

Projekat iz predmeta Digitalni sistemi otporni na otkaz

Tema projekta : Na FIR filter implementirati dve hardverske tehnike, *N modular redundancy* na MAC module I *N modular redundancy with spares* na voting logiku.

Asistent : Jana Jankovic

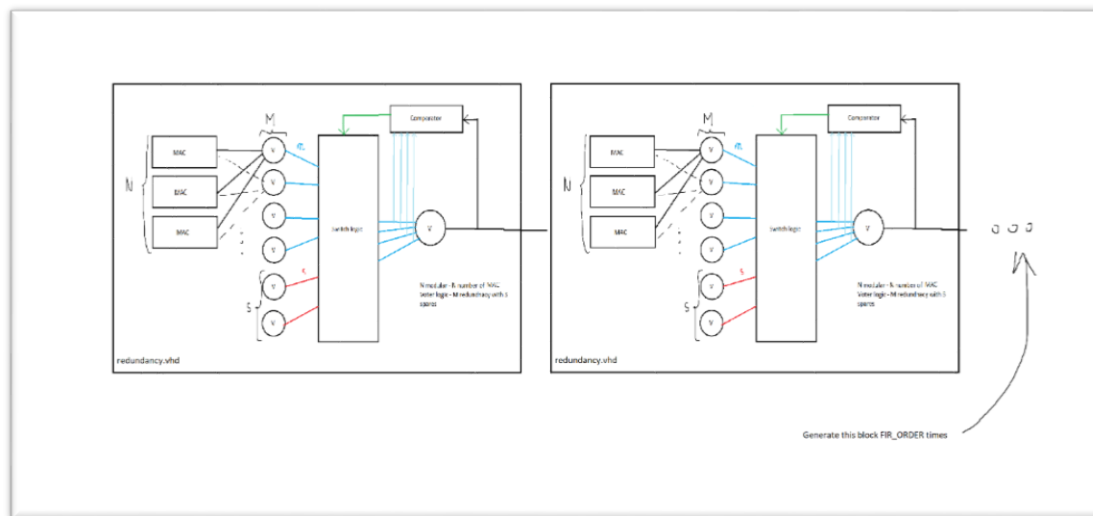
Student : Petar Stamenkovic E1-11/2023

Sadržaj dokumentacije

1. Uvod I ideja o realizaciji celog sistema.....	3
2. Implementacija komponenti	4
2.1. Implementacija komparatora	4
2.2. Implementacija votera	5
2.3. Implementacija switch logike	5
2.4. Implementacija redundancy komponente	5
3. Analiza resursa.....	6
4. Testiranje tehnika (Forsiranje)	6
4.1. Tehnika <i>N modular redundancy</i>	7
4.2. Tehnika <i>N modular redundancy with spares</i>	7

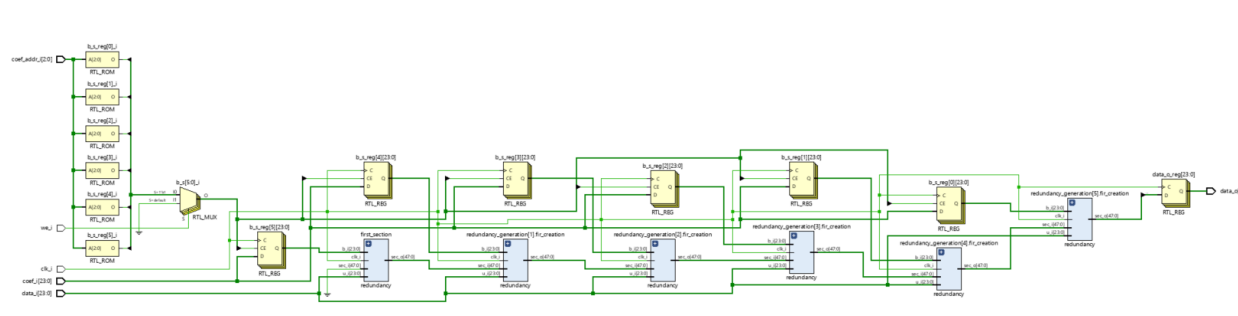
1. Uvod I ideja o realizaciji celog sistema

Sledeci document predstavlja kratku dokumentaciju I opis rada sistema koji je otporan na greske. Radi se o N modularnoj (N modular redundancy) tehnici za MAC modul I N modular with spares tehnici za voting logiku. Ideja sistema je prikazana na sledecoj slici 1:

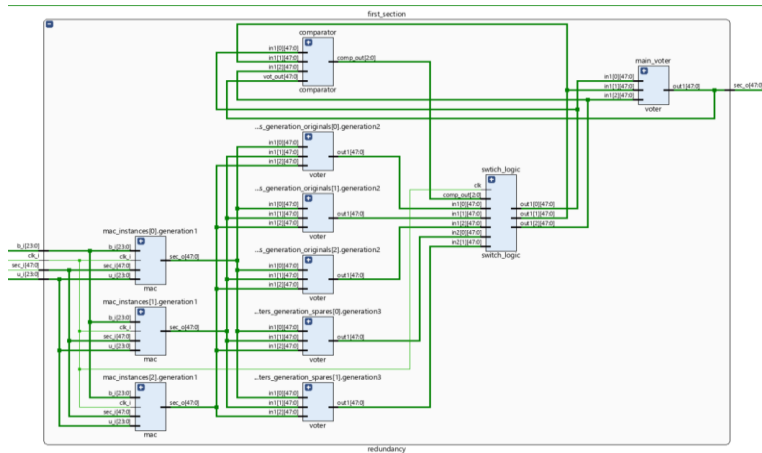


Slika 1 : Skica celokupnog Sistema

Cela ideja je spakovana u komponenti **redundancy** koja u sebi generise N MAC modul jedinica ciji se izlazi vode na $M + K$ voter komponenti. Voter tehnika koja se koristi je majority vote tj. prosledjivanje ulaza koji se najviše pojavljuje. Na izlazu iz switch logike imamo finalni voter ciji je cilj zajedno sa komparatorom otkrije koji modul ne radi kako treba kako bi ista switch logika mogla da obavi zamenu. Na izlaz finalnog votera se vezuje **sec_o** kao signal **mac_inter**. Ova redundancy komponenta se ponavlja FIR_ORDER puta tj. red filtra puta. Model ne koristi generic mape uglavnom (samo za finalni voter), jer je ceo sistem parametrizovan iz paketa *variable_io_package*. Izlaz iz poslednjeg **redundancy** modula je **data_o** tj izlaz samog filtra.



Slika 1a : Elaboriran dizajn celokupnog filtra (reda 5)



Slika 1b) Elaboriran redundancy modul – primetno poklapanje sa sl.1

2. Implementacija komponenti

2.1. Implementacija komparatora

Komparator prima M ulaza kao izlaze iz switch_logike I izlaz iz finalnog votera. Kroz for petlju se vrši poredjenje I ukoliko dodje do neslaganja na *i-toj* komponenti, odgovarajuci *i-ti* bit izlaza komparatora se dize na '1' cime se daje informacija switch logici o tome **koja** komponenta ne radi I koja treba da se zameni.

```
architecture Behavioral of comparator is
begin
process(vot_out, in1)
begin
for i in 0 to NUM_MODULAR-1 loop
if(vot_out /= in1(i)) then
comp_out(i) <= '1';
else
comp_out(i) <= '0';
end if;
end loop;
end process;
end architecture;
```

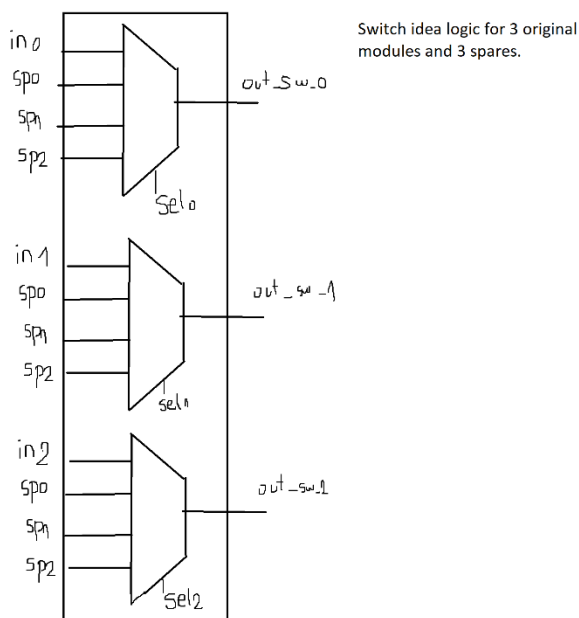
Slika 2 : Realizacija komparatora u Vivadu

2.2. Implementacija votera

Ideja za voter je drugačija od standardne implementacije sa prevadnja. Ovde se porede redom svi ulazi koje dobije I smesta informacije o podaranju istih u poseban niz **cnt_v**. Za svako podudaranje *i-ta* pozicija u nizu se povećava za 1, I na kraju se u drugom procesu trazi najveća vrednost tog niza. Recimo da se najveća vrednost nalazi na *j-toj* poziciji, to znaci da cemo proslediti *j-ti* signal na ulazu.

2.3. Implementacija switch logike

Switch logika je implementirana u vidu M broja multipleksera od koji svaki ima M+1 ulaza, gde je svakome prvi ulaz *i-ti* signal iz originalnog modula a ostalih M redom rezevere, tako da kada se promeni selekcionni ulaz svakog dobijamo drugačiji izlaz na jednom od M izlaza. To zavisi sve od izlaza komparatora koji ce, ukoliko se on razlikuje od 0, uticati na povećanje *fail* signala I time I promenu selekcionih ulaza pomenutih multipleksera.



Skica implementacije switch logike

2.4. Implementacija redundancy komponente

Sve ove komponente su povezane u skladu sa inicijalnom skicom kako bi se formirao *redundancy* blok koji sadrži obe tehnike za tolerantnost. Redundancy se u top modulu instancira FIR ORDER puta kako bi se formirao FIR filter.

3. Analiza resursa

U sistemu MAC module je implementiran na DSP, a na sledecim slikama ce biti prikazana potrosnja resursa za vecu modularnost MAC modula.

LUT	FF	BRAM	URAM	DSP	Start	Elapsed
17574	264	0.0	0	54	1/30/24, 4:26 PM	00:02:02

MAC modularity 3

LUT	FF	BRAM	URAM	DSP	Start	Elapsed
29938	264	0.0	0	72	1/30/24, 4:30 PM	00:02:50

MAC modularity 4

Sto se tice frekvencije sledi njen proracun : (Unutar *constraint* fajla je clk podesen za 0.000 8.000)

$$16 - 0.890 = 15.110 \rightarrow f = 1 / 15.110 = 66.1 \text{ MHz}$$

Setup

Worst Negative Slack (WNS): 0.890 ns

WNS za dizajnirani system

Informacije o kriticnoj putanji su date na sledecoj slici :

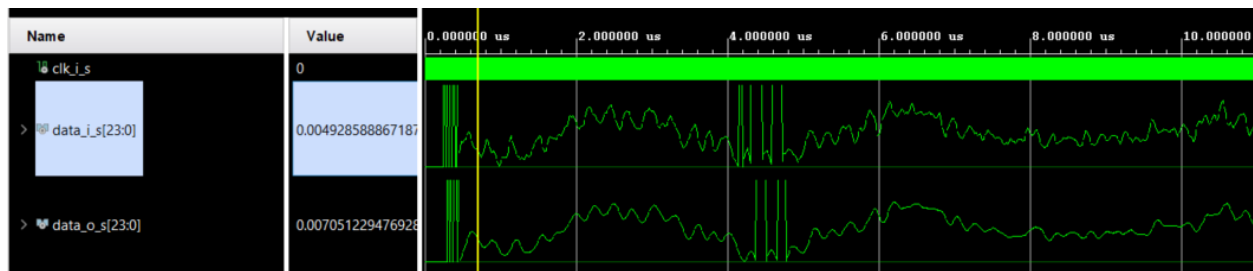
Max Delay Paths

```
-----
Slack (MET) :          0.890ns (required time - arrival time)
Source:          redundancy_generation[3].fir_creation/mac_instances[0].generation1/sec_o0/CLK
                  (rising edge-triggered cell DSP48E1 clocked by clk_i  {rise@0.000ns fall@8.000ns period=16.000ns})
Destination:     redundancy_generation[4].fir_creation/mac_instances[1].generation1/sec_o__0/C[19]
                  (rising edge-triggered cell DSP48E1 clocked by clk_i  {rise@0.000ns fall@8.000ns period=16.000ns})
Path Group:       clk_i
Path Type:        Setup (Max at Slow Process Corner)
Requirement:      16.000ns (clk_i rise@16.000ns - clk_i rise@0.000ns)
Data Path Delay:  14.668ns (logic 8.173ns (55.718%) route 6.495ns (44.282%))
```

Informacije o kriticnoj putanji generisane nakon elaboracije

4. Testiranje tehnika (Forsiranje)

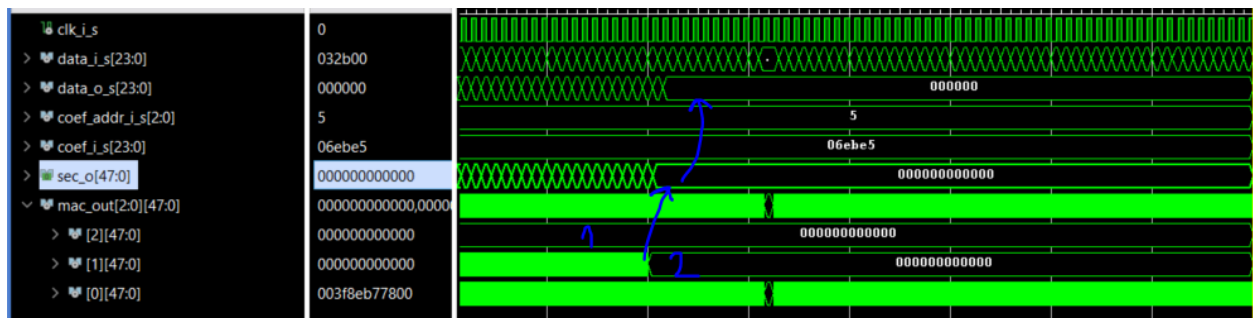
U sledecem poglavlju ce biti testirane pomenute tehnike. Najpre ce biti prikazan sistem bez greski I njegov rad. Pustena je simulacija na 10000ns I sa testbenchom I expected vrednostima se potvrđuje korektan (ocekivan) rad sistema.



Slika 4 : Prikaz rada filtra u simulaciji

4.1. Tehnika *N modular redundancy*

Na sledecoj slici je prikazano kako forsiranjem dva od 3 MAC modular na vrednost 0 prouzrokuje 0 na izlazu te (4te) redundancy komponente a time i 0 na izlazu iz filtra. Pokrenuti skriptu *n_mod_force*. Pokretanjem skripte vidimo da dolazi do *missmatcha* na 850ns, jer je drugi MAC (po skripti) ugasen u 800ns. Preostalih 50ns se ta greska propagira do izlaza obzrim da skripta narušava rad 4-tog redundancy bloka. Za *unforcovanje* pokrenuti *remove_force_n_mod.tcl*. Za proveru ove tehnike otvoriti simulacioni fajli *top_tb_behav.wcfg*.



Slika 5 : Prikaz 2 / 3 MAC modula ne rade – data_o je nula.

4.2. Tehnika *N modular redundancy with spares*

Ova tehnika ce biti testirana forsiranjem nula vrednosti na izlazima voter komponenti, i dalje u 4-tom redundancy bloku. Potrebno je pokrenuti skriptu *force_num_mod_sp.tcl* gde se ocekuje *mismatch* jer smo zadatli previse greske (u ovom primeru konkretno 4). Na talasnim oblicima se vidi reakcija **comp_out** signala cija je uloga da obavesti o greski na izlazu iz *switch* logike i odradi zamenu, sto vidimo time sto se on nakon jednog ciklusa odmah vraca na 0. (jer je sada sve okej – zamena je izvorsena). Za *unforce* se pokrece skripta *remove_force_num_mod_spare.tcl*. Na sledecoj slici 6 vidimo opisanu situaciju. Pokvarili smo prvi modul (indeks 0) i komparator prijavljuje tu gresku sa 001 (jedinica na nultom indeksu). Za proveru ove tehnike otvoriti simulacioni fajl *top_tb_behav2.wcfg*.

switch_in_original[2:0][47:0]	ffffd857a800,ffffd857a800	000000...	ffffea7...	00002b1...	000056f...	ffffeb2...	ffffa79...	00003e8...	0000838...	ffff...
> [2][47:0]	ffffd857a800	000000...	ffffea7...	00002b1...	000056f...	ffffeb2...	ffffa79...	00003e8...	0000838...	ffff...
> [1][47:0]	ffffd857a800	000000...	ffffea7...	00002b1...	000056f...	ffffeb2...	ffffa79...	00003e8...	0000838...	ffff...
> [0][47:0]	000000000000	000000...	ffffea7...	00002b1...	000056f...	ffffeb2...	ffffa79...	00003e8...	0000838...	ffff...
switch_in_spares[1:0][47:0]	ffffd857a800,ffffd857a800	000000...	ffffea7...	00002b1...	000056f...	ffffeb2...	ffffa79...	00003e8...	0000838...	ffff...
> [1][47:0]	ffffd857a800	000000...	ffffea7...	00002b1...	000056f...	ffffeb2...	ffffa79...	00003e8...	0000838...	ffff...
> [0][47:0]	ffffd857a800	000000...	ffffea7...	00002b1...	000056f...	ffffeb2...	ffffa79...	00003e8...	0000838...	ffff...
comp_out_t[2:0]	000	000	000	000	001	000	000	000	000	000

Slika 6 : Prikaza rada switch logike I komparatora