

LABORATORIJSKI ZADATAK 1

Kombinacione mreže

Potrebno predznanje

- Urađena pokazna vežba 2
- Standardne kombinacione mreže – pun sabirač, dekodler, koder, multiplexer, demultiplexer

Šta će biti naučeno tokom izrade vežbe?

Nakon urađene vežbe:

- Primenićete znanje standardnih kombinacionih mreža i iskoristiti ih za projektovanje složenih kombinacionih mreža koje sadrže više standardnih komponenti,
- Savladaćete kompletan ciklus projektovanja digitalnih sistema za FPGA – od specifikacije, preko logičke šeme, VHDL opisa, simulacije, do implementacije,
- Imaćete iskustvo u projektovanju kombinacionih mreža koje rešavaju uslovnu logiku.

Apstrakt i motivacija

Nakon što smo naučili osnove projektovanja kombinacionih mreža i njihovih opisa logičkom šemom i VHDL jezikom, vreme je da stečeno znanje primenimo u rešavanju problema projektovanja složenijih kombinacionih mreža od onih koje smo do sada videli. U ovoj vežbi ćete, kroz niz koraka, projektovati kombinacionu mrežu koja implementira uslovnu logiku. Ova mreža će, u zavisnosti od selekcionih bita, računati različite funkcije ulaznih signala. Ova vežba će takođe pokazati kako se sistem može proširivati dodajući komponente u sistem. Za početak projektovaćete veoma jednostavnu kombinacionu mrežu, pun sabirač, i kroz ovaj zadatak ponoviti korake simulacije i implementacije sistema u Xilinx ISE alatu. Nakon toga ćete proširiti ovaj sistem sa još nekoliko komponenata, a krajnji rezultat vežbe će biti kombinacioni digitalni sistem sastavljen od nekoliko standardnih kombinacionih mreža.

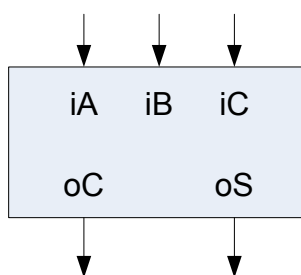
Šta treba doneti na termin laboratorijske vežbe?

- Tražene istinitosne tablice, logičke i blok šeme na papiru ili računaru.
- VHDL opis **krajnjeg koraka** u realizaciji sistema (međukoraci nisu potrebni) – krajnji korak je poslednji korak koji uspete da realizujete.
- Testbench za krajnji korak sistema.
- Generisanu *.bit* datoteku za konfigurisanje E2LP platforme za krajnji korak sistema.

ZADACI

1. Trobitni sabirač – logička šema

Pun sabirač (slika 1-1) je standardna kombinatorna mreža sa 3 ulaza i 2 izlaza. Ulazi su jednobitni celi brojevi koji su međusobno ravnopravni (dva predstavljaju bitove operanada, a treći predstavlja ulazni prenos). Zadatak punog sabirača je da sabere tri bita sa ulaza i na izlazu prikaže njihovu sumu. Suma tri binarna broja može uzeti vrednosti od 0 do 3, tako da izlaz punog sabirača sadrži 2 bita – jedan predstavlja sumu, a jedan predstavlja prenos. Izlazi posmatrani kao dvobitni broj, takav da je prenos na višoj poziciji od sume, predstavljaju sumu tri jednobitna ulaza.



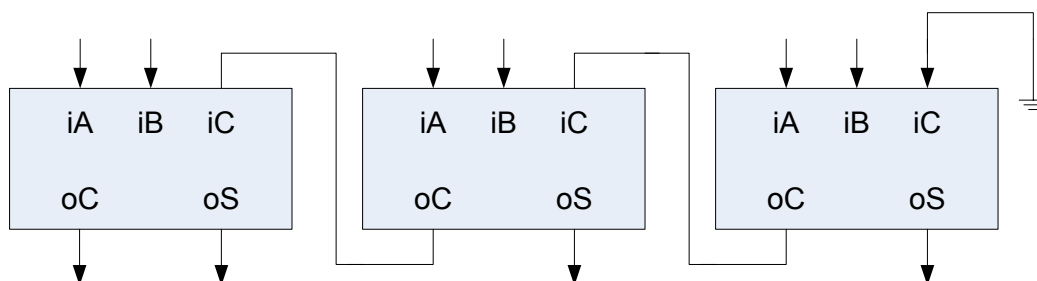
Slika 1-1. Pun sabirač

Vaš zadatak je da projektujete pun sabirač prateći sve korake projektovanja koje smo učili u prethodnih vežbama:

1. Napisati istinitosnu tablicu punog sabirača i napisati Bulove funkcije svakog od izlaza ove kombinatorne mreže.
2. Na papiru nacrtati logičku šemu punog sabirača (na nivou logičkih kola).
3. Na računaru, u Xilinx ISE alatu, nacrtati logičku šemu punog sabirača (schematic).
4. Napisati testbench za proveru punog sabirača prolazeći kroz **sve** mogućnosti ulaza.
5. Simulirati pun sabirač i proveriti mu funkcionalnost.

Proveriti funkcionalnost punog sabirača u simulaciji pre nego što pređete na sledeći korak.

Nakon što smo (uspešno) napravili jednobitni pun sabirač, sistem ćemo proširiti tako da predstavlja trobitni sabirač kao na slici 2-1.



Slika 2-1. Trobitni sabirač

Trobitni sabirač ima sledeće prolaze:

- Ulazi – trobitni operand iA i trobitni operand iB
- Izlazi – trobitni zbir oS i prenos oC.

Koristeći Bulove funkcije izvedene u prethodnom zadatku, proširiti sistem sa jednobitnog na trobitni sabirač. Uraditi sledeće:

1. Dopuniti šemu punog sabirača tako da dobijemo trobitni sabirač (replicirati pun sabirač tri puta i povezati na odgovarajući način).
2. Napisati test bench za trobitni sabirač prolazeći kroz nekoliko slučajeva sabiranja.
3. Simulirati sistem i proveriti mu funkcionalnost.

Ovaj korak nećemo implementirati na E2LP platformi, tako da ne morate prolaziti kroz postupak dodeljivanja pinova ulazima i izlazima i formiranja .bit datoteke.

2. Uslovna logika paljenja dioda

U nastavku vežbe napravite novi projekat u kome ćemo realizovati drugi sistem. Ovaj sistem će uslovno paliti diode na E2LP platformi u zavisnosti od stanja srednjeg tastera.

Sistem treba da ima sledeće prolaze:

- Ulazi – osmobarbitni iSW [7:0] kojeg ćemo povezati na sve prekidače, jednobitni iSEL kojeg ćemo povezati na srednji taster,
- Izlazi – osmobarbitni oLED [7:0] kojeg ćemo povezati na LED diode.

Sistem treba da se ponaša na sledeći način:

- Ako je taster iSEL *pritisnut* (taster na E2LP ima vrednost 0 kada je pritisnut), diode treba stalno da budu upaljenje,
- Ako je taster iSEL *pušten*, na izlazu treba da se pojavi sledeće:
 - diode LED3 – LED0 treba da sadrže zbir trobitnih brojeva prikazanih na prekidačima SW5 – SW3 (jedan broj) i SW2 – SW0 (drugi broj),
 - diode LED6 – LED4 treba da sadrže indeks jednog od upaljenih prekidača (od 0 do 7); ukoliko je upaljeno više od jednog prekidača izabrati onaj sa najvećim indeksom,
 - dioda LED7 treba da sadrži vrednost 1 ako je bar jedan prekidač upaljen, a vrednost 0 u suprotnom.

Za ovaj zadatak uraditi sledeće:

- Nacrtati blok šemu sistema (na nivou standardnih komponenti – ne mora biti na nivou logičkih kola).
- Opisati u VHDL jeziku sistem koristeći proizvoljne VHDL konstrukcije za opis kombinacionih mreža – uslovne dodele ili procese (ili ukoliko baš želite, logička kola).
- Napisati testbench za sistem, proveravajući karakteristične slučajeve.
- Simulirati i implementirati sistem za E2LP platformu.

Dodeliti pinove prema tabeli 3-1.

Tabela 3-1. Dodela pinova

| Prolaz | Smer | Komponenta na E2LP platformi |
|-----------|------|------------------------------|
| iSW[7:0] | in | SW7-SW0 |
| iSEL | in | JOY2 |
| oLED[7:0] | out | LED7-LED0 |

Preporučujemo korišćenje uslovnih dodela i procesa u rešavanju ovog (a i svih kasnijih zadataka) – rešavanje na nivou logičkih kola je veoma mukotrpno i dugotrajno već kod ovog primera.

OPŠTE NAPOMENE

Prilikom crtanja šeme na papiru ili računaru, koristiti blokove za standardne kombinacione mreže – nema potrebe da poznate komponente crtate na nivou logičkih kola. Bitno je da logička šema ispravno opiše logiku kombinacione mreže, nije neophodno spuštati se u detalje i crtati svako logičko kolo. (Ova napomena ne važi samo za prvi zadatak ove vežbe jer se u njemu traži da šemu prikazete na nivou logičkih kola)

Implementaciju sistema izvršiti za poslednji urađen korak zadatka. Nije neophodno imati urađenu implementaciju za svaki korak. Simulaciju treba raditi posle svakog koraka, jer na sledeći korak treba preći jedino ukoliko je prethodni funkcionalno proveren.

Na termin vežbe doneti **jedan** VHDL opis i **jedan** testbench (nije neophodno donositi ni opis ni testbench međukoraka) za svaki zadatak. Međukoraci su tu da bi vama olakšali put ka kompletnom sistemu i omogućili da vaš sistem projektujete inkrementalno – počev od jednostavnijih komponentata ka složenijim.

Broj poena koje možete osvojiti na vežbama zavisi od koraka do kojeg ste doveli sistem – svaki zadatak nosi 1 poen (do maksimalno 2), a viši poeni (3, 4) se mogu dobiti izradom modifikacija i malih dopuna vašeg sistema koje ćete dobiti na terminu vežbe. Dodatni zadatak na terminu vežbe vam može doneti i dodatni, peti poen.

ZAKLJUČAK

U ovoj vežbi ste napravili složeni kombinacioni sistem koji se sastoji od bar 2 tipa standardnih kombinacionih mreža. Ova vežba je imala za cilj da vas uvežba da projektujete složene kombinacione mreže, ali i da vam pokaže kako se sistem može projektovati inkrementalno, počev od jednostavnijih delova sistema, inkrementalno dodajući komponente sa ciljem projektovanja složenog sistema.