Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Digitalna logika

Laboratorijske vježbe korištenjem sklopovskih pomagala

Upute za 2. laboratorijsku vježbu

Marko Zec

Studeni 2015.

1 Zadatak: shematski opis kombinacijskih sklopova

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. Sve izlaze vlastite kombinacijske mreže potrebno je dovesti i na LED indikatore kako bi mogli vizualno provjeriti ispravnost generiranja zadanih ASCII kodova.

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 studentovo ime ili prezime kraće od četiri znaka umjesto raspodjele znakova 4+4 potrebno je odabrati takvu raspodjelu da se popuni svih 8 znakova, npr. `MarkoZec`. Svaki znak niza treba generirati pobudom (pritiscima na tipke) prema tablici:

Ulaz (tipke): down left center up right	Izlaz: kodna riječ	Primjer: MarkoZec
-	ASCII NULL	(NULL)
down	ASCII NULL	(NULL)
left	1. znak	M
center	2. znak	a
up	3. znak	r
right	4. znak	k
down AND left	5. znak	o
down AND center	6. znak	Z
down AND up	7. znak	e
down AND left	8. znak	c

Za sve ostale moguće kombinacije tipki (one koje nisu obuhvaćene tablicom) rad kombinacijskog modula nije specificiran, odnosno dozvoljeno je na izlazu generirati bilo kakav 8-bitni ASCII kod. Po vlastitoj želji možete odabrati i drugačije kombinacije tipki od predloženih, u kojem slučaju te kombinacije morate ispravno dokumentirati, odnosno označiti u tablici u sklopu pripremnog dijela vježbe.

Na laboratorijsku vježbu trebate doći s napisanom pripremom (provedena minimizacija logičkih funkcija za sve bitove kodne riječi), te po mogućnosti s izvedbom sklopa koja odgovara funkcijskim zahtjevima zadatka.

2 Primjer rješenja

Na početku projektiranja kombinacijske mreže potrebno je definirati njeno ponašanje tablicom kojom se za svaku od predviđenih kombinacija ulaznih signala (pritisnutih

tipki) određuje vrijednost ASCII kodne riječi na izlazu. U ovom primjeru pretpostavit ćemo da se student zove Gilgameš Svamiđi Ramajana Ratan Brahmaputra Uš. Kombinirajući početna slova imena i prezimena te uz zamjenu dijakritičkih znakovima engleske abecede dobivamo slijed znakova `GilgamUs`, te popunjavamo tablicu:

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	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
down	NULL	00	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
left	G	47	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**1**
center	i	69	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**
up	l	6c	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**0**
right	g	67	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**1**
down AND left	a	61	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**
down AND center	m	6d	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**
down AND up	U	55	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
down AND right	s	73	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**

Prema podacima iz tablice potrebno je pronaći logičke funkcije za svaki od 8 bitova izlazne kodne riječi code = f(down, left, center, up, right). Minimalne oblike svake od 8 funkcija code[i] može se dobiti tako da se u zasebnu K-tablicu za svaki bit kodne riječi upišu vrijednosti funkcije (0 ili 1) za svaku od 10 tablicom specificiranih kombinacija ulaznih signala, dok se ostala polja u K-tablici ostave prazna ili označe s "x" kao nespecificirane vrijednosti funkcije. Za bitove najveće težine (7 i 6) u ovom primjeru već samim pogledom na gornju tablicu može se isčitati minimalni oblik funkcija:

code[7] = 0

code[6] = left + center + right + up

U nastavku je prikazan primjer pronalaženja minimalnog oblika logičke funkcije za bitove 5 i 4 kodne riječi korištenjem K-tablica.

code[5]		down left center								
		000	001	011	010		100	101	111	110
up right	00	0	1	x	0		0	1	x	1
	01	1	x	x	x		1	x	x	x
	11	x	x	x	x		x	x	x	x
	10	1	x	x	x		0	x	x	x

code[5] = right + center + $\overline{\text{down}}$ · up + down · left

code[4]		down left center								
		000	001	011	010		100	101	111	110
up right	00	0	0	x	0		0	0	x	0
	01	0	x	x	x		1	x	x	x
	11	x	x	x	x		x	x	x	x
	10	0	x	x	x		1	x	x	x

code[4] = down ▪ (up + right)

Logičke funkcije za preostala 4 bita kodne riječi su ove:

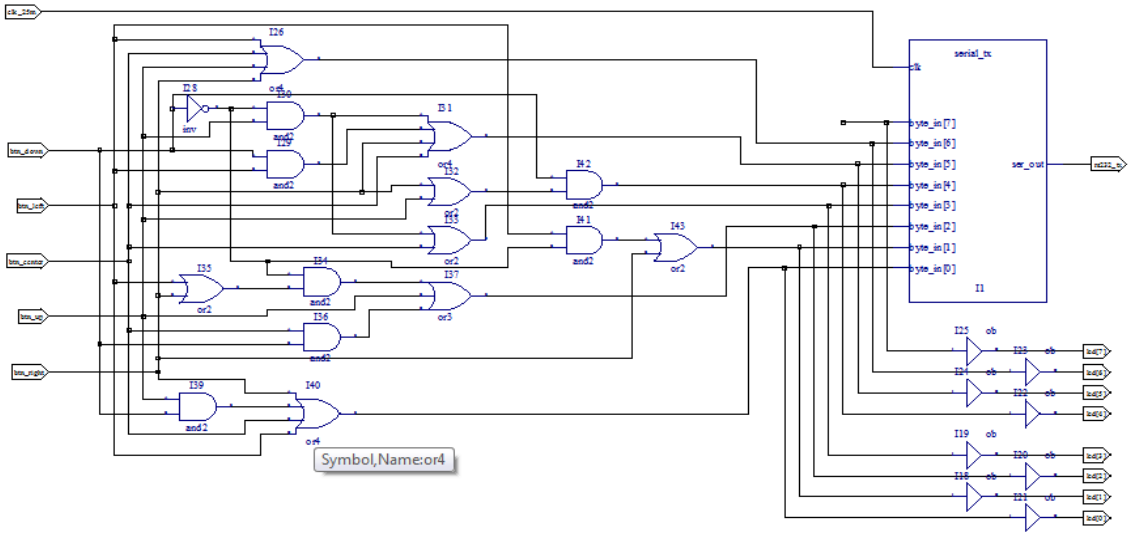
code[3] = center + $\overline{\text{down}}$ ▪ up

code[2] = up + $\overline{\text{down}}$ ▪ (left + right) + down ▪ center

code[1] = right + $\overline{\text{down}}$ ▪ left

code[0] = left + center + right + down ▪ up

Temeljem funkcija za svih osam bitova kodne riječi može se konstruirati shema kombinacijske mreže:



3 Priprema – minimizacija logičkih funkcija

Popunite tablicu kombinacija ulaznih signala (tipki) i pripadajućih izlaznih kodnih riječi u heksadekadskom i binarnom zapisu.

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

Izvedite i napišite minimalne oblike funkcija code[i] = f(up, down, left, right, center) za svaki pojedini bit izlazne kodne riječi. Za minimizaciju koristite K-tablice.

code[6]		down left center								
		000	001	011	010		100	101	111	110
up right	00									
	01									
	11									
	10									

code[6] =

code[5]		down left center								
		000	001	011	010		100	101	111	110
up right	00									
	01									
	11									
	10									

code[5] =

code[4]		down left center								
		000	001	011	010		100	101	111	110
up right	00									
	01									
	11									
	10									

code[4] =

code[3]		down left center								
		000	001	011	010		100	101	111	110
up right	00									
	01									
	11									
	10									

code[3] =

code[2]		down left center								
		000	001	011	010		100	101	111	110
up right	00									
	01									
	11									
	10									

code[2] =

code[1]		down left center								
		000	001	011	010		100	101	111	110
up right	00									
	01									
	11									
	10									

code[1] =

code[0]		down left center								
		000	001	011	010		100	101	111	110
up right	00									
	01									
	11									
	10									

code[0] =

4 Opis i sinteza sklopa

Stvorite novi prazni direktorij na disku, te u njega pohranite slijedeće datoteke koje možete dohvatiti iz repozitorija na sustavu Ferko ili s www.nxlab.fer.hr/dl:

- `ulx2s.lpf`: definicije ulazno-izlaznih priključaka FPGA sklopa na pločici ULX2S
- `lab2.sch`: shematski opis oglednog sklopa, instancira modul "`serial_tx`", određuje povezivanje kombinacijske mreže s vanjskim (fizičkim) ulaznim i izlaznim priključcima
- `serial_tx.vhd`: VHDL opis sekvencijskog modula za serijsko odašiljanje 8-bitnog podatka prema računalu putem USB sučelja
- `serial_tx.sym`: grafički simbol sučelja modula "`serial_tx`"

Datotekom `lab2.sch` opisan je ogledni sklop iz primjera iz poglavlja . Već gotovu shemu možete koristiti kao predložak za izradu vlastite vježbe.

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. Prilikom stvaranja projekta preporuča se odmah u projekt uključiti datoteke `ulx2s.lpf`, `lab2.sch` i `serial_tx.vhd`. Za *synthesis tool* treba **obavezno odabrati *Synplify Pro***!

Prije nego što počnete uređivati shemu vlastite implementacije kombinacijske mreže, preporuča se sintetizirati ogledni sklop korištenjem nepromijenjenih datoteka dohvaćenih iz sustava Ferko, isprogramirati FPGA sklop dobivenom konfiguracijskom datotekom, te ispitati rad sklopa.

VAŽNO: prije **svakog** pokretanja postupka sinteze u alatu Lattice Diamond obavezno provjeriti da je uključena opcija "**View -> Show views -> Hierarchy**", čime alat za sintezu automatski odabire glavni (*top-level*) modul koji se povezuje s vanjskim priključcima FPGA sklopa. Ako se preskoči ovaj korak alat ne će ispravno sintetizirati konfiguraciju FPGA sklopa!

Vlastitu implementaciju najbrže možete specificirati tako da iz demonstracijske implementacije izbrišete kombinacijsku mrežu te ju zamijenite vlastitom. Ako u Vašoj implementaciji neki od izlaznih signala trebaju imati konstantnu vrijednost, iz biblioteke možete odabrati komponente VHI za '1' odnosno VLO za '0'.

5 Ispitivanje rada sklopa

Konfigurirajte FPGA sklop korištenjem alata ujprog. Ispitajte rad sintetiziranog sklopa opažanjem stanja LED indikatora. Odziv na pobudu možete pratiti i na računalu korištenjem programa za emulaciju asinkronog terminala, npr. Hyper Terminal ili Putty, uz parametre: 115200 bauds, 8-bitni podaci, 1 start bit, 1 stop bit, bez pariteta. Alat ujprog također može poslužiti kao emulator asinkronog terminala:

```
$ ujprog lab2_impl1.jed
ULX2S JTAG programmer v 2.beta2 (built Jul  7 2015 14:50:37)
Using USB JTAG cable.
Programming: 100%
Completed in 2.65 seconds.

$ ujprog -t
ULX2S JTAG programmer v 2.beta2 (built Jul  7 2015 14:50:37)
Using USB JTAG cable.
Terminal emulation mode, using 115200 bauds
Press ENTER, ~, ? for help
GilgamUs
```

Ogledni sklop iz 2. poglavlja kao odziv na pritisnutu tipku `btn_left` na računalu će u prozoru emulatora asinkronog serijskog teminala prikazati znak “G” (ASCII 0x47), a isti kod bit će vidljiv i na LED indikatorima razvojne pločice: