

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Digitalna logika

Laboratorijske vježbe korištenjem sklopovskih pomagala

Upute za 3. laboratorijsku vježbu

Marko Zec

Studen 2017.

1 Uvod

VHDL (*Very high speed integrated circuits hardware description language*) je jezik za opis digitalnih sklopova i simulacijskih ispitnih okruženja za digitalne sklopove. U izvođenju ove vježbe koristiti će se podskup jezika VHDL koji omogućuje opis funkcionalnih digitalnih sklopova (tzv. *synthesizable VHDL*).

Prilikom upoznavanja sa strukturom, sintaksom i semantikom VHDL-a izrazito je bitno cijelo vrijeme imati u vidu da se **ne radi o programskom jeziku**, već o jeziku kojim se **opisuje struktura i ponašanje skupa međusobno povezanih digitalnih komponenti koje rade paralelno**, za razliku od programskih jezika kojima se formuliraju pravila za vremenski slijedno izvršavanje naredbi na računalu.

Glavni cilj vježbe je upoznavanje studenata s jezikom VHDL kroz praktične primjere projektiranja i ispitivanja rada jednostavnih kombinacijskih sklopova. U prvom koraku potrebno je korištenjem jezika VHDL opisati sklop koji je strukturno i funkcijski potpuno jednak sklopu kojeg je u 2. laboratorijskoj vježbi trebalo specificirati crtanjem sheme. U drugom dijelu vježbe sklop se proširuje instanciranjem dodatnih modula.

Korištenjem već gotovog VHDL opisa ispituje se ponašanje temeljnih sekvencijskih elemenata (bistabila) u trećem dijelu vježbe.

2 Jednostavni kombinacijski sklop (1 bod)

Vaš je zadatak projektirati i ispitati sklop koji će 10 različitih kombinacija pritisaka na tipke razvojne pločice (`btn_up`, `btn_down`, `btn_left`, `btn_right` i `btn_center`) preslikati u ASCII kodove prema unaprijed zadanoj tablici. Izlaze iz vlastite kombinacijske mreže povežite na ulaz već gotovog modula `"serial_tx"` koji 8-bitne podatke s ulaza slijedno (bit po bit) šalje putem USB sučelja prema računalu, na kojem je pomoću odgovarajućeg programa moguće pratiti odziv sklopa na pobudu. Izlaze vlastite kombinacijske mreže potrebno je dovesti i na LED indikatore kako bi mogli vizualno provjeriti ispravnost generiranja zadanih ASCII kodova. Rad modula `"serial_tx"` upravljan je signalom takta, pa na ulaz modula `"clk"` treba dovesti vanjski signal takta `"clk_25m"`.

Odaberite slijed od 8 ASCII znakova na način da prva četiri znaka slijeda sačinjavaju prva četiri slova Vašeg imena, a druga četiri znaka slijeda tvore prva četiri znaka Vašeg prezimena. Hrvatski dijakritički znakovi zamjenjuju se slovima engleske abecede (Č i Ć u C, Đ i DŽ u D, Š u S te Ž u Z). Početna slova imena i prezimena trebaju biti kodirana velikim slovima, dok za ostatak kodnih riječi treba koristiti mala slova (ASCII znakove). U slučaju da je ime ili prezime kraće od četiri znaka umjesto raspodjele znakova 4+4 potrebno je odabrati takvu raspodjelu da se popuni svih 8 znakova.

Za sve ostale moguće kombinacije tipki (one koje nisu obuhvaćene tablicom) rad kombinacijskog sklopa nije specificiran, te je dozvoljeno da kombinacijski sklop na izlazu generira bilo kakvu kodnu riječ.

Popunite tablicu kombinacija ulaznih signala (tipki) i pripadajućih izlaznih kodnih riječi u dekadskom i binarnom zapisu. **Na laboratorijsku vježbu trebate doći s ispravno popunjenom tablicom**, te po mogućnosti s izvedbom sklopa koja odgovara funkcijskim zahtjevima prvog dijela zadatka. Ako ste sačuvali pripremu s 2. laboratorijske vježbe, tablicu ne trebate ponovo popunjavati, nego je dovoljno donijeti pripremu s prethodne vježbe.

Ulaz (tipke): down left center up right	Izlaz: znak	Izlaz: ASCII kod (heksadekadski)	Izlaz - binarno							
			7	6	5	4	3	2	1	0
-	NULL	00								
down	NULL	00								
left										
center										
up										
right										
down AND left										
down AND center										
down AND up										
down AND right										

2.1 Sinteza i ispitivanje rada sklopa

Stvorite novi prazni direktorij na disku, te u njega pohranite slijedeće datoteke koje možete dohvatiti s web sjedišta laboratorijskih vježbi:

- `ulx2s.lpf`
- `slova.vhd`
- `serial_tx.vhd`

Pokrenite razvojnu okolinu Lattice Diamond, te stvorite novi projekt. Za radni direktorij projekta odaberite upravo stvoreni direktorij u kojem se nalaze datoteke dohvaćene iz repozitorija. FPGA sklop ugrađen na pločicu ULX2S je Lattice XP2-5E LFXP2-5E u TQFP-144 kućištu, o čemu treba voditi računa prilikom odabira ciljanog FPGA sklopa.

Već gotov predložak "`slova.vhd`" možete koristiti za izradu vlastite vježbe. Multipleksore i ostale komponente ne treba izvoditi kao zasebne module, nego ih se može opisati korištenjem konkurentnih izraza `with-select` ili `when-else` direktno u bloku `architecture`. Nije potrebno ni preporučljivo koristiti blokove tipa `process`.

Primjer opisa 32-ulaznog multipleksora s podatkovnim ulazima širine 7 bita, na kojem su na 10 ulaza dovedeno konstante, dok su preostali ulazi nespecificirani:

```
--
-- Visebitni multipleksor 7 * 32 -> 7 s nepotpuno specificiranim izlazima
-- Signal za odabir (select) je "btns"
--
with btns select
code <=
    "0000000" when "00000",
    "0000000" when "10000",
    "0111100" when "01000",
    "0111110" when "00100",
    "1000000" when "00010",
    "1000001" when "00001",
    "1000011" when "11000",
    "1000101" when "10100",
    "1000111" when "10010",
    "1001000" when "10001",
    "-----" when others ; -- don't care
```

Programirajte FPGA sklop sintetiziranom konfiguracijom te ispitajte njegov rad za svih 10 definiranih ulaznih kombinacija pritisnutih tipki promatranjem odziva na LED indikatorima, te praćenjem ispisa znakova na zaslonu programa za emulaciju asinkronog terminala (ujprog -t).

Usporedite kompleksnost rješenja u VHDL-u sa shematski opisanom implementacijom istog sklopa iz 2. laboratorijske vježbe.

3 Proširenje sklopa (1 bod)

Konstruirajte modul "brojke" koji će različite kombinacije pritisaka na tipke razvojne pločice preslikati u ASCII kodove zadnjih osam znamenaka Vašeg JMBAG identifikatora prema unaprijed zadanoj tablici. **Na laboratorijsku vježbu trebate doći s već popunjenom tablicom!**

Ulaz (tipke): down left center up right	Izlaz: znamenka JMBAG-a	Izlaz: ASCII kod (heksadekadski)	Izlaz - binarno							
			7	6	5	4	3	2	1	0
-	NULL	00								
down	NULL	00								
left										
center										
up										
right										
down AND left										
down AND center										
down AND up										
down AND right										

Sučelje modula "brojke" deklarirajte na slijedeći način:

```
entity brojke is
  port (
    xy_left, xy_right, xy_up, xy_down, xy_center: in std_logic; -- ulazi
    xy_code: out std_logic_vector(7 downto 0) -- izlaz
  );
end brojke;
```

Prefiks "xy" u nazivima signala zamijenite svojim inicijalima. Ovakvom deklaracijom definirano je ime modula (brojke), te u bloku port nazivi (xy_left, xy_right, xy_up, xy_down, xy_center) i tip (std_logic) ulaznih signala (in), te naziv i tip izlaznog signala (xy_code). Uočimo kako ovakva deklaracija sučelja, baš kao ni "crna kutija" koja simbolizira modul "brojke" u shematskom prikazu, ni na koji način ne određuje njegovu funkciju ili ponašanje, već samo određuje na koji se način modul može povezivati s drugim komponentama.

Proširite sklop "slova" na način da u njemu instancirate modul "brojke". Sučelje sklopa proširite četverobitnim vanjskim ulaznim signalom "sw" koji je priključen na četveropolni DIP prekidač na razvojnoj pločici. Zavisno od položaja prekidača sw(0),

pritisak na tipke razvojne pločice treba rezultirati prikazom znakova Vašeg imena i prezimena, odnosno JMBAG identifikatora, na zaslonu programa za emulaciju asinkronog terminala (ujprog -t), te prikazom odgovarajućeg ASCII koda na LED indikatorima razvojne pločice.

U prazni okvir na ovoj stranici ucrtajte blok-shemu proširenog sklopa "slova". U shemi moraju biti vidljivi svi vanjski priključci, dva multipleksora, instance modula "brojke" i "serial_tx", te svi signali koji ih međusobno povezuju. Module "brojke" i "serial_tx" ucrtajte kao "crne kutije" odnosno kao prazne pravokutnike, bez rasčlambe njihove unutarnje strukture, ali na njima obavezno označite nazive njihovih ulaznih odnosno izlaznih priključaka. **Na laboratorijsku vježbu trebate doći s već nacrtanom blok-shemom!**



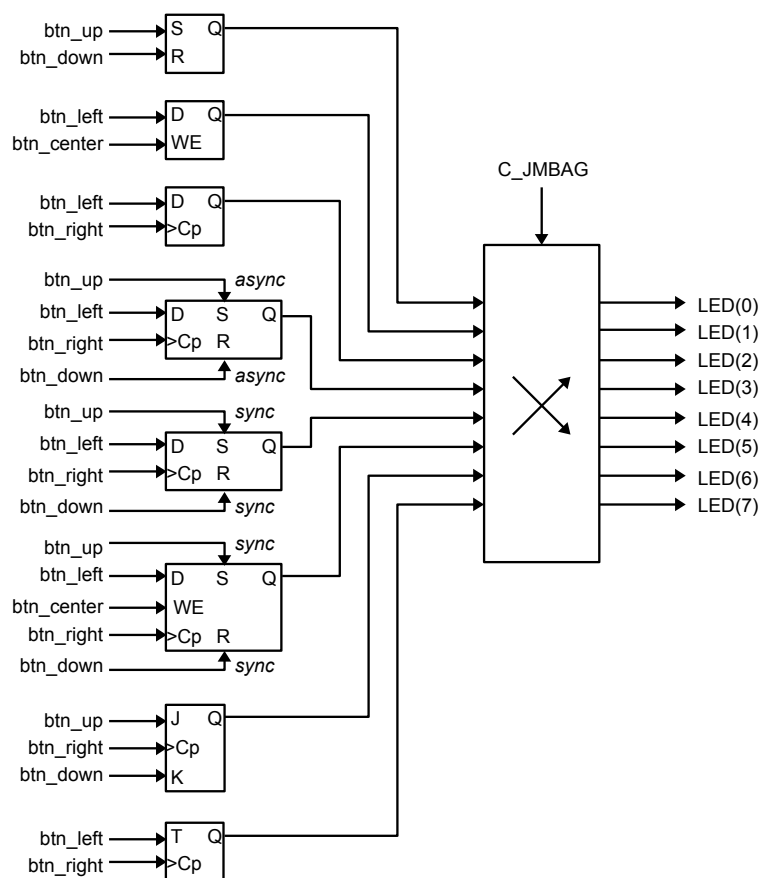
Nakon što ste isitali rad sklopa, u sustav Ferko trebate prenijeti (upload) slijedeće datoteke:

- slova.vhd
- brojke.vhd
- lab3.jed

4 Identifikacija bistabila (1 bod)

Već gotov ispitni sklop nazvan `pogodi_bistabil` povezuje izlaze iz osam različitih tipova bistabila s osam LED indikatora putem programirljivog prospojnog sklopa.

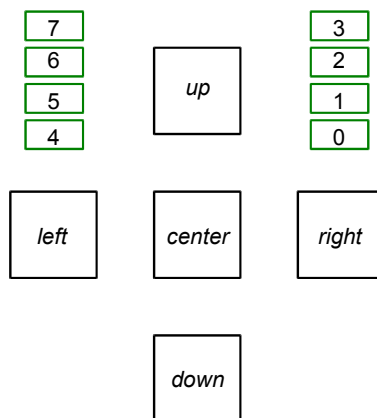
Prospojni sklop svaki od svojih osam ulaza prospaja na točno jedan od svojih osam izlaza, na način koji ovisi o studentovom JMBAG identifikatoru. Blok-shema ispitnog sklopa prikazana je na slici 1.



Slika 1: blok-shema ispitnog sklopa

S web sjedišta laboratorijskih vježbi dohvatite opis ispitnog sklopa pogodi_bistabil.vhd. U deklaraciji sučelja sklopa **podesite vrijednost generic parametra c_jmbag u skladu s vlastitim JMBAG identifikatorom**, sintetizirajte sklop, te dobivenom konfiguracijskom datotekom programirajte svoju razvojnu pločicu. Osim podešavanja vrijednosti generic parametra c_jmbag, u VHDL opisu ispitnog sklopa **nije potrebno ni dopušteno raditi bilo kakve druge preinake!**

Ulazima bistabila ispitnog sklopa upravlja se putem tipki btn_up, btn_down, btn_left, btn_center i btn_right, u skladu s blok-shemom prikazanom na slici 1. Raspored tipki i LED indikatora na razvojnoj pločici prikazan je na slici 2.



Slika 2: raspored i numeracija LED indikatora na razvojnoj pločici

Vaš je zadatak identificirati bistabile promatranjem njihovog odziva (izlaza vidljivih na LED indikatorima) na odgovarajuću pobudu, koju morate sami osmisliti. Kako je za svakog studenta konfiguracija prospojnog sklopa drugačija (ukupno je 8! odnosno 40320 mogućih konfiguracija prospojnog sklopa), vrlo je vjerojatno da će izlaziti iz bistabila kod većine studenta biti prospojeni na LED indikatore na različit način. Prilikom prepoznavanja bistabila odnosno načina prospajanja njihovih izlaza osim korištenja blok-sheme sa slike 1 možete potražiti i proučiti i opise bistabila u datoteci `pogodi_bistabil.vhd`.

Rezultate opažanja rada bistabila potrebno je u **točno propisanom formatu** zapisati u datoteku `pogodi_bistabil.txt`, pri čemu treba voditi računa da su pozicije LED indikatora numerirane brojevima u rasponu od 0 do 7. Predložak datoteke prikazan je u sljedećem okviru, a dostupan je i na web sjedištu laboratorijskih vježbi.

```
JMBAG: xxxxxxxxxxxx
LED(x): SR
LED(x): D latch
LED(x): D flip-flop
LED(x): D flip-flop, asinkroni SR
LED(x): D flip-flop, sinkroni SR
LED(x): D flip-flop s enable ulazom, sinkroni SR
LED(x): JK
LED(x): T
```

U skladu s Vašim opažanjima rada ispitnog sklopa, sva polja označena malim slovom `x` potrebno je zamijeniti odgovarajućim podacima, odnosno vašim JMBAG identifikacijskim brojem, te pozicijom LED indikatora na kojima su opaženi izlazi iz određenih tipova bistabila. Ostale znakove, kao ni redoslijed redaka u predlošku **nije dopušteno mijenjati!** Datoteku nazvanu točno `pogodi_bistabil.txt` potrebno je spremići (upload) u sustav Ferko najkasnije do završetka ciklusa laboratorijskih vježbi (petak, 8. prosinca).

Ocjenjivanje ovog dijela vježbe provodit će se isključivo strojnom provjerom rješenja, koja ne dopušta nikakva odstupanja od propisane forme!