

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Digitalna logika

Laboratorijske vježbe korištenjem sklopovskih pomagala

Upute za 2. laboratorijsku vježbu

Marko Zec

Listopad 2024.

1 Zadatak: shematski opis kombinacijske logike

Vaš je zadatak projektirati i ispitati kombinacijsku logiku kojoj je ulaz četverobitni binarni a izlaz osambitni ASCII kod. Izlazni kodovi trebaju odgovarati ASCII znakovima zadnjih osam znamenaka Vašeg JMBAG identifikatora, te sadržavati dodatne oznake kraja niza. Kombinacijsku logiku potrebno je prije ucrtavanja u shemu minimizirati korištenjem K tablica, zasebno za svaki bit izlaznog ASCII koda.

Kombinacijsku logiku treba ugraditi u unaprijed pripremljen kostur ispitnog sekvencijskog sklopa. Pomoćna logika ispitnog sklopa sastoji se od:

- četiri bridom okidana D bistabila (*D flip-flop*);
- zbrajala, koje trenutnu vrijednost binarnog koda spremljenog u bistabilima kombinacijskom logikom uvećava za jedan;
- okidačkih modula, koji pritiske na tipke razvojne pločice pretvaraju u impulse duljine točno jednog ciklusa takta, kojima se omogućuje upis novih vrijednosti u D bistabile, odnosno njihov reset;
- izlaznog modula, koji 8-bitni ASCII kod iz kombinacijske logike odašilje kao niz (seriju) bitova brzinom od 115200 bit/s u formatu pogodnom za prijenos putem USB sučelja na računalo i ispis ASCII znakova na zaslonu.

Svi sekvencijski moduli ispitnog sklopa okidaju se na rastući brid takta iz oscilatora ugrađenog na razvojnu pločicu (25 MHz). Kako bi se omogućilo vizualno praćenje stanja pobude i odziva kombinacijske logike, ispitni sklop uključuje i prospoj izlaza iz D bistabila na četiri LED indikatora razvojne pločice, te prospoj četiri najniža bita ASCII koda na dodatna četiri LED indikatora.

Na laboratorijsku vježbu trebate doći **s napisanom pripremom** prema uputama iz poglavlja 2 .

2 Priprema: minimizacija logičkih funkcija

Popunite tablicu kombinacija pobude i pripadajućih izlaznih kodnih riječi u heksadekadskom i binarnom zapisu:

Ulaz: Q3 Q2 Q1 Q0	Izlaz: JMBAG znamenka	Izlaz: ASCII kod (hex)	Izlaz: ASCII kod (binarno)
0 0 0 0			
0 0 0 1			
0 0 1 0			
0 0 1 1			
0 1 0 0			
0 1 0 1			
0 1 1 0			
0 1 1 1			
1 0 0 0	<ASCII CR>	0x0d	0 0 0 0 1 1 0 1
1 0 0 1	<ASCII LF>	0x0a	0 0 0 0 1 0 1 0

Tablica se popunjava s osam od ukupno deset znakova JMBAG identifikatora, pri čemu se odbacuju prve dvije znamenke s lijeve strane (dvije nule za većinu studenata). Popunite tablicu na način da za ulaznu kombinaciju $Q[3..0] = 0000$ logika daje ASCII kod znamenke najveće težine, a za kod $Q[3..0] = 0111$ ASCII kod znamenke najmanje težine. Za ulazni kod $Q[3..0] = 1000$ Vaša kombinacijska logika mora na izlazu dati ASCII kod za povratak na početak retka (*carriage return - CR*), a za ulaz $Q[3..0] = 1001$ generirati ASCII kod za prelazak u novi red (*line feed - LF*).

Za sve ostale kombinacije tipki (one koje nisu obuhvaćene tablicom) dozvoljeno je na izlazu generirati bilo kakav 8-bitni ASCII kod.

Izvedite i napišite minimalne oblike funkcija $ascii[i] = f(q3, q2, q1, q0)$ za svaki pojedini bit izlazne ASCII kodne riječi na temelju popunjene tablice. Ne treba koristiti postupak minimizacije višezlazne funkcije, nego se svaki bit izlaznog ASCII koda može minimizirati kao zasebna jednobitna funkcija.

Za bitove najveće težine (7 i 6) odmah bi trebalo biti vidljivo da će im vrijednost biti 0 za sve ulazne kombinacije. Za sve ostale bitove ASCII kodne riječi, kao alat za pronalaženje minimalnog oblika logičke funkcije koristite priložene K-tablice.

		Q1 Q0			
		00	01	11	10
Q3 Q2	00				
	01				
	11				
	10				

$ascii[5] =$

		Q1 Q0			
		00	01	11	10
Q3 Q2	00				
	01				
	11				
	10				

$ascii[4] =$

		Q1 Q0			
		00	01	11	10
Q3 Q2	00				
	01				
	11				
	10				

$ascii[3] =$

		Q1 Q0			
		00	01	11	10
Q3 Q2	00				
	01				
	11				
	10				

$ascii[2] =$

		Q1 Q0			
		00	01	11	10
Q3 Q2	00				
	01				
	11				
	10				

ascii[1] =

		Q1 Q0			
		00	01	11	10
Q3 Q2	00				
	01				
	11				
	10				

ascii[0] =

3 Opis i sinteza sklopa

Stvorite novi prazni direktorij na disku, te u njega raspakirajte arhivu lab2.zip koju možete dohvatiti s www.nxlab.fer.hr/dl.

Pokrenite razvojnu okolinu Lattice Diamond, te otvorite projekt lab2_ulx2s.ldf ili lab2_ulx3s.ldf, zavisno od razvojne pločice s kojom radite. Provjerite odgovara li tip FPGA sklopa onom koji je ugrađen na Vašu razvojnu pločicu, te prema potrebi u kartici "File" dvostrukim klikom na tip FPGA sklopa putem iskočnog izbornika odaberite ispravni.

Prije nego što počnete uređivati shemu vlastite implementacije kombinacijske mreže, preporuča se sintetizirati ogledni sklop korištenjem nepromijenjenih datoteka dohvaćenih s www.nxlab.fer.hr/dl, isprogramirati FPGA sklop dobivenom konfiguracijskom datotekom, te ispitati rad sklopa.

Vlastitu implementaciju najbrže možete specificirati tako da korištenjem shematskog editora **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'.

4 Primjer rješenja zadatka

Primjer tabličnog zadavanja kombinacijske funkcije za (fiktivni) JMBAG 0087654321:

Ulaz: Q3 Q2 Q1 Q0	Izlaz: JMBAG znamenka	Izlaz: ASCII kod (hex)	Izlaz: ASCII kod (binarno)
0 0 0 0	8	0x38	0 0 1 1 1 0 0 0
0 0 0 1	7	0x37	0 0 1 1 0 1 1 1
0 0 1 0	6	0x36	0 0 1 1 0 1 1 0
0 0 1 1	5	0x35	0 0 1 1 0 1 0 1
0 1 0 0	4	0x34	0 0 1 1 0 1 0 0
0 1 0 1	3	0x33	0 0 1 1 0 0 1 1
0 1 1 0	2	0x32	0 0 1 1 0 0 1 0
0 1 1 1	1	0x31	0 0 1 1 0 0 0 1
1 0 0 0	<ASCII CR>	0x0d	0 0 0 0 1 1 0 1
1 0 0 1	<ASCII LF>	0x0a	0 0 0 0 1 0 1 0

U nastavku je prikazan primjer pronalaženja minimalnog oblika logičke funkcije za bitove 3, 2, 1 i 0 ASCII kodne riječi korištenjem K-tablica:

		Q1 Q0			
		00	01	11	10
Q3 Q2	00	1	0	0	0
	01	0	0	0	0
	11	x	x	x	x
	10	1	1	x	x

$$\text{ascii}[3] = q_3 + \overline{q_2} \cdot \overline{q_1} \cdot \overline{q_0}$$

		Q1 Q0			
		00	01	11	10
Q3 Q2	00	0	1	1	1
	01	1	0	0	0
	11	x	x	x	x
	10	1	0	x	x

$$\begin{aligned} \text{ascii}[2] = & \overline{q_2} \cdot q_1 + \overline{q_3} \cdot \overline{q_2} \cdot q_0 \\ & + q_2 \cdot \overline{q_1} \cdot \overline{q_0} + q_3 \cdot \overline{q_0} \end{aligned}$$

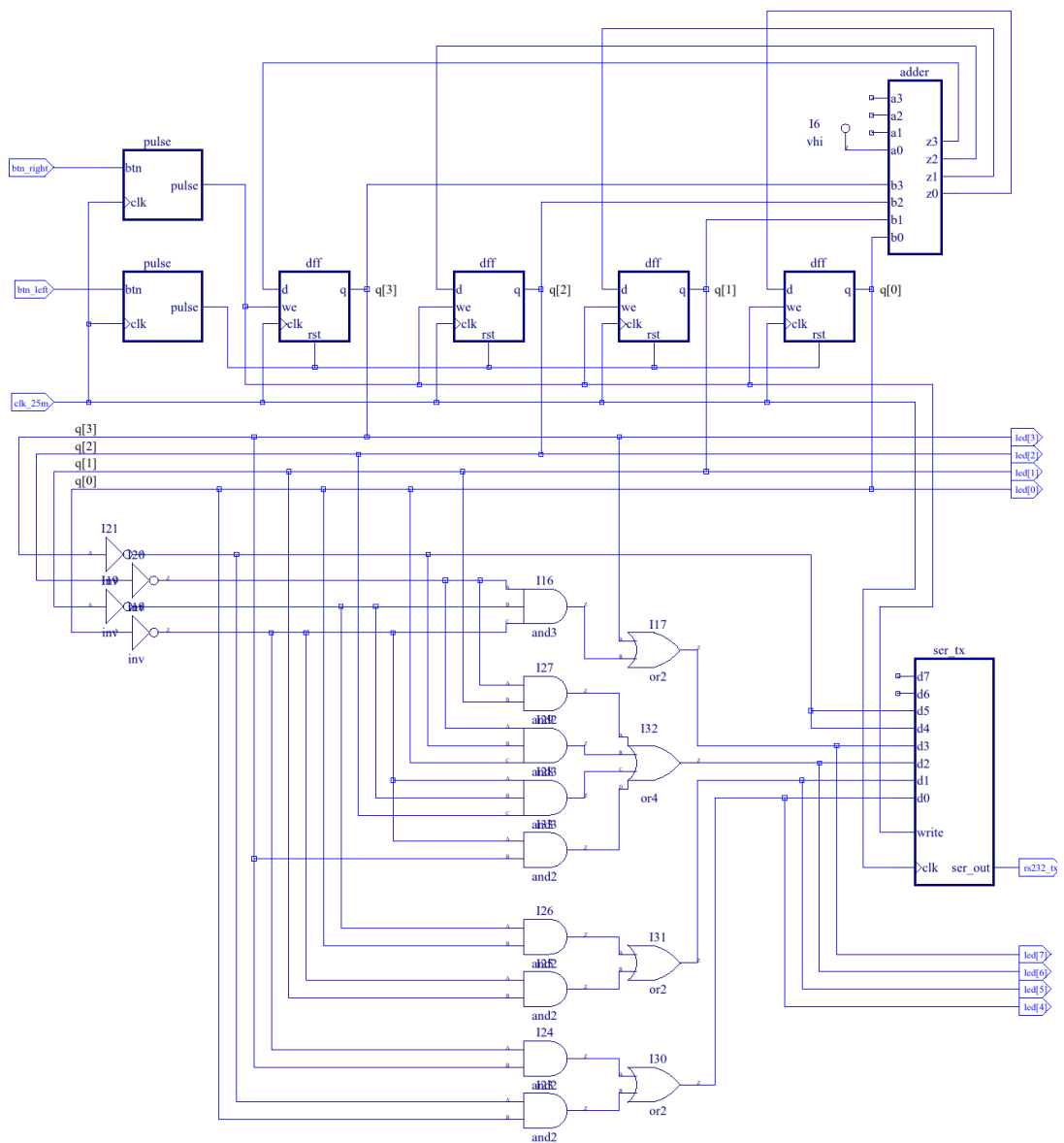
		Q1 Q0			
		00	01	11	10
Q3 Q2	00	0	1	0	1
	01	0	1	0	1
	11	x	x	x	x
	10	0	1	x	x

		Q1 Q0			
		00	01	11	10
Q3 Q2	00	0	1	1	0
	01	0	1	1	0
	11	x	x	x	x
	10	1	0	x	x

$$\text{ascii}[1] = \overline{q1} \cdot q0 + q1 \cdot \overline{q0}$$

$$\text{ascii}[0] = q3 \cdot \overline{q0} + \overline{q3} \cdot q0$$

Temeljem funkcija izvedenih za svih osam bitova kodne riječi može se konstruirati shema kombinacijske mreže, uklopljena u ispitni sklop:



5 Ispitivanje rada sklopa

Konfigurirajte FPGA sklop korištenjem alata uprog. Tipkom btn_right okida se upis nove vrijednosti u bistabile, a tipkom btn_left stanje svih bistabila postavlja se na nulu. Ispitajte rad sintetiziranog sklopa opažanjem stanja LED indikatora i praćenjem odziva na računalu:

```
$ uprog -t lab2_impl1.bit
ULX2S / ULX3S JTAG programmer v 3.0.92 (built Jun 17 2024 03:11:54)
Using USB cable: ULX3S FPGA board
Programming: 100%
Completed in 1.65 seconds.
Terminal emulation mode, using 115200 bauds
Press ENTER, ~, ? for help
87654321

87654321
```

Uočite da se na LED indikatorima prikazuje stanje bistabila i bitovi niže težine ASCII kodne riječi koja tek treba biti poslana prema računalu putem USB sučelja u trenutku pritiska na tipku btn_right.