

Izveštaj za projekat iz UPV-a

Elektrotehnički fakultet Univerziteta u Beogradu 2020/2021

Aleksandar Jevtić 0523/2016

| | |
|---|---|
| ABSTRACT | 1 |
| UVOD | 1 |
| SYSTEM SPECIFICATION AND BLOCK SCHEME | 2 |
| Generator | 2 |
| Specification and block scheme | 2 |
| Roll dice | 3 |
| Specification and block scheme | 3 |
| Bcd to 7 segment 3x8 decoder | 4 |
| Specification and block scheme | 4 |
| Yamb | 4 |
| Specification and block scheme | 4 |
| ZAKLJUČAK | 4 |
| LITERATURA | 6 |

Abstract

Biće predstavljen način realizacije digitalnog sistema koji treba da simulira društvenu igru jamb. Na raspolaganju je *DE1 – SoC* ploča na kojoj se nalazi čip *dual-core Cortex-A9* iz familije “ARM”, dok je softver koriscen za programiranje navedenog čipa “*Altera Quartus Prime Lite Edition 18.1.1.646*”.

Pristup realizaciji sistema se svodi na razlaganje sistema na manje komponente čijim će se krajnjim povezivanjem na odgovarajući način postignuti željena funkcionalnost celokupnog sistema. Periferije koje sistem zahteva su 6 led displeja, 6 prekidača, 1 taster i 10 LE dioda. Ako se pogleda blok dijagram navedene FPGA ploče videćemo da su ove komponente označene sa *7-segment Display*, *Slide-Switch*, *Push Button* i *LED*, respektivno. Potrebno je još samo odrediti način funkcionisanja navedenih periferija i može se pristupiti realizaciji sistema. Blok šema kao i način funkcionisanja se mogu pronaći u uputstvu¹ za korišćenje odgovarajuće FPGA ploče.

Uvod

Potrebno je napraviti digitalni sistem koji će da simulira društvenu igru jamb. Dok je sistem resetovan i u inicijalnom stanju, na 6 LED displeja se ispisuje vrednost 0. Pritiskom na taster se na displejima prikazuje vrednost 6 generatora pseudo slučajnih brojeva koji generišu vrednosti između 1 i 6 (bacanje kockica). Drugim pritiskom na taster se zaustavlja obrtanje kockica i na displejima se prikazuju vrednosti zaustavljenih generatora slučajnih brojeva. Postavljanjem prekidača na stanje logičke jedinice, zadržavaju se kockice za naredno bacanje. Ponovnim pritiskom na taster, ponovo se pokreću generatori slučajnih brojeva, ali ovog puta samo za kockice čiji su prekidači postavljeni na logičku nulu. Isti postupak se ponavlja i za treće bacanje. Ako je nakon trećeg bacanja dobijen jamb (5 kockica ima istu vrednost), potrebno je sve diode uključiti da blinkaju nekoliko sekundi, a zatim preći u inicijalno stanje u kome je sistem spreman za novo bacanje.

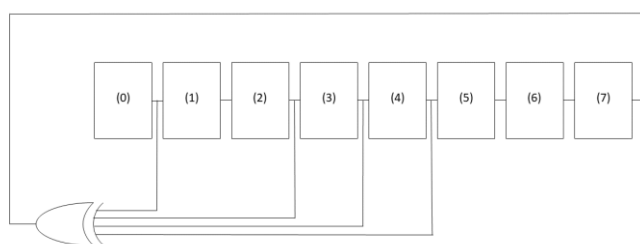
System specification and block scheme

Na osnovu opisa sistema koji treba realizovati je najpre potrebno uočiti da sistem zahteva generator slučajnih brojeva. Uloga generatora jeste da u sistem unese sekvence slučajnih brojeva koji se nalaze u određenom opsegu. Ideja je da se takav jedan generator nalazi u komponenti koja će da predstavlja kockicu za bacanje. Sa tim u vezi je jasno da će generator generisati sve cele brojeve iz opsega 1 do 6. Kockica za bacanje je realizovana kao zasebna komponenta da bi lako mogla da se proširi njena primena i na druge sisteme koji zahtevaju istu. Ako se nastavi sa daljom analizom opisa sistema, zaključuje se da je potrebno obezbediti pravilan prikaz brojeva na displejima. Komponenta koja će ispuniti traženu funkcionalnost jeste 3x8 dekodler. U nastavku će detaljnije biti opisana struktura i način funkcionisanja svake od komponenti kao i način na koji je potrebno te komponente povezati da bi se ispunile tražene funkcionalnosti.

Generator

Specification and block scheme

Komponenta generator se realizuje kao mašina stanja gde se promena stanja vrši po principu 8-bitnog *linear feedback shift registra*ⁱⁱ (*lfsr*). Na Slika 1 je prikazan način realizacije 8-bitnog *lfsr*-a. Teorijski je dovoljno iskoristiti i *lfsr* sa 3 bita s'

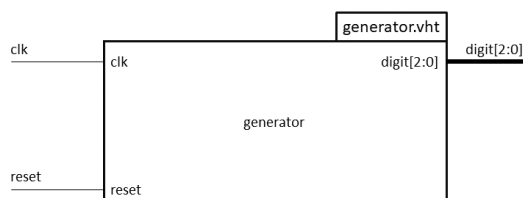


Slika 1. 8-bitni *lfsr*

obzirom da je potrebno izgenerisati cele brojeve u opsegu 1 do 6 (sa 3 bita je moguće predstaviti 8 različitih brojeva), međutim zbog bolje raspodele se koristi 8-bitni *lfsr*. Ovo znači da će postojati 2^8 različitih stanja, ali kako je potrebno izgenerisati samo 6 različitih brojeva, svakom broju će biti dodeljen određeni opseg stanja. U ovome se i ogleda bolja raspodela sekvence generisanih brojeva. Naime, može se desiti da generator izgeneriše dva ili više istih uzastopnih slučajnih brojeva što je u saglasnosti sa realnim bacanjem kockice (na primer kada neko dva puta zaredom dobije 6). Mana ovakve realizacije se ogleda u neravnomernoj raspodeli, odnosno u tome da nemaju svi brojevi iz opsega istu verovatnoću generisanja. Kako broj stanja nije deljiv sa 6 jednostavno se dolazi do prethodnog zaključka. Da se uzme *lfsr* sa bilo kojim brojem bita takođe ne bi bilo moguće usvojiti ravnomernu raspodelu iz istog razloga. Ono što je moguće uraditi da bi se poboljšala raspodela jeste da se u konkretnom slučaju od 2^8 stanja uzme najveći broj stanja deljiv sa 6 i on ravnomerno dodeli svakom od brojeva na odgovarajući način, dok bi ostatak stanja predstavljao stanja u kom se prepisuje prethodno generisani broj. Za ovaj primer se uzima takva raspodela da svi celi brojevi iz opsega 1 do 5 imaju istu verovatnoću generisanja, dok broj 6 imaju manju verovatnoću generisanja.

Treba uvideti pseudo karakteristiku ovako realizovanog generatora. Nakon svakog reseta sistema, generator će preći u inicijalno stanje. Na osnovu Slika 1 se jasno vidi da postoji jednoznačnost u prelasku stanja. Iz jednog određenog stanja se može preći samo u neko drugo određeno stanje, zatim iz tog drugog u neko treće itd. Dakle svakim resetom sistema, generator će na isti način menjati stanja, odnosno generisaće istu sekvencu "slučajnih brojeva".

Na Slika 2 je prikazana blok šema generatora. Kao što se vidi sa prikazane blok šeme, generator će po uključanju napajanja



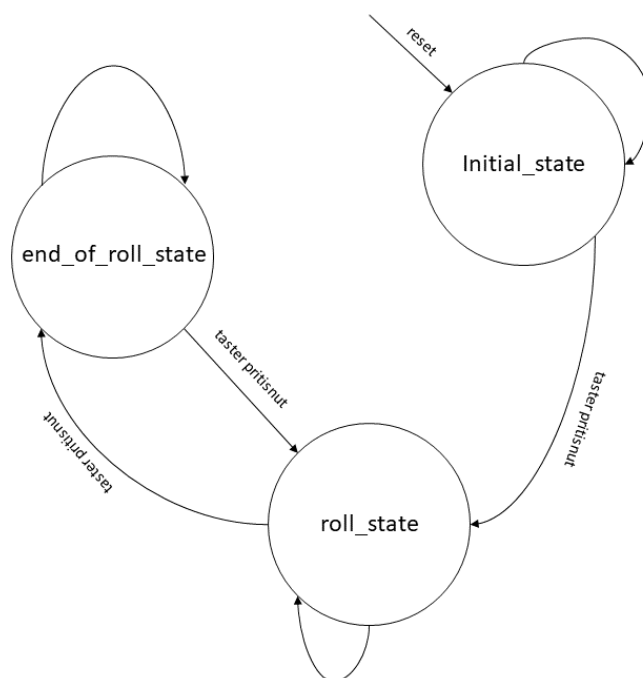
Slika 2. Blok šema generatora

početi sa generisanjem slučajnih brojeva. U ovome i leži razlog što se generator i kockica realizuju kao zasebne komponente, ali više o tome u narednom odeljku.

Roll dice

Specification and block scheme

Komponenta koja simulira kockicu se realizuje kao mašina stanja. Na Slika 3 je prikazan dijagram stanja. Postoje 3 stanja.



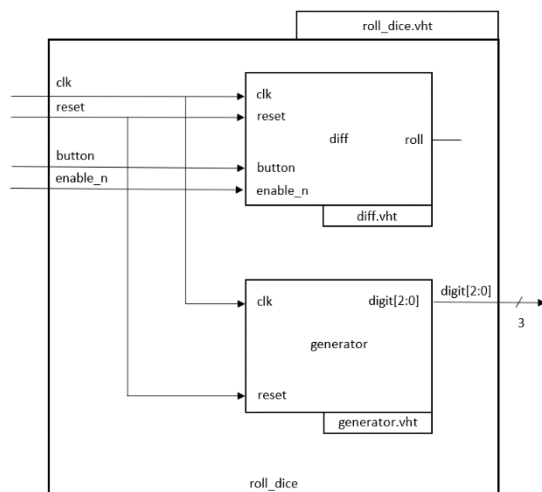
Slika 3. Dijagram stanja za komponentu roll_dice

Inicijalno stanje (**initial_state**), stanje u kome se kockica vrti (**roll_state**) i stanje kada je kockica stala (**end_of_roll_state**). U inicijalno stanje se prelazi pojavom signala **reset**. Kada je kockica u inicijalnom stanju na izlazu će biti prikazana 0. Iz inicijalnog stanja se može preći samo u stanje u kome se kockica vrti. Prelazak se vrši pritiskom na taster. Ovo označava da je sistem bio u početnom stanju i da je igrač pritisnuo taster čime je pokrenuo bacanje kockice. Tada se na izlaz prosleđuje izlaz generatora, odnosno biće prikazane vrednosti slučajno generisanih brojeva. U stanju u kom se kockica vrti se ostaje sve dok igrač ponovo ne pritisne taster što označava kraj bacanja. Tada se prelazi u stanje koje označava da je kockica stala i na izlaz se prosleđuje vrednost slučajnog broja koju je generator izgenerisao u trenutku pritiska na taster. U stanju koje označava da je kockica stala se ostaje sve do novog pritiska tastera što označava da je igrač započeo novo bacanje.

Kockica sadrži generator slučajnih brojeva. Dokle god ima napajanja generator generiše slučajne brojeve bez obzira u kom se stanju kockica nalazi. Tek kada se kockica nađe u odgovarajućem stanju, ona počinje da se ponaša kao generator, odnosno na svoj izlaz prosleđuje izlaz generatora. Ovime je postignuto da izlaz kockice zavisi i od prvog i od drugog pritiska na taster a ne samo od drugog. Prethodna analiza predstavlja glavni razlog što su kockica i generator realizovane kao zasebne komponente.

Kockica takođe sadrži i diferencijator silazne ivice. Naime, kako taster ima aktivnu vrednost u logičkoj nuli, pritisak tastera se detektuje silaznom ivicom signala koji šalje taj taster. Diferencijator diferencira tu silaznu ivicu i kao rezultat daje signal u trajanju jedne periode signala takta i koji je aktivan u logičkoj jedinici (na Slika 3 je taj signal nazvan **taster pritisnut**).

Na Slika 4 je prikazana blok šema kockice. Na **enable** signal se dovodi signal sa jednog od prekidača. S obzirom da je



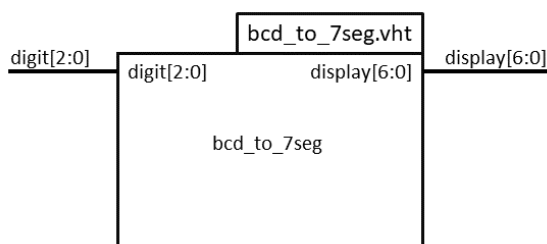
Slika 4. Blok šema komponente roll_dice

uslov da bacanje kockice bude omogućeno kada je prekidač postavljen na vrednost logičke nule na signal **enable** je dodat postfix **_n**. Signal **roll** je interni i on predstavlja signal **taster pritisnut** na Slika 3.

Bcd to 7 segment 3x8 decoder

Specification and block scheme

Komponenta koja ima ulogu da na sedmosegmentnom displeju ispravno prikaže izlaze kockica jeste 3x8 dekodera. Ulaz ovog dekodera je trobitna cifra prikazana u BCD formatu. Na osnovu kombinacije ulaza će se na odgovarajući način generisati izlaz. Jedan bit izlaza nije od značaja i on se nigde ne povezuje. Na Slika 5 je prikazana blok šema ovako realizovanog 3x8 dekodera.



Slika 5. Blok šema bcd_to_7seg

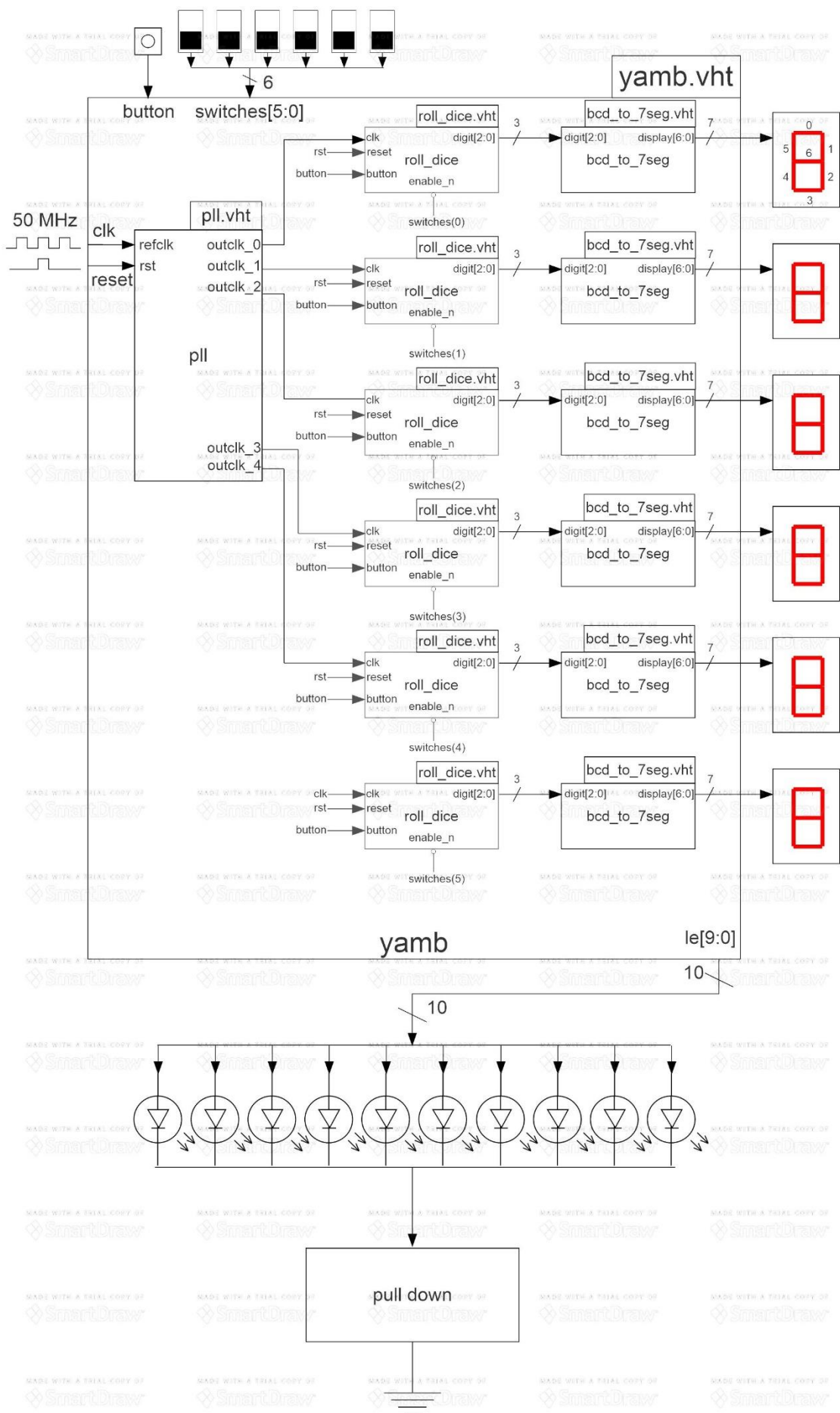
Yamb

Specification and block scheme

Krajnji korak u realizaciji sistema jeste povezivanje prethodno definisanih komponenata. Kako je jamb igra koja se igra sa šest različitih kockica biće potrebno 6 komponenata koje predstavljaju kockice kao i 6 3x8 dekodera za dekodovanje svakog od izlaza kockica u formu pogodnu za pravilan prikaz na displejima. Potrebno je ubaciti i komponentu *pll*. Razlog za uvođenje *pll*-a jeste taj što se time omogućuje vremenska različitost generatora. Na svaki generator se dovodi različit signal takta jer bi u suprotnom svaki generator u svakom trenutku generisao isti slučajni broj. Potrebno je da svi taktovi generatora budu brži od takta celog sistema. Blok šema celog sistema je data na Slika 6.

Zaključak

Da bi se realizovao jedan ovakav sistem potrebno je uraditi prvo detaljnu analizu hardvera, zatim samu analizu opisa sistema i tek onda razlagati sistem na manje celine. Nije imeprativno razložiti sistem na što više manjih celina već je potrebno pronaći što funkcionalniji način za realizaciju sa jedne i što praktičniji način za realizaciju sa druge strane. Nakon toga se može pristupiti pisanju koda i po potrebi se vratiti i modifikavati određene delove u ideji za realizaciju.



Slika 6. Blok šema celog sistema

Literatura

ⁱ Copyright © 2015Terasic Technology. (2015). *DE1-SoC User Manual*. www.terasic.com: Altera.

ⁱⁱ (<https://www.engineersgarage.com/vhdl/feed-back-register-in-vhdl/>)