

PROJEKAT

Medijan filtriranje slike

Potrebno je metodu za popravku kvaliteta slike na FPGA korišćenjem VHDL jezika i demonstrirati rad na FPGA razvojnom sistemu. U ovom dokumentu je opisan zadatak koji se sastoji iz nekoliko celina koje se nadovezuju jedna na drugu.

1. deo – Slanje slike na PC računar (15 poena)

Memorija za čuvanje slike

Grayscale slika veličine 256×256 8-bitnih piksela je smeštena u inicijalizovanoj BRAM memoriji koja je data kao prateći fajl ovog dokumenta (*im_ram.vhd*). Za inicijalizaciju se koristi fajl *lenaCorrupted.dat* koji je potrebno smestiti na istu lokaciju kao i *im_ram.vhd*. Memorija je realizovana kao “Simple Dual Port” memorija, s obzirom na to da će se u kasnijim fazama zadatka koristiti jedan port za čitanje, a drugi za upis u memoriju. Dubina memorije je $256 \times 256 = 65536$, a na svakoj memorijskoj lokaciji se nalazi 8-bitna vrednost koja predstavlja intenzitet piksela. Pikseli su poređani po redovima, tj. u prvih 256 lokacija se nalazi 256 piksela prvog reda slike, u drugih 256 lokacija se nalaze pikseli drugog reda slike itd.

Sintetisati memoriju *im_ram* i uočiti da ona zauzima 16 blokova BRAM memorije, po dva za svaki bit. Dva bloka su kaskadirana na način opisan u odeljku “Cascadable Block RAM”, na 23. strani dokumenta dostupnog na linku https://docs.xilinx.com/v/u/en-US/ug473_7Series_Memory_Resources.

UART predajnik

Na raspolaganju je i fajl *uart_tx.vhd*, u okviru koga je implemenitan UART predajnik. Ova komponenta prima 8-bitni podatak sa porta *tx_data*, kada je signal *tx_dvalid* na “1”, koji označava da je podatak validan. Ako slanje prethodnog podatka nije u toku, započinje se slanje podatka *tx_data* i prelazi u zauzeto stanje, tokom koga je signal *tx_busy* na “1”. Tokom zauzetog stanja, na izlaznom pinu *tx* se jedan po jedan šalju biti brzinom koja je podešena generikom *SER_FREQ*. Prijemnik će za potrebe ovog projekta biti USB-UART konvertor koji treba povezati sa FPGA pločom. Preporučuje se brzina od 115 200 bps. Kada signal *tx_busy* padne na “0”, kursorik može postaviti novi podatak i *tx_dvalid* na “1”, čime se započinje novo slanje. Da bi se ostvarila brzina definisana generikom *SER_FREQ*, neophodno je podesiti generik *CLK_FREQ* na vrednost koja odgovara učestanosti signala takta koji se koristi u sistemu. Signal *par_en* se može postaviti na “0”.

Zadatak

Korišćenjem navedenih komponenti potrebno je implementirati čitanje slike iz memorije i slanje svih piksela slike na PC računar. TX port UART-a je mapiran na GPIO port A0 na FPGA ploči. Ovo je podešeno u pratećoj .xdc skripti koju treba po potrebi izmeniti. Na raspolaganju je Pajton skripta koja prima

podatke na strani računara i iscrtava sliku. Pajton skripta se može pokrenuti iz PyCharm-a ili bilo kog drugog okruženja, samo je neophodno instalirati paket *pyserial*.

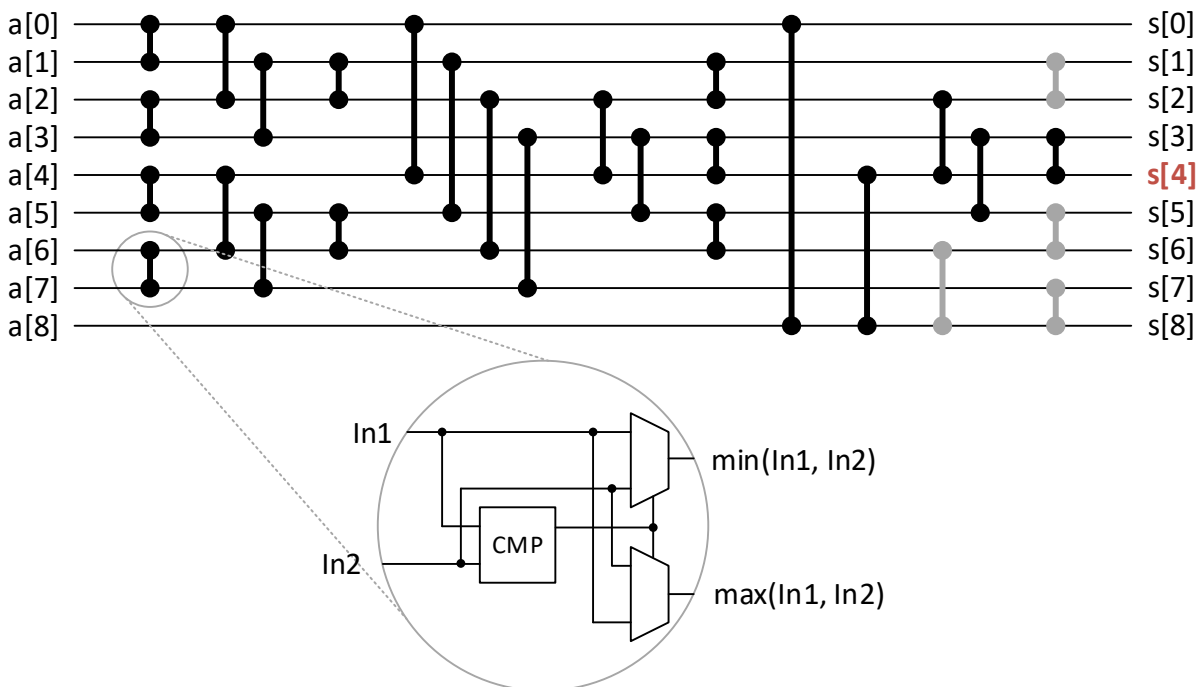
Slanje se startuje pritiskom na taster i traje sve dok se ne pošalju svi pikseli slike. Svaki put kada se pritisne taster započinje se novo slanje slike, a ako se taster pritisne u toku slanja, pritisak se ignoriše.

- Implementirati komponente potrebne za detekciju pritiska tastera.
- Implementirati komponentu za slanje podataka na PC računar.
- Simulirati sve komponente i priložiti *testbench* fajlove.
- Testirati slanje slike na računar na FPGA platformi. Potrebno je najpre pokrenuti Pajton skriptu, a zatim startovati transfer. Imajući u vidu da se šalje 65536 piskela, slanje bi trebalo da traje nešto duže od pet sekundi.

Povezivanje USB-UART modula sa FPGA pločom i računarom, kao i instrukcije za instalaciju drajvera, prikazani su na kraju ovog dokumenta.

2. deo – Mreža za izdvajanje medijane (10 poena)

Za potrebe filtriranja slike, potrebno je uraditi sortiranje devet susednih piksela i izdvojiti središnji element (medijanu). Mreža koja treba da vrši sortiranje se realizuje korišćenjem “*Odd-Even Merge*” arhitekture koja je prikazana na slici. Svaka vertikalna linija označava blok koji određuje manju i veću vrednost dva ulaza i prosleđuje ih na izlaz tako da manja vrednost bude bliža indeksu nula, a veća indeksu 8. Na izlazu cele šeme dobija se sortirani niz. Imajući u vidu da je za dalju obradu u ovom projektu potrebna samo medijana, tj. vrednost $s[4]$, blokove za poređenje označene sivom bojom nije neophodno realizovati. Čak i ako bi bili realizovani u VHDL kodu, nakon sinteze će automatski biti izoptimizovani, tj. eliminisani iz dizajna, jer nisu potrebni za generisanje izlaza.

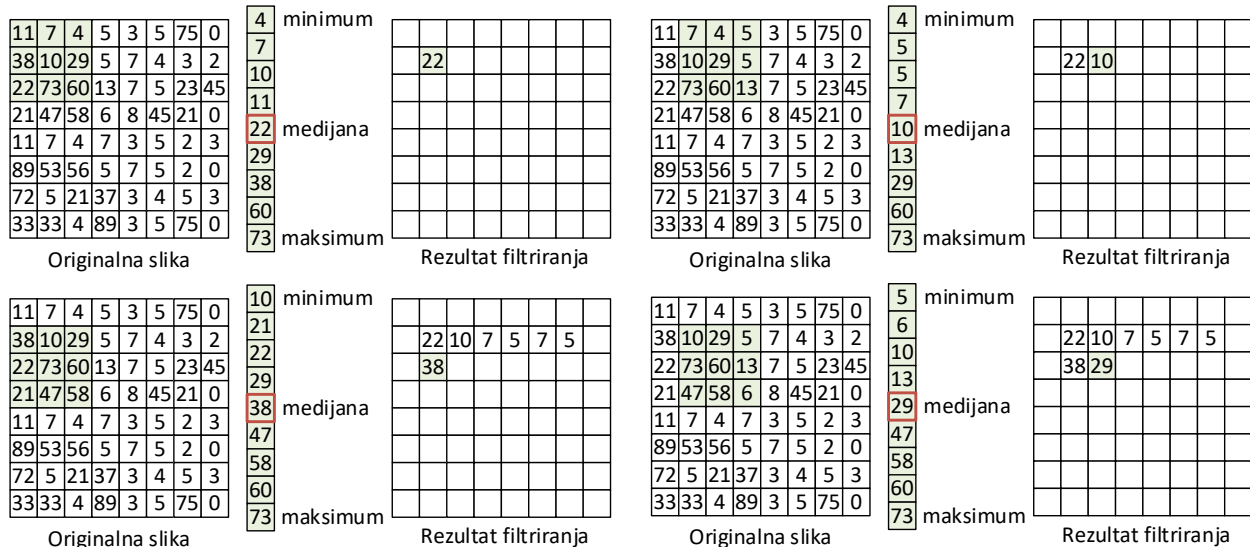


- Implementirati mrežu za izdvajanje medijane iz niza od 9 osmobičnih elemenata.
- Postaviti registre na ulaz i izlaz i proceniti maksimalnu učestanost rada ove mreže, nakon out_of_context implementacije.

- Na odgovarajuća mesta postaviti pajplajn registre kako bi se postigla veća maksimalna učestanost rada. Ispitati maksimalnu učestanost rada pajplajnovane komponente.
- Simulirati sve komponente i priložiti *testbench* fajlove.

3. deo – Medijan filter (25 poena)

Slika u memoriji je pogođena impulsnim šumom koga je potrebno ukloniti medijan filtrom. Medijan filter je filter koji piksel na slici menja medijanom vrednosti svih piksela iz unapred definisanog susedstva, uključujući i vrednost piksela koja se menja. Preporučujemo za čitanje članak sa linka: https://en.wikipedia.org/wiki/Median_filter. Primer obrade nekoliko piksela prikazan je na slici ispod.

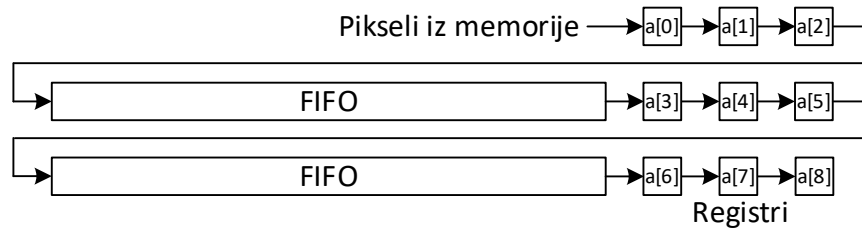


Treba obratiti pažnju da je uz ivice slike susedstvo piksela manje, tj. da neki pikseli nemaju sve susedne piksele sa svake strane. Zbog toga je slika u memoriji zapravo slika dimenzija 254x254 piksela, proširena nulama, što odgovara operaciji *medfilt2d* iz *Scipy* paketa. U Pajton skripti je, pored čitanja sa serijskog porta, dato učitavanje slike iz fajla, njeno filtriranje i prikaz rezultata.

Modul koji implementira medijan filtriranje treba da čita piksele iz RAM memorije koja je dostupna nakon završenog prvog dela zadatka. Rezultat filtriranja se mora upisati u istu memoriju, čime se originalni sadržaj memorije uništava. Rezultat filtriranja se nakon toga može lako prikazati na računaru korišćenjem serijske komunikacije, korišćenjem iste infrastrukture kao u 1. delu projekta. Na UART se mogu slati svi pikseli, uključujući i proširene piksele, dakle svih 256x256 piksela. Nema potrebe raditi isecanje značajnih piksela slike u hardveru.

Filtriranje se može realizovati na više načina, ali se preporučuje da se najpre dve linije slike baferuju u dva FIFO bafera i da se kreira struktura kao na slici. Pikseli iz memorije se čitaju na svaki takt po jedan. FIFO baferi su dubine $256 - 3 = 253$. Korišćenjem ovakve strukture postiže se da se lokalno susedstvo pomera na svaki takt i omogućava se obrada jednog piksela po taktu. Preporučuje se studentima da najpre analiziraju tok podataka pre nastavka projekta.

Implementirati medijan filtriranje slike opisano u ovom odeljku, a rezultate uporediti sa filtriranjem iz Pajton skripte. Odrediti maksimalnu učestanost rada sistema. Priložiti sve simulacione fajlove.



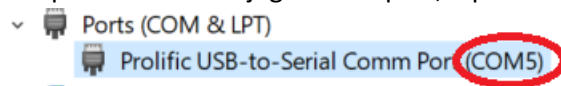
(opciono) Fleksibilna veličina maske: Modul koji filtrira sliku može da bude generički u vreme sinteze. Modul treba da ima odgovarajući generik/parametar kojim se podešava veličina maske, i to pre sinteze. Ovo nije neophodno uraditi, ali grupe koje podrže ovu fleksibilnost mogu dobiti nagradne poene u slučaju da im je potrebno za veću ocenu. Ako se podržava fleksibilnost, treba podržati tri veličine maske: 3x3, 5x5 i 7x7. **Napomena:** Imati u vidu da je fleksibilnost dodatna komplikacija projekta, pa preporučujemo da se najpre realizuje filter za veličinu maske 3x3, a tek onda uradi nadgradnja.

Povezivanje i instalacija

USB – UART drajver

Konvertor USB na UART koji ste dobili u svojim paketićima služi za komunikaciju PC računara i FPGA ploče. Za njega je potrebno instalirati drajver i to u sledećim koracima:

1. Raspakovati arhivu *Prolific PL2303 driver v3.3.2.102 (2008-24-09) Win8_x64_x86.7z*.
2. Windows + X i odabrati Device Manager;
3. U okviru *Ports (COM & LPT)* pronaći *pl2303hxa* uređaj i kliknuti desni klik na njega i odabrati *Properties*;
4. Odabrati *Driver* tab pa *Update Driver*, pa *Browse My Computer for Drivers*, pa *Allow me to choose from a list of available drivers on my computer*.
5. Kliknuti na "Have disk ..." dugme.
6. Pronaći folder u kome se nalaze fajlovi koji su raspakovani u tački 1 i odabrati fajl "ser2pl".
7. Kliknuti *OK* i nastaviti sa instalacijom.
8. Nakon instalacije u *Device Manager*-u bi trebalo da se pojavi uspešno povezan uređaj, kao na slici ispod. Zabeležiti njegov COM port, u primeru sa slike, to je COM5.



USB – UART Povezivanje sa FPGA pločom

FPGA ploča šalje podatke, pa se na USB – UART pločici koristi pin za prijem (RX). Takođe, neophodno je povezati i masu jedne i druge ploče kako bi referentni potencijal bio isti. Primeri .xdc skripti za obe ploče su dati kao prateći fajlovi ovog dokumenta, a fotografije u nastavku prikazuju povezivanje. Čip na obe ploče je isti, i to xc7z020clg400-1.

