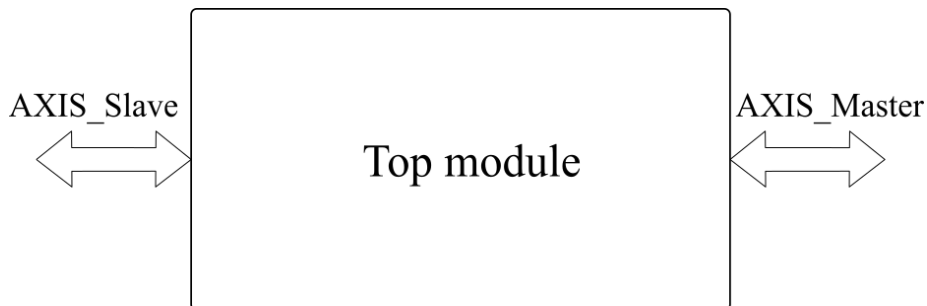


FIR filter otporan na otkaz

Zadatak

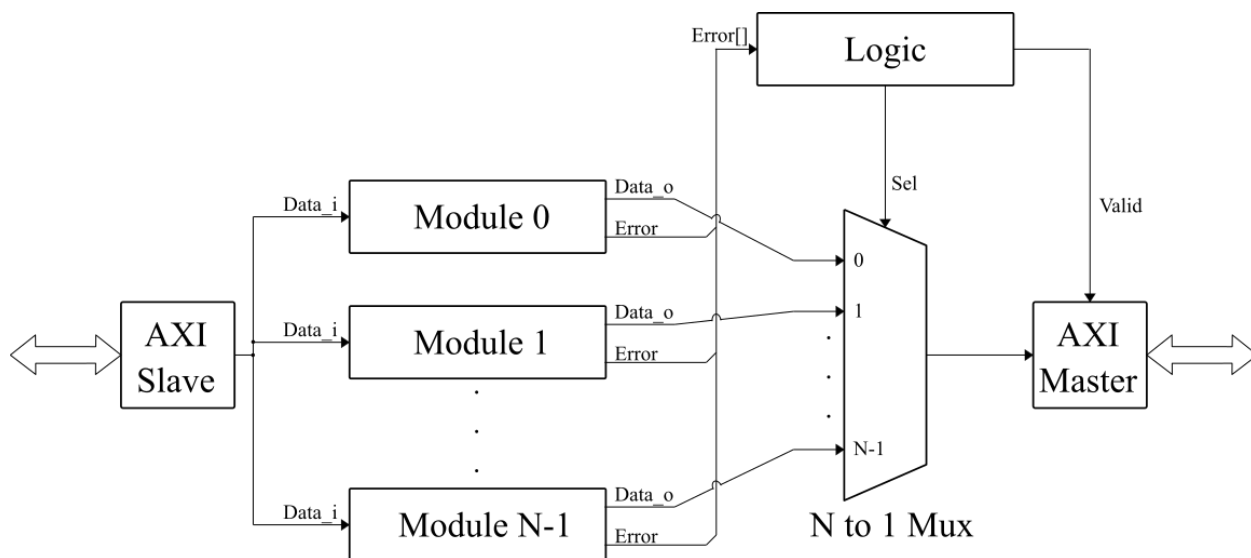
Zadatak ovog projekta je bio da se implementira FIR filter proizvoljnog reda na koji je primenjena *Standby Sparing* tehnika otpornosti na otkaz. Redundansu je trebalo primeniti na nivou FIR filtra. Top modul je trebalo da poseduje AXI stream slave i master interfejs.



Slika 1 Blok dijagram top modula

Implementacija

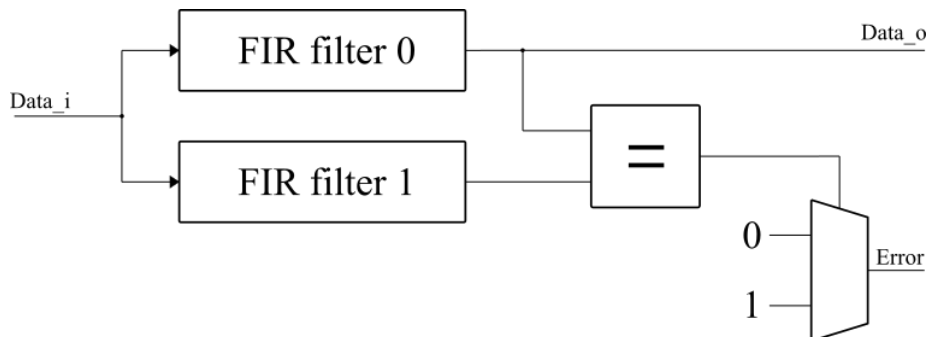
Na slici 2 se nalazi šema implementiranog sistema. Moduli su replicirani N (*module_ord* u kodu) puta i odbirci koji stižu sa AXI slave ulaza u sistem se šalju na ulaze svih repliciranih modula. Jedan od *data_o* izlaza modula se multipleksira na AXI master izlaz sistema.



Slika 2 Šema implementiranog sistema

Na slici 3 su prikazani najbitniji detalji unutrašnjeg izgleda jednog modula. Upotrebljena je tehnika *duplication with comparison*. Unutar modula su instancirana dva ista FIR filtra koji istovremeno dobijaju iste odbirke na ulazu. Izlaz jednog filtra se prosleđuje van modula, a

pomoću *Error* signala se prati da li ta dva filtra imaju iste rezultate, odnosno, da li se desio kvar na jednom od njih. Ako je *Error* signal postavljen na jedinici to znači da je modul u kvaru.

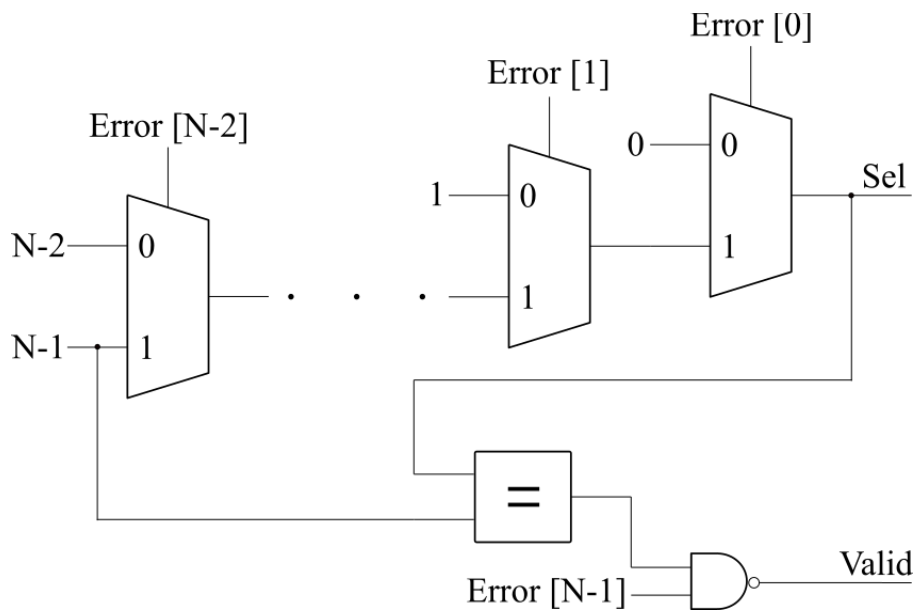


Slika 3 Izgled jednog modula

Svi *Error* signali se spajaju u N-bitni signal (*Error[]*) koji se šalje u komponentu koja određuje sa kog modula će da prosledi odbirke na izlaz sistema i da li će oni biti validni (povezano na *axi_tvalid_o_s* signal u kodu). tako da se prosleđuju odbirci sa modula sa najmanjim indeksom koji još radi. Npr. ako su modul 0 i modul 1 u kvaru, a modul 2 radi, njegov izlaz će se proslediti (*Sel=2*). Dok bar neki moduli rade, izlazni odbirci sistema će biti validni, ako su svi u kvaru, odbirci će biti nevalidni. Rad ove komponente je prikazan na tabeli 1, a način na koji je ta logika implementirana se nalazi na slici 4. Ovaj sistem može da toleriše kvarove N-1 modula, a detektuje greške na svih N modula.

Error[]	Sel	Valid
XX...XX0	0	1
XX...X01	1	1
XX...011	2	1
⋮	⋮	⋮
01...111	N-1	1
11...111	N-1	0

Tabela 1 Rad logic komponente za računanje Sel i Valid signala

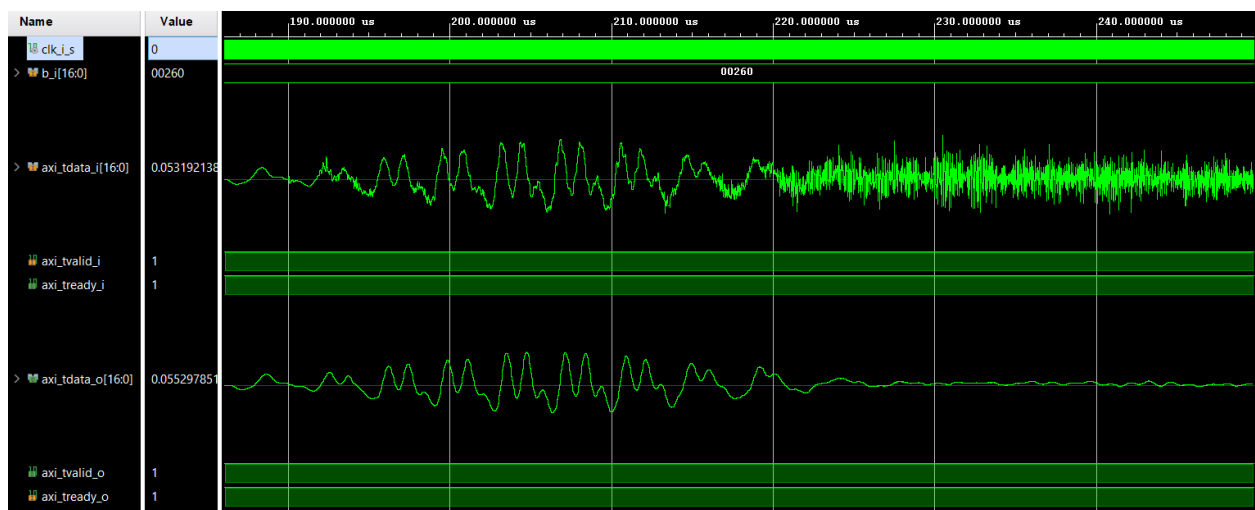


Slika 4 Izgled logike za računanje Sel i Valid signala

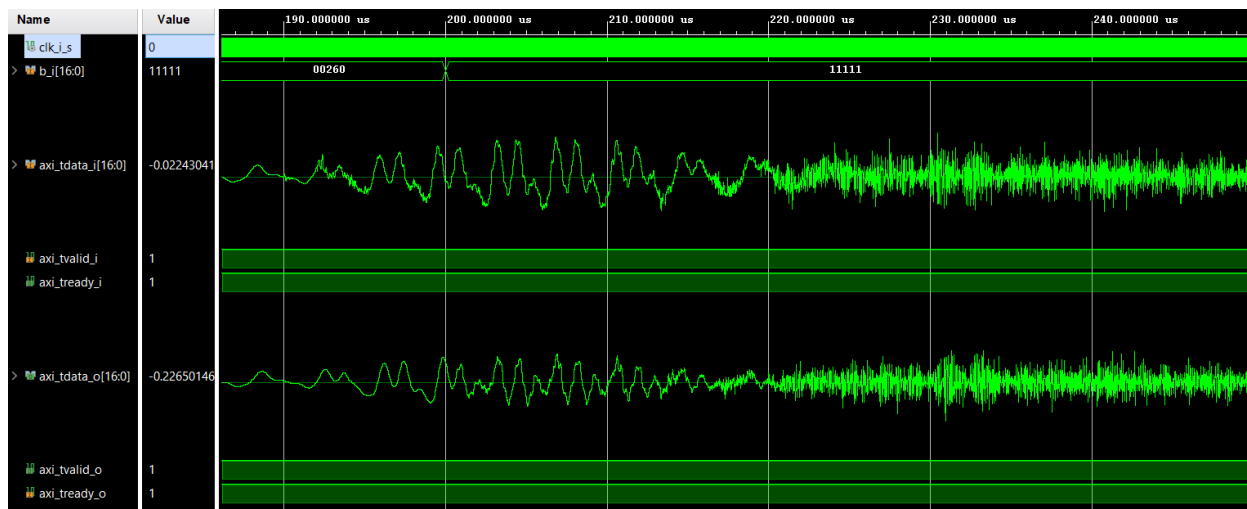
U konačnoj implementaciji su u celom sistemu dodati registri da bi se očuvala frekvencija.

Simulacija

Nakon implementacije je pomoću simulacije pokazano da sistem radi. Posmatran je FIR filter 20. reda sa podacima dužine 17 bita (1 za celobrojni i 16 za razlomljeni deo). Greške su forsirane pomoću .tcl komandi. Najpre je posmatran uticaj pojavljivanja greške na filter bez *fault tolerant* tehnika. Na slici 5 su prikazani ulazni i izlazni signali kada nema greške, a na slici 6 je prikazan uticaj greške na izlazni signal. Greška se pojavljuje na koeficijentu za množenje b u prvoj MAC jedinici posle 200µs: promena sa 0x002600 na 0x111111.

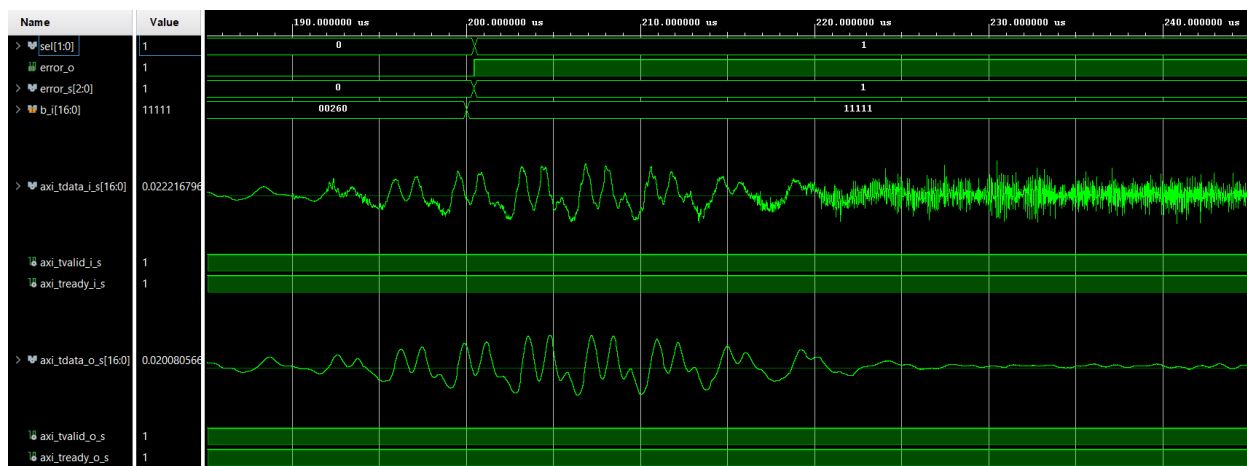


Slika 5 Ulazni i izlazni signali na filteru koji nije otporan na otkaz kada nema greške



Slika 6 Izlazni signal na filtru koji nije otporan na otkaz kada postoji greška

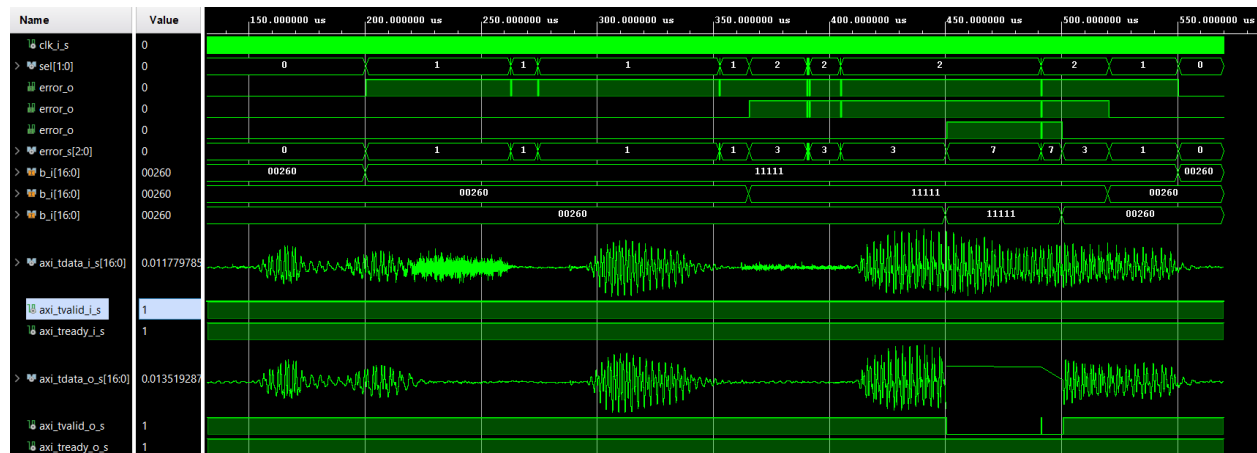
Ista greška je replicirana i za slučaj *fault tolerant* FIR filtra. U početku modul 0 radi ispravno i njegov izlaz se šalje na AXI Stream izlaz sistema. Nakon što se detektuje greška u modulu 0, *sel* signal se menja na 1 i na dalje se izlaz modula 1 prosleđuje dalje. Kako je implementirana *hot standby* verzija, modul 1 je spreman da odmah nastavi gde je modul 0 stao i uticaj greške se ne vidi na izlazu sistema. Na slici 7 je ovaj slučaj prikazan.



Slika 7 Implementirani filter otporan na otkaze može da detektuje i toleriše grešku

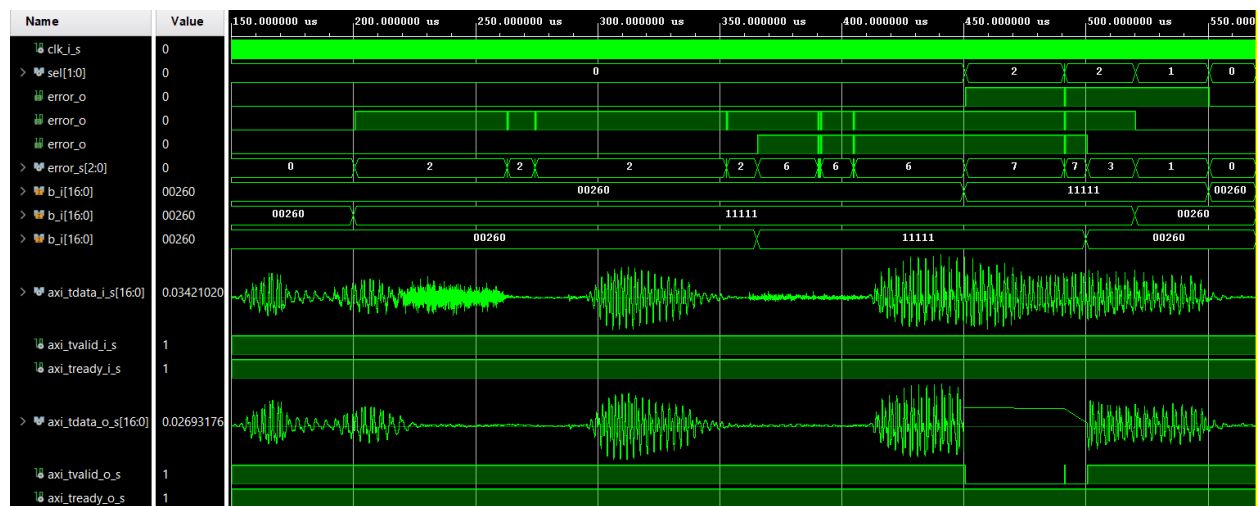
Standby Sparing tehnika može da toleriše $N-1$ grešaka i da detektuje N grešaka. U sledećim primerima je ovo pokazano. U simulaciji je korišćen sistem sa 3 modula ($N=3$). Na slici 8 je prikazan prvi primer (.tcl skripta *force_signals1.tcl*). Greške se pojavljuju redom na modulima 0 do 2 (*b_i* signali sa 002600 prelaze u 11111). U početku radi modul 0 i kada detektuje grešku ($t = 200 \mu s$) će setovati svoj *error_o* signal. Kako je modul 1 ispravan *sel* signal će dobiti vrednost 1 i glavni modul će postati modul 1. Zatim se pojavljuje greška na modulu 1 ($t = 365 \mu s$), *sel* signal dobija vrednost 2 i modul 2 postaje glavni modul. Ovako je tolerisano $3-1=2$ grešaka. Sledeća greška na modulu 2 ($t = 450 \mu s$) će se detektovati, ali neće se moći tolerisati jer nema više

standby modula. Sel signal ostaje na vrednosti 2, ali biće signalizirano da izlazni podaci sistema nisu više validni ($\text{axi_tvalid_o_s} \leq '0'$). Na kraju simulacije su obrnutim redom uklanjane greške.



Slika 8 Prvi slučaj kada greške se pojavljuju redom

Na slici 9 je prikazan drugi primer kada se greške ne pojavljuju redom na modulima (.tcl skripta force_signals2.tcl).. U ovom primeru se prvo pojavljuje greška na modulu 1 pa na modulu 2, a modul 0 je sve vreme glavni modul. Nakon što se pojavi i poslednja greška, na modulu 0, izlazni podaci postaju nevalidni.



Slika 9 Drugi slučaj kada se greške ne pojavljuju redom

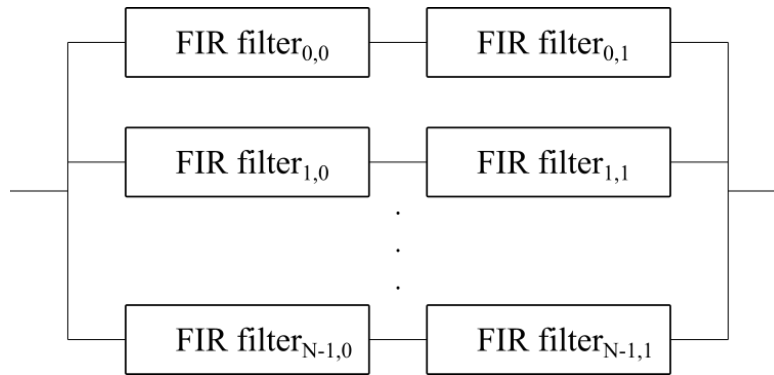
Procena pouzdanosti

Može se dati brza, ali gruba procena pouzdanosti implementiranog sistema pomoću dijagrama pouzdanosti. Ako se uzme FIR filter bez redundansi kao komponenta koja se replicira, njegova pouzdanost se može predstaviti formulom $R(t) = e^{-\lambda t}$. Kako svaki modul radi ispravno ukoliko njegova oba filtra rade, u dijagramu može se predstaviti kao serijska veza dva filtra, a njegova pouzdanost je $R_m(t) = R^2$. Prednost korišćenja *duplication with comparison* tehnike je da je

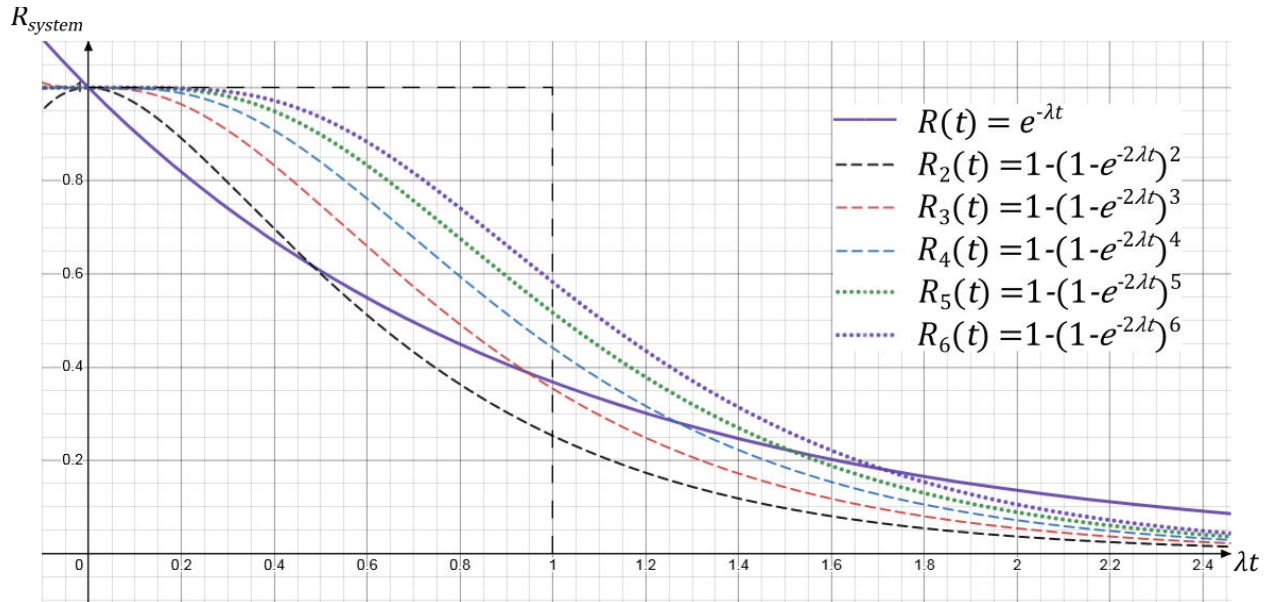
lako implementirati detekciju greške, ali mana je da joj serijska veza daje manju pouzdanost od komponente koja se replicira. Sistem se onda može predstaviti kao paralelna veza tih N modula. Pouzdanost celog sistema se onda može izračunati kao:

$$R_N(t) = 1 - \prod_{i=1}^N (1 - R_m) = 1 - \prod_{i=1}^N (1 - e^{-2\lambda t})$$

Dijagram pouzdanosti je prikazana na slici 10, a na slici 11 je prikazano poređenje pouzdanosti za filter komponentu i za implementirani *Standby Sparing* sistem za različite vrednosti N.



Slika 10 Dijagram pouzdanosti za implementirani sistem



Slika 11 Poređenje pouzdanosti sa i bez *Standby Sparing* tehnike

Što se tiče sigurnosti sistema, kada dođe do kvara svih modula, sistem javlja da su podaci nevalidni, što predstavlja njegovo fail-safe stanje. Ipak, kao što se može videti na slici 3, moguća je pojava glica na axi_tvalid_o_s signalu. Ovo se dešava ukoliko se rezultati dva filtra u modulu

poklope slučajno. Ukoliko su oba filtra jednog modula u kvaru, sistem će potencijalno slati pogrešne podatke kao validne. Ipak, ovo se može desiti samo jedan takt, neka duža sekvenca je praktično nemoguća.

Analiza resursa

U tabeli 2 je prikazano poređenje sistema sa i bez ft (*fault tolerant*) tehnika koji implementiraju FIR filter 20 reda za podatke veličine 17 bita. Primećuje da je DSP jedinica najkritičniji resurs za implementaciji FIR filtra, pa će na njemu biti fokus ove analize. Ako je FIR filter reda M, on koristi M DSP jedinica. Kako svaki redundantni modul koristi dva FIR filtra, i ako je replicirano N modula, broj utrošenih DSP jedinica za implementirani *fault tolerant* FIR filter iznosi $2 \cdot M \cdot N$. Ovaj broj ne može da prevazilazi 80, koliko ima Zybo ploča, pa je red filtra u konfliktu sa maksimalnom mogućom redundansom. U tabeli 3 je prikazan maksimalan moguć red filtra M za različite vrednosti broja modula N, koji se dobija iz nejednačine $2 \cdot N \cdot M \leq 80$. Gledajući primer kada je red filtra 20, u poređenju sa filtrom bez ft tehnika, ukoliko želimo da tolerišemo jednu jedinu grešku potrebno je utrošiti 4 puta više DSP jedinica, a ukoliko želimo bolju pouzdanost od toga, potrebno je smanjiti red filtera.

Sistem	Red filtra	LUT	FF	DSP
Bez ft tehnika	20	20 (0.11%)	357 (1.01%)	20 (25%)
<i>Standby Sparing</i>	20	112 (0.64%)	1500 (4.26%)	80 (100%)

Tabela 2 Poređenje za FIR filter sa i bez *Standby Sparing* tehnike

Broj modula N	Maksimalan red filtra M	DSP
2	20	80 (100%)
3	13	78 (97.5%)
4	10	80 (100%)
5	8	80 (100%)
6	6	72 (90%)
7	5	70 (87.5%)
8	5	80 (100%)

Tabela 3 Maksimalan moguć red filtra za različit broj modula

Analiza frekvencije

Za slučaj filtra bez ft tehnika je korišćena frekvencija od 200MHz. Jedan od zahteva za implementaciju verzije otporne na otkaz je bio da može da radi na istoj frekvenciji. U tabeli 4 je prikazan *worst negative slack* izračunat za frekvenciju 200MHz za slučaj bez ft tehnika i za slučajeve sa implementiranom ft tehnikom. Korišćeni su redovi filtra kao iz tabela 2 (bez ft) i 3 (sa ft). Vidi se da je frekvencija održana za sve $N \leq 6$.

Broj modula N	<i>Worst negative slack</i> za frekvenciju 200MHZ (ns)
Bez ft tehnika	1.080
2	0.526
3	0.398
4	0.446
5	0.157
6	0.391
7	-0.048
8	-0.219

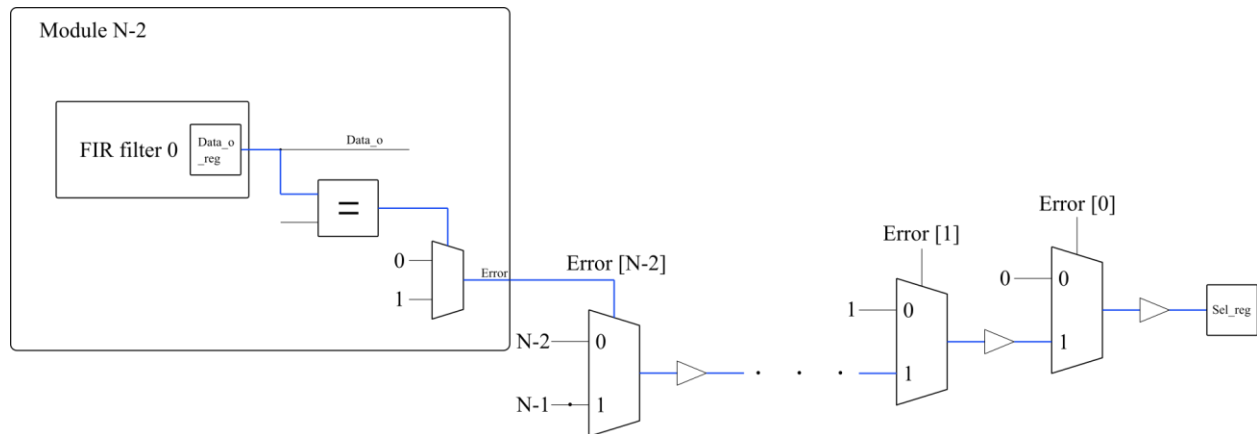
Tabela 4 Analiza frekvencije rada za različit broj modula

Može se zaključiti da broj modula ima uticaja na frekvenciju. U *Vivado* alatu, je analizirana kritična putanja za primer sistema sa 7 modula, gde prvi put nije zadovoljena ciljane frekvencija rada. Rezultati su prikazani na slici 12. Na osnovu ovog je određen opšti izgled kritične putanje, prikazan na slici 13. Putanja počinje u modulu N-2 (5 u primeru sa slike 12), u registru za odbirke koji izlaze iz filtra 0 (data_o_reg). Na svom putu signal prolazi kroz jedan komparator i jedan multiplexer 2 na 1 da bi izašao iz tog modula. Potom prolazi kroz celu logiku za izračunavanje sel signala, kroz svih N-1 multipleksera 2 na 1 i N-1 bafer (nastalih zbog *dont_touch* atributa), da bi stigao do registra sel_reg, što predstavlja kraj kritične putanje. Sa povećanjem broja modula N, linearno se povećava i broj multipleksera kroz koji signal mora da prođe te se povećava i kritična putanja. Jedno rešenje za poboljšanje frekvencije rada je predstavljeno na slici 14. Potrebno je podeliti logiku za izračunavanje sel signala (N-1 multiplexer) na dve ili više faza protočene obrade ubacivanjem registara. Takođe nije loše da se registruje Error signal na izlazu iz modula. Na kraju, dodatnim registrima treba zakasniti, po potrebi, ostale signale u sistemu.

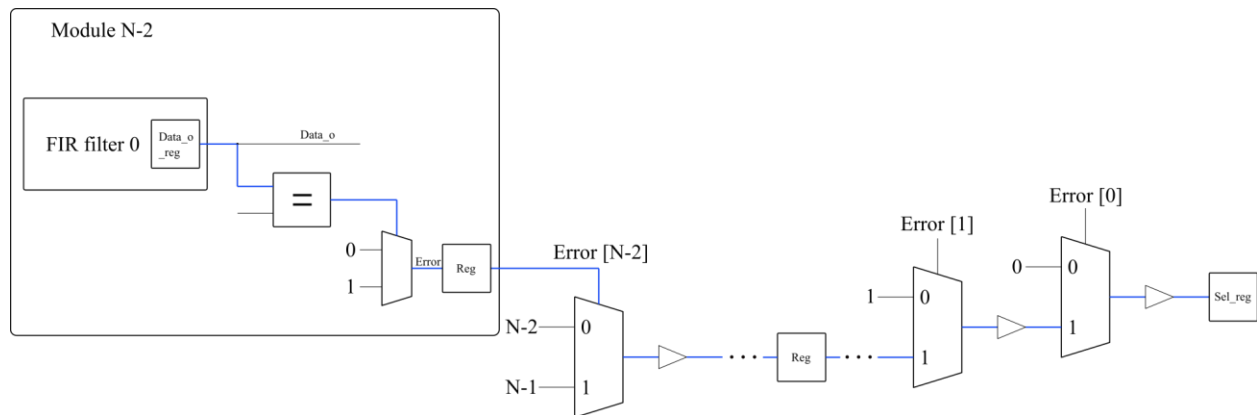
Max Delay Paths

```
Slack (VIOLATED) :      -0.048ns  (required time - arrival time)
Source:               module_gen[5].modules/fir_filter_0/data_o_reg[7]/C
                      (rising edge-triggered cell FDRE clocked by sys_clk_pin  (rise@0.000ns fall@2.500ns period=5.000ns))
Destination:          sel_reg[0]/D
                      (rising edge-triggered cell FDRE clocked by sys_clk_pin  (rise@0.000ns fall@2.500ns period=5.000ns))
```

Slika 12 Timing summary - Route design prozor u Vivado alatu gde je izračunata kritična putanja



Slika 13 Izračunata kritična putanja



Slika 14 Rešenje za poboljšanje frekvencije rada