# GENERATOR PSEUDOSLUČAJNE SEKVENCE BAZIRAN NA MIKROKONTROLERU

**M. Nedeljković[1], M.Stojčev[2]**

***Rezime:*** *Generatori pseudoslučajnih sekvenci se koriste za testiranje složenih integrisanih kola i sistema u elektronici. U odnosu na način generisanja test sekvenci razlikuju se dva osnovna tipa. Prvi je baziran na Fibonacci a drugi na Galois konceptu. Uglavnom se ova kola realizuju kao hardverski blokovi specijalne namene, koji imaju test sekvence fiksnih dužina. Napretkom elektronike omogućeno je da se ova kola deanas realizuju i kao programibilna. U ovom radu opisana je struktura jednog programabilnog generatora pseudoslučajnih sekvenci baziranog na 8-bitnom mikrokontroleru PIC16F877. Osnovna karakteristika opisanog rešenja sastoji se u fleksibilnom programskom definisanju dužine sekvenci.*

***Ključne reči:*** *Generator pseudo slučajne sekvence, PRNG, LFSR*

## 1. UVOD

Generator oblika (Pattern Generator-PG) je kolo tipa konačni automat (Finite State Machine-FSM) koje osim taktne pobude ne prihvata druge spoljašnje ulaze. PG se standardno konfiguriše kao pomerački registar sa izvedenim povratnim spregama. Za PG kažemo da je linearno kolo ako su njegove grane za povratnu spregu implementirane samo pomuću isključivo-ili logičkih kola (Exclusive Or , XOR, logic gates). U ostalim slučajevima za PG kažemo da je nelinearno kolo. Zbog svoje relativno niske složenosti, velike brzine rada, i pogodnosti za implementaciju u VLSI (Very Large Scale of Integration) tehnologiji najveći broj linearnih PG-ova se realizuju danas kao aplikaciono specifična integrisana kola (Application Specific Integrated Circuits- ASICs) poznatih pod imenom pseudoslučajni binarni generatori sekvenci (Pseudo Random Binary Number Generators- PRNGs) [1]. Najjednostavniji i najčešći metod realizacije PRNG-a je onaj koji se zaniva na njegovoj implementaciji pomoću pomeračkog registra sa linearnim povratnim spregama (Linear Feedback Shift Register). LFSR-ovi su ključni gradivni blokovi velikog broja digitalnih sistema kod kojih je potrebno generisati pseudoslučajne bit sekvence. Aplikacije koje pokrivaju ova kola su tipična za kriptografiju, procenu BER-a (Bit Error Rate) kod bežičnih komunikacionih sistema koji se baziraju na tehnici prenosa signala u proširenom spektru, testiranju integrisanih kola, itd [2], [3].

Napredkom tehnologije u izradi integrisanih kola, a posebno zahvaljujući tehnici skaliranja dimenzija tranzistora stvorili su se ralni uslovi da elektronska kola rade u multigigahercnom frekventnom području. Posebno je ovaj napredak evidentan na polju primene mikroprocesora. Ilustracije radi Pentium IV radi na 4 GHz, što zači da je njegova taktna pobuda 250 ps, a zahvaljujuće primeni protočnosti, ugradnje keša i predikcije grananja, on u proseku jednu instrukciju izvršava za 300 ps ( CPI ≈ 1,2). Ovakva brzina rada u izvršenju instrukcija, u najvećem broju slučajeva, opravdava projektovanje digitalnih

[1] *Dip. Ing. Miodrag Nedeljković. Elektronski fakultet, 18000 Niš, A. Medvedeva 14a*
[2] *Prof. Dr. Mile Stojčev. Elektronski fakultet, 18000 Niš, A. Medvedeva 14a*

sistema korišćenjem softverskih tehnika. Naime, umesto da PRNG-ove bazirane na LFSR-ovima realizujemo kao ASIC kola svrsishodnije je realizovati ih pomoću mikroprocesora, tj programski [4]. Evidentno je pri ovome da će brzine rada softverski realizovanih LFSR-ova biti niže, ali značajna prednost ovakvog pristupa ogleda se u većoj fleksibilnosti rešenja, tj lakoj izmeni strukture kola a time i generatora sekvenci.

U ovom radu razmatraće se jedno rešenje programske realizacije LFSR-a baziranog na mikrokontroleru PIC16F877. Cilj autora u toku realizacije ovog kola je bio prvenstveno usmeren ka sagledavanju složenosti rešenja iskazanog preko obima kôda što je od primarne važnosti kod embedded sistema koji imaju ograničeni broj resursa.

## 2. TIPOVI GENERATORA PSEUDOSLUČAJNIH BINARNIH SEKVENCI

Već smo naglasili da je PRNG autonomni FSM. Na početku rada (faza inicijalizacije) stanje njegovih flip-flopova (gradivni blokovi LFSR-a) se postavljaju u poznata stanja. Nakon toga, u normalnom režimu rada, njegovo funkcionisanje se karakteriše generisanjem repetitivnih sekvenci prelaznih stanja. Vrednosti koje definišu promenljive stanja se koriste kao test vektori u sledea dva načina rada: U paralelnom načinu rada (parallel mode), u datom taktnom intervalu, vrednosti promenljivih koje su prisutne na $v$-out-of $n$, za $v \leq n$, izlazima LFSR-ovoh flip-flopova se koriste kao test vektor. U serijskom načinu rada (serial mode), vrednosti koje se u toku v uzastopnih taktnih intervala generišu na izlazu specifičnog LFSR-ovog flip-flopa se koriste kao $v$-bitni test vektor.

Sa aspekta taksonomije PRNG-ove možemo klasifikovati na osnovu dva ravnopravna medjusobno nezavisna kriterijuma. Prvi se bazira na raspoloživosti promenljivih u paralelnom ili serijskom obliku, a drugi u odnosu na metod implementacije kola za generisanje pseudoslučajnih sekvenci. Shodno tome razlukujemo sledeca dva pristupa koji se odnose na generisanje pseudoslučajnih brojeva: Prvi se bazira na *Fibonacci* pristupu, a drugi na *Galois* konceptu. Na slici 1a) prikazan je način formiranja PRNG-a koji je karakterističan za *Fibonacci* koncept , a na slici 1 b) PRNG realizovan koristeći *Galois*-ov pristup.

Na slici 1 elemenat označen sa $Ri, i=1,.., n$, odgovara memorijskom elementu, tj. D flip-flopu, elemenat označen sa $Cj, j=1,..,n-1$, označava prekidač koji može biti u stanju ON (postoji povratna veza) ili OFF (ne postoji povratna veza), i element označen simbolom $\oplus$ koji predstavlja XOR logičko kolo.

Analizom slike 1 može se uočiti sledeće: Svaka *Fibonacci*-jeva implementacija LFSR-a može se jednostavno transformisati u *Galois* LFSR, na sledeći način.
Označimo skup povratnih veza za *Galois* generator kao:

$$[f_1, f_2, \ldots, f_i]_g$$

gde $i$ označava broj povratnih sprega (izuzev $g_0$), $f_1=m$ je povratna veza najvišeg reda, a indeks $g$ označava *Galois* LFSR. Skup povratnih veza za ekvivalentni *Fibonacci*-ev LFSR glasi:

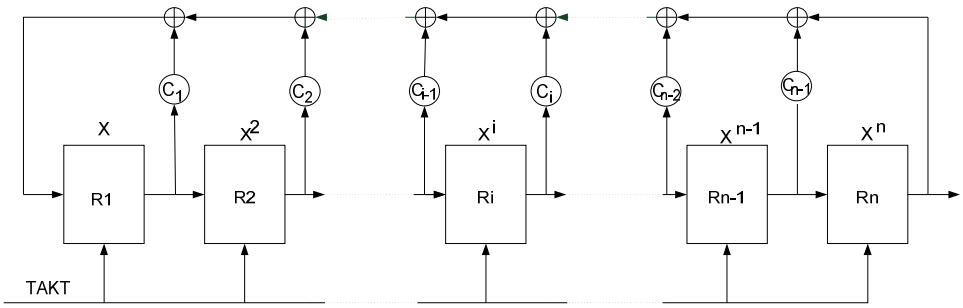
$$[f_1, m-f_2, m-f_3, \ldots, m-f_i]_f.$$

Na primer za $m=3$ i $g_8, g_6, g_5, g_4$ i $g_0$ imamo:

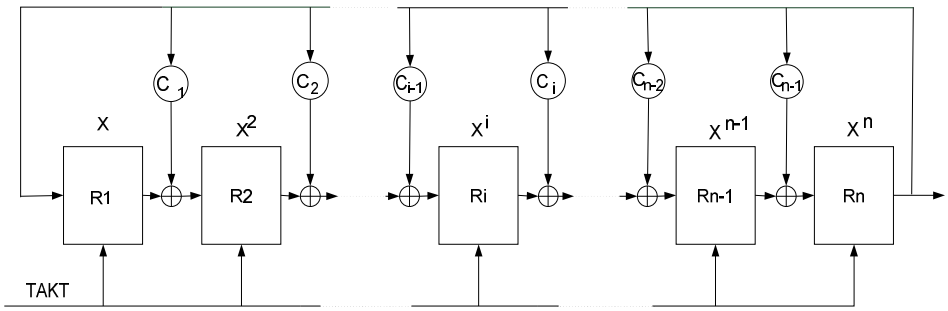
$$[8,6,5,4]_g \text{ i } [8,8-6,8-5,8-4]_f = [8,2,3,4]_f = [8,4,3,2]_f.$$

Koju implementaciju ćemo da koristimo zavisi od projektanskih zahteva. *Fibonacci*-jeva implementacija je koncepcijski lakša zato što se samo ulazni bit u pomeračkom

registru menja sa promenom taktnog signala. Međutim *Galois* implementacija se standardno koristiti za rad pri višim frekvencijama. Brzina rada je posebno kritična kada u




***Slika 1a).*** *PRNG-a koji je karakterističan za Fibonacci koncept*




***Slika 1b).*** *PRNG-a koji je karakterističan za Galois koncept*

lancu povratne sprege postoji veći broj XOR kola. Kod *Fibonacci* pristupa XOR kola su kaskadno povezana pa je propagaciono kasnjenje signala kroz LFSR veće, a kod *Galois* izlazi XOR kola se vezuju paraleno pa je zbog toga propagacija signala manja, tj brzina rada veća.

## 3. SOFTVERSKA REALIZACIJA LFSR-a

Kao što smo već napomenuli, u ovom radu biće prezentirano jedno rešenje LFSR-a bazirano na mikrokontroleru PIC16F877. Osnovna karakteristika predloženog rešenja ogleda se u tome što je dizajn implementiran softverski. Softversko rešenje ima veći broj prednosti pri čemu su najbitnije sledeće:

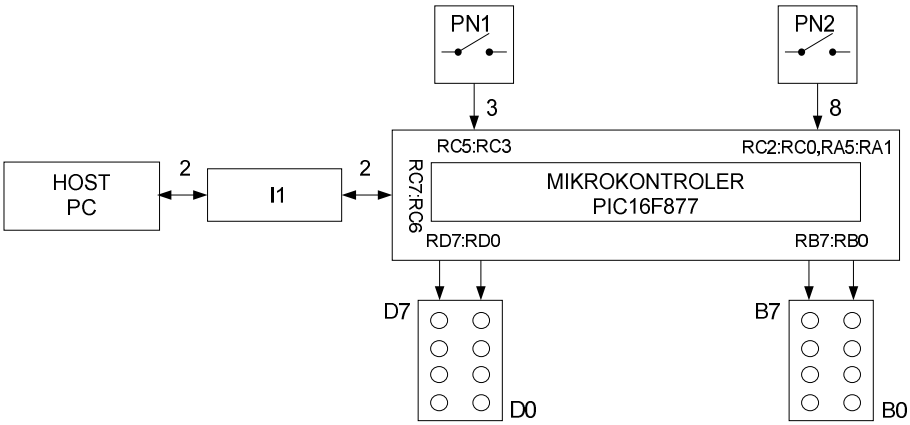
1. Izbor modela LFSR-a baziran na konceptu *Fibonacci* ili *Galois* se izvodi programski bez potrebe da se vrše bilo kakve modifikacije ili intervencije u hardveru.
2. Izbor reda polinoma i broja članova kojima se definišu povratne veze se izvodi softverski na zahtev korisnika.
3. Definisanje inicijalnog stanja se odredjuje programski.

Nedostatak softverske realizacije je:

1. Sporiji rad

## 4. BLOK ŠEMA

Hardverska realizacija PRNG-a prikazana je na slici 2.




***Slika 2.*** *Blok šema PRNG-a*

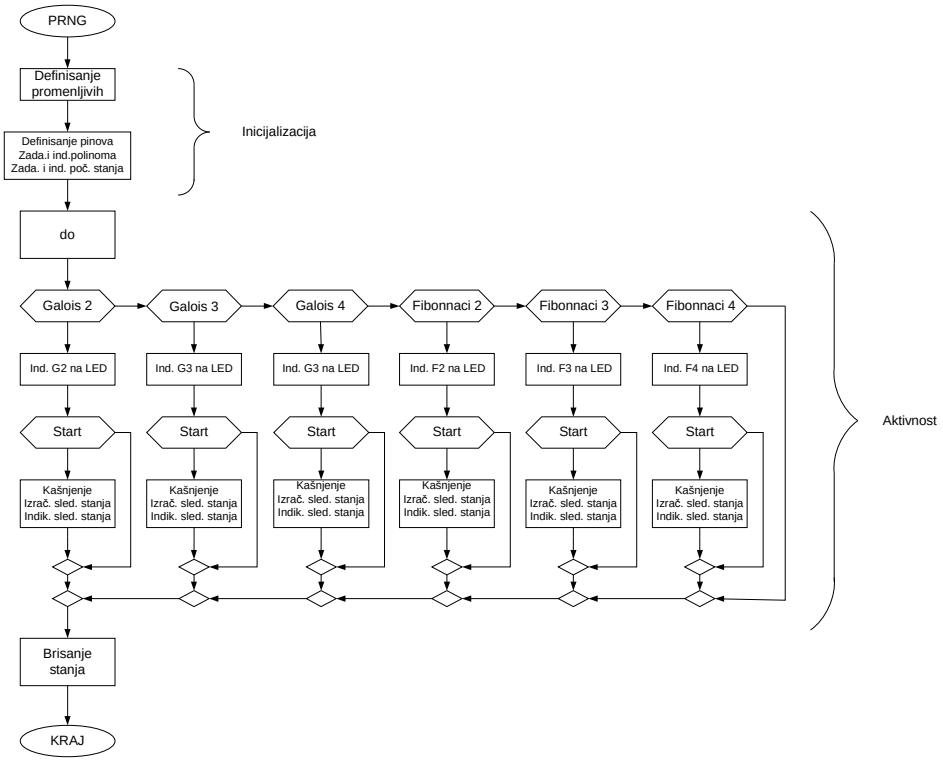
Sistem čine sledeći blokovi: mikrokontroler, kao centralna procesorska jedinica, koji izvršava program za generisanje pseudoslučajnih sekvenci, interfejs I1, host- tipa PC mašina, sistem lokalnog upravljanja PN1 i PN2, i LED indentifikatori. Sprega sa host računarom se ostvaruje preko interfejsne logike I1 (RS232). Komande koje se izdaju preko hosta se odnose na sledeće aktivnosti:

1. Postavljanje flip-flopova u inicijalno stanje.
2. Definisanje tipa LFSR-a.
3. Definisanje reda polinoma.
4. Iniciranje test sekvence.
5. Prihvat konačnih rezultata.

Pored daljinskog, preko hosta, postoji i lokalno upravljanje PRNG-om koje se realizuje u zavisnosti od stanja u kojima su postavljeni prekidači PN1 i PN2. Preko grupe prekidača PN1 se zadaje stepen polinoma i *Fibonacci* ili *Galois*, kao i koncept implementacije, a preko grupe prekidača PN2 definišu se inicijalno stanje i povratne sprege. Kao elementi za prikazivanje toka generisane sekvence se koriste LED diode D7-D0 (ukazuju na povratne sprege i početno stanje), a LED diode B7-B0 (signaliziraju da li je implementiran princip *Fibonacci*/*Galois*, stepen polinoma, i sekvencu).

## 5. PROGRAMSKA PODRŠKA

Dijagram toka rada sistema prikazan je na slici 3.




***Slika 3.*** *Dijagram toka sistema*

Na početku rada vrši se definisanje promenljivih koje uključuje postavljanje izlaza D flip-flopova i definisanje vrednosti pomoćno promenljivih. Nakon ovoga sledi dalje izvršenje faze `Inicijalizacija` koja obuhvata sledeće aktivnosti:

a) definisanje pinova (ulaz/izlaz) mikrokontrolera,
b) unos reda polinoma,
c) unos povratnih sprega,
d) postavljanje inicijalnog stanja,
e) izbor metoda, i
f) i aktiviranje indikatora.

Faza `Aktivnost` predstavlja struktura tipa višestruko grananje (multiway-branch). U okviru svake grane obavljaju se slične aktivnosti, dok se grananja odnose na izbor metoda generisanja pseudoslučajnih sekvenci odgovarajućeg stepena. Bez umanjenja opštosti u konkretnom rešenju su prikazane samo aktivnosti koje se odnose na sekvence koje su tipične za *Galois* stepena 2, 3, 4 i Fibonacci stepena 2, 3, 4. Imajući u vidu da su aktivnosti u okviru generatora pseudoslučajne sekvence indentične za sve prikazane slučajeve, ukazaćemo samo na jednu od njih.

Nakon grananja na *Galois* metod realizovan polinomom drugog stepena vrši se indikacija metoda (*Galois*) i indikacija stepena polinoma (2) na LED indikatorima. Potom se vrši čekanje dozvole za generisanje pseudoslučajne sekvence, a kada se ona aktivira prelazi se u proces izračunavanja sledećeg stanja flip-flopova. Nakon vremenskog kašnjenja

od 250 ms (iznos kašnjenja je izabran radi bolčje vizielizacije rezultata) novo generisano stanje se prikazuje na LED indikatorima. Za slučaj da se želi promena bilo kog parametra vrši se prvo brisanje tekućeg stanja, a zatim zadavanje novih.

Program je kreiran u mikroC-u i sadrži 171 liniju uzimajući u obzir i komentare. Kompajliran program zauzima 1734 bajta memorije, odnosno 21% od 8 kB raspoloživog memorijskog prostora, što je prihvatljivo kao rešenje za embedded aplikaciju.

## 6. ZAKLJUČAK

U radu je opisana praktična realizacija PRNG-a baziranog na mikrokontroleru 16F877.

U odnosu na klasična rešenja koja se baziraju na ASIC konceptima,predloženo rešenje je softverski implementirano.

Osnovne prednosti ovakvog rešenja su fleksibilnost u pogledu jednostavne i lake izmene programske strukture koja se odnosi na izmenu tipa LFSR-a (baziran na konceptu *Fibonacci* ili *Galois*), stepena polinoma, oblika polinoma (broj članova polinoma), i definisanja inicijalnog stanja flip-flopova LFSR-a koje neće dovesti do blokiranja rada PRNG-a.

S obzirom da je rešenje softverski implementirano ono je pogodno za testiranje i ugradnju u sistemima čije je vreme odziva reda mili-sekundi kakvi su ono koji se koriste za testiranje elektronskih sistema koji rade u interaktivnom radu, sistemima koji se primenjuju u kriptografiji, sistemima za edukaciju studenata, i td.

## LITERATURA


[1] Wang L.-T., Wu C.-W., Wen X., eds, "*VLSI Test Principles & Architectures*", Morgan Kaufmann, 2006, San Francisco

[2] Wang L.-T., Stroud C., Touba N., eds, "*System-on-Chip Test Architectures*", Morgan Kaufmann, 2007, San Francisco

[3] Derickson D., Müller M., eds, "*Digital Communications Test and Mesurement*", Prentice Hall, 2008, New York

[4] Vahid F., "*Digital Design*", John Wiley& Sons, 2007, New York


# Pseudo-Random Sequence Generator Based on Microcontroller


***Abstract:*** *Pseudorandomsequence generators are standardly used today for testing complex integrated electronic circuits and systems. In respect to the principle of test sequence generation we distinguish two types of sequence generators. The first one is based on Fibonacci, while the second on Galois concept. In general these circuits are realized as special purpose hardware blocks with test sequences of fixed wifth. Recent advance in digital electronics allows us to implement these circuits as programmable structures. A structure of one programmable pseudorandom sequence generator is described in this paper. It is based on a microcontroller PIC16F877. The main feature of the proposed solution deals with fast operatio and flexibility in respect to test sequence width definition in a program way.*



***Key words:*** *Pseudorandom number generator, PRNG, LFSR*