Facial recognition

*Dokumentacija za predmet: Projektovanje složenih digitalnih sistema*

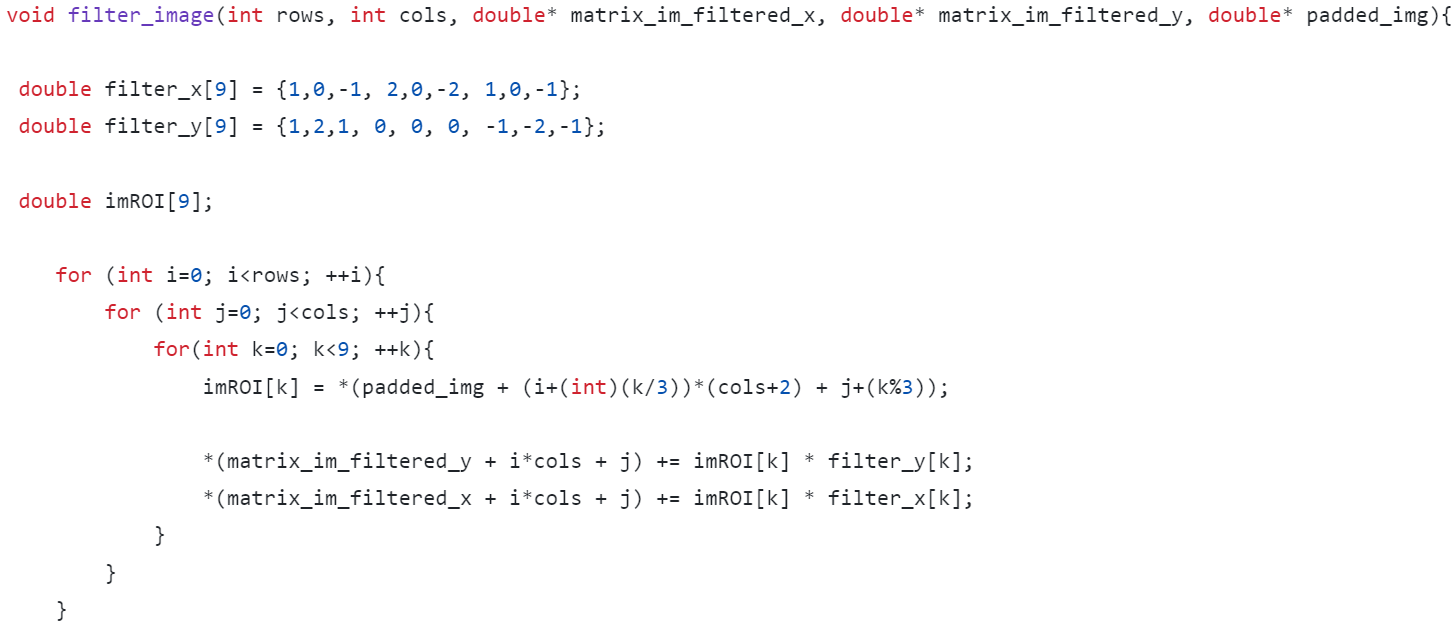
1. Opis algoritma

Ideja projekta je detektovanje ljudskog lica na fotografiji. Fotografija se prosleđuje programu, a on iscrtava okvir oko detektovanog lica. Centralni dio algoritma predstavlja filtriranje slike (po x-osi i y-osi) koje podrazumijeva množenje manjih dijelova slike sa filterima. Naime, dimenzije filtera su 3x3, te se svaki komad slike dimenzija 3x3 množi sa njime. Šablon se ponavlja sve dok se ne prođe kroz čitavu sliku pomjerajući se prvo po x-osi za 1 piksel, a nakon obrađenog reda za 1 piksel po y-osi. Ostatak algoritma sprovodi niz transformacija nad slikom čiji je krajnji rezultat skup HOG vektora. Zatim je potrebno pronaći skup HOG vektora na dijelu originalne slike koji je sličan skupu HOG vektora šablona slike. Algoritam ekstrahuje HOG šablona slike kojoj se mijenjaju dimenzije počevši od minimalne dimenzije pa do trećine minimalne dimenzije originalne slike da bi se omogućilo prepoznavanje lica bez obzira na njegovu veličinu na originalnoj slici. Sa različitim veličinama HOG-a šablona slike algoritam prolazi kroz originalnu sliku i pamti one koje su iznad određenog praga (threshold).

*Slika 1.1 Originalna slika Slika 1.2 Šablon slika Slika 1.3 Izlazna slika*

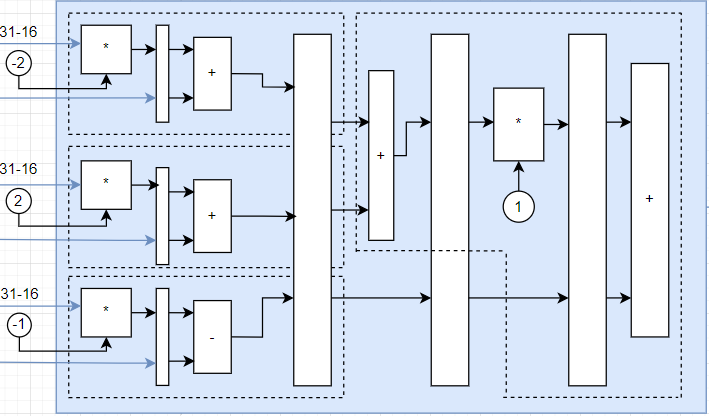
1. Implementacija algoritma u C jeziku

**

*Slika 2. Implementacija algoritma u C jeziku*

1. Optimizacija algoritma

Filter funkcija sadrži operacije množenja, sabiranja i oduzimanja, te je idealna za mapiranje na DSP module. Da bi se optimizovao algoritam, ideja je da se obrađuje 8 tačaka u paraleli što predstavlja unutrašnju k petlju. S obzirom da je kod oba filtera peti član niza nula, nema potrebe obrađivati devetu tačku jer neće uticati na rezultat. Takođe, za član filter niza koji je jednak nuli nije potrebno koristiti množač, već se piksel direktno vodi na sabirač. Da bi se izračunao jedan filter potrebno je 4 DSP bloka (*Slika 3.1*). A za dodatnu optimizaciju obrađivaćemo istovremeno 16 filtera (po 8 za oba filtera). Ovo ograničenje je posljedica nedostatka većeg broja DSP blokova na Zybo ploči.

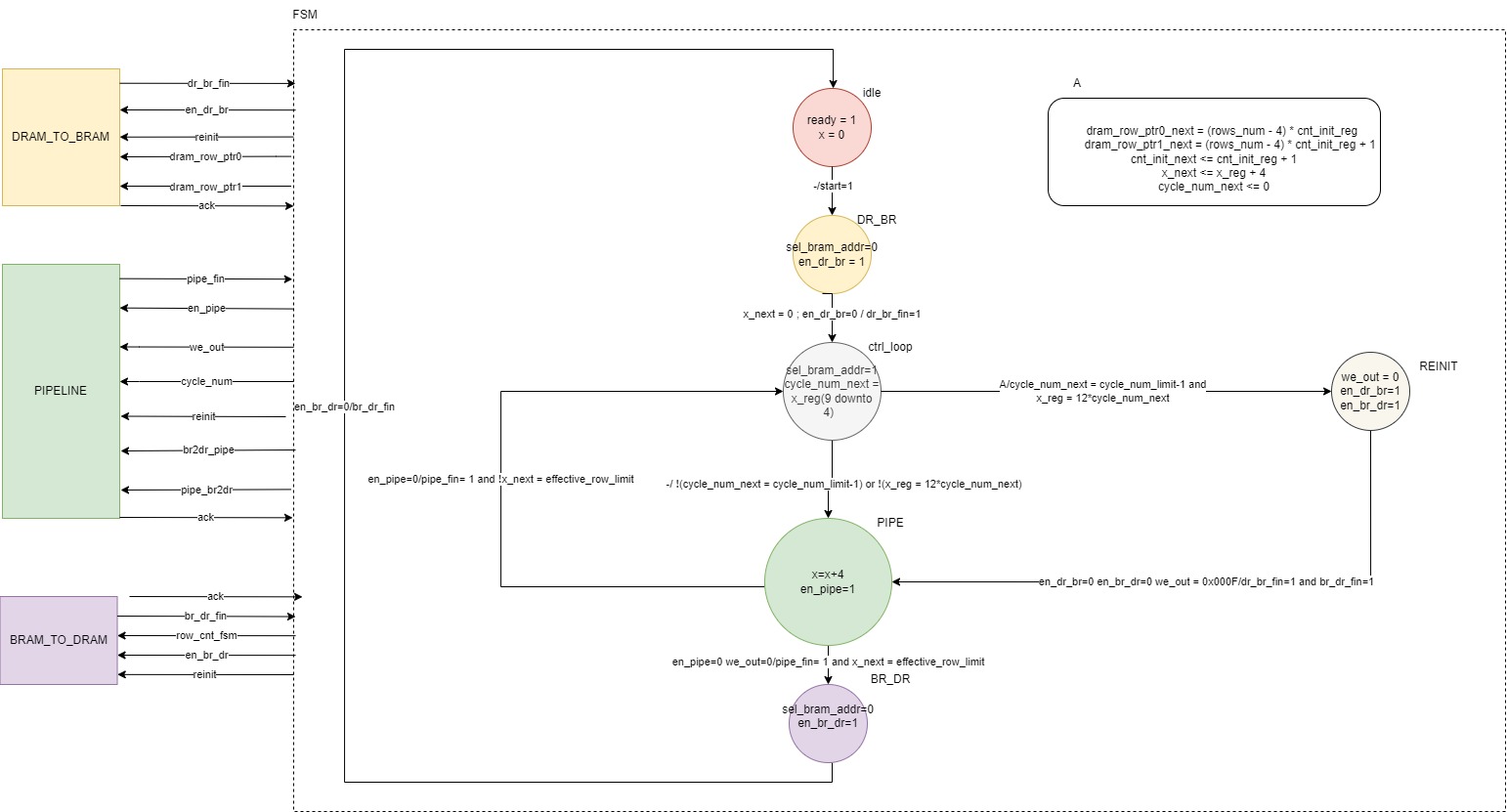


*Slika 3.1 Filter blok*

Dodatno, obrada podataka se vrši u 5 faza pipeline-a. U prvoj fazi se čitaju podaci iz BRAM blokova i uz pomoć rutirajuće mreže prosleđuju na odgovarajuće ulaze filter blokova. Zatim se filtriranje slike odvija u 3 faze, dok posljednja faza podrazumijeva upis u BRAM blokove, što je takođe kontrolisano rutirajućom mrežom.

1. Kontrola sistema

Unutar kontrole postoje 3 jedinice koje će biti usklađene od strane FSM-a. Blok šema je prikazana na *Slici 4.1*, a u nastavku će svaki modul biti posebno objašnjen.



*Slika 4.1 Blok šema kontrolne jedinice*

Sa teorijske tačke gledišta DUT može da radi nad neograničenom slikom, preciznije, maksimalne dimenzije slike ograničene su veličinom DRAM-a. Ideja je da se prebacuju redovi slike redom u BRAM blokove (njih 16), sve dok se ne popuni njihov kapacitet. Istovremeno se vuče 8 podataka iz DRAM-a preko 2 linije od 64 bita. Zatim se vode na rutirajuću mrežu koja će odrediti BRAM blok u koji se upisuju odgovarajući pikseli. Ideja je da se nakon svakog upisanog reda prelazi na upis u naredni BRAM blok. Takođe, upis se vrši preko 2 porta BRAM bloka. Cjelokupnu kontrolu nad ovom fazon obrade ima modul DRAM\_TO\_BRAM. Kontroliše adrese sa kojih se čita iz DRAMA, adrese na koje se upisuje u BRAM blokove i kontroliše mrežu za rutiranje podataka. ASM dijagram ovog bloka prikazan je na *Slici 4.2*.

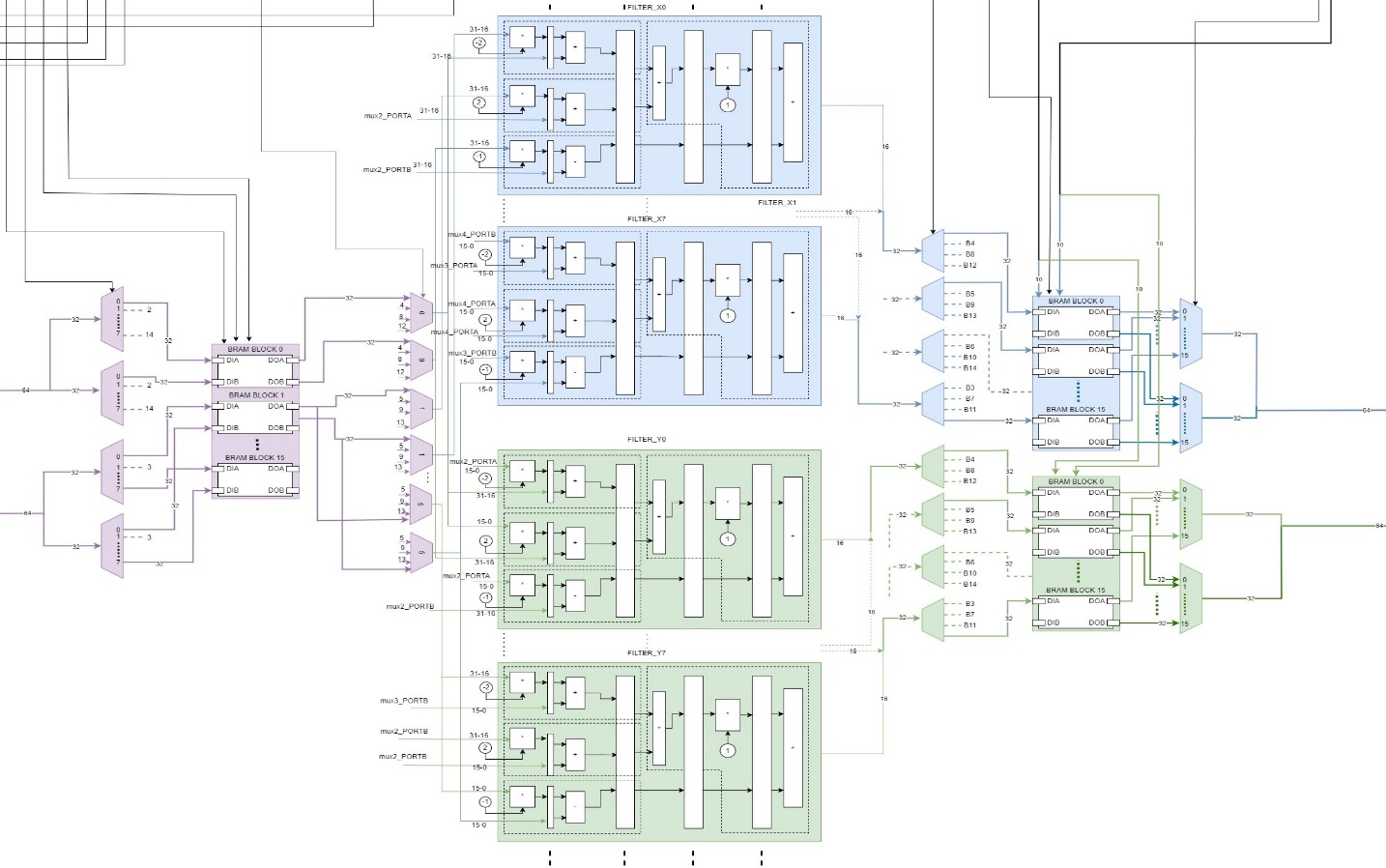
Modul koji kontroliše protočnu obradu je CONTROL\_LOGIC. Svrha ovog bloka je kontrola rutirajuće mreže ulaznih piksela, obrađenih piksela te formiranje adrese za upis filtriranih piksela u izlazne BRAM blokove. ASM dijagram CONTROL\_LOGIC modula prikazan je na *Slici 4.3*.

Modul koji kontroliše upis izlaznih piksela u dram je BRAM\_TO\_DRAM. Ovaj modul je odgovran za generisanje adrese sa koje će podaci biti pročitani iz BRAM blokova, adrese na koju će podaci biti upisani u DRAM, kao i za kontrolu rutirajuće mreže koja će odrediti iz kog BRAM bloka će podaci biti pročitani. ASM dijagram BRAM\_TO\_DRAM modula prikazan je na *Slici 4.4*.

FSM je modul koji kontroliše redoslijed izvršavanja prethodnih modula. Ukoliko je slika većih dimenzija od kapaciteta BRAM-a potrebno je nakon obrade preostale neobrađene piksele smjestiti u BRAM i obraditi. Proces se ponavlja sve dok postoje neobrađeni pikseli u DRAM-u. Ova faza nosi naziv REINIT i podrazumijeva pokretanje DRAM\_TO\_BRAMA da bi se smjestili novi podaci iz DRAMA u BRAM i BRAM\_TO\_DRAMA da bi se obrađeni podaci poslali iz BRAMA u DRAM. Ukoliko nema potrebe za reinicijalizacijom faze se izvršavaju sekvencijalno. ASM dijagram FSM modula prikazan je na *Slici 4.5.*

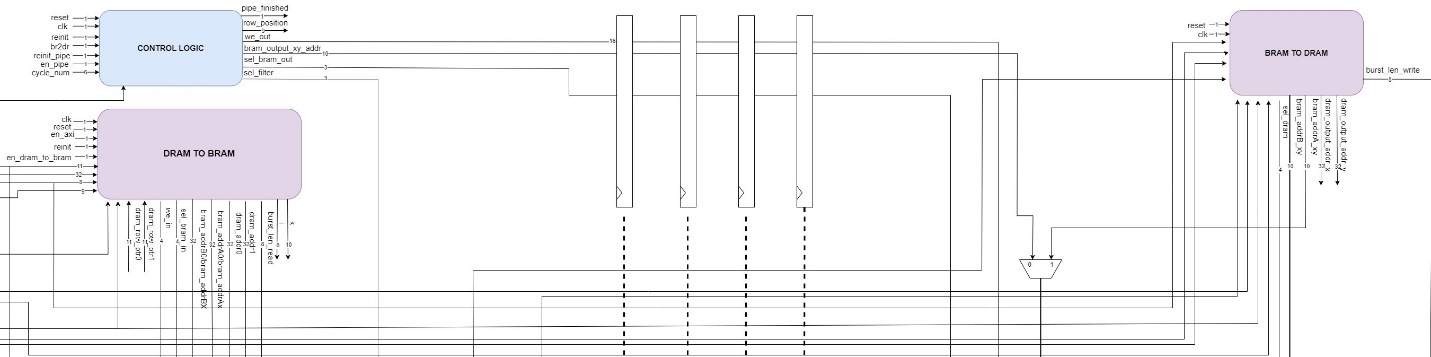
1. Izgled DUT-a

Izgled jedinice za obradu podataka prikazan je na *Slici 5.1*. Demultiplekseri na ulazu sistema služe za kontrolu izbora BRAM bloka u koji će podaci biti upisani. Nakon što su podaci upisani u BRAM blokove i pročitani iz istih, multiplekseri će iskontrolisati iz kog BRAM bloka će se podaci slati za obradu na filtere. Da bismo dobili 8 tačaka u paraleli na izlazu, potrebna su nam 24 ulazna piksela (na svaki filter blok se prosleđuju 3 piksela, ukupan broj filter blokova je 8). Analogno tome se obrađuje i drugi filter. Filter blokovi su sastavljeni od 4 DSP modula i izvršavaju neophodne operacije sabiranja i množenja. Nakon što su pikseli obrađeni multiplekseri će odrediti u koji bram blok se smještaju obrađeni podaci. Nakon što se podaci pročitaju iz BRAM blokova, demultiplekser određuje koji obrađeni red se prosleđuje na izlaz *Datapath*-a.



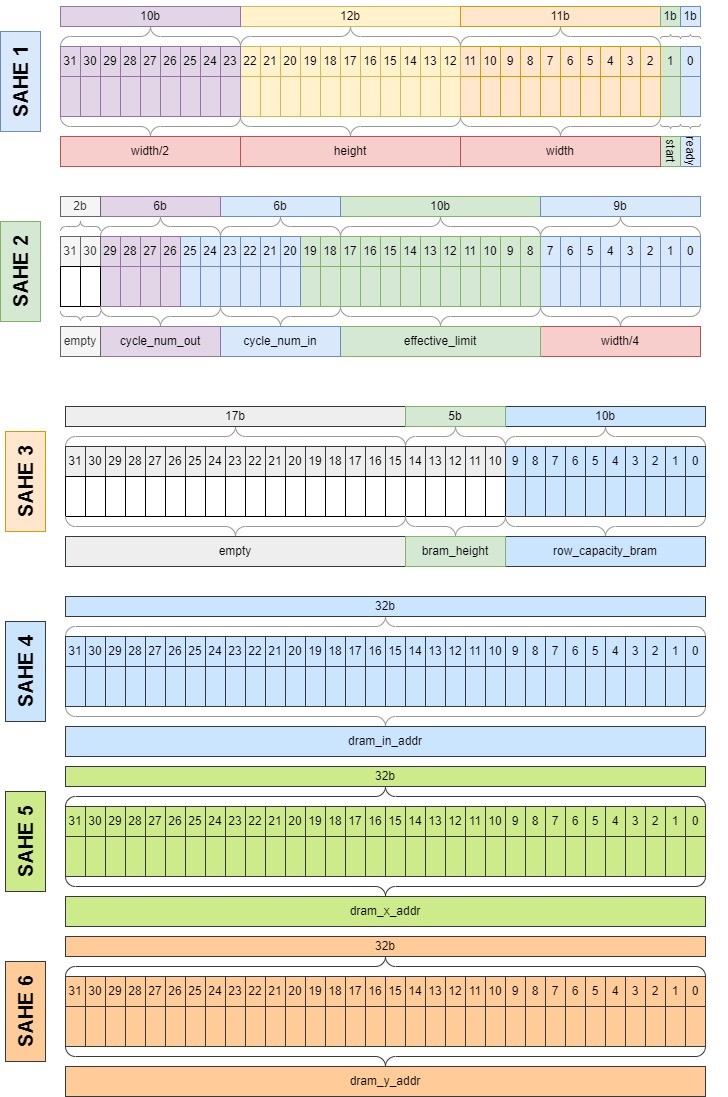
*Slika 5.1 Izgled jedinice za obradu podataka*

Kontrolna jedinica prikazana je na *Slici 5.2,* opisana detaljno u sekciji 4.



*Slika 5.2 Izgled kontrolne jedinice*

Unutar sistema postoji registarska banka u kojoj su smještene konstante potrebne za izvršavanje algoritma. Ove konstante se šalju prema programabilnoj logici u vidu SAHE-a da bismo smanjili kašnjenje prilikom konfiguracije sistema. Izgled SAHE-a prikazan je na *Slici 5.2.*



*Slika 5.2 Izgled SAHE-a*