

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Digitalna logika

Laboratorijske vježbe korištenjem sklopovskih pomagala

Upute za 3. laboratorijsku vježbu

Marko Zec

Prosinac 2015.

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.

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 treba proširiti instanciranjem dodatnih modula. U trećem dijelu korištenjem jezika VHDL opisuje se sklop za daljinsko upravljanje maketa vozila.

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 diakritički 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 (1.5 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

prezimenam, 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!**



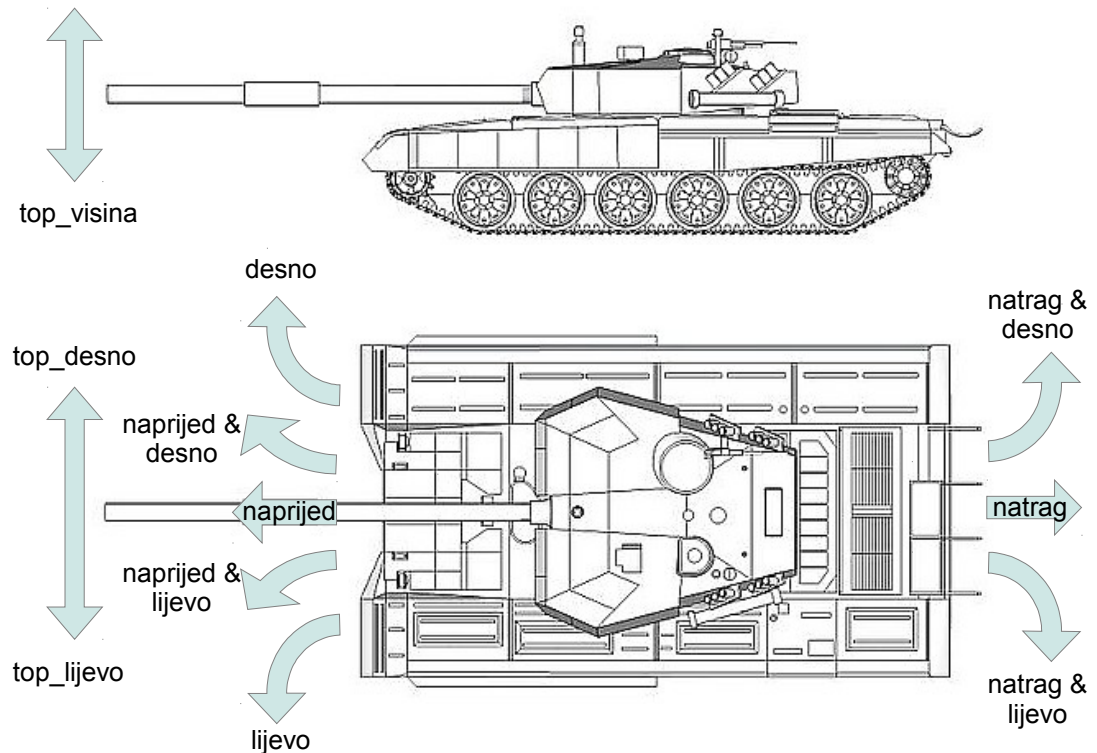
Nakon što ste ispitali rad sklopa, u sustav Ferko trebate prenijeti (upload) slijedeće datoteke:

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

4 Daljinski upravljač makete vozila (0.5 boda)

Vaš je zadatak konstruirati kombinacijski sklop koji će omogućiti upravljanje maketom tenka s većim brojem upravljačkih signala pomoću samo pet tipki na razvojnoj pločici ULX2S. Ciljani sklop je funkcijski potpuno identičan primjeru iz 1. laboratorijske vježbe, kojeg umjesto shematskim putem ovaj put treba opisati VHDL-om.

Pri izradi zadatka koristite već gotovi modul "rf_modulator" koji ima slijedeće ulaze: clk, kanal, brzina, naprijed, natrag, lijevo, desno, top_lijevo, top_desno, top_visina, top_zvuk, top_granata, strojnica i motor (slika 1). Jedini izlaz iz modula "rf_modulator" je radiofrekvencijski signal rf koji služi za prenošenje kodiranih upravljačkih informacija maketi tenka bežičnim putem na odabranom kanalu u frekvencijskom pojasu 27 MHz.



Slika 1: funkcije glavnih upravljačkih signala modula "rf_modulator"

Jezikom VHDL opišite sklop koji instancira modul "rf_modulator" te na njega dovedite sve signale potrebne za upravljanje maketom, a koji trebaju biti izvedeni kao logičke funkcije vanjskih ulaznih signala povezanih s pet tipki razvojne pločice: btn_up, btn_down, btn_center, btn_left i btn_right:

```
motor = btn_up AND btn_down AND NOT btn_center
naprijed = btn_up AND NOT btn_center
natrag = btn_down AND NOT btn_center
lijevo = btn_left AND NOT btn_center
desno = btn_right AND NOT btn_center
top_lijevo = btn_left AND btn_center
top_desno = btn_right AND btn_center
top_visina = btn_down AND btn_center
top_zvuk = btn_up AND btn_center
strojnica = btn_left AND btn_right
```

Izlazni signal `rf` modula "`rf_modulator`" treba povezati na priključnice višenamjenske stereo utičnice `p_tip` i `p_ring`, putem kojih se modulirani radiofrekvencijski signal odašilje u eter. Na dvobitni ulaz `kanal` modula "`rf_modulator`" treba dovesti vanjski signal s mikroprekidača `sw_1` i `sw_0`, a na ulaz `brzina` dovesti vanjski signal s mikroprekidača `sw_3` i `sw_2`.

Pri izradi zadatka možete se poslužiti predloškom za opis sklopa koji instancira i povezuje modul "`rf_modulator`" sa samo nekima od zadanih upravljačkih signala. Predložak je prikazan u okviru na slijedećoj stranici, a možete ga preuzeti i s web-sjedišta laboratorijskih vježbi.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
use ieee.numeric_std.all;

entity daljinski_upravljac is
    port (
        clk_25m: in std_logic;
        sw: in std_logic_vector(3 downto 0);
        btn_left, btn_right, btn_up, btn_down, btn_center: in std_logic;
        p_tip: out std_logic_vector(3 downto 0);
        p_ring: out std_logic
    );
end daljinski_upravljac;

architecture behavioral of daljinski_upravljac is
    signal rf: std_logic;
    signal naprijed, natrag, motor: std_logic;

begin

    motor <= btn_up and btn_down and not btn_center;

    naprijed <= btn_up and not btn_center;
    natrag <= btn_down and not btn_center;

    odasiljac: entity rf_modulator
    port map (
        clk => clk_25m,
        kanal => sw(1 downto 0), brzina => sw(3 downto 2),
        naprijed => naprijed, natrag => natrag,
        lijevo => '0', desno => '0',
        top_lijevo => '0', top_desno => '0',
        top_zvuk => '0', top_visina => '0',
        top_granata => '0', strojnica => '0',
        motor => motor, rf => rf
    );

    -- Izlazni FM signal
    p_tip <= rf & rf & rf & rf; -- grupiranje jednobitnih u visebitni signal
    p_ring <= rf;

end;
```

Više detalja o funkcijama modula "`rf_modulator`", smjernice za ispitivanje rada sklopa za daljinsko upravljanje, te smjernice postupanje s radioupravljanim maketama možete pronaći u uputama za 1. laboratorijsku vježbu.

Po želji, umjesto daljinskog upravljača makete tenka, možete korištenjem jezika VHDL konstruirati daljinski upravljač LEGO makete vozila, koristeći modul "`lego_ir`" iz 1. laboratorijske vježbe.