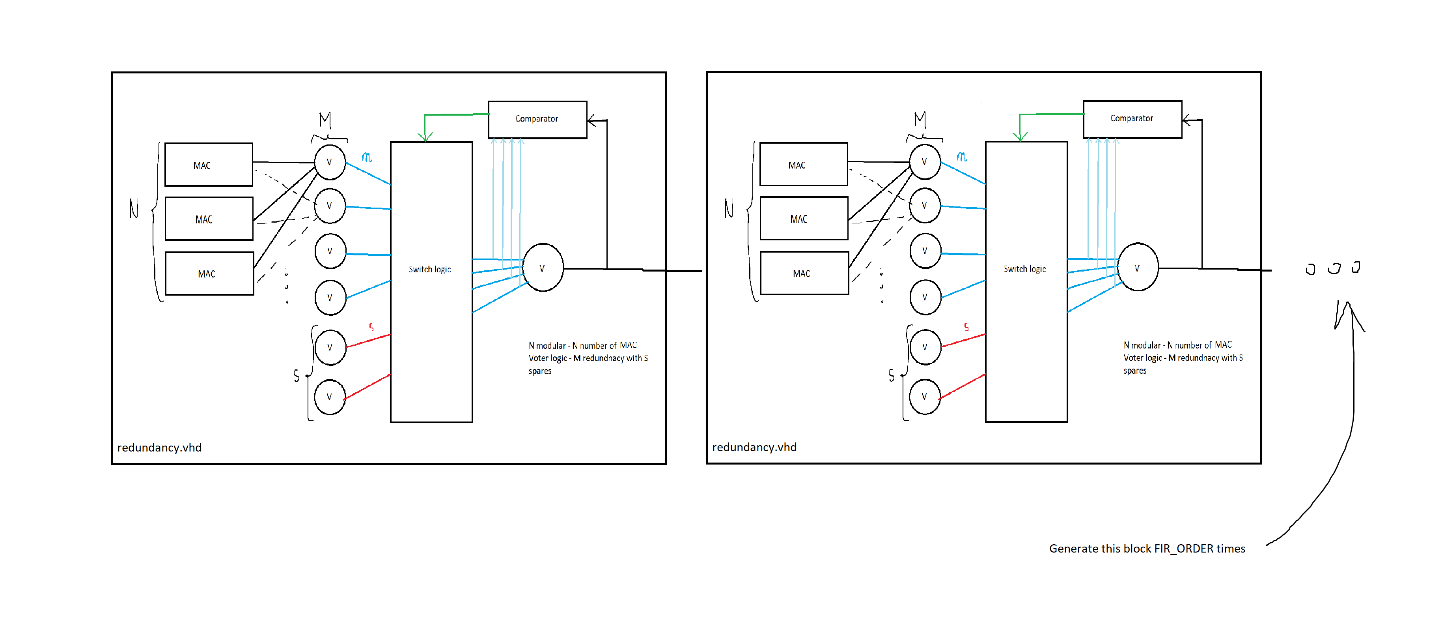
# Dokumentacija za projekat iz predmeta DSONG

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



Cela ideja je spakovana u komponentu **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 najvise 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 system parametrizovan iz paketa *variable\_io\_package*. Izlaz iz poslednjeg **redundancy** modula je data\_o tj izlaz samog filta.

* **Implementacija komparatora**

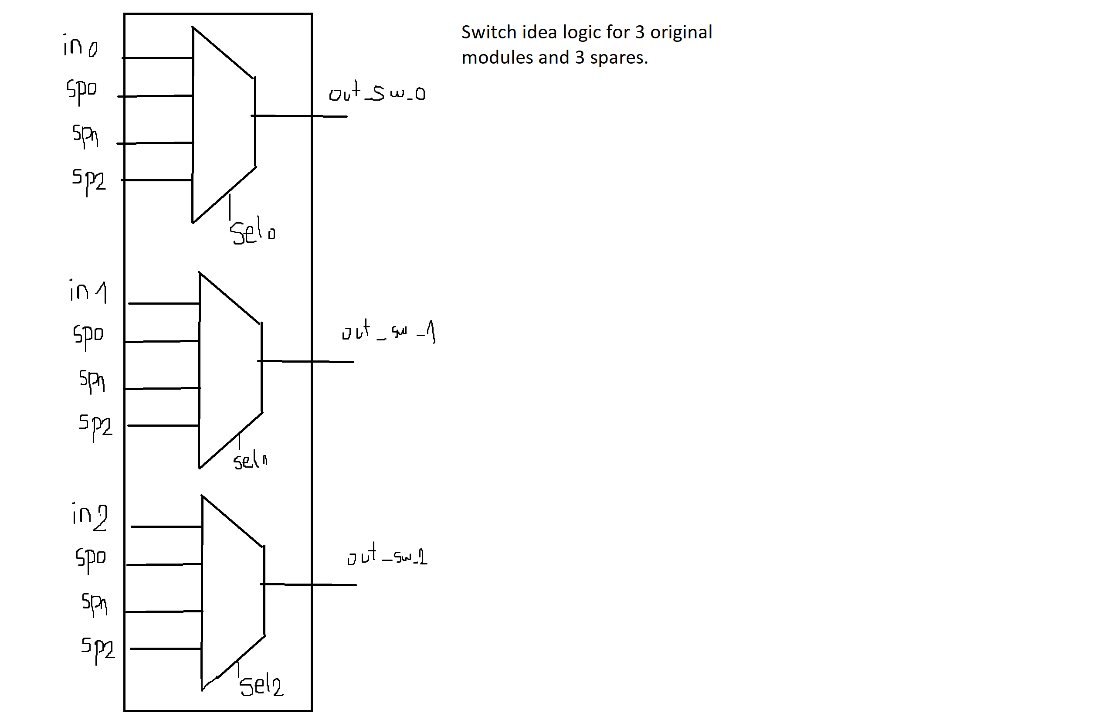
Komparator prima M ulaza kao izlaze iz switch\_logike I izlaz iz finalnog votera. Kroz for petlju se vrsi poredjenje I ukoliko dodje do neslaganja na *i-toj* komponenti, odgovarajuci *i-ti* bit izlaza komparatora se dize na ‘1’ cime se daje informacija switch logici o tome **koja** komponenta ne radi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

* **Implementacija votera**

Ideja za voter je drugacija 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 povecava za 1, I na kraju se u drugom procesu trazi najveca vrednost tog niza. Recimo da se najveca vrednost nalazi na *j-toj* poziciji, to znaci da cemo proslediti *j-ti* signal na ulazu.

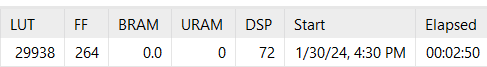
* **Implementacija switch logike**

Switch logika je implementriana 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 selekcioni ulaz svakog dobijamo drugaciji izlaz na jednom od M izlaza. To zavisi sve od izlaza komparatora koji ce, ukoliko se on razlikuje od 0, uticati na povecanje *fail* signala I time I promenu selekcionih ulaza pomenutih multipleksera.



* **Analiza resursa**

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



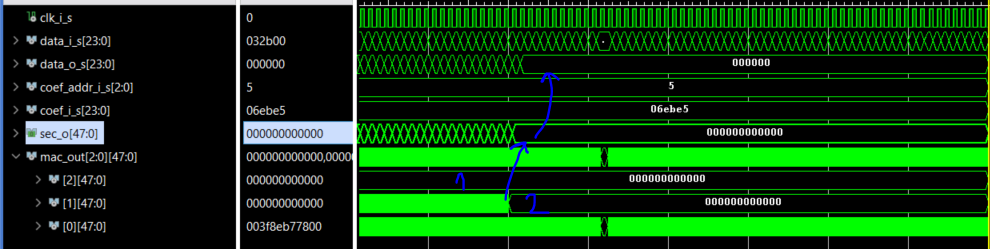


MAC modularity 3

MAC modularity 4

* **Forsiranje za N modularnost**

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*.



* Problemi sa sistemom

Ovaj sistem ima nekoliko problema koje nisam bio u stanju da resim. Kao prvo, u tb za top modul nemam podudaranje sa “expected.txt” fajlom, iz meni nepoznatog razloga. Deluje mi da je sve dobro povezano I smisleno, svaka komponenta individualno radi kako treba (za ovo promeniti bilo koju komponentu da bude top I njen tb I pokrenuti simulaciju). Na sledecoj slici sam forsirao nulu na jednom od izlaza iz votera I imam neko cudno ponasanje selekcionih ulaza, kakvo nemam kada je individualno switch testiran. Ovde prikazujem u sustini da NmodWithSpares isto ne radi kako treba.

