

# Mikroračunarski sistemi za rad u realnom vremenu Projekat 21.3

## Kvadratni koren (*Square root, SQRT*)

Studenti: Nenad Petrović EE69/2018, Dušan Vincilir EE227/2018

**Tekst zadatka projekta:** Razviti Linux drajver i odgovarajuću aplikaciju koja koristi drajver i demonstrira kontrolu i upravljanje *sqr*t IP jezgrom.

**Opis strukture hardvera:** Četiri identična *sqr*t IP jezgra (zelena boja na blok dijagramu) su memorijski mapirana u adresni prostor ZYNQ procesora. *Sqr*t jezgro služi za računanje celobrojnog kvadratnog korena. IP jezgro koristi *brute-force* algoritam, pri čemu jezgru treba "n" mašinskih ciklusa za računanje kvadratnog korena broja "n" [vreme izvršavanja je linearno  $O(n)$ ]. Sva jezgra na fpga čipu rade na frekvenciji od 100MHz. Nakon što *sqr*t ip jezgro izračuna kvadratni koren, ono šalje *interrupt* prekidni signal procesoru (ljubičasta boja na blok dijagramu), čime signalizira da je obrada podatka završena.

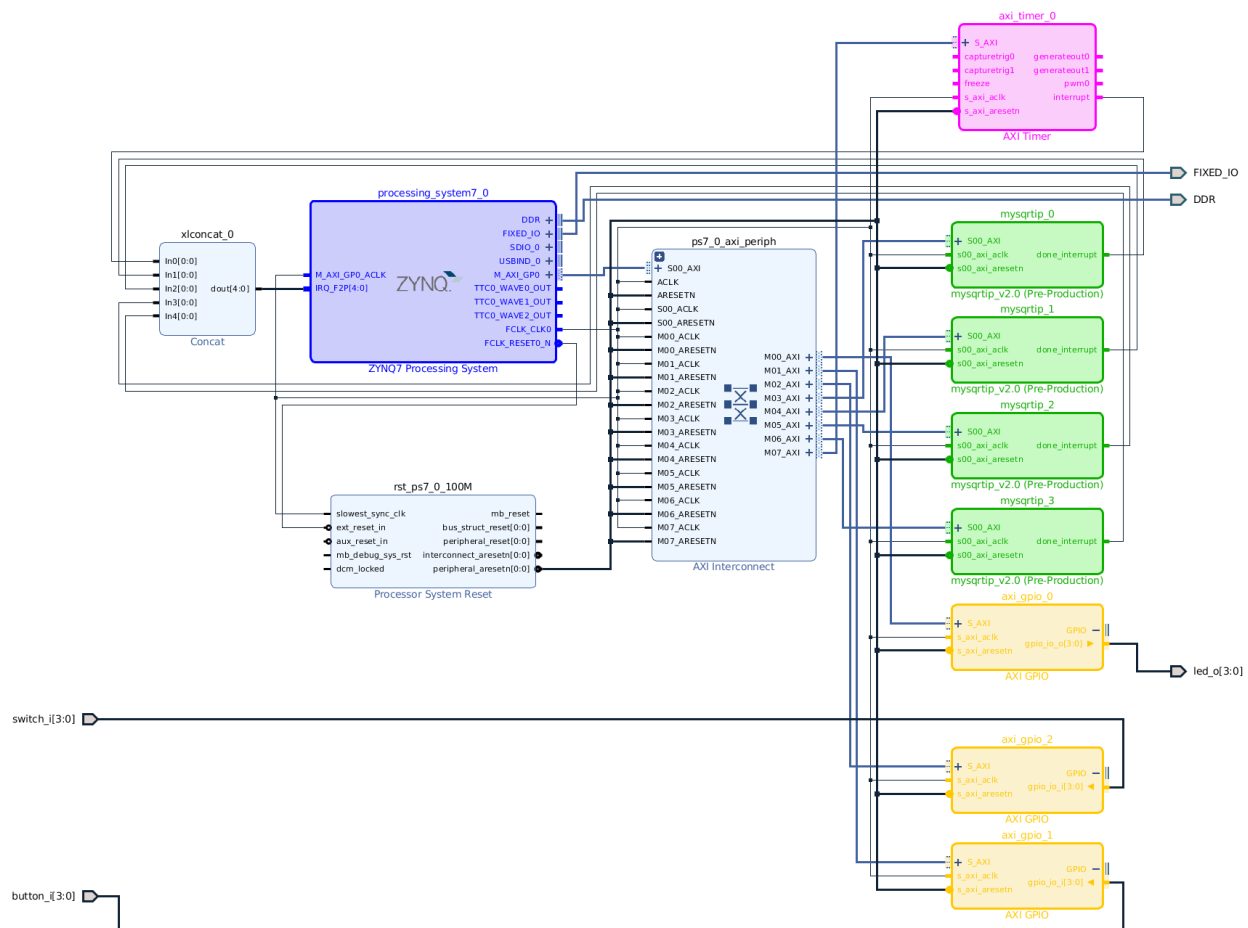
Offset	Registar
0	X
4	start
8	Y
12	ready
16	interrupt_flag

Sqr jezgro ima četiri registra: "X", "start", "Y" i "ready" koja su memorijski mapirana i kojima se pristupa uz baznu adresu *sqr*t modula i ofseta za određeni registar (videti ofsete registara u datoj tabeli). Procesor može da upisuje i čita iz ovih registara preko AXI Lite interfejsa. U "X" registar se smešta 32-bitni broj čiji se koren traži. "Start" registar služi za pokretanje jezgra i početak računanja korena. Kontrolni registar "start" je potrebno postaviti na jedinicu kako bi računanje otpočelo, a neposredno nakon toga na nulu kako ne bi po završetku

računanja *sqr*t ponovno krenuo da računa koren za isti broj "X". U "Y" registar će, po završetku računanja, biti smešten rezultat, tj. kvadratni koren broja "X". Statusni registar "ready" govori da li je *sqr*t završio sa računanjem, tj. da li je spreman da krene sa novim računom. Kako bi se znalo da li je *sqr*t završio sa računom, i da je vrednost u "Y" registru validna, može se ili periodično čitati "ready" registar dok se u njemu ne pojavi jedinica (*polling*) ili se može detektovati prekidni signal (*interrupt*). Ukoliko se koriste prekidi, potrebno je u prekidnoj rutini resetovati *interrupt\_flag* upisom jedinice u njega.

Sqr kao "X" prihvata bilo koji celobrojni broj do  $2^{30}$ . Ukoliko se prosledi broj koji nema celobrojan kvadratni koren, rezultat će biti zaokružen na niže (npr.  $X=12 \rightarrow Y=3$ ).

Sem ovakva četiri jezgara, postoji i jednostavno jezgro za paljenje/gašenje dioda (zelena boja).



0. Koristeći uputstvo koje se može pronaći na stranici predmeta, pripremiti SD karticu sa svim potrebnim datotekama (BOOT.bin, devicetree.dtb i ulmage) koje omogućavaju podizanje Linux operativnog sistema, sa gore opisanim dizajnom uključenim u BOOT.bin datoteku

2. Napisati Linux drajver za *sqr*t modul i propratnu aplikaciju sa sledećom funkcionalnostima:

Drajver dobija slobodne upravljačke brojeve (MAJOR, MINOR) od operativnog sistema. Potrebno je pri pokretanju insmod funkcije, automatski kreirati node fajl /dev/sqr. Drajver treba da istovremeno kontroliše četiri *sqr*t modula, kao i gpio modul koji se koristi za kontrolu dioda.

Drajveru se preko terminala šalje niz celih brojeva u decimalnom formatu odvojeni zarezom (niz može biti proizvoljne dužine):

```
echo "16,255,9,10000,176400,16000000,25,1073741824" > /dev/sqr
```

Drajver treba da pomoću četiri *sqr*t jezgra u što kraćem vremenskom periodu izračuna kvadratne korene ovih brojeva i rezultate sačuva kao niz brojeva. Koristiti prekidni signal za detektovanje kada je određeno

jezgro završilo sa računanjem broja, te ukoliko postoji još brojeva za računanje, odmah dodeliti novi broj tom jezgru. Sva četiri jezgra trebaju da rade istovremeno u paraleli, sem u slučaju kada ostane manje od četiri broja za obradu.

Budući da postoje četiri *sqrt* jezgra i četiri diode, ukoliko neko jezgro *sqrt* krene da procesira podatak, njemu dodeljenu diodu je potrebno upaliti. Po detektovanju završetka obrade podatka je potrebno istu ugaziti. Sa ugrađenom ovom funkcionalnošću, moguće je pomoću dioda videti koja su jezgra aktivna u bilo kom datom trenutku.

Čitanje vrednosti iz drajvera pomoću *cat /dev/sqrt* vraća izračunati niz podataka u sledećem formatu (npr. za prethodni primer):

```
cat /dev/sqrt
```

```
16:4, 255:25, 9:3, 10000:100, 176400:420, 16000000:4000, 25:5, 1073741824:32768
```

Pri pozivu *cat* komande se ispisuju samo brojevi koji su do tog trenutka proračunati.

Ako drajver trenutno obrađuje niz podataka koji mu je prethodno prosleđen, te mu se prosledi novi niz, drajver treba biti spreman da prihvati te brojeve, te da ih obradi i sačuva pored brojeva koje trenutno obrađuje.

**2.** Napraviti aplikaciju u C programskom jeziku koja stres-testira drajver tako što u nasumičnim intervalima od 1-5 sekundi šalje nizove nasumične dužine (4-8 članova) nasumičnih brojeva (koji imaju celobrojne kvadrate, te su u dozvoljenom opsegu). Aplikacija treba takođe u nasumičnim intervalima od 3-5 sekundi čita do tog trenutka izračunate brojeve i proverava dobijene rezultate (parove x:y).

*Napomena: Kompletan razvoj drajvera održavati na proizvoljnom git repozitorijumu.*