



**UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA U
NOVOM SADU**



Implementacija Self-purging redundancy tehnike na IIR filtru

Projekat iz predmeta Digitalni sistemi otporni na greške

Mentor:
Nikola Kovačević

Student:
Jelena Marinković
E1/46-2020

Novi Sad, jul 2022.

Sadržaj

1.	Uvod	3
1.1	<i>Self-purging redundancy</i>	3
1.2	IIR filter	4
2.	VHDL model IIR filter	5
3.	Verifikacija IIR filter	7
3.1	MATLAB simulacija	7
3.2	Vivado simulacija bez redundanse	8
3.3	Vivado simulacija sa redundansom	9
4.	Analiza performansi sistema	12
5.	Zaključak	13
6.	Literatura	14
	Dodatak	15

1. Uvod

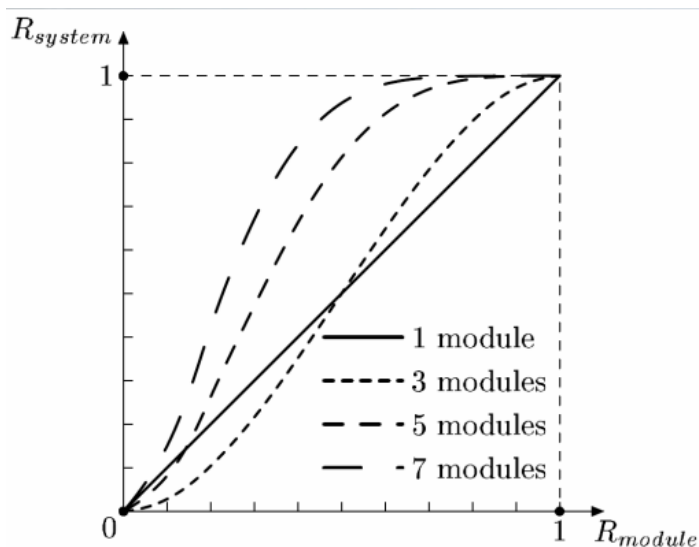
Da bi sistem bio tolerantan na greške neophodno je uvođenje redundantnih modula koji bi očuvali funkcionalnost celokupnog sistema u slučaju otkaza pojedinih komponenti. Uvođenje redundanse je efikasnije na nižim nivoima hijerarhije nego na nivou sistema, zbog velikog broja nesavršenih komponenti. Redundantni moduli nemaju ulogu kada kvar ne postoji, a značajno utiču na cenu projektovanog sistema, stoga je važno sagledati sve aspekte i odlučiti se za tehniku tolerancije koja ispunjava zahteve u pogledu pouzdanosti, a opet bespotrebno ne usložnjava sistem. Tehnike tolerancije se dele u tri velike grupe: pasivne, aktivne i hibridne redundanse. Kada se greška maskira to je pasivna redundansa, za razliku od aktivne gde je cilj detektovati grešku. Hibridna redundansa je kombinacija prethodne dve tehnike, greška se maskira, ali se i detektuje gde je do nje došlo, tako da neispravna komponenta može da se zameni redundantnom ili rekonfiguriše.

1.1 *Self-purging redundancy*

Self-purging redundancy je hibridna tehnika hardverske tolerancije na greške. Sadrži N identičnih modula, N prekidača (*eng. switch*) i glasač (*eng. voter*). Svi moduli istovremeno vrše proračune i preko prekidača prosleđuju dobijeni rezultat glasaču. Ukoliko se vrednost koju je modul prosledio poklopi sa izlazom glasača taj modul radi ispravno, u suprotnom je došlo do greške. Kada prekidač detektuje grešku isključuje modul tako što umesto njegovog izlaza ubuduće glasaču prosleđuje nulu. Na ovaj način moguće je maskirati N-2 greške. Moguće je detektovati i N-1 neispravan modul, ali u tom slučaju grešku više nije moguće maskirati jer glasač ne zna koja od dve pristigle vrednosti je tačna. Ako pretpostavimo da su prekidači i glasač idealne komponente, kao i da svi moduli (od 1 do N) imaju istu pouzdanosti R, pouzdanost sistema može da se iskaže preko formule:

$$R = 1 - ((1 - R)^N + NR(1 - R)^{N-1})$$

Na slici 1., vidi se pouzdanosti sistema R_{system} u zavisnosti od broja redundantnih modula N. Očekivano, veći broj redundantnih modula daje veću pouzdanost.



Slika 1. Pouzdanost sistem sa jednim, tri, pet i sedam redundantnih modula [1]

1.2 IIR filtar

IIR (eng. *Infinite impulse response*) filtar je tip digitalnog filtra koji zbog postojanja povratne sprege ima beskonačni impulsni odziv. Filtar ima za cilj da vrši diskriminaciju signala prema frekvenciji, signal učestanosti koja odgovara propusnom opsegu se propušta, dok se signal iz nepropusnog opsega slabi. Izlazni signal dobija vrednost po sledećoj formuli:

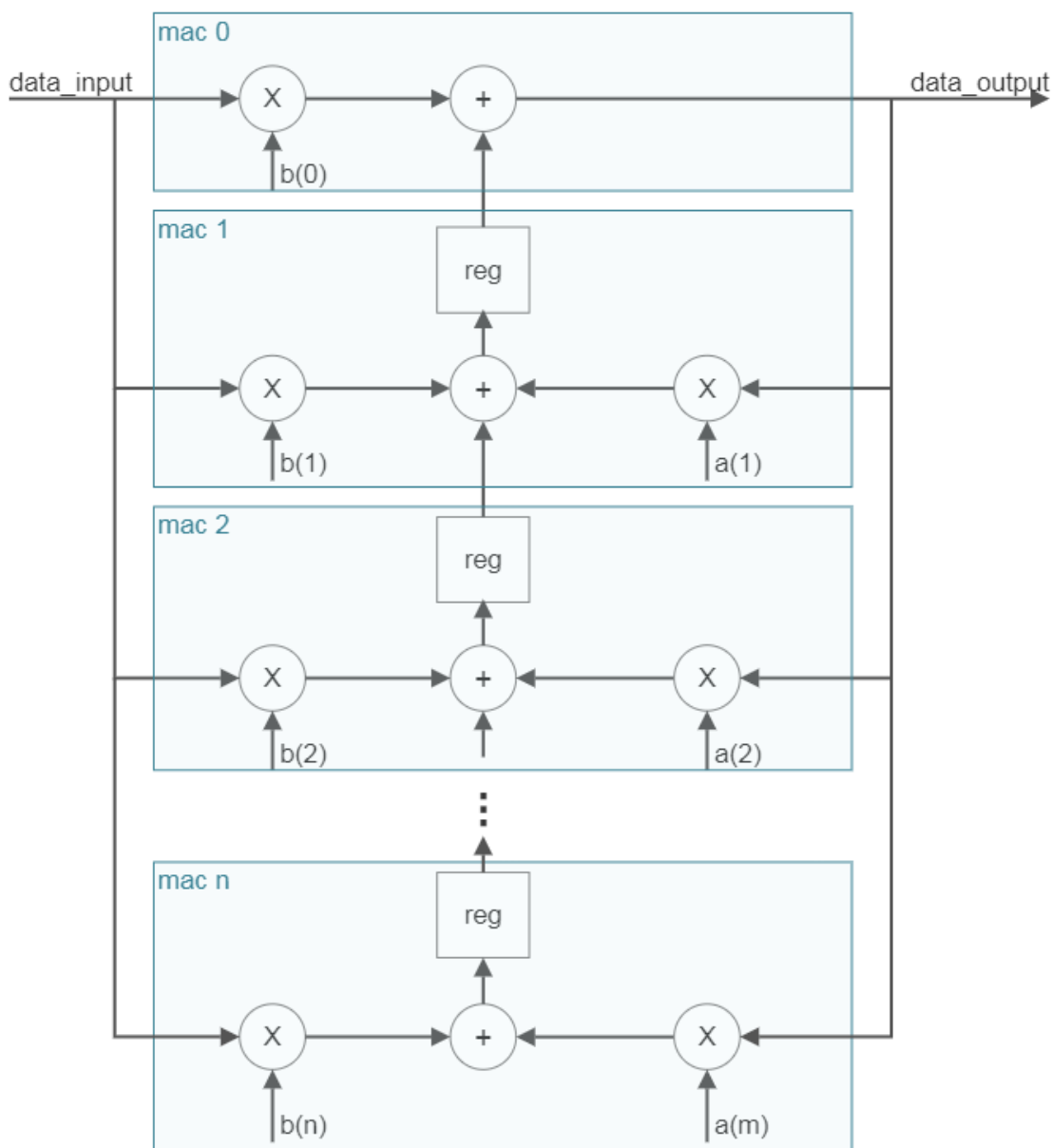
$$y(n) = \frac{1}{a_o} \left[\sum_{k=0}^M b_k u(n-k) + \sum_{k=1}^N (-a_k) y(n-k) \right]$$

Zbog postojanja povratne sprege, trenutna vrednost izlaza zavisi od prethodnih vrednosti i ulaznog signala $u(n)$ i izlaznog signala $y(n)$. Ukoliko se filtar realizuje iz dva podsistema, gde svaki direktno računa po jednu sumu iz prethodne jednačine reč je o direktnoj (formi) realizaciji I. Promenom redosleda suma, odnosno podsistema koji računaju sumu po ulaznom $u(n)$ odnosno izlaznom $y(n)$ signalu dobija se direktna forma II. Transponovanjem direktnih formi, dobijaju se transponovana direktna forma I i II koje imaju istu složenost kao i direktne forme, ali kraći najduži propagacioni put. U ovom projektu odabrana je realizacija IIR filtra preko transponovane direktne forme II jer njen najduži propagacioni put ima samo jedan množač (b_o) i jedan sabirač.

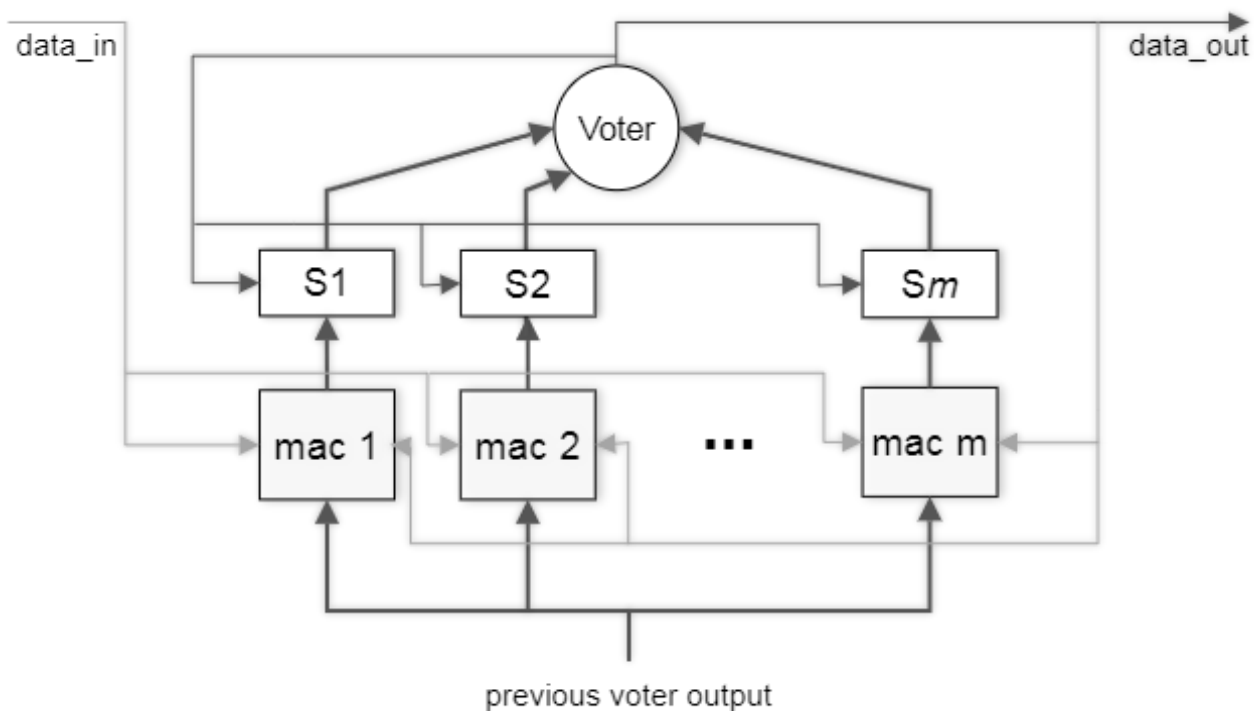
Ovaj rad se sastoji iz pet celina. U prvom delu, uvodu, dat je kratak podsetnik o načinu rada IIR filtra i konceptu redundanse sa akcentom na *Self-purging* tehniku. U drugom delu biće reči o implementaciji IIR filtra u VHDL-u, blokovima koji su korišćeni i strukturi sistema. Treći deo je posvećen verifikaciji samog modula i analizi rezultata dobijenih simulacijom. Četvrti deo bavi se analizom utrošenih resursa i performansama sistema. Poslednji, peti deo, sadrži kratak osvrt na dobijene rezultate.

2. VHDL model IIR filtra

IIR filter dobija se povezivanjem osnovnih gradivnih blokova, odnosno množača, sabirača i pomerača, u odgovarajućem redosledu. Blokovi koji se nalaze u istom redu filtra mogu da se grupišu u okviru MAC (*eng. Multiply Accumulate*) podmodula, tako da svaki red filtra zapravo čini jedna MAC jedinica. Na slici 2. prikazna je blok šema filtra n -tog reda. Prvi MAC modul, *mac_0*, sastoji se samo od jednog množača i sabirača, dok svaki sledeći modul od *mac_1* do *mac_n* čine po dva množača (jedan množi ulaz sa koeficijentom $b(x)$, drugi izlaz sa koeficijentom $a(x)$, $x \in [1, n]$), sabirača i registra, koji ima ulogu pomerača.

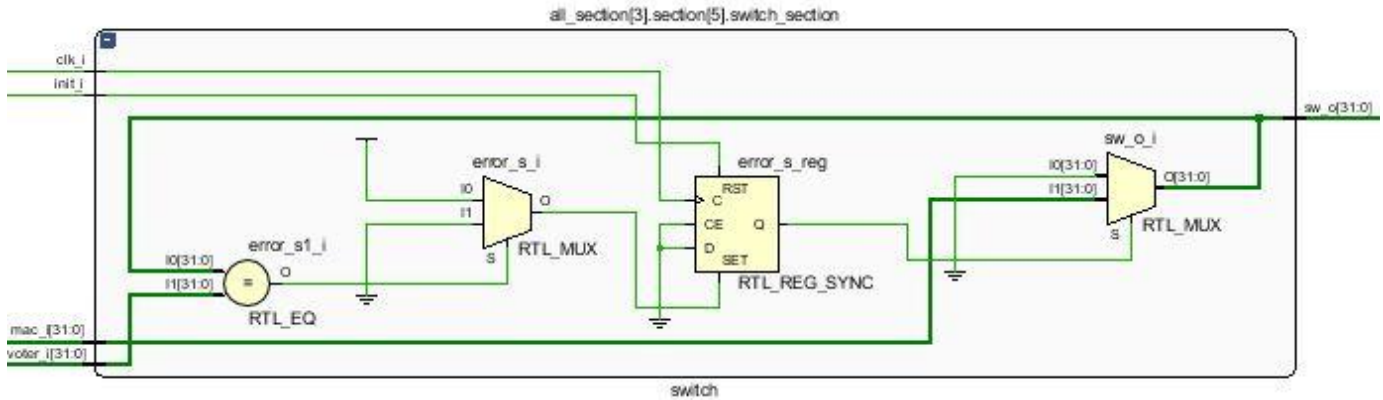


Slika 2. Struktura IIR filtra sa označenim MAC podmodulima

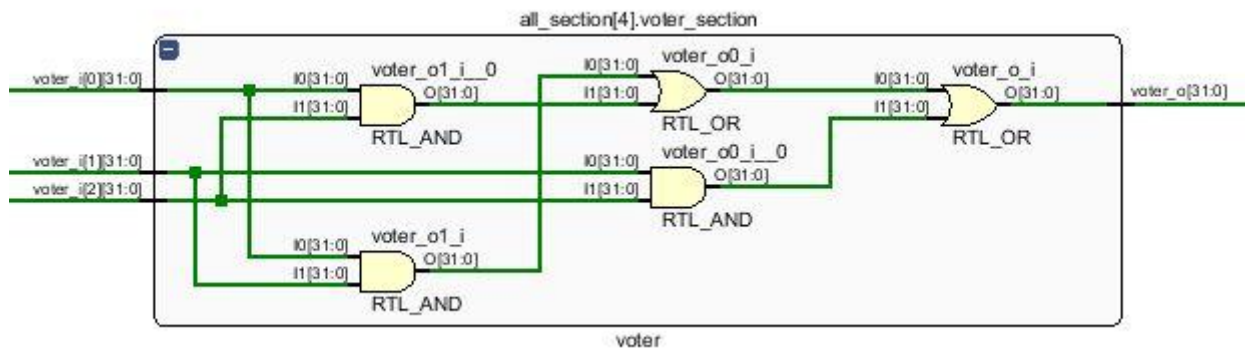


Slika 3. Uvođenje redundanse u jedan red filtra

Da bi dati sistem bio otporan na greške neophodno je uvesti određeni nivo redundanse. Pouzdanost pojedinačnih blokova jeste velika, ali je i broj komponenti u sistemu velik (tačan broj zavisi od reda filtra), stoga se verovatnoća da sve komponente ispravno rade značajno smanjuje. U slučaju otkaza samo jedne od njih rezultat se trajno narušava. Na slici 3. prikazan je koncept uvođenja redundanse na nivou MAC modula. Do sad je svaki red filtra činila jedna MAC jedinica, umesto toga sad imamo m MAC komponenti (mac_{1-m}), m prekidača (S_{1-m}) i glasač (*Voter*) u svakom redu. Na ulaz svakog MAC-a dovodi se izlaz sa prethodnog glasača, svaki podmodul nezavisno vrši proračune i prosleđuje vrednost prekidaču. Prekidač ima funkciju da u slučaju nepoklapanja vrednosti koju je glasač proglasio tačnom i vrednosti koju je izračunao MAC, isključi MAC komponentu i glasaču ubuduće prosleđuje nulu. Detaljan izgled prekidača prikazan je na slici 4. Ulaz mac_i predstavlja vrednost dobijenu od MAC jedinice, a $voter_i$ vrednost sa izlaz glasača koji se dovodi nazad na prekidač radi provere ispravnosti komponente. Ako je došlo do neslaganja interni signal *error_s* dobija vrednost "1" čime signalizira grešku, te se do sledećeg resetovanja prekidača glasaču prosleđuje vrednost "0" umesto vrednosti sa ulaza mac_i .



Slika 4. Realizacija prekidača, izgled Vivado schematic-a



Slika 5. Realizacija glasača, izgled Vivado schematic-a

Glasač ima onoliko ulaza koliko ima redundantnih modula, m . Zbog strukture glasača nula sa nespravnog modula ne utiče na odabir rešenja, stoga je moguće maskirati $m-2$ greške. U primeru glasača sa slike 5. postoje tri ulaza, pa je moguće neutralisati postojanje samo jedne greške.

3. Verifikacija IIR filtra

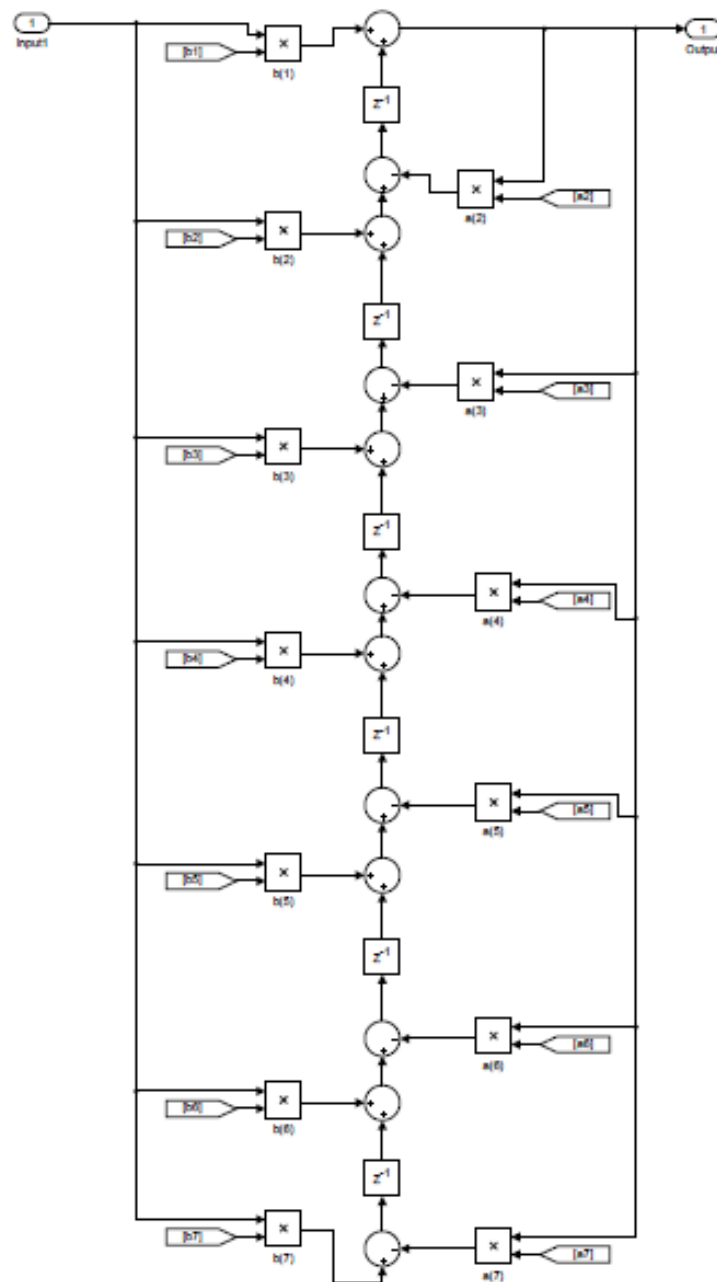
3.1 MATLAB simulacija

Da bi se izvršila provera korektnog rada dizajniranog filtra neophodno je projektovati neki IIR filter, odrediti red filtra, koeficijente a i b , ulazni i izlazni test vektor, a potom dobijene vrednosti uneti u VHDL model i vršiti poređenje dobijenog i očekivanog odziva. Referentni model dizajniran je u MATLAB alatu sa sledećim parametrima:

NF filter: $f_p = 5000 \text{ Hz}$, $f_s = 8000 \text{ Hz}$, $a_p = 1 \text{ dB}$, $a_s = 40 \text{ dB}$, $f_{sr} = 22050 \text{ Hz}$

pri čemu je korišćena Batervortova (eng. *Butterworth*) aproksimacija. Minimalni red filtra za date parametre je šest. Ulaz i izlaz filtra je 32-bitni, gde 5 brojeva čine celobrojni, a 27

razlomljeni deo. Na slici 6. prikazan je model referentnog IIR filtra sa navedenim osobinama dizajniran u SIMULINK-u.

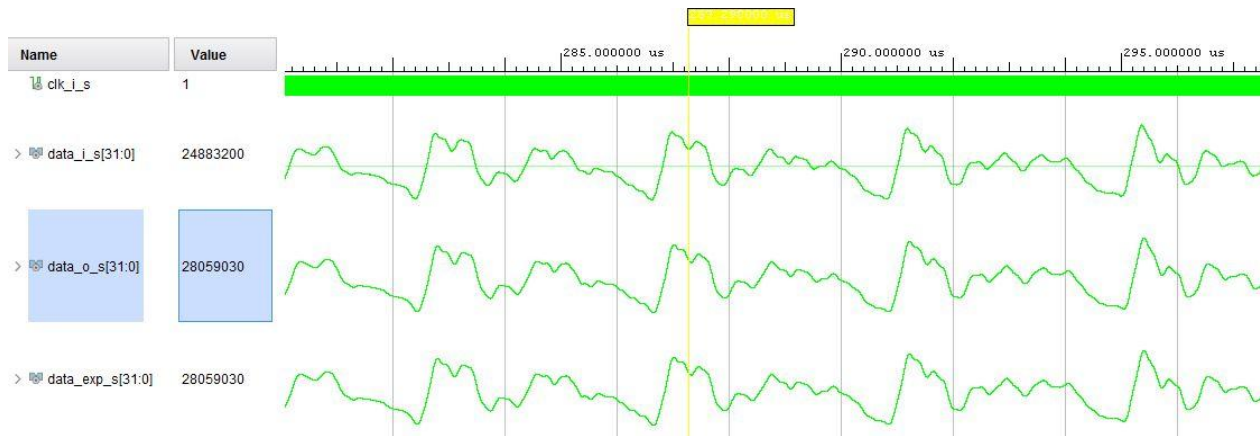


Slika 6. IIR filter, SIMULINK

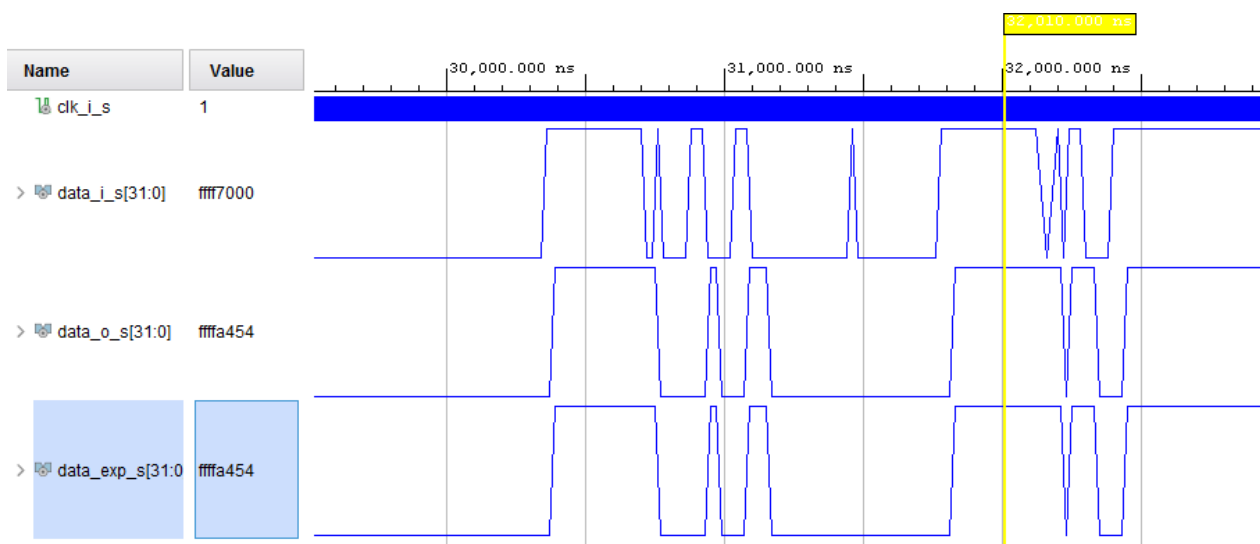
3.2 Vivado simulacija bez redundanse

Simulacija je obavljena u Vivado simulatoru, prvo je izvršena provera funkcionalnosti modelovanog IIR filtra bez redundantnih modula, a potom i sa 3, 5 i 7 redundantnih modula u svakom redu filtra. Na slici 7. nalaze se redom ulazni signal, izlazni signal dobijen simulacijom i

referentni izlazni signal iz MATLAB-a. Slika 8. Prikazuje iste signale samo u heksadecimalnom formatu radi lakšeg uočavanja razlike između ulaza i izlaza.



Slika 7. Ulazni, izlazni i referentni izlazni signal

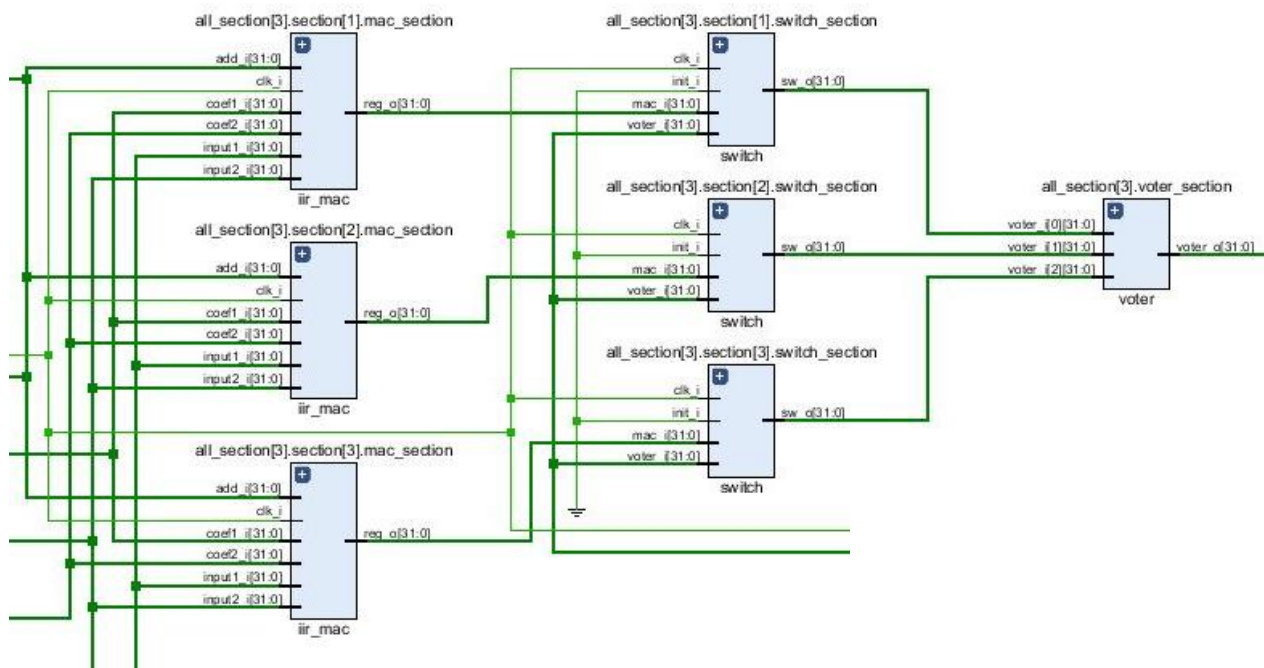


Slika 8. Ulazni, izlazni i referentni izlazni signal, hex format

U slučaju odstupanja između dobijene i očekivane vrednosti u okviru testbenč fajla detektuje se neslaganje i u tom trenutku šalje upozorenje u Tcl konzolu (*report "Result mismatch!" severity warning*).

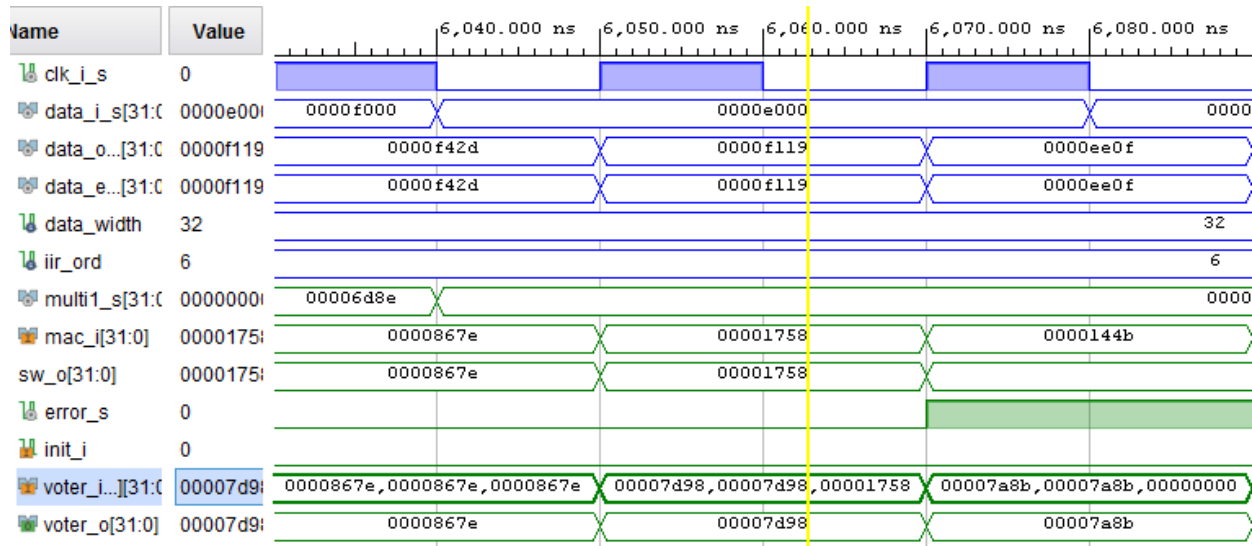
3.3 Vivado simulacija sa redundansom

Nakon što je ustanovljeno da modul daje očekivane rezultate, uvodi se redundansa. U dodatku na slici A nalazi se izgled filtra sa redundantnim modulima. U slučaju da je redundansa tri, jedan red filtra ima izgled prikazan na slici 9.



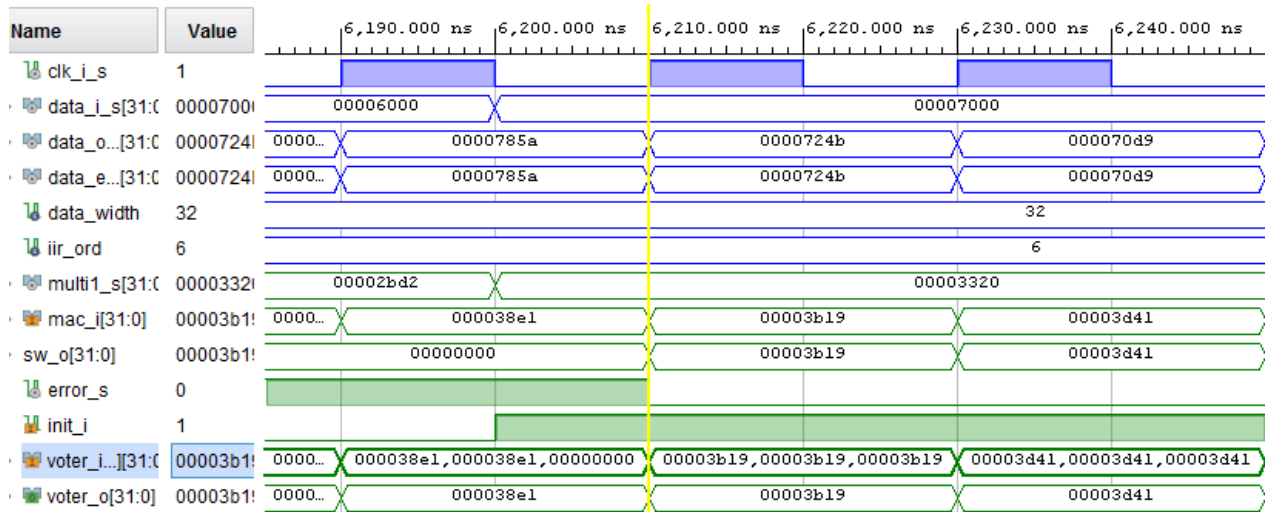
Slika 9. Jedan red filtra nakon uvođenja redundanse

Da bismo proverili da li se sistem sa redundantnim modulima očekivano ponaša, tj. da li prekidači i glasači obavljaju svoju funkciju, neophodno je simulirati grešku u MAC modulu i posmatrati da li će biti detektovana, odnosno maskirana. Najlakši način da se to uradi, jeste napisati .tcl skriptu koja će forsirati vrednosti na signalima od interesa u odabranim vremenskim trenucima. Ubacivani su SA kvarovi u nasumično odabrane MAC module, na signale sabirača, množača ili registara. Slike 10. i 11. demonstriraju ispravan rad sistema, na primeru sa tri redundantna modula. Na signalu *multi1_s* forsira se vrednost "0", što za posledicu ima da MAC modul prosleđuje neispravan rezultat glasaču. U trenutku 6,050 ns (označeno žutim pokazivačem) vidimo da glasač *voter_i* sadrži tri vrednosti, dve od ispravnih modula i jednu vrednost koja se razlikuje. Signal *voter_o* dobija vrednost za koju su se odlučili moduli bez greške, i taj rezultat prosleđuje narednim modulima, odnosno MAC jedinicama u sledećem redu filtra, te je greška uspešno maskirana. Prekidač MAC-a sa greškom detektuje razliku u rešenju modula i glasača, te aktivira signal *error_s* čime isključuje modul. Vidimo da su pri sledećem odabiru rešenja glasaču na raspolaganju samo dva rezultata, jer se sa modula koji je prethodno imao grešku prosleđuje nula. U slučaju da neki od preostala dva modula otkazu, glasač više ne bi mogao da utvrdi koji je modul otkazao te greška ne bi mogla da se maskira.

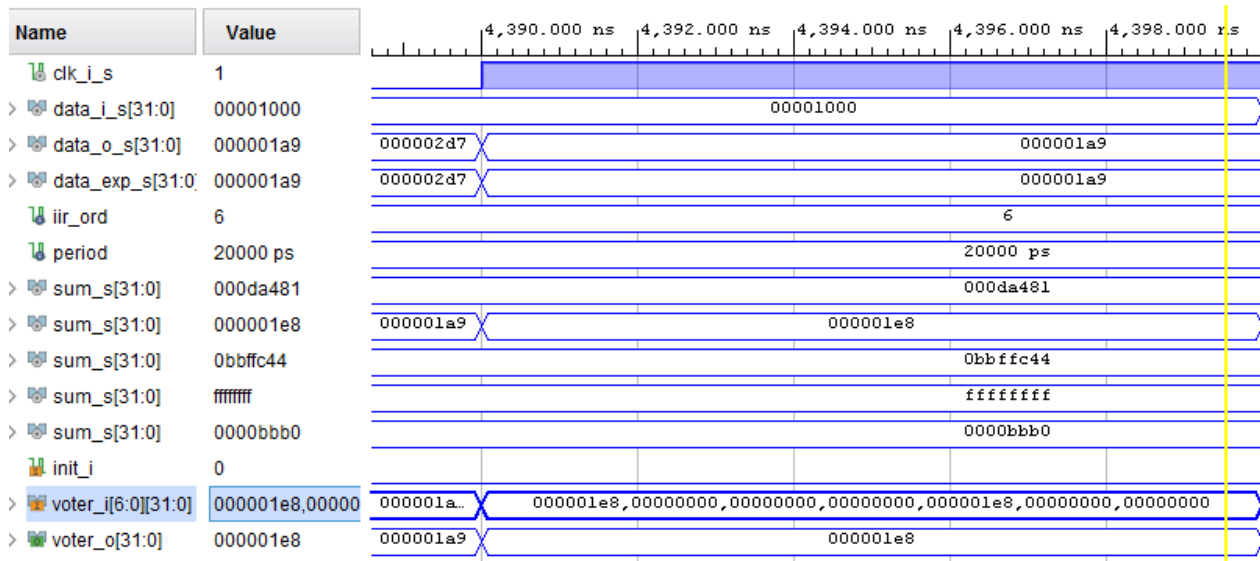


Slika 10. Pojava greške u modulu na množaču

Isključeni modul može da postane validan tek nakon resetovanja, kao što je prikazano na slici 11. *init_i*, signal za reset, se aktivira u trenutku 6,200 ns, što za posledicu ima da se signal za detekciju greške *error_s* vraća na "0", kao da se greška nije ni dogodila, pa prekidač ponovo dozvoljava prosleđivanje vredosti na glasač. Od trenutka 6,210 ns vidimo da je glasaču ponovo dostupna vrednost prethodno isključenog MAC modula.



Slika 11. Restartovanje isključenog modula



Slika 12. Sistem sa sedam redundantnih modula po redu filtra

U slučaju više redundantnih modula moguće je tolerisati veći broj grešaka. Na slici 12. demonstriran je slučaj kad pet od sedam modula otkazu. Vidimo da signal *voter_i* od samo dva modula prima vrednosti, kod su ostali moduli u kvaru, ali to ne utiče na rad glasača pa se greške uspešno maskiraju.

4. Analiza performansi sistema

U prethodnom poglavlju urađena je verifikacija modula i prikazan način na koji se tolerišu greške u sistemu. Videli smo da što se uvede veći broj redundantnih modula, veća je i otpornost na otkaze, međutim treba sagledati i koliko košta uvođenje redundanse sa stanovišta preformansi sistema i potrebnih resursa. U tabeli 1. dat je prikaz utrošenih resursa u zavisnosti od broja redundantnih modula. Vidimo da je u slučaju uvođenja sedam redundantnih MAC modula po redu filtra, potrebno skoro deset hiljada LUT-ova, osam stotina FF-a i dve stotine DSP-eva više nego da smo uveli po tri redundantna modula.

	LUT	FF	DSP
No redundancy	2395	608	52
Redundancy = 3	7601	1013	148
Redundancy = 5	12385	1411	244
Redundancy = 7	17392	1809	340

Tabela 1. Utrošeni resursi u zavisnosti od broja redundantnih modula

Tabela 2. sadrži podatke o potrošnji u zavisnosti od broja redundantnih modula. Razlika nije velika, ali postoji i očekivano - potrošnja raste sa porastom broja kopONENTI na čipu.

	Total On-Chip Power	Dynamic[signals,logic,DSP...]	Device Static
No redundancy	0.123 W	11%	89%
Redundancy = 3	0.123 W	18%	82%
Redundancy = 5	0.137 W	27%	73%
Redundancy = 7	0.151 W	34%	66%

Tabela 2. Potrošnja u zavisnosti od broja redundantnih modula

	WNS [Worst Neg. Slack]	WHS [Worst Hold Slack]	Max Frequency
No redundancy	46.731 ns	0.619 ns	18.77 MHz
Redundancy = 3	42.629 ns	0.238 ns	17.42 MHz
Redundancy = 5	41.948 ns	0.250 ns	17.20 MHz
Redundancy = 7	39.380 ns	0.263 ns	16.40 MHz

Tabela 3. Frekvencija u zavisnosti od broja redundantnih modula

Sistem je projektovan da radi na 10MHz, a kolika je maksimalna frekvencija rada zavisi od broja komponenti, što je naznačeno u tabeli 3. *Slack* se definiše kao potrebno vreme da neki signal pristigne. Tako da *WNS* (eng. *Worst Negative Slack*) govori koliko je slobodnog vremena ostalo nakon što se ispune vremenska ograničenja (tj. za koliko možemo da skratimo periodu takta, a da vremenska ograničenja ostanu ispoštovana). Ako je T perioda jednog ciklusa takta, maksimalnu frekvenciju dobijamo po formuli : $V_{\max} = 1/(T-WNS)$. Na primer, u slučaju pet redundantnih modula, $WNS = 41.948$ ns, perioda je $T = 100$ ns, tako da $V_{\max} = 1/58.052\text{ns} \approx 17.2$ MHz. Vidimo da je maksimalna frekvencija najveća kad se ne uvođe redundantni moduli ($\approx 18.77\text{MHz}$), dok sa uvođenjem dodatnih modula postepeno opada.

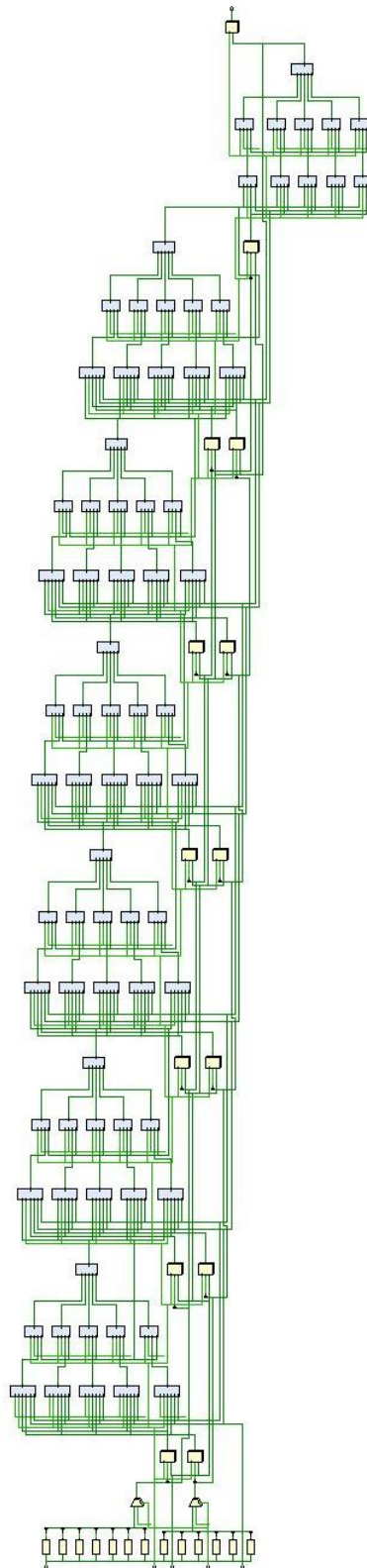
5. Zaključak

IIR filter sa implementiranom *Self-purging redundancy* tehnikom hardverske tolerancije na otkaz pokazao je mogućnost efikasnog maskiranja velikog broja grešaka. Simulacije su vršene sa pretpostavkom da su prekidači i glasači idealne komponente, međutim realna pouzdanost nijedne komponente nije stopostotna, što može da dovede do nešto lošijih performansi od očekivanih. Takođe, uvođenjem redundanse značajno se povećava broj komponenti u sistemu. Na primer, IIR filter koji ima 3 MAC modula, posle uvođenja minimalne redundanse, sadrži 9 MAC modula, 9 prekidača i 3 glasača. U slučaju projektovanog sistema moduli sadrže samo osnovne gradivne komponente: množače, sabirače, registre, međutim u sistemima sa velikim brojem skupih komponenti (poput memorija), treba pažljivo birati kojim podmodulima želimo povećati pouzdanost i za koliko.

6. Literatura

- [1] <https://www.elektronika.ftn.uns.ac.rs/digitalni-sistemi-otporni-na-otkaz/specifikacija/specifikacija-predmeta/> posećeno jul 2022.
- [2] <https://www.elektronika.ftn.uns.ac.rs/diskretni-sistemi/specifikacija/specifikacija-predmeta/> posećeno jun 2022.
- [3] <http://i.stanford.edu/pub/ctr/reports/csl/tr/73/62/CSL-TR-73-62.pdf> posećeno maj 2022.
- [4] <https://www.mathworks.com/help/signal/ug/the-filter-function.html> posećeno maj 2022.
- [5] <https://docs.xilinx.com/v/u/2020.1-English/ug835-vivado-tcl-commands> posećeno jun 2022.
- [6] <http://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html> posećeno jun 2022.

Dodatak



Slika A IIR filter- redundansa pet, izgled Vivado schematic-a