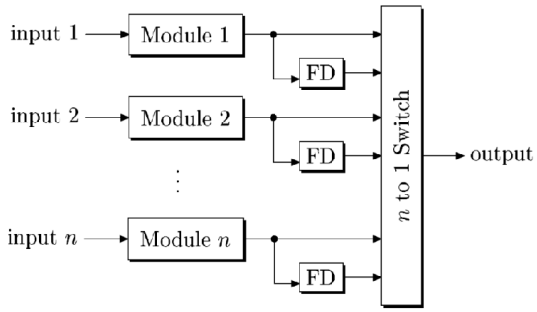
UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA

PROJEKAT IZ DIGITALNIH SISTEMA OTPORNIH NA GREŠKE

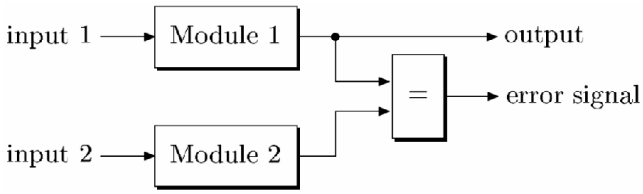
Primena *pair-and-a-spare* tehnike otpornosti na greške na primeru FIR filtra

Asistent : Janković Jana
Student : Milin Ivan E1-79/2023

1. Uvod



1. Standby sparing tehnika

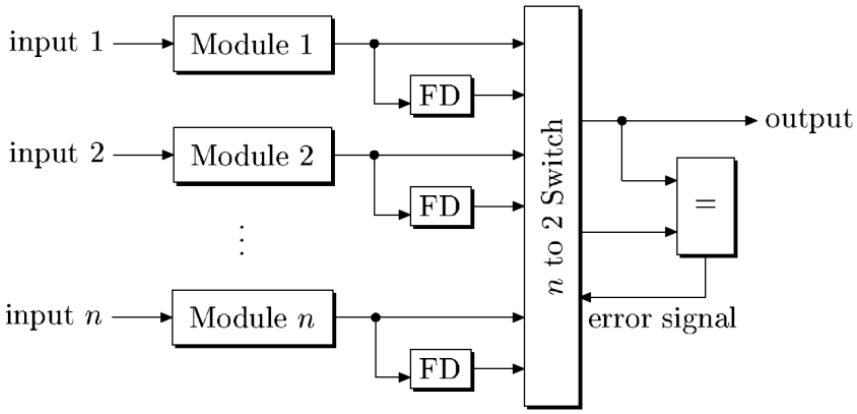


2. Tehnika dupliciranja uz poređenje

Pair-and-a-spare je aktivna tehnika koja kombinuje dobre osobine *standby sparing* tehnike i tehnike dupliciranja uz poređenje.

Na slici 1. prikazana je idejna blok šema *standby sparing* tehnike, jedan od modula je aktivan i snabdeva izlaz sa podacima, dok ostalih n-1 modula je u *standby* režimu i čeka da se taj modul pokvari kako bi mogla *switch* logika da ga zameni sa modulom koji nije u kvaru.
Kvar se detektuje u FD (eng. *fault detection*) bloku koji se sastoji od bloka koji obavlja istu funkcionalnost koju obavlja i Module 1 i komparatora koji postavlja jedinicu na portu *error signal* ukoliko dođe do razlike između vrednosti podatka iz *Module 1* i *Module 2*, što je prikazano na slici 2.

U svakom trenutku svih n modula je aktivno i prima podatke sa ulaza, dok samo jedan od tih modula prosleđuje podatke na izlaz, mana je povećana potrošnja jer svi istovremeno rade.



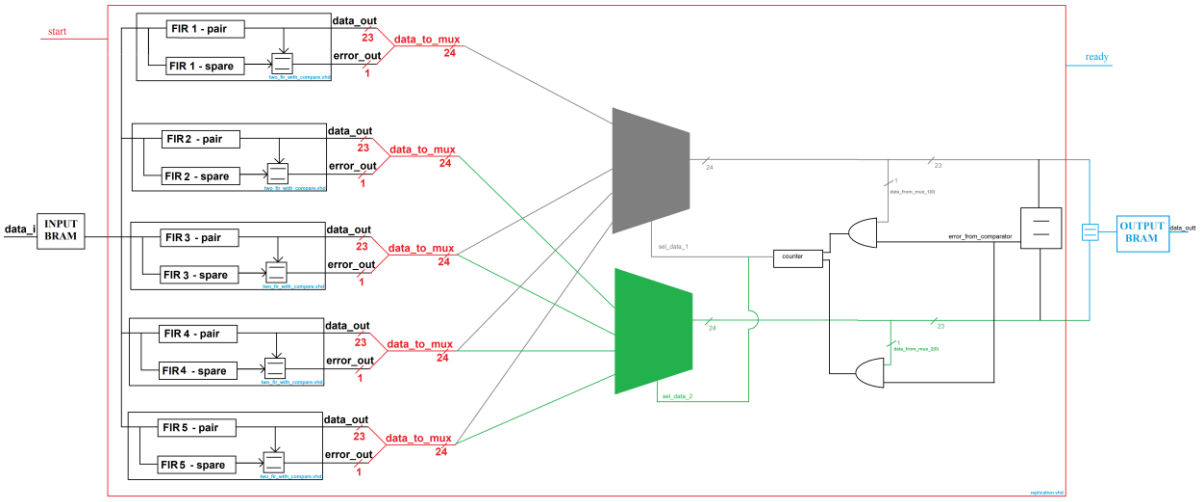
3. Pair-and-a-spare tehnika

Na slici 3. prikaza je blok šema *pair-and-a-spare* tehnike, ideja *pair-and-a-spare* je slična, uz razliku što se ovde iz *switch* logike istovremeno prosleđuju dva izlaza iz modula čiji se podaci proveravaju u izlaznom komparatoru,

ukoliko se pojavi jedinica na portu *error signal switch* logika treba da utvrdi koji od ta dva modula nije ispravan i da ga zameni nekim od *standby* modula.

Ova tehnika otpornosti na greške sa n-modula toleriše n-1 grešku nakon čega sistem prestaje ispravno da radi.

2. Implementacija sistema



4. Blok šema bez ulaznog i izlaznog BRAMa

Na slici 4. prikazana je blok šema po kojoj je implementirana tehnika.

Opis strukture poređano po fajlovima :

1. Unutar fajla ***fir_param.vhd*** implementiran je FIR filtar (filtar je petog reda, a podaci su širine 24 bita).
2. Unutar fajla ***two_fir_with_compare.vhd*** implementirana je tehnika dupliciranja FIR filtra uz poređenje.
3. Unutar fajla ***replication.vhd*** implementirana je *pair-and-spare* tehnika.
4. Unutar fajla ***top.vhd*** instanciran je ulazni i izlazni BRAM kao i modul kreiran u fajlu *replication.vhd*.

Switch logika implementirana je pomoću dva parametrizovana multipleksera, prvi multiplekser prima izlaze svih modula osim od drugog modula, dok drugi multiplekser prima izlaze svih modula osim od prvog modula.
Inicijalno je postavljeno da je selekcioni signal oba multipleksera na nuli i podaci oba modula se šalju kroz oba multipleksera ka izlaznom komparatoru.

Ukoliko se ispostavi da podaci nisu isti, izlazni komparator postavlja jedinicu na žicu *error_from_comparator*, čime signalizira *switch* logici da je jedan od modula neispravan nakon toga se proverava koji od modula je neispravan, taj se isključuje i vrednost unutrašnjeg brojača se uvećava za jedan i selekcioni signal tog multipleksera koji je prosleđivao podatke modula koji je neispravan dobija vrednost tog brojača.

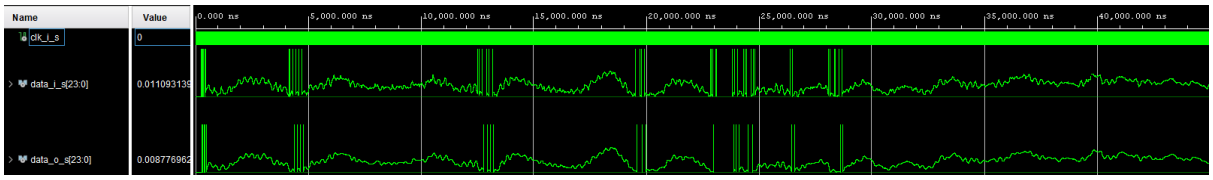
Podaci se prvo smeštaju u ulazni BRAM, nakon što se *start* postavi na jedinicu počinje obrada podataka iz ulaznog BRAMa, kada sistem završi obradu i smesti podatke u izlazni BRAM postavlja se jedinica na portu *ready*.

3. Rezultati simulacije i testiranje ispravnosti

Prilikom pokretanja simulacije prvo je proverena ispravnost zasebnih podsistema čitave hijerarhije kojom je sistem implementiran, nakon čega je dodat ulazni i izlazni BRAM kojim je ispoštovan tekst zadatka.

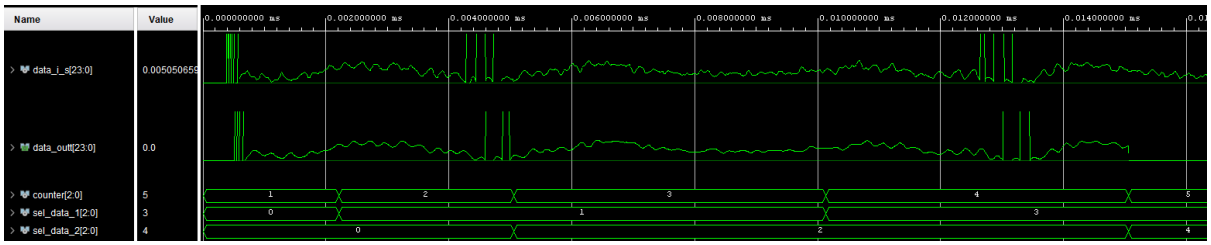
Ulazni signal koji je iskorišćen za testiranje prvobitno je izgenerisan u matlab programskom jeziku i sačuvan u fajl kojim je testirana funkcionalnost sistema.

Prelaskom sa nižeg na viši nivo hijerarhije, unutar testbenča vrednosti podataka koje je sistem generisao su se poredile sa očekivanim vrednostima, koje su takođe prvobitno izgenerisane u matlab-u.



5.Talasni oblik FIR filtra bez tehnika otpornosti na greške

Na slici 5. prikazani su talasni oblici ulaznog i izlaznog signala kod FIR filtra kod kog nije primenjena ni jedna tehnika otpornosti na greške.



6. Tolerancija na grešku kod sistema sa 5 repliciranih FIR filtara

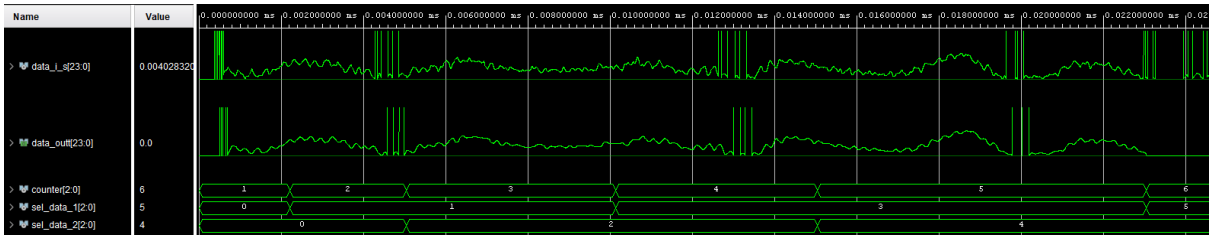
Na slici 6. prikazano je ponašanje sistema u kom se nalazi pet repliciranih FIR filtara, namernim umetanjem greški u sistem brojač *„counter"* se uvećavao do 5, nakog čega sistem prestaje ispravno da radi i izlazni signal postaje nula.

Provera ispravnosti sistema vršila se namernim umetanjem greški što je postignuto pomoću skripte *force_error.tcl, slika 7.*

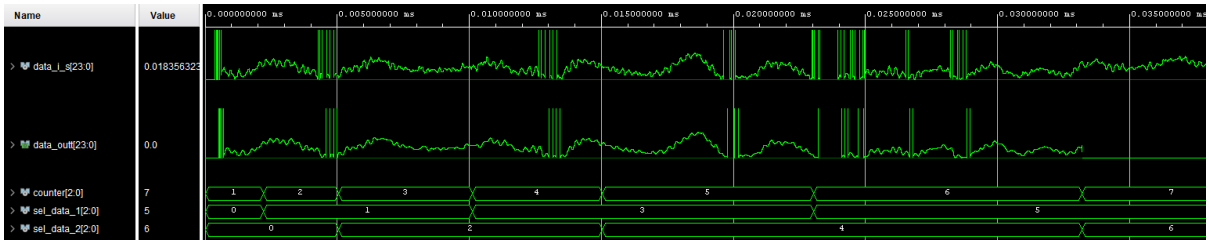
Skripte koje su korišćene u ovom projektu nalaze se u folderu *scripts* :
1. *create_prj.tcl* služi za kreiranje Vivado projekta.
2. *force_error* služi sa forsiranje greški za fajl *replication.vhd.*
3. *remove_force* služi sa uklanjanje greški za fajl *replication.vhd.*
4. *force_error_top* služi sa forsiranje greški za fajl *top.vhd*
5. *remove_force_top* služi sa uklanjanje greški za fajl *top.vhd.*

```
add_force {/tb/uut_fir_filter/\replication_of_fir(0)\/replication/first_data_o_s[10]} -radix hex {1 2000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(1)\/replication/first_data_o_s[10]} -radix hex {1 5000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(2)\/replication/first_data_o_s[10]} -radix hex {1 10000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(3)\/replication/first_data_o_s[10]} -radix hex {1 15000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(4)\/replication/first_data_o_s[10]} -radix hex {1 23000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(5)\/replication/first_data_o_s[10]} -radix hex {1 33000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(6)\/replication/first_data_o_s[10]} -radix hex {1 36000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(7)\/replication/first_data_o_s[10]} -radix hex {1 38000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(8)\/replication/first_data_o_s[10]} -radix hex {1 41000ns} -cancel_after 400ms
add_force {/tb/uut_fir_filter/\replication_of_fir(9)\/replication/first_data_o_s[10]} -radix hex {1 53000ns} -cancel_after 400ms
```

7.tcl skripta za forsiranje greški za fajl replication.vhd

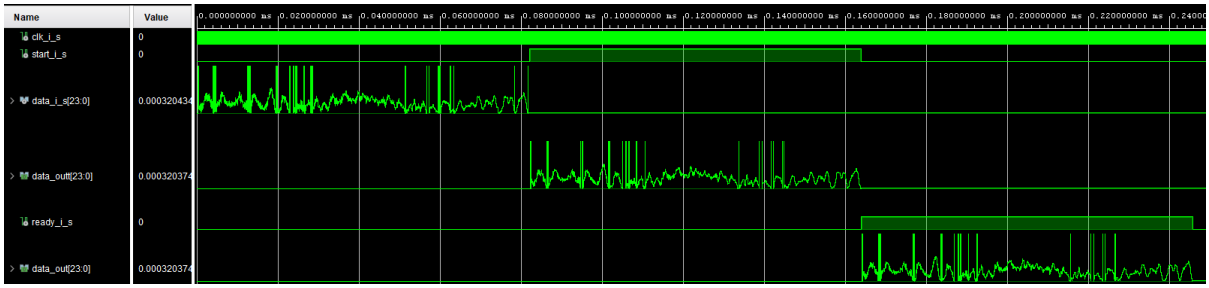


8.Tolerancija na grešku kod sistema sa 6 repliciranih FIR filtara



9.*Tolerancija na grešku kod sistema sa 7 repliciranih FIR filtara*

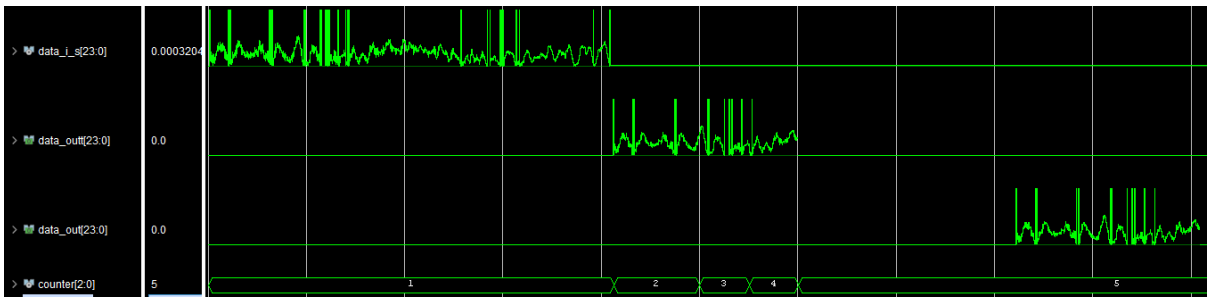
Na slikama 8. i 9. prikazano je ponašanje sistema kod kog je prisutno 6, odnosno 7 repliciranih FIR filtara, pri uvećavanju *counter-a* na 6 odnosno 7 vrednost izlaznog signala postaje nula.



10. Ulazni i izlazni signal nakon dodavanja ulaznog i izlaznog BRAMa

Na slici 10. prikazano je kako se ulazni signal *data_i_s* prvo smešta u ulazni BRAM, zatim postavljanjem jedinice na *start_i_s* taj ulazni signal se propušta kroz čitav sistem da bi pojavom *ready_i_s* na jedinicu sistem signalizirao da je obradio sve podatke i smestio ih u izlazni BRAM, nakon čega se šalju adrese na port *addr_read* da bi se smešteni podaci isčitali preko porta *data_out*.

Na slici postoji i signal *data_outt* koji predstavlja izlaz iz redudantnog FIR filtra tj. ulazni signal u izlazni BRAM.



11. Ulazni i izlazni signal nakon dodavanja ulaznog i izlaznog BRAMa i forsiranja greške

Na slici 11. prikazani je ulazni i izlazni signal kako izgleda kada se u FIR sistem unose greške i istroše svi zamenski filtri.

4. Analiza utrošenosti resursa

1. Analiza utrošenosti resursa i frekvencija

Broj modula	LUT	FF	BRAM	DSP	IO	BUFG	Potrošnja [W]	Frekvencija [MHz]
Bez tehnike	4	24	/	18	77	1	0.12	103
3	198	272	6	108	104	1	0.204	63.083
4	278	344	6	144	104	1	0.234	63.015
5	376	419	6	180	104	1	0.26	63.035
6	424	491	6	216	104	1	0.293	63.111

Analiza resursa izvršena je za razvojni sistem Xilinx Zynq 7000, Zybo Z7-20.

U tabeli 1. može se videti utrošenost resursa kao i najveća frekvencija na kojoj sistem može da radi ukoliko se poveća broj repliciranih zamenskih blokova.

Može se uočiti da se frekvencija ne menja skoro uopšte sa povećanjem broja redudantnih blokova, ukoliko su svi ispravni u svakom trenutku će samo dva biti istovremeno aktivna dok će ostali biti u *standby* režimu.

Skaliranjem sistema utrošenost resursa raste, ukoliko bi se broj redudantnih modula replicirao 7 ili više puta MAC operacija bi morala da se implementira pomoću LUT ćelija jer ne bi bilo dovoljno DSP ćelija za impelementaciju.

Skaliranjem sistema takođe i potrošnja raste jer kod *pair-and-a-spare* tehnike svi moduli istovremeno rade.

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.148ns  (required time - arrival time)
  Source:                 INPUT_bram/memory_reg_1/CLKBWRCLK
                            (rising edge-triggered cell RAMB36E1 clocked by clk  {rise@0.000ns fall@8.000ns period=16.000ns})
  Destination:            pair_and_spare_FIR/replication_of_fir[0].replication/second_module/other_sections[1].fir_section/reg_s1/A[15]
                            (rising edge-triggered cell DSP48E1 clocked by clk  {rise@0.000ns fall@8.000ns period=16.000ns})
  Path Group:             clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            16.000ns  (clk rise@16.000ns - clk rise@0.000ns)
  Data Path Delay:        15.366ns  (logic 2.454ns (15.970%)  route 12.912ns (84.030%))
```

12. Kritična putanja

Na slici 12. prikazana je kritična putanja za tri replicirana modula.

5. Litaratura

[1] https://www.elektronika.ftn.uns.ac.rs/digitalni-sistemi-otporni-na-greske/specifikacija/specifikacija-predmeta/ - januar 2024.