Mikrokontrolerski Računarski Sistemi

Projekat

Viktor Zivkovic

June 2024

1 Zadatak

Napisati program kojim se vrši izvlačenje nasumičnih brojeva (**loto**). Korišćenjem Linear Feedback Shift Register (LFSR) logike implementirati generator pseudo random brojeva u opsegu 0-31. Generisati brojeve odgovarajućom brzinom i ispisivati u BCD formatu na LED displeju. Kada se pritisne taster prekinuti generisanje i zadržati trenutni broj na displeju, i proslediti izvučeni broj preko UART-a na računar. Pristikom na drugi taster ponovo se pokreće generisanje i ispis pseudo random brojeva. U narednom izvlačenju obezbediti da se već izvučeni brojevi ne mogu ponovo pojaviti. Nakon 7 izvučenih brojeva, pritiskom na drugi taster se ispisuje poruka "En", a na UART se šalje poruka poruka koja sadrži karakter za novi red. Tokom izvlačenja brojeva moguće je pritisnuti treći taster koji se ponaša kao reset i koji pokreće izvlačenje ispočetka i šalje na UART karakter za novi red.

2 Generisanje slučajnih brojeva

Potrebno je generisati slučajne brojeve izmedju 0 i 31 konstantno, pa pritiskom na taster zadržati("izvući") trenutno generisan broj.

2.1 PRBS i hardver koji imamo

Generisaćemo pseudo-slučajne brojeve koristeći metod PRBS (Pseudo Random Binary Sequence), zato što je brz i implementira se pomocu Linear Feedback Shift Registra, što je i traženo u zadtaku. MSP430 ima implementiranu logiku LFSR-a u modulu za CRC (User's guide, 14. poglavlje), koji ćemo i koristiti. Medjutim, CRC modul ima implementiran samo polinom za CRC-CCIT standard. $x^{16} + x^{12} + x^5 + 1$ I to koristeći Galois implementaciju. Vikipedija kaže da možemo koristiti i Galois implementaciju za PRBS, samo neće krenuti iz iste tačke, sto nama i nije bitno. Ako dovedemo bajt podatak $0x00$ na *Data In*, dobijamo 8 cifara prbs sacuvanih u *CRCINIRES* registru, i tako cemo generisati PRBS.

2.2 Slučajni broj od PRBS

PRBS odnosno pseudo slučajna binarna sekvenca naravno nije slučajni broj, ali ako uzmemo samo 5 njenih bita, mozemo ih posmatrati kao slučajan broj izmedju 0 i 31, što cemo i da uradimo.

2.3 Koliko je ovo dobro rešenje?

Koristeći ovaj sajt, ubacio sam polinom koji imamo u binarnoj formi (1000100000010000, bez poslednje 1) u kalkulator i dobio periodu ponavljanja od 32767, što je tačno pola od maksimalne ($2^6-1 = 65535$). Mislim da je ovaj rezultat zadovoljavajući. Za našu primenu ovo je dovoljno velika perioda ponavljanja, pošto mi koristimo 8 po 8 bita za generisanje slučajnih brojeva, ponvaljaće se sa periodom od oko 40000 brojeva. Ovo je totalno u redu, čak i za državnu lutriju, s tim da ima jedno izvlačenje nedeljno, sa 7 brojeva, to znači da bi se ponovili brojevi posle 10 godina.
Naravno, PRBS nije dobro rešenje za državnu lutriju jer je iz dovoljno dugačke njene sekvence moguće rekonstruisati pocetne parametre (source) i polinom koji se koristi za generišanje, što bi dalo prilike za varanje. :)

Sa druge strane velika prednost ove implementacije je što je neverovatno brza, generisanje slučajnog broja traje samo jednu periodu signala takta!

2.4 Nedostatak ove implementacije

Jedan veliki nedostatak ove implementacije je to što je pocetno stanje deterministički odredjeno u kodu i isto je prilikom svakog pokretanja programa. Što znači da svaki put kada se nas mikrokontroler resetuje, brojevi će biti generisani istim redom, što u zavisnosti od primene može predstavljati veliki problem.

3 Reference

- Ben Eater - How do CRCs work?
- Ben Eater - Hardware build: CRC calculation
- LFSR
- CRC - Wikipedia
- PRBS - Wikipedia
- LFSR - Wikipedia
- PRBS kalkulator