

Projektni zadatak

8x8 Banyan mreža sa provjerom ispravnosti ulaza

Odgovorni nastavnik: R. prof. dr Mirko Škrbić, dipl.ing.el.
Asistent: V. asis. Enio Kaljić, Mr. dipl. ing.

Almin Repeša 17550
Nejra Velić 17313
Medina Hodžić 17350
Semina Nurkić 17522

Tema 5: 8x8 Banyan mreža sa provjerom ispravnosti ulaza

Signali:

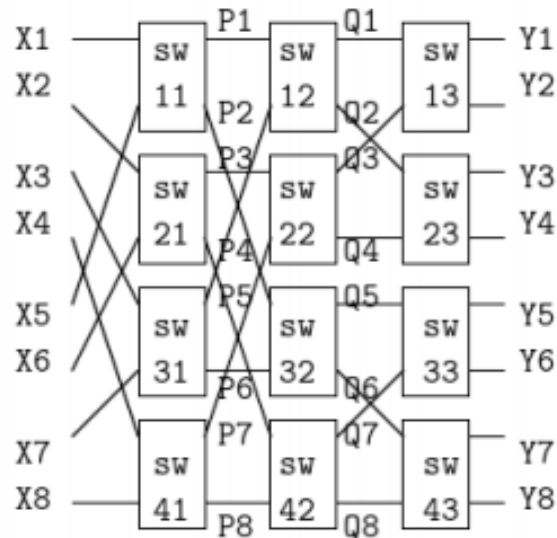
- X1 – ulaz, 4-bitna ćelija,
- ...
- X8 – ulaz, 4-bitna ćelija,
- Y1 – izlaz, 4-bitna ćelija,
- ...
- Y8 – izlaz, 4-bitna ćelija

Prva tri bita iz kodne riječi svakog ulaza označavaju odredišnu adresu, a četvrti bit predstavlja bit pariteta. Na ulazu u komutator potrebno je provjeriti ispravnost kodne riječi, te dozvoliti komutaciju samo ispravnim ćelijama. Potrebno je na konkretnim primjerima analizirati ponašanje komutatora, te simulirati analizirane primjere upotrebom zasebnih VHDL testbench-a. Komutator je potrebno realizirati primjenom tehnika hijerarhijskog modeliranja.

Zadatak:

- Šematski prikaz dizajna komponent/e/i uz kratki opis funkcionalnosti i definiciju ulaznih i izlaznih signala
- Ilustracija principa rada komponent/e/i na više primjera koji pokrivaju sve slučajeve upotrebe
- Realizacija testbench/a/ova za testiranje rada komponent/e/i na jednom ili više primjera koji pokrivaju sve slučajeve upotrebe.
- Verifikacija funkcionalnosti komponent/e/i pomoću verifikacijskog procesa testbench/a/ova i vremenskih dijagrama iz ModelSim-a.

Uvod: Banyan mreža



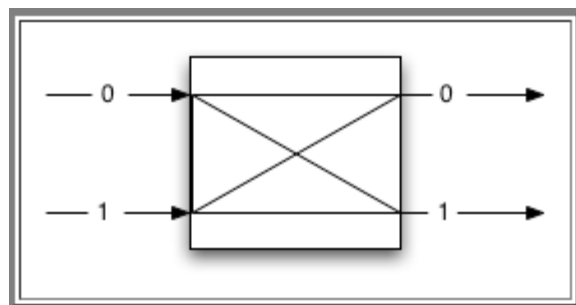
Slika 1. 8x8 Banyan mreža

Na slici 1. data je blok shema 8x8 Banyan komutatora sa naznačenim nazivima instanci komponenti i unutrašnjim signalima.

Složenost Banyan mreža, sa stanovišta putanja i komutacionih elemenata, je reda $N \lg N$. Zbog toga su mnogo pogodnije od crossbar struktura za konstruisanje komutatora velikih dimenzija.

Banyan mreža je interno blokirajuća. Interno blokiranje se javlja ako se više od jednog paketa nadmeće, istovremeno, za isti resurs unutar komutacionog uređaja. U tom slučaju, svi paketi koji učestvuju u nadmetanju, osim jednog, moraju se privremeno uskladištiti u baferne.

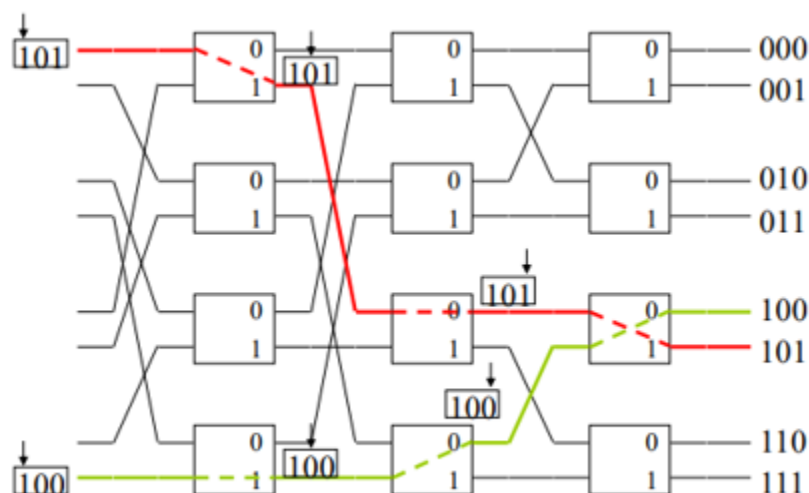
Banyan mreža je implementirana interkonekcijom 2×2 komutacionih mreža u višestrukim i rekurzivnim fazama. 2x2 Banyan komutator ima dva ulaza, 0 i 1, i dva izlaza (slika 2) i zahtjeva $(2/2) \log_2 2 = 1$ komutaciona elementa.



Slika 2. 2x2 Banyan komutator

Svaki paket koji dolazi, u paketu ima zaglavlje koje sadrži jedan bit, koji pokazuje koje je njegovo odredište (0 ili 1). Ako komutator čita bit i ima vrijednost 0, šalje paket na svoj veći izlaz (što je u ovom slučaju 0), te na njegov niži izlaz ako je bit usmjeravanja jedan. Povezivanjem ovih komutacionih elemenata u seriji i paralelno moguće je, dakle, rutirati pakete na komplikovanije načine, u zavisnosti od željenih ruta koje treba uspostaviti.

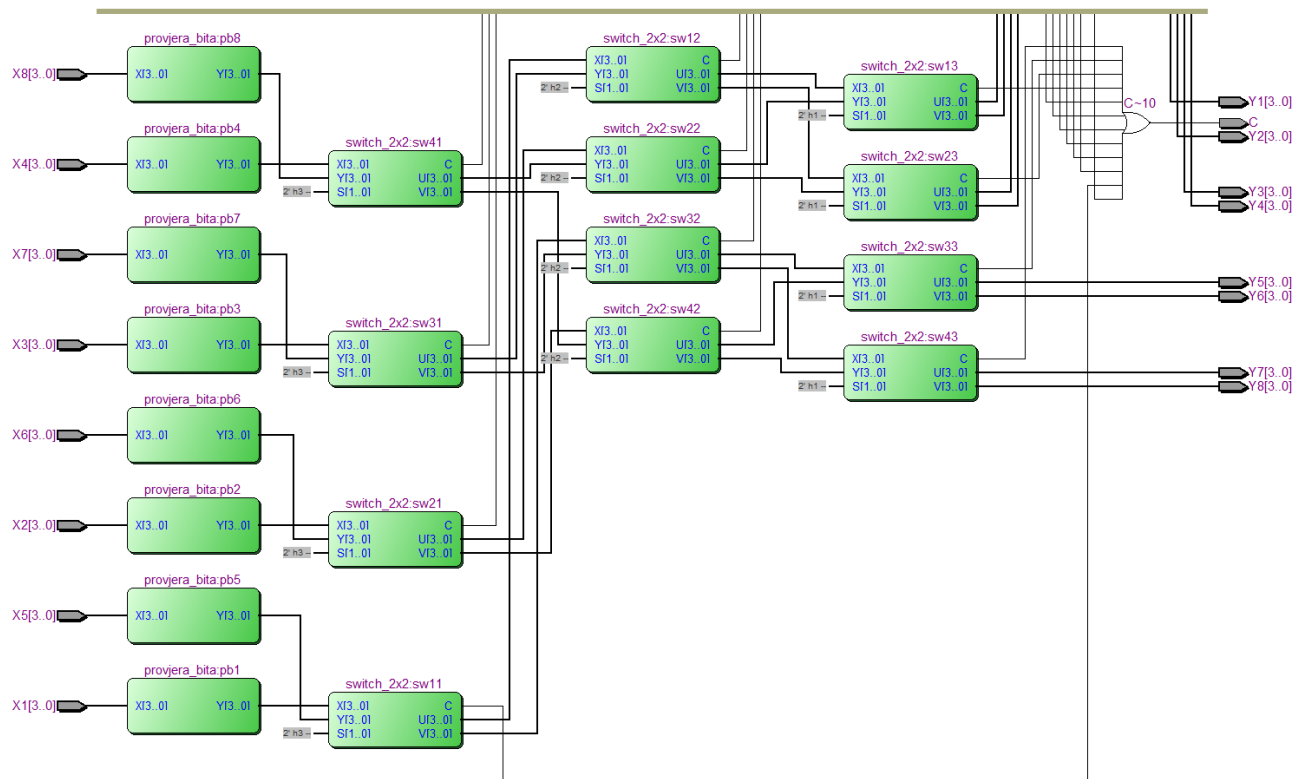
Ilustracija principa rada



Slika 3. Samoproslijeđivanje u Banyan mreži: 0 - gornji izlaz, 1 - donji izlaz komutatora 2x2

Princip rada 8x8 Banyan komutatora prikazan je na slici 2 i na slici 3 u sklopu prosljeđivanja. Ukoliko na ulaz komutatora dolazi vrijednost logičke jedinice, ona se prosljeđuje na donji izlaz, dok se vrijednost logičke nule prosljeđuje na gornji izlaz. Ako se istovremeno šalju dvije kodne riječi te dođe do slanja istog bita (logičke jedinice ili nule) na izlazu iz komutatora dolazi do konflikta.

Prikaz realizovane Banyan 8x8 mreže u VHDL-u



Slika 4. RTL prikaz mreže

Na slici 4 je prikaz Banyan 8x8 mreže sa provjerom ispravnosti ulaza. Kao što se vidi sa slike, funkcija za provjeru bita je postavljena prije same mreže, jer se neispravni ulazi odbacuju i ne puštaju kroz mrežu. Samo ispravne kodne riječi se propuštaju dalje u prva 4 switch-a. Dalje se one komutiraju po pravilima objašnjenim prije u radu Banyan mreže.

Logika izrade zadatka:

Prilikom izrade projektnog zadatka slijedili smo logiku Banyan 8x8 mreže kao što je opisano u uvodnom dijelu. Pri čemu je potrebno ispitati ispravnost ulaza u komutator.

Ispravnost kodne riječi koja ulazi u komutator je ispitana uz pomoć paritetnog bita. Što znači da ako riječ koja ulazi u komutator ima paran broj jedinica predstavlja ispravnu kodnu riječ, analogno tome riječ koja ima neparan broj jedinica je neispravna kodna riječ.

Na osnovu ove logike, u VHDL-u se ispitivanje ispravnosti vrši u fajlu koji smo nazvali „provjera bita“ gdje je xor-ovan svaki bit pojedinačne kodne riječi te ako je rezultat tog xor-a 0, tada je kodna riječ ispravna te se propušta dalje kroz komutator. U suprotnom se kodna riječ blokira i ne ide kroz mrežu, tada se na izlazu pojavi prethodni izlaz, koji je izazvan prethodno odabranom ispravnom kodnom riječi.

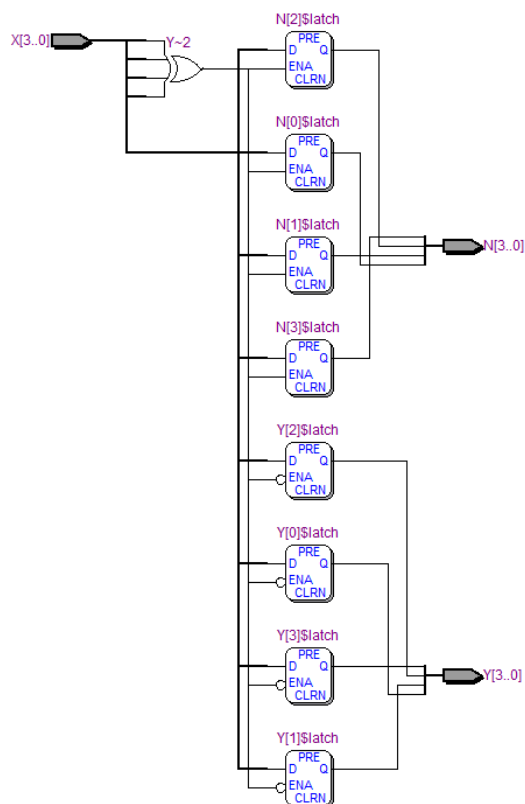
Pored ulaznih i izlaznih signala u VHDL-u potrebna su nam i signal C, koji predstavlja signal za provjeru da li je došlo do konflikta, kao i signal S koji određuje bit koji se poredi, pri čemu se provjera vrši sa desna na lijevo. Tako npr. ako je $S=10$ poređenje dvije kodne riječi se vrši na osnovu drugog bita sa desne strane.

Kada se provjerom bita ustanovi da je kodna riječ ispravna ona ide dalje kroz mrežu. Biti se šalju jedan po jedan, pri čemu se paritetni bit više ne gleda. Ukoliko je bit jednak logičkoj jedinici šalje se na donji izlaz, u suprotnom se šalje na gornji izlaz. Kao što je i prethodno objašnjeno za rad Banyan komutatora.

U VHDL-u smo dali nekoliko primjera, ukoliko su sve kodne riječi ispravne, na izlazu dobijemo očekivane riječi koje su poredane u skladu sa radom cjelokupne mreže. Dalje, ukoliko se desio konflikt na nekom od bita, te kodne riječi se ne prosljeđuju na izlaze, a ukoliko imamo neispravnu kodnu riječ ne prenosi se kroz mrežu te se na izlazu pojavljuje prethodno dobiveni izlaz od ispravne riječi.

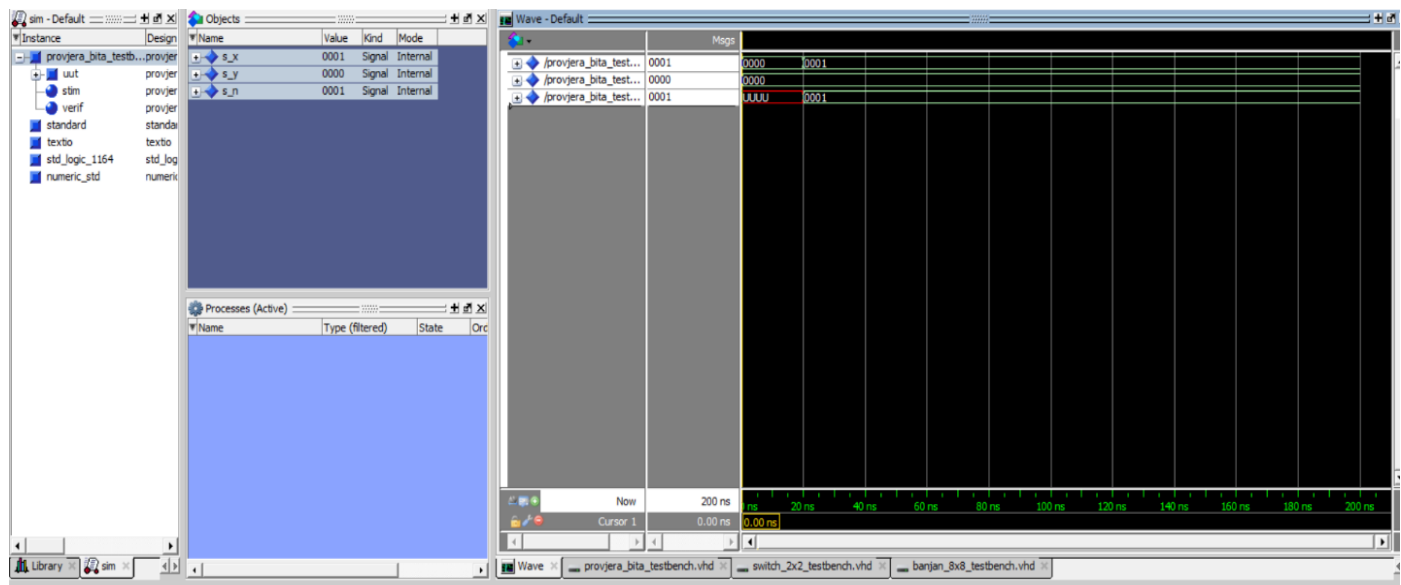
Realizovan je i testbench kojim smo potvrdili ispravnost VHDL programa.

Analogno tome, ako je rezultat prethodne operacije 1 tada znamo da je broj jedinica u kodnoj riječi neparan, desila se greška na jednom bitu, sama kodna riječ se vodi na neispravan izlaz (u kodu označen kao izlaz N). Nakon toga, urađen je testbench kojim smo provjerili ispravnost napisanog koda. Rezultat je dat na slici 6.



Slika 5. Prikaz sklopa za provjeru bita

Slika 5 prikazuje na koji način se provjerava ispravnost kodnih riječi. Postupak je odrađen preko XOR kola, a detaljan način izrade i logika je navedeno u sljedećem poglavlju „Logika izrade zadatka“.



Slika 6. Demonstracija rada VHDL koda za provjeru ulaza

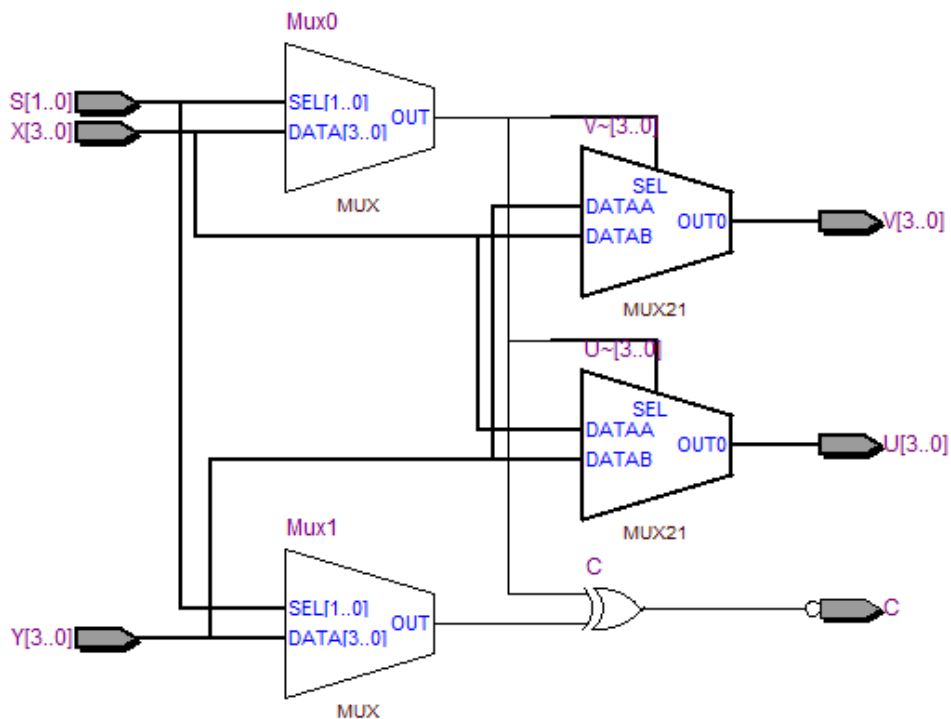
Na slici 6 prikazani su izlazi samo iz prve funkcije provjera izlaza. Vidimo da smo dobili očekivani izlaz, odnosno za ispravnu kodnu riječ imamo ispravan izlaz, dok je u slučaju neispravne kodne riječi izlaz jednak prethodnom (dobiveni za ispravi ulaz).

Nakon implementirane funkcije za provjeru izlaza slijedi standardna Banyan 8x8 mreža. Ona se sastoji iz 12 switch-eva, kako nam treba mreža sa 8 ulaza i 8 izlaza slijedi da je:

$$\text{Broj switch - eva} = \frac{N}{2} \lg N = 4 * \lg 8 = 4 * 3 = 12$$

koji su realizovani kao zasebne komponente i to na sljedeći način:

Prije svega switch treba da se sastoji iz 2 ulaza i 2 izlaza. Definisana su dva ulaza u switch kao četverobitne kodne riječi, analogno tome su definisana i dva izlaza iz switcha. Pored signala ulaza i izlaza potrebna su nam još dva signala, signal za indikator konflikta kao i signal koji će određivati to po kojem bitu se vrši poređenje dvije kodne riječi. Potrebno je još napomenuti i to da se paritetni bit koji nam je bio potreban prije samog switch-a više ne uzima u obzir.

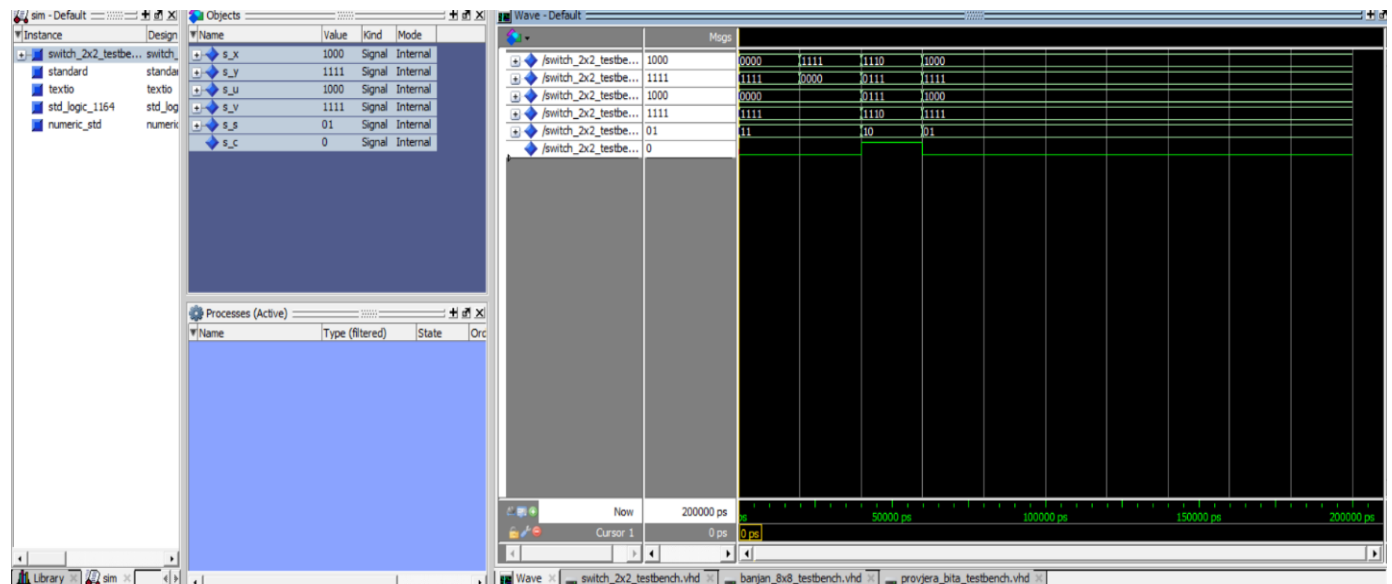


Slika 7. Prikaz sklopa switch-a

Signal po kojem se porede ulazi realizovan je kao dvobitni, dok je signal za indiciranje konflikta STD_LOGIC.

Kada dva signala dođu na ulaz switch-a porede si biti dvaju signala pojedinačno. Ito ako na ulaz dolazi logička 0 ona se prosljeđuje na gornji izlaz, dok se logička jedinica prosljeđuje na donji izlaz. U ovom slučaju, dakle ako su dva različita logička stanja došla na switch ne dolazi do konflikta te je C=0. U suprotnom, ako se na ulazu pojave dvije logičke nule ili dvije logičke jedinice tada će na isti izlaz pokušati izaći oba ulaza te dolazi do konflikta te je C=1.

Realizovan je testbench, te je prikaz signala pokazan na slici 8.



Slika 8. Demonstracija rada switch-a u VHDL-u

Na slici 8 se vidi da ćemo dobiti očekivane izlaze u slučaju bez konflikta, te će i signal konflikta biti jednak 0, u suprotnom je on 1. U samom testbench-u signal S="11".

Pošto smo definisali funkciju za provjeru ispravnosti ulaza kao i funkciju switcha slijedi spajanje svega u jednu cjelinu i realizovanje 8x8 Banyan mreže.

Prije svega definisano je 8 signala ulaza kao četverobitnih vektora analogno tome definirano je 8 signala izlaza. Pošto smo već implementirali switch, u dijelu programa COMPONENT, pozivamo tu komponentu tako da definisane ulaze možemo vezati na ulaze switcha.

Dalje se definišu signali na ulazima kao i signali između samih switch-eva, odnosno unutar same Banyan mreže.

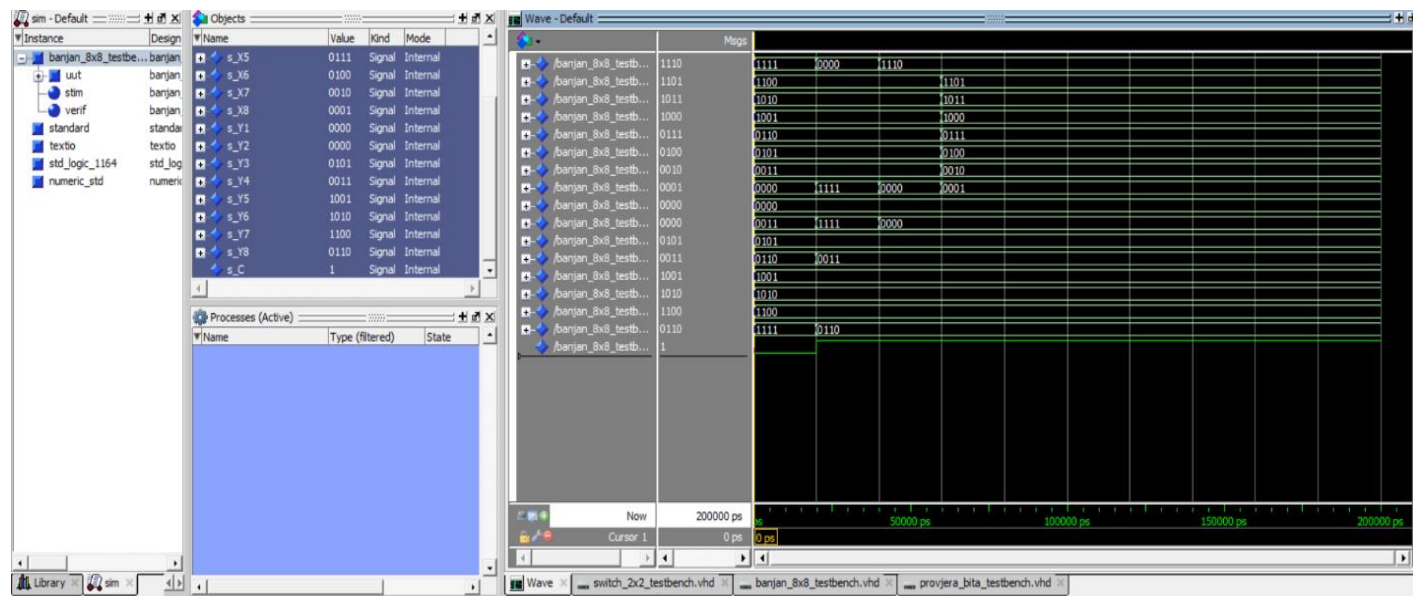
Pored toga, potrebno je još definirati 12 signala konflikta, koji će biti na izlazima svakog od switch-eva te provjeravati da li je došlo do konflikta na njihovom izlazu.

Nakon definiranja svih potrebnih signala kao i komponenata switch-a, provjeravamo ispravnost svake od ulaznih riječi tako što ulaze u switch povezujemo sa implementiranim kodom koji vrši ovu funkciju.

Nakon toga, svaku ispravnu kodnu riječ dovodimo na ulaz switch-a tako da nam je za prvi stepen potrebno 4 switcha za 8 ulaznih riječi, pri čemu se gleda jedan po jedan bit kodne riječi. Analogno tome za sljedeći bit se dodaje još 4 switcha u mrežu koji se povezuju sa prethodnim, te se njihovi izlazi vežu na ulaze novododanih switch-eva. Analogno tome i za zadnji stepen, odnosno za dodana zadnja 4 switch-a.

Nakon ove procedure, vrši se provjera da li je igdje u mreži došlo do konflikta, te se provjerava 12 signala konflikta definisanih prethodno.

Nakon same realizacije, realizovan je i testbench za provjeru rada samog programa, koji je dao očekivane izlaze.



Slika 9. Prikaz izlaza za ulaze određene u VHDL kodu

Na slici 9. prikazani su izlazi, odnosno krajnji rezultat realizovane mreže. Za jedan primjer koji smo postavili u našem testbench-u, a to je da su nam ulazne kodne riječi sve ispravne, dobili smo očekivani izlaz. Također prikazani su primjeri za slučaj kada dođe do konflikta za sve ispravne kodne riječi, kao i kada su kodne riječi neispravne, te i kada dođe do konflikta pri neispravnim kodnim riječima.

Zaključak

Tokom realizacije projekta i rada na njemu mnogo bolje smo se upoznali sa radom u VHDL-u. Osim implementacije Banyan mreže, testiranja različitih slučajeva i provjere ispravnosti ulaznih kodnih riječi, uočili smo značaj paketskih komutacija i komutacionih mreža. Banyan mreža je jedna od mreža u kojima se ostvaruje prostorna komutacija. Mana Banyan mreža je ta što njene performanse brzo opadaju povećanjem dimenzija. A prednost je što se bazira na topologiji stabla i ima mnogo manje prespojnih tačaka nego u crossbar-u. Banyan komutatori su jednostavni elementi, skalabilni, vrše paralelno prosljeđivanje i rade istom brzinom. Konflikti koji nastaju u mreži mogu biti interni i eksterni. Postoji još mnogo mreža koje koriste Banyan komutatore kao što su kopirajuće mreže, proširene Banyan mreže, višestruke mreže u paraleli, produžene Banyan mreže, mreže sa recirkulacijom itd.

Literatura

[1]Predavanja i slajdovi sa web stranice predmeta Komutacioni sistemi

<https://c2.etf.unsa.ba/course/view.php?id=135#section-0>

[2] Uvod u VHDL <https://c2.etf.unsa.ba/mod/resource/view.php?id=50071>

[3] M. Škrbić, *Čvorišta u telekomunikacionoj mreži*, Elektrotehnički fakultet u Sarajevu, 2007.