Mikroračunarski sistemi za rad u realnom vremenu

Projekat 1: Aritmetičko Logička Jedinica (ALU)

Primenjena elektronika

Studenti: Jovan Slavujević, Lazar Vučković, Bogdan Todorović

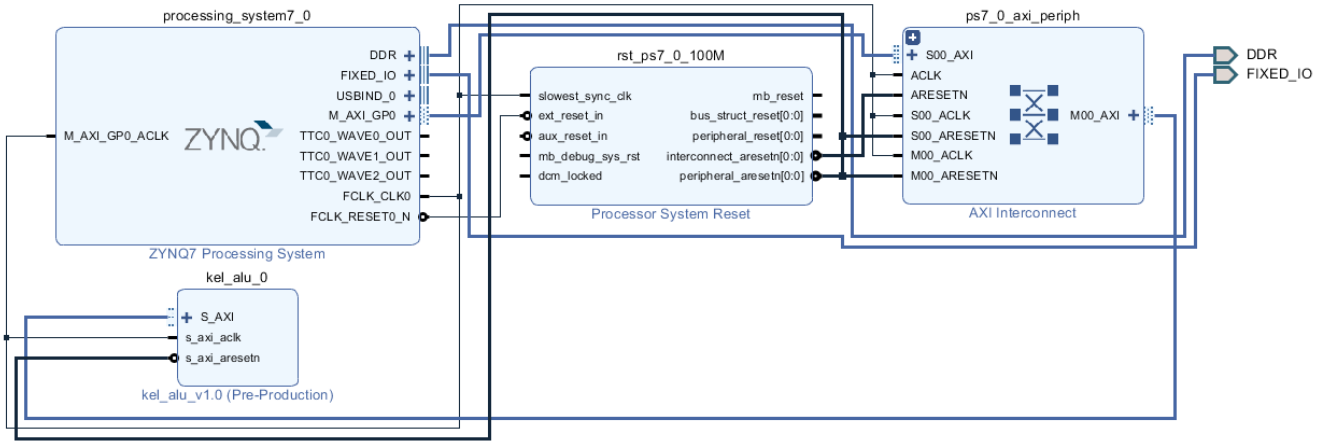
Tekst zadatka projekta: Razviti Linux drajver i aplikaciju koja koristi drajver u cilju kontrole jednostavne aritmetičko logičke jedinice (ALU), čija je funkcionalnost opisana u tabeli ispod.

op_sel	Operacija	Opis
000	Res = A & B	Logičko AND
001	Res = A | B	Logičko OR
010	Res = A + B	Sabiranje
011	Res = A - B	Oduzimanje
100	Res = shl A	Pomeranje u levo registra A
101	Res = shr A	Pomeranje u desno registra A
110	Res = rol A	Rotiranje u levo registra A
111	Res = ror A	Rotiranje u desno registra A

Opis funkcionalnosti: *A*, *B, op_sel* i *Res* su registri ALU mapirani na adresama 0x43C00000, 0x43C00001, 0x43C00002 i 0x43C00003, respektivno. Kao što se može videti iz tabele iznad, prve četiri operacije zahtevaju oba operanda, dok poslednje četiri koriste samo operand *A*. ALU je realizovan kao kombinaciono kolo, bez kašnjenja, tako da se rezultat operacije može pročitati neposredno nakon promene bilo kog operanda i/ili operacije. Svi registri ALU su 32-bitni sa sledećim dodatnim informacijama:

- u slučaju operacije oduzimanja, rezultat će biti u formatu komplementa-2 (rezultat operacije može biti negativan i pozitivan);
- kod pomeranja u levo/desno na upražnjeno mesto krajnje desnog/levog bita se upisuje 0.

Potrebne datoteke: Zajedno sa ovim zadatkom projekta, dobićete i *bitstream* datoteku dizajna, prikazanog na slici ispod (*design.bit*), kao i *hardware description file* datoteku kreiranu za ZYBO razvojnu ploču (*design_1_wrapper.hdf*). ALU za koji se razvija drajver je blok *kel_alu_v1.0*, mapiran preko AXI4-lite magistrale na adresu 0x43C00000.



Zadatak:

1. Koristeći dostavljene datoteke i uputstvo koje se može pronaći na stranici predmeta, pripremiti SD karticu sa svim potrebnim datotekama (*BOOT.bin*, *devicetree.dtb* i *uImage*) koje omogućavaju podizanje Linux operativnog sistema, sa gore opisanim dizajnom uključenim u *BOOT.bin* datoteku. (5p)
2. Napisati Linux drajver za ALU sa sledećom funkcionalnosti (ukupno 25 poena):
   - drajver dobija slobodne upravljačke brojeve (MAJOR, MINOR) od operativnog sistema, a *device file* datoteka se automatski kreira u okviru fajl sistema (/dev/alu); (2p)
   - slanje podataka ka ALU vrši se *write* operacijom nad datotekom /dev/alu. String koji se upisuje u datoteku treba da bude sledećeg formata:

     operand1;operand2;operacija

     gde su operacije *and* (za logičko and), *or* (logičko or), *add* (sabiranje), *sub* (oduzimanje), *shr* (pomeranje levo), *shl* (pomeranje desno), *rol* (rotiranje levo) i *ror* (rotiranje desno). Na primer sabiranje brojeva 10 i 15 bi se izvršilo sa:

     echo “10;15;add” > /dev/alu

     (8p)

     Operandi koji se prosleđuju mogu da budu u decimalnom i heksadecimalnom formatu, a drajver treba da prepozna u kom su formatu prosleđeni operandi. Npr, sabiranje brojeva 10 i 15 bi moglo da se izvrši i sa upisom

     echo “0x0a;0x0f;add” > /dev/alu

     (7p)

     Za operacije koje koriste samo jedan operand, format naredbe je sličan:

     operand1;operacija

     (2p)

     Na primer, rotiranje u levo broja 0x80000000 bi se izvršilo komandom

     echo “0x80000000;rol” > /dev/alu

     Napomena: Drajver treba da proveri da li se prosleđeni operandi mogu predstaviti kao 32-bitni podaci, da li je zahtevana operacija neka iz skupa {*and, or, add, sub, shl, shr, rol, ror*} i da li je, generalno, format poruke zadovoljavajući (može biti praznih mesta, ; kao delimiter). U slučaju da bilo koji od uslova od gore nije ispunjen, treba prikazati adekvatnu poruku korisniku. (3p)

- čitanje dobijenog rezultata vrši se read operacijom na datoteci /dev/alu, npr

  cat /dev/alu

  Rezultat se uvek čita kao heksadecimalni broj. Na primer, kao rezultat sabiranja brojeva 10 i 15 od gore, prethodna komanda bi vratila rezultat 0x19. (3p)

3. Napisati Linux GCC aplikaciju (ukupno 20p) koja omogućava testiranje ALU drajvera. Program se zove ALU_test i poziva se sa ./ALU_test "izraz_za_izracunavanje". Izraz koji se prosleđuje prilikom poziva će se izračunavati u iteracijama, sukcesivnim pozivanjem funkcija Linux drajvera opisanih ranije. Izraz za izračunavanje može da bude jedan od navedenih (operandi *operand1* i *operand2* prosleđeni kao decimalni ili heksadecimalni brojevi slično kao u slučaju drajvera od gore):

   operand1 & *operand2* (logičko AND dva operanda)
   operand1 | *operand2* (logičko OR dva operanda)
   operand1 + *operand2* (sabiranje dva operanda)
   operand1 - *operand2* (oduzimanje dva operanda)
   operand1 * *operand2* (množenje dva operanda)
   operand1 << *operand2*(pomeranje operand1 operand2 bita u levo)
   operand1>>*operand2*(pomeranje operand1 operand2 bita u desno)
   operand1 rol *operand2* (rotiranje operand1 operand2 bita u levo)
   operand1 ror *operand2* (rotiranje operand1 operand2 bita u desno)

   Sve gore navedene izraze treba izračunati **isključivo koristeći ALU** i odgovarajući drajver.

   (10p)

   Izraz koji se izračunava može da bude složeniji i da sadrži i zagrade (ne ugnježdene). Tako, na primer, sledeći izrazi su takođe prihvatljivi:

   0x05050505 >> (2+3)
   5*(4+3)
   8ror(7-0x03)
   (0x0F&0x02)+0x03
   (10-0x02)*(0x08>>2)

   (10p)