tb/alu_ft_c/correct...	0000000000000000		

Slika 5.1: Signali u aritmetičko logičkoj jedinici tokom simulacije

6 Zaključak

U radu je detaljno proučena i analizirana tolerancija na hardverske greške pristupom *Triplex-Duplex*. Data tolerancija na hardverske greške implementirana je u okviru aritmetičko logičke jedinice mikroprocesora MIPS r2000. Za korist verifikacije dizajna, razvijena su verifikaciona okruženja i uradjena je simulacija.

Pomoću simulacije pokazano je pomoću ove tehnike moguće detektovati greške u sistemu i ispraviti ih.

Dalji rad na toleranciji za greške mikroprocesora MIPS r2000 mogao bi biti dodavanje tolerancije na hardverske greške na registarskim bankama kodovima parnosti.