Programska podrška za upravljanje kamerama uređaja za snimanje podataka

Mitrović, Dominik

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:022711

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Programska podrška za upravljanje kamerama uređaja za snimanje podataka

Diplomski rad

Dominik Mitrović

Osijek, 2018.

Sadržaj

1. UVOD 1
1.1. Pregled postojećih sličnih rješenja 2
1.1.1. ARCCORE Arctic Fusion Board 2
1.1.2. AMV Alpha Board 3
2. OPIS AMV GRABBER RAZVOJNOG OKRUŽENJA 5
2.1. AMV Grabber platforma 5
2.2. I^2C protokol 7
2.3. Ulazni i izlazni portovi 8
2.4. *Deserializer*/*serializer* 10
2.5. *Switch* i *expander* 11
2.6. Kamere 11
2.7. VDMA 13
3. KONCEPT RJEŠENJA 15
3.1. Razvojna okruženja 15
3.2. Razrada rješenja 16
4. IMPLEMENTACIJA RJEŠENJA 18
4.1. Inicijalizacija kamere 20
4.2. Konfiguracija kamere 21
4.3. Pokretanje i zaustavljanje snimanja 21
4.4. Upravljanje VDMA upravljačima 22
5. REZULTATI TESTIRANJA 24
5.1. Maksimalni broj kamera 24
5.2. Prijedlog za povećanje maksimalnog broja kamera 27
6. ZAKLJUČAK 28
LITERATURA 29
ABSTRACT 30
SAŽETAK 30
ŽIVOTOPIS 31

1. UVOD

Automobilska industrija je kroz povijest bila jedan od najvećih pokretača tehnoloških inovacija. Od dizajniranja i usavršavanja najranijih motora, sve do autonomne vožnje koja je danas još uvijek u ranim stadijima razvoja. Kroz proteklih nekoliko godina automobilska industrija doživljava još rapidniji napredak. Jedan od razloga zbog kojeg je taj ubrzani razvoj moguć je uporaba sve složenijih hardverskih komponenti koje se ugrađuju u vozila. Drugi razlog je korištenje sve kompleksnijih softverskih algoritama koji taj hardver iskorištavaju optimalno. Bilo koji sustav koji pomaže vozaču na neki način naziva se napredni sustav za pomoć vozaču (engl. *Advanced Driver-Assistance System, ADAS*). Ova tehnologija ima svoje korijene u značajkama koje su se počele pojavljivati prije više od desetljeća, poput tempomata, ili čak još ranije, uvođenjem jednostavnih sustava upozoravanja vozača, kao što su svjetla upozorenja na kontrolnoj ploči automobila. Moderni ADAS sustavi temelje se pretežno na uporabi različitih senzora koji su postali sastavni dijelovi vozila, primjerice kamere, senzori za mjerenje udaljenosti, radar, lidar, itd. Pri tome mora postojati i centralna procesorska jedinica koja će primiti podatke sa senzora, obraditi ih pomoću softverskih algoritama te na kraju dati određeni odgovor.

Neki tipični primjeri današnjih ADAS tehnologija su: [1]

- Adaptivni tempomat – Funkcionalnost koja omogućava jedinici za upravljanje motorom manipuliranje akceleratorom i kočnicom automobila na temelju informacija s vanjskog senzora. Koristi se za automatsko kontroliranje brzine vozila.
- Upozorenje na frontalne kolizije – Sustav koji upozorava vozača ukoliko detektira mogućnost frontalne kolizije. Temelji se na mjerenju brzine vozila i udaljenosti do prepreke ispred njega.
- Prepoznavanje pješaka – Sustav koji pomoću informacija s kamera detektira pješake i obavještava vozača na njihovu prisutnost.
- Prepoznavanje prometnih znakova – Podsustav koji vrši prepoznavanje prometnih znakova i upozorava vozača na njih.
- Sustav za održavanje vozila u traci – Sustav koji prati putanju vozila i upozorava vozača ako predvidi kretanje prema traci u kojoj dolazi promet iz suprotnog smjera.
- Mnogi drugi napredni sustavi.

Najveće prednosti današnje ADAS tehnologije su općenito unaprjeđenje ugodnosti vožnje i povećavanje sigurnosti vozača i svih ostalih sudionika u prometu. Također, ove tehnologije postaju sve pristupačnije širokoj javnosti zbog sve veće dostupnosti i postepenog smanjivanja cijena. Pretpostavka je da će unutar nekoliko godina napredne ADAS tehnologije biti sastavni dio gotovo svakog novog vozila.

Kako bi se omogućilo lakše razvijanje, testiranje i verifikacija ispravnosti algoritama korištenih u automobilskoj industriji postoji potreba za posebnim uređajima koji su namijenjeni upravo tome. Uređaj *AMV Grabber* (engl. *Automotive Machine Vision Grabber*) na kojem je razvijano rješenje u sklopu ovog diplomskog rada spada u tu kategoriju.

U okviru zadatka ovog diplomskog rada bilo je potrebno implementirati mehanizam upravljanja kamerama na AMV Grabberu. Video sadržaj snimljen kamerama kasnije se koristi za vršenje testiranja i verifikacije ADAS algoritama. Rješenje je pisano u programskom jeziku C. Temelj AMV Grabber ploče je *ARM* (engl. *Advanced RISC Machines*) jezgra centralne procesorske jedinice Zynq XC7Z030. Uređaj u teoriji podržava snimanje s najviše 9 kamera istovremeno. U praksi se javljaju određena hardverska ograničenja koja su pojašnjena u kasnijim poglavljima ovog rada. Kontrola kamera podrazumijeva detekciju prisutnosti kamera (nije napravljeno iz praktičnih razloga koji će biti objašnjeni), konfiguriranje kamera, kontrolu odgovarajućih *VDMA* (engl. *Video Direct Memory Access*) upravljača, pokretanje i zaustavljanje snimanja. Bilo je potrebno napraviti podršku za dvije vrste kamera: kameru koja daje RAW format na izlazu i kameru koja daje YUV format na izlazu.

1.1. Pregled postojećih sličnih rješenja

Već postoje uređaji slični AMV Grabberu koji rješavaju probleme testiranja ADAS algoritama ili njihove obrade. U ovom potpoglavlju navedeno je nekoliko primjera.

1.1.1. ARCCORE Arctic Fusion Board

Ploča „Arctic Fusion“ proizvođača ARCCORE dizajnirana je za testiranje tehnologija i koncepata koji se koriste u nadolazećim aktivnim sigurnosnim sustavima u automobilima. Ima

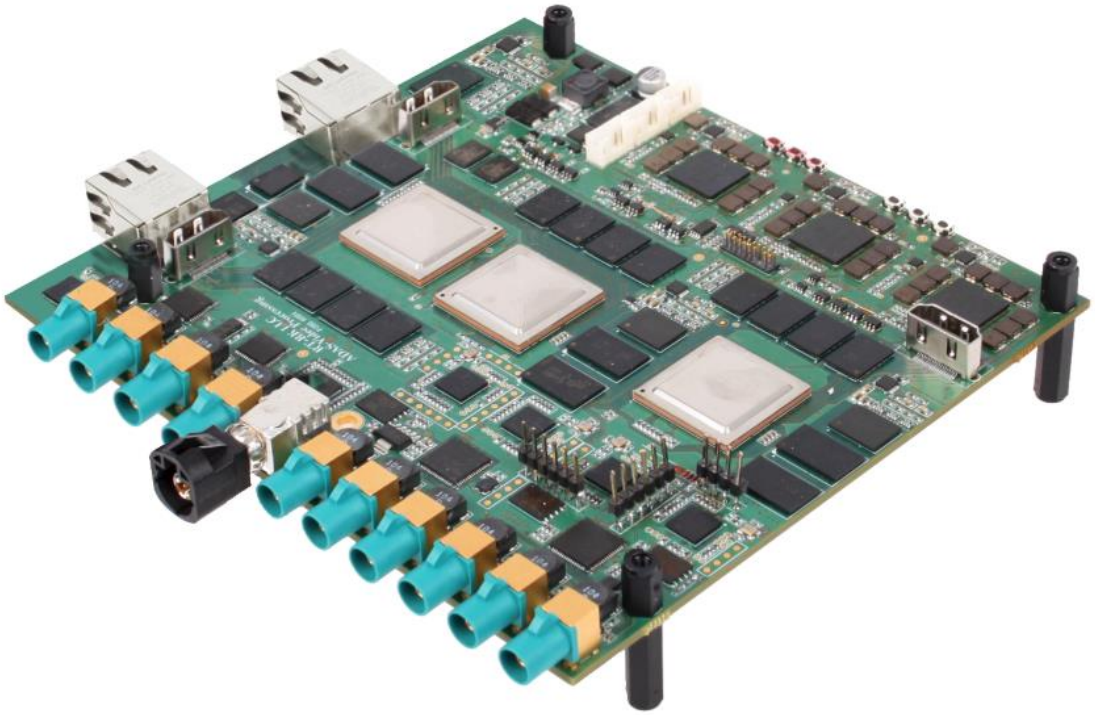
modularnu izvedbu koja omogućava jednostavno spajanje na druge uređaje. Temelji se na Aurix procesoru sa tri jezgre koji obrađuje podatke sa kamera koje su spojene na ulaz ploče. Uređaj je još uvijek u fazi dizajniranja, ali se u budućnosti planira proizvodnja i puštanje u prodaju [2]. Na slici 1.1. nalazi se prikaz spomenutog uređaja. Ovaj uređaj je izravna konkurencija AMV Grabber ploči zbog vrlo slične namjene.



Slika 1.1. Uređaj „Arctic Fusion“. [3]

1.1.2. AMV Alpha Board

AMV Alpha ploča je, kao i AMV Grabber, dizajnirana na institutu RT-RT. Njezina namjena nije pomoć pri testiranju, nego izvršavanje ADAS algoritama u realnom vremenu. Također sadrži ulaze na koje se mogu spojiti kamere ili neki drugi uređaj koji će dostavljati tok video podataka. Princip kontrole kamera je izveden na sličan način kao i na AMV Grabberu te zbog toga predstavlja dobru referencu za ovaj projekt. **Slika 1.2.** prikazuje navedeni uređaj.



Slika 1.2. Prikaz AMV Alpha ploče. [4]

2. OPIS AMV GRABBER RAZVOJNOG OKRUŽENJA

U ovom poglavlju opisana je hardverska platforma koja je korištena u realizaciji diplomskog rada, njene osnovne komponente i njihova funkcija. Ova teorijska podloga pomaže pri razumijevanju rada uređaja u cjelini.

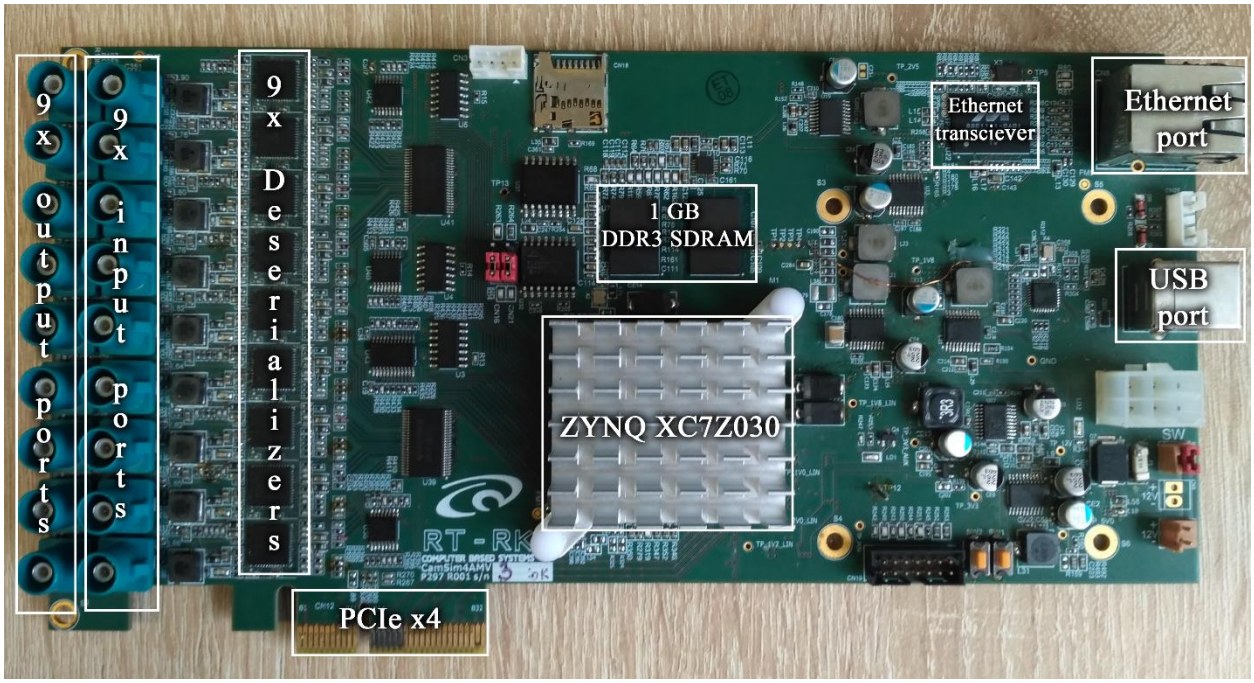
2.1. AMV Grabber platforma

AMV Grabber ploča dizajnirana je na institutu RT-RK i zamišljena je kao platforma za razvijanje, testiranje i verifikaciju ADAS algoritama. Često, prilikom razvijanja takvih algoritama treba vršiti veliki broj sukcesivnih testova, što nikako nije pogodno činiti na cesti u realnim uvjetima. Korištenjem ovakvog uređaja povećava se sigurnost u prometu jer se eliminira mogućnost da se u prometu nađe vozilo sa nedovoljno testiranim ili nedovršenim algoritmima. Ovaj uređaj ima veliki potencijal jer stvara laboratorijske uvjete za nešto što bi se inače testiralo puno duže i u potencijalno riskantnim okolnostima.

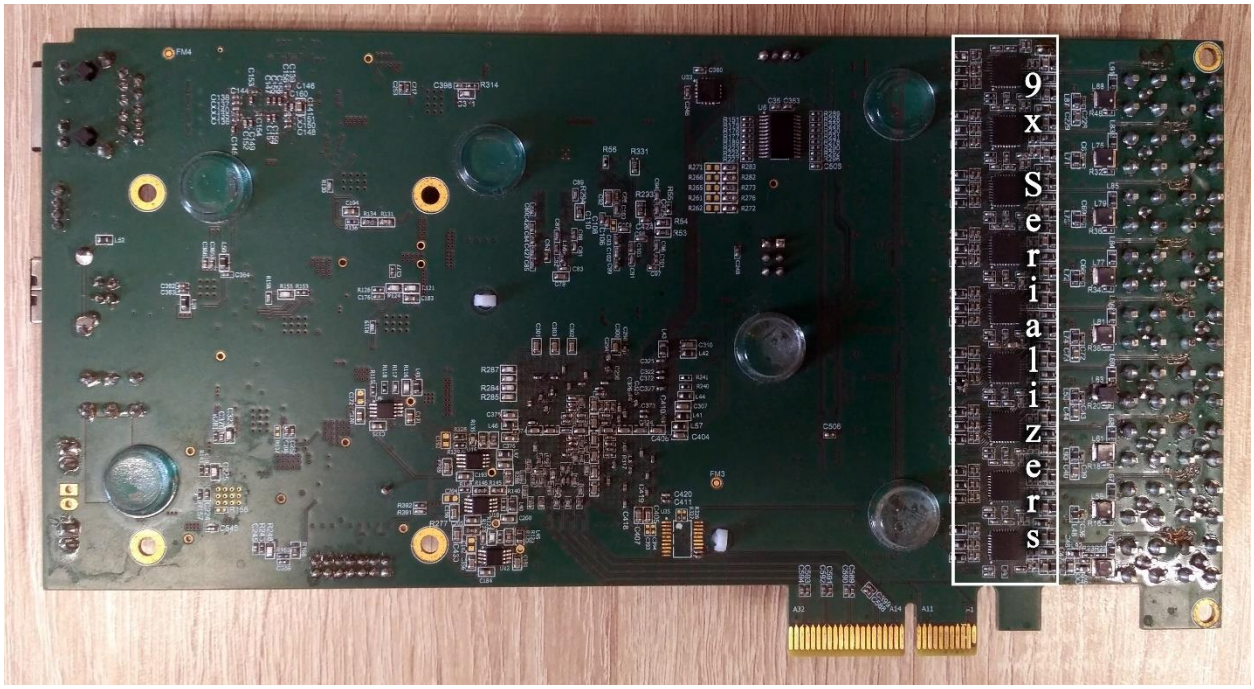
Platforma je opremljena Zynq sustavom na čipu (engl. *System on chip, SoC*), koji je baziran na ARM Cortex-A9 procesoru s dvije jezgre i Xilinx programibilnoj logici (engl. *Programmable Logic, PL*). Procesor, zajedno sa memorijom na čipu, vanjskim memorijskim sklopom i setom sklopova za periferno povezivanje čini sustav za obradu podataka (engl. *Processing System, PS*). Programibilna logika se temelji na *FPGA* (engl. *Field Programmable Gate Array*). FPGA je u osnovi sklop koji korisnik može programirati i reprogramirati za obavljanje različitih funkcija i operacija. Nakon proizvodnje ne obavlja specifičnu zadaću nego je na programeru da implementira funkcionalnosti koje su mu potrebne. Povezivanje procesora s programibilnom logikom odrađeno je preko *AXI* sabirnica, što znači da sklopovi implementirani u programibilnoj logici mogu komunicirati sa softverom koji se izvršava na procesoru.

Na uređaj je ugrađeno 1 *GB DDR3 SDRAM* (engl. *Double Data Rate type three of Synchronous Dynamic RAM*) memorije koja služi kao međuspremnik u koji se spremaju podaci sa kamere i koje je potrebno obraditi. Od ostalih komponenti, bitno je spomenuti *serializere*, *deserializere* i *PCIe x4* (engl. *Peripheral Component Interconnect Express*) sučelje za povezivanje sa računalom. Kako bi se mogao shvatiti tok podataka od kamere do tvrdog diska na kojeg se spremaju podaci koje ona šalje potrebno je detaljnije objasniti ulogu pojedinih

komponenti. Na **slikama 2.1.** i **2.2.** se nalaze prikazi AMV Grabber platforme sa označenim glavnim komponentama [5].



Slika 2.1. Prikaz prednje strane AMV Grabber platforme sa označenim glavnim komponentama.



Slika 2.2. Prikaz stražnje strane AMV Grabber platforme sa označenim *serializerima*

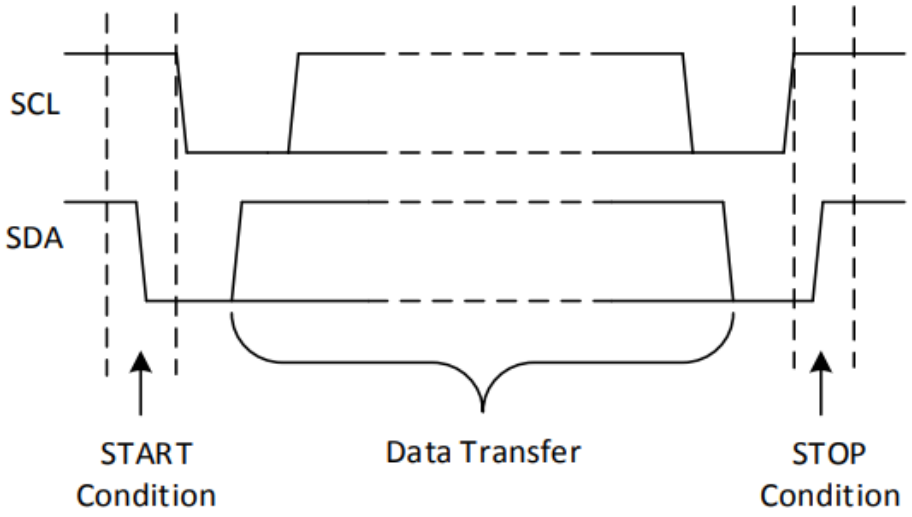
2.2. I^2C protokol

I^2C (engl. *Inter-Integrated Circuit*) protokol je iznimno popularan protokol koji omogućava komunikaciju između jednog ili više *master* uređaja sa jednim ili više *slave* uređaja. Pri tome se koriste samo 2 vodiča, što je jedna od najvećih prednosti u odnosu na druge protokole. Konkretno, u ovom projektu, I^2C se koristi za komunikaciju između pojedinih komponenti na platformi i njihovu konfiguraciju. *Slave* uređaj može slati podatke jedino ako *master* uređaj to zatraži. Svaka komponenta na I^2C sabirnici ima jedinstvenu adresu koja ju razlikuje od ostalih uređaja na istoj sabirnici. *Slave* uređaji obično zahtijevaju konfiguriranje prilikom pokretanja kako bi se definiralo njihovo ponašanje. Ovo se u pravilu događa kada *master* uređaj pristupa mapi registara *slave* uređaja, koji također imaju svoje adrese.

Fizička I^2C sabirnica se sastoji od 2 linije – linije za serijski takt (engl. *Serial Clock, SCL*) i linije za serijske podatke (engl. *Serial Data, SDA*). Prijenos podataka može započeti samo kada se sabirnica nalazi u stanju mirovanja. Općeniti postupak kojim *master* pristupa *slave* uređaju je sljedeći:

1. Ako *master* želi poslati podatke *slave* uređaju:
    - *Master* šalje START uvjet i adresu *slave*-a
    - *Master* šalje podatke *slave*-u
    - *Master* prekida prijenos slanjem STOP uvjeta
2. Ako *master* želi primiti/pročitati podatke iz *slave* uređaja
    - *Master* šalje START uvjet i adresu *slave*-a
    - *Master* šalje adresu registra čiju vrijednost treba pročitati
    - *Master* prima podatke od *slave* uređaja
    - *Master* prekida prijenos slanjem STOP uvjeta

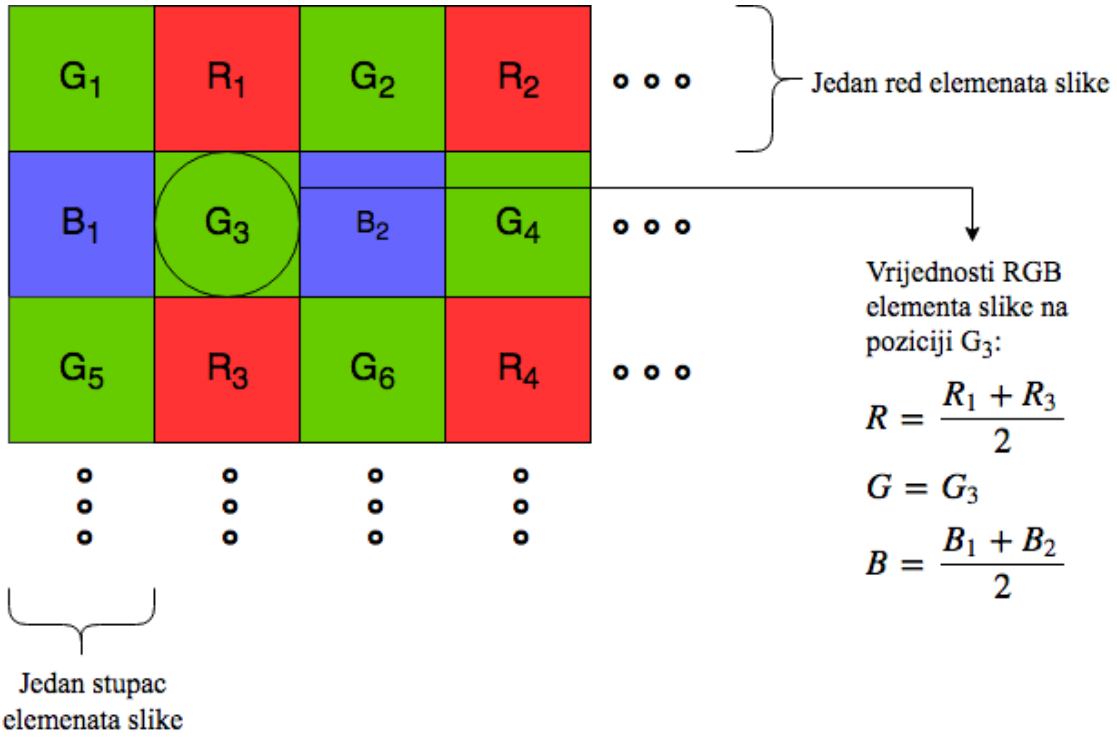
Na **slici 2.3.** prikazani su START i STOP uvjeti u I^2C protokolu.



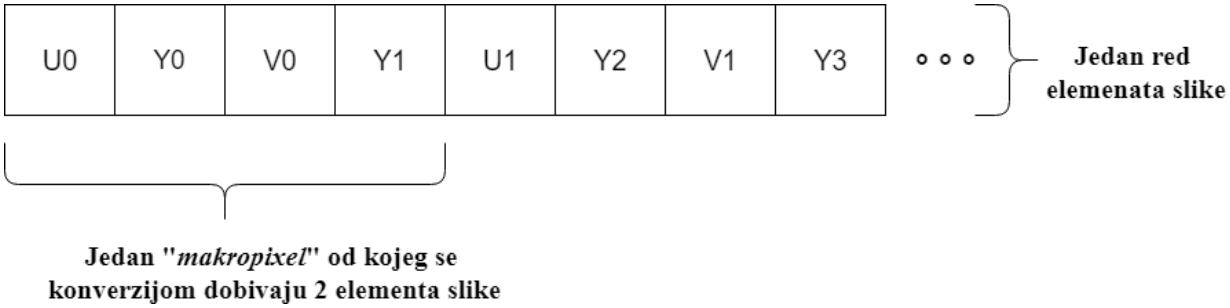
Slika 2.3. Prikaz START i STOP uvjeta u I^2C protokolu. [6]

2.3. Ulazni i izlazni portovi

Na ploči je implementirano 9 ulaznih i 9 izlaznih portova. Njihova glavna svrha je da budu pristupne točke za ostale periferne uređaje. Za AMV Grabber, tipični uređaji koji se spajaju na ulazne portove su kamere koje opskrbljuju platformu sa podacima koje treba obraditi. Implementirana je podrška za 2 vrste kamera koje šalju podatke u dva različita formata – RAW i YUV. Kada se okviri prenose u RAW formatu, za svaki element slike se šalju podatci o količini crvene, zelene i plave boje. Kombinacijom tih triju boja moguće je dobiti bilo koju drugu boju. Na **slici 2.4.** se nalazi pojednostavljen prikaz slanja podataka s ovakvim formatom. Kada se prijenos okvira vrši u YUV formatu, svaki element slike je predstavljen kombinacijom tri elementa. Y određuje svjetlinu boje, dok U i V elementi određuju samu boju. Na **slici 2.5.** se nalazi pojednostavljeni prikaz slanja podataka YUV formatom.

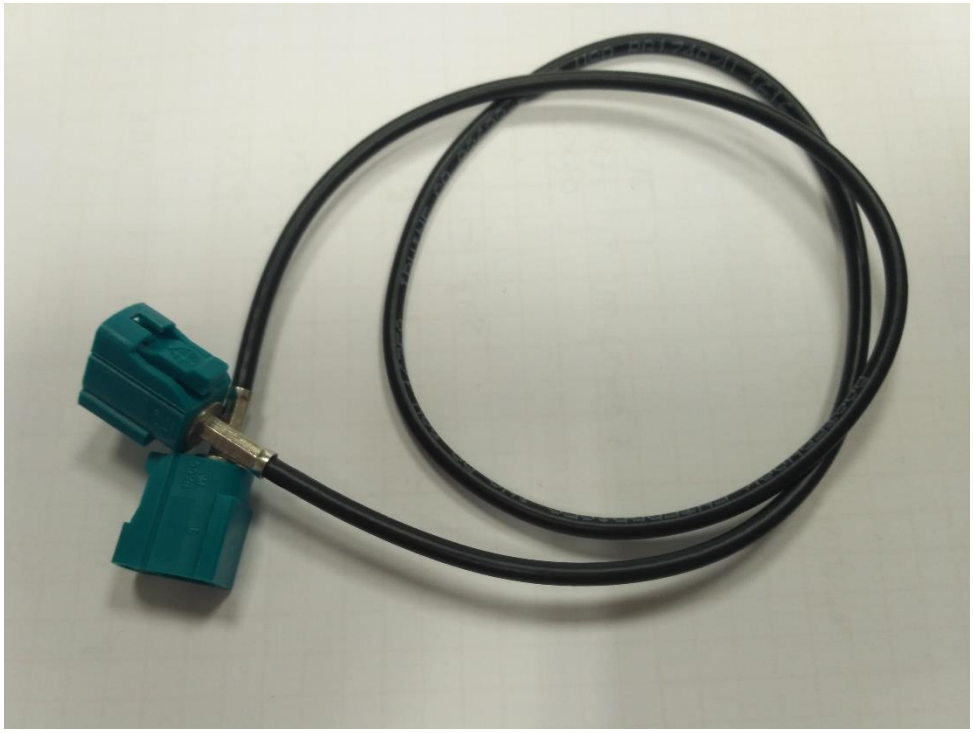


Slika 2.4. Pojednostavljen prikaz slanja podataka RAW formatom.



Slika 2.5. Pojednostavljen prikaz slanja podataka YUV formatom.

Uređaji koji se mogu spojiti na izlazne portove nisu striktno definirani. U pravilu su to uređaji koji mogu iskoristiti video podatke koje je ploča prikupila sa kamera. Povezivanje perifernih uređaja i portova se ostvaruje koaksijalnim kablovima impedancije 50 Ω i *FAKRA* (njem. *Facharbeitskreis Automobil*) konektorima na krajevima. Ova vrsta konektora je standard u automobilskoj industriji jer pruža posebni mehanizam zaključavanja spoja između uređaja i kabla. Namijenjeni su za korištenje u zahtjevnim scenarijima jer su pouzdani i vrlo otporni na vibracije, temperature i ostale mehaničke stresove. Na **slici 2.6.** prikazan je primjer koaksijalnog kabla koji je korišten.

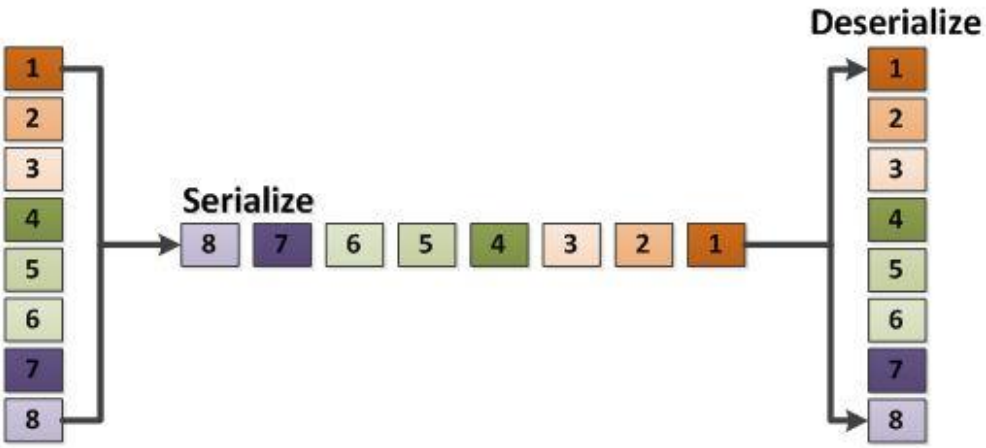


Slika 2.6. Prikaz koaksijalnog kabla.

2.4. *Deserializer/serializer*

Deserializer je sklop koji na ulazu prima podatke serijski te ih na izlaz šalje paralelno. *Serializer* ima obrnutu funkciju. Na ulazu prima podatke paralelno, a na izlaz šalje serijski niz podataka. Ovi sklopovi imaju veoma važnu ulogu zbog toga što se preko njih vrši prijenos podataka od kamere prema centralnoj procesorskoj jedinici. Pri tome se koristi samo jedna fizička veza za svaki ulaz i izlaz na ploči što uvelike olakšava hardversku izvedbu platforme te smanjuje potrošnju energije i troškove izrade. Kako se povećava brzina prijenosa podataka, problemi koji se javljaju korištenjem paralelnih sabirnica se također naglo povećavaju. Kada se prenosi veća količina podataka, koristi se više snage, a hardverski dizajn postaje kompliciraniji i teži za implementaciju. Kod paralelnog prijenosa podataka se također javlja veća mogućnost pogreške. Zbog sve veće težnje za smanjivanjem veličine uređaja bitno je prostor na ploči iskoristiti što efikasnije. To su neki od problema koji se rješavaju korištenjem ovih sklopova. Brzina prijenosa serijski poslaganih podataka je tipično mnogo veća nego kada se podaci prenose preko paralelne sabirnice. Takt (engl. *Signal Frequency*) korišten za serijski prijenos se može

povećati puno više nego kod paralelnog prijenosa jer ne treba vršiti sinkronizaciju podataka koji dolaze sa pojedinih paralelnih linija te ne postoji interferencija između njih. Na primjer, puno je lakše povećati brzinu prijenosa 16 puta na jednoj liniji, nego 2 puta na 8 paralelnih linija. Na **slici 2.7.** se može vidjeti pojednostavljeni prikaz rada *serializera* i *deserializera*.



Slika 2.7. Pojednostavljeni prikaz rada *serializera* i *deserializera*. [7]

2.5. *Switch* i *expander*

Switchevi i *expanderi* su sklopovi pomoću kojih procesorska jedinica šalje I^2C naredbe do svakog pojedinačnog *deserializera* i *serializera*. U slučaju kada se vrši inicijalizacija kamera od strane procesorske jedinice omogućena je tzv. „*Pass Through*“ komunikacija u oba smjera na putanji procesor ↔ *serializer* ↔ *deserializer* ↔ kamera. Oni također omogućavaju razlikovanje i korištenje istih perifernih uređaja, u ovom slučaju kamera, na istoj I^2C sabirnici.

2.6. Kamere

Iako su kamere periferni uređaji i nisu sastavni dio ploče, one čine jednu od najvažnijih komponenti. Cijeli projekt se bazira na njihovom osposobljavanju i manipuliranju tokom podataka koji one generiraju. Implementirana je podrška za kamere koje koriste senzore OV10640 [8] i OV10635 [9] proizvođača *OmniVision* [10]. Oba senzora koriste *CMOS* (engl. *Complementary Metal Oxide Semiconductor*) tehnologiju. Ovi senzori se sastoje od velikog

broja foto-detektora koji pretvaraju svjetlost u napon koji je proporcionalan njezinom intenzitetu. Broj foto-detektora je jednak broju elemenata slike (engl. *pixel*), s tim da se na senzoru nalaze i dodatni neiskorišteni detektori koji služe za prepoznavanje crne boje. Dobiveni napon se pojačava pomoću pojačala i formiraju se digitalni podatci koji predstavljaju sliku. Pomoću *serializera*, koji je u ovom slučaju sastavni dio same kamere, dobiveni podatci se slažu u serijski niz i putem koaksijalnog kabla šalju prema ploči. Kamere podržavaju različite rezolucije, broj okvira po sekundi i formate. Usporedba i kratak pregled navedenog nalazi se u **tablici 2.1.**

Tablica 2.1. Pregled i usporedba korištenih senzora

	OV10635	OV10640
Podržane rezolucije	1280x800, 1280x720, 752x480, 640x480, 352x288, 320x240	1280x1080, 640x480, 320x240
Podržani formati	RAW, YUV	RAW
FPS	30, 60	30
Ostale funkcionalnosti	Automatska kontrola ekspozicije, korekcija leća, ispravljanje defektivnih elemenata slike, automatska kontrola crne boje, mogućnost rezanja slike na ne-standardnu rezoluciju	

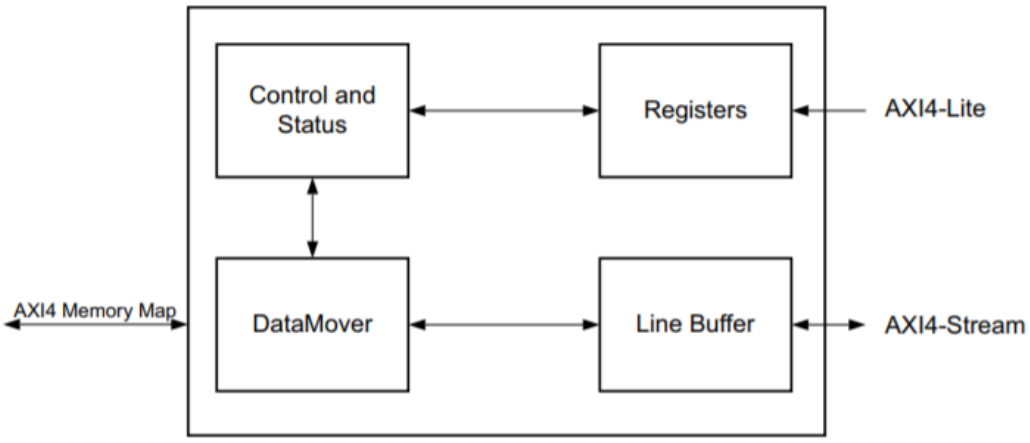
Unutar kamere se nalaze registri za konfiguraciju senzora. Svaki registar ima svoju adresu i vrijednost koja je upisana u njega. Ovisno o vrijednostima koje se nalaze zapisane u njima, mijenja se ponašanje kamere. Primjerice, upisivanjem određenih vrijednosti u odgovarajuće registre, moguće je promijeniti rezoluciju kojom će senzor snimati video. Ukoliko u registre nisu upisane pravilne vrijednosti mogu se dogoditi nepredviđene greške, šum, artefakti na slici ili kamera uopće neće funkcionirati. Upisivanje vrijednosti se izvršava putem I^2C protokola. Na **slici 2.8**. se nalazi prikaz jedne od korištenih kamera.



Slika 2.8. Prikaz jedne od korištenih kamera.

2.7. VDMA

VDMA nije fizički sklop na ploči, već je dizajniran na programibilnoj logici procesora. Glavna svrha ovog sklopa je da prima okvire koji dolaze s kamera i smisleno ih slaže u memoriju. Posebno je optimiziran za rad sa video podacima i brine se da svi dolazni podaci budu sinkronizirani. Moguće ga je konfigurirati da radi u smjeru pisanja i smjeru čitanja. U ovom slučaju je konfiguriran za zapisivanje podataka u memoriju. Na **slici 2.9.** prikazan je blok dijagram VDMA sklopa. Vidljivo je da sadrži registre za konfiguraciju kojima se upravlja putem *AXI4-Lite* sučelja. Podatci koji dolaze putem AXI4 toka se pomoću *DataMovera* zapisuju u memoriju. Blok za kontrolu i status upravlja *DataMoverom* [11].



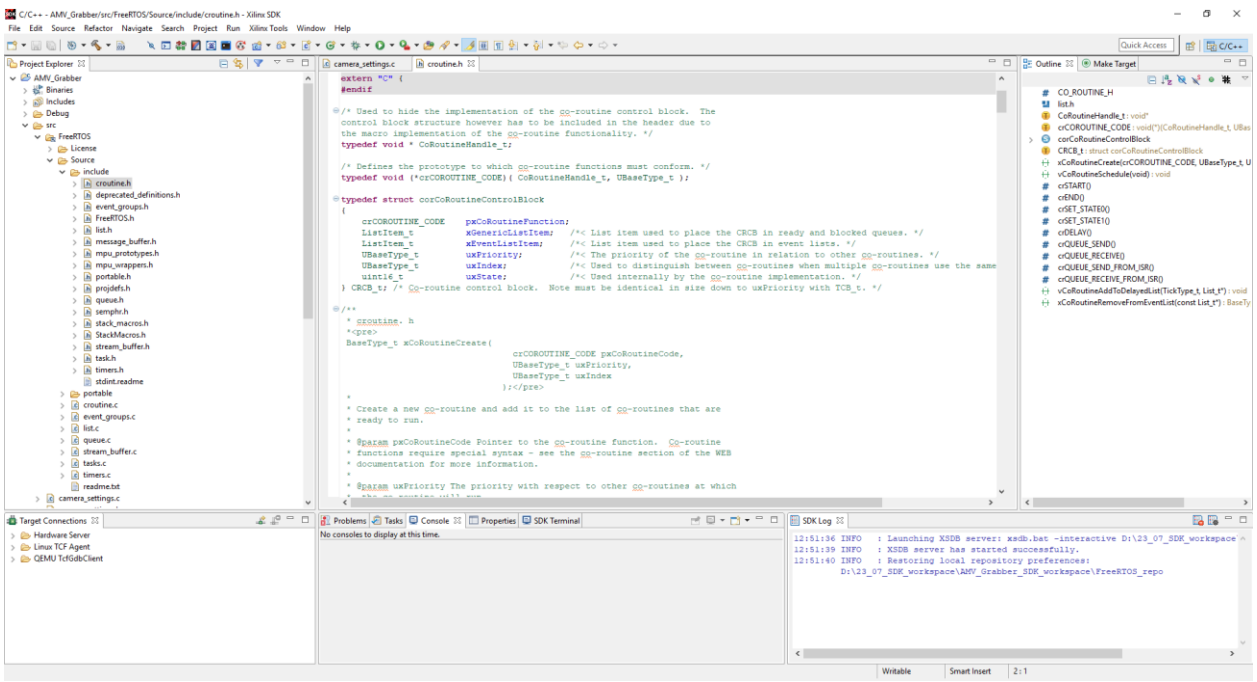
Slika 2.9. Blok dijagram VDMA bloka implementiranog na programibilnoj logici procesora [12].

3. KONCEPT RJEŠENJA

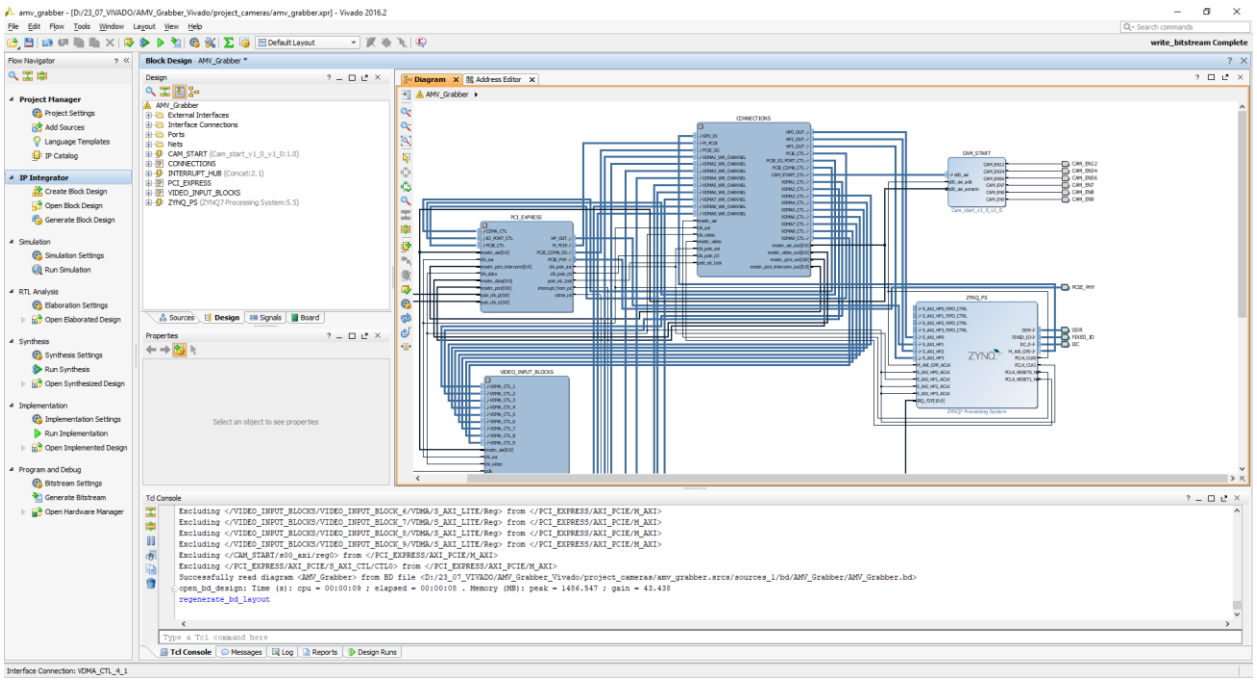
U ovom poglavlju opisana su razvojna okruženja u kojima je rješenje napravljeno i iznesen je koncept rješenja.

3.1. Razvojna okruženja

Prilikom izrade diplomskog rada korištena su 2 razvojna okruženja: *Xilinx Software Development Kit* [13] i *Vivado Design Suite* [14]. *Vivado* je korišten za programiranje FPGA dijela Zynq sustava na čipu. Sklop koji se „simulira“ programibilnom logikom ne postoji fizički na ploči, već se njegova funkcionalnost u potpunosti temelji na dizajniranju FPGA. Jedni od važnijih sklopova koji su dizajnirani ovim putem su VDMA i *CDMA* (engl. *Central Direct Memory Access*). *SDK* se koristi za razvoj logike koja se implementira na fizičkim sklopovima ploče, ali i za interakciju sa simuliranim hardverom u programibilnoj logici. Oba razvojna okruženja sadrže moćne alate za uklanjanje softverskih i hardverskih pogrešaka (engl. *Debug* – proces uklanjanja grešaka u programskom kodu ili hardveru) koji su bili od velike pomoći tijekom razvijanja projekta. Primjer korisničkih sučelja *SDK*-a se nalazi na **slici 3.1.**, a korisničkog sučelja *Vivada* na **slici 3.2.**



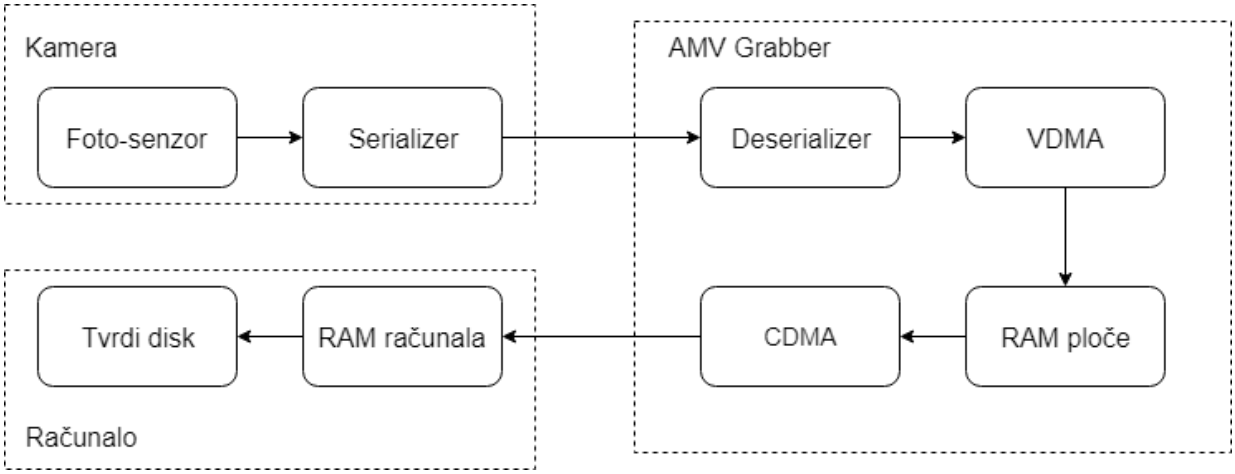
Slika 3.1. Korisničko sučelje *SDK* razvojnog okruženja.



Slika 3.2. Korisničko sučelje *Vivado* razvojnog okruženja.

3.2. Razrada rješenja

Dijagram na **slici 3.3.** predstavlja tok podataka od kamere do tvrdog diska računala i prikazuje koje sve komponente sudjeluju pri tome. Također, pruža dobar uvid u problematiku koja se rješava ovim radom i predodžbu o radu ploče.



Slika 3.3. Tok podataka od kamere do tvrdog diska računala i prikaz komponenti koje se pri tome koriste.

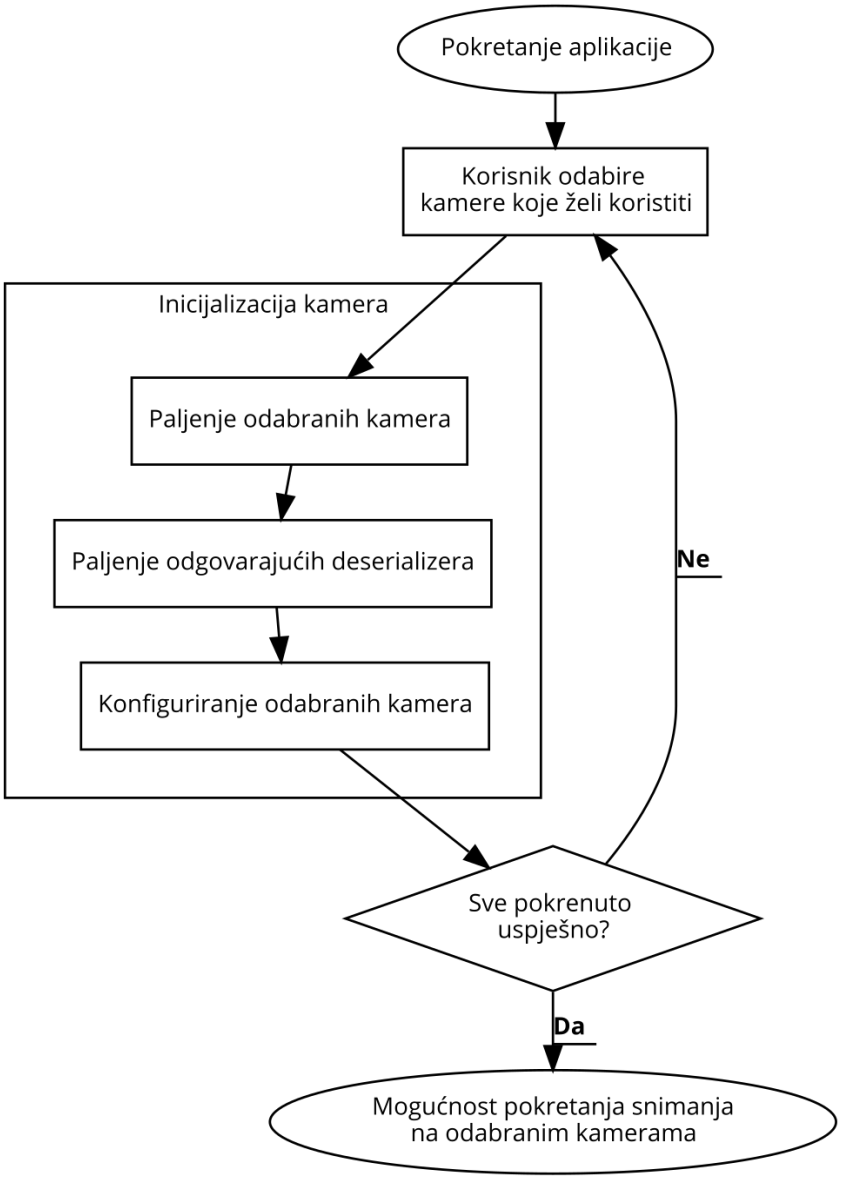
Prvi korak koji je bilo potrebno napraviti je detekcija prisutnosti kamera. Za vrijeme razvoja projekta odlučeno je da se mehanizam za automatsko prepoznavanje na kojem portu su priključene kamere neće implementirati. Razlog za to je sljedeći: Budući da korisnik mora samostalno priključiti kamere na pojedini ulaz, unaprijed se zna na koje portove će kamere biti priključene. Umjesto da se troše računalni resursi i vrijeme na izvršavanje algoritma koji provjerava prisutnost kamere, korisnik unutar aplikacije odabire ulaze na koje je spojio kamere. Nakon toga se izravno prelazi na inicijalizaciju priključenih kamera. Ipak, bilo je potrebno napraviti provjeru uspješnog konfiguriranja. Korisnik se u sučelju aplikacije obavještava ukoliko kamera nije uspješno konfigurirana. Razlozi zbog kojih može doći do neuspješne konfiguracije su: neispravna kamera, kamera nije ispravno spojena na ploču ili je korisnik odabrao krivi port u aplikaciji.

Nakon uspješnog paljenja i konfiguracije kamere, potrebno je uključiti i konfigurirati odgovarajuće *deserializere* kako bi oni podatke vratili iz serijskog niza nazad u paralelni. Podatci zatim stižu do VDMA upravljača koji sakuplja okvire videa s kamere i zapisuje ih na RAM ploče. CDMA upravljač uzima podatke s RAM-a ploče i preko PCIe sučelja ih šalje u RAM računala. Zadnji korak prijenosa podataka je zapisivanje na tvrdi disk računala. Ovisno s kojom vrstom kamere je napravljeno snimanje, zapisani podaci se trebaju pretvoriti iz RAW ili YUV formata u format čitljiv s konvencionalnim programima za video reprodukciju.

Pokretanje snimanja se izvršava slanjem određenih I^2C komandi u specifičan registar na kameri. Zaustavljanje snimanja radi na istom principu, ali potrebno je poslati naredbu u drugi registar.

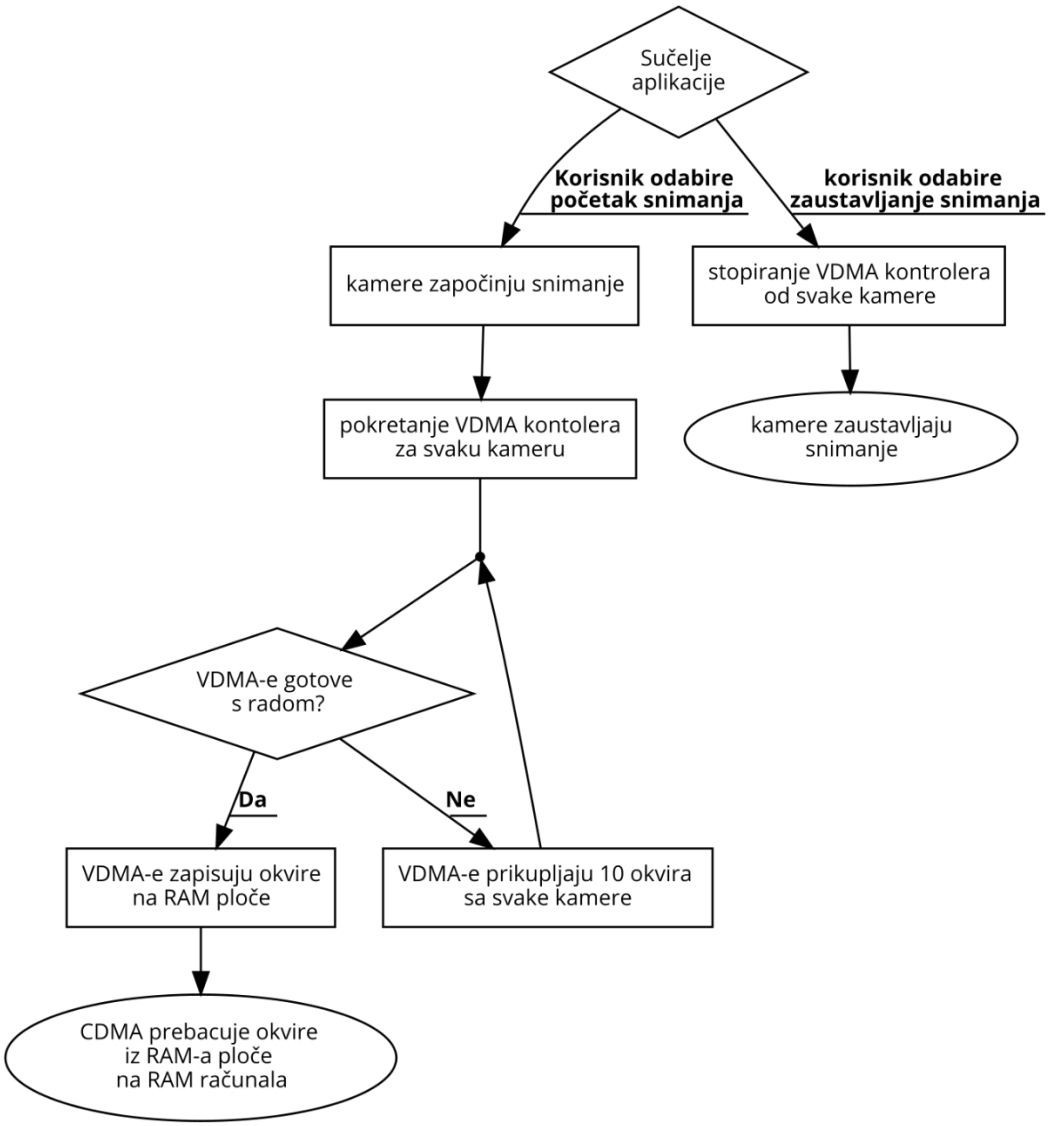
4. IMPLEMENTACIJA RJEŠENJA

U ovom poglavlju detaljnije je objašnjeno na koji način je rješenje implementirano u projekt. Sva rješenja su predočena dijagramima i pseudo kodom. Na **dijagramu 4.1.** može se vidjeti da je prva funkcionalnost koju treba implementirati inicijalizacija samih kamera. Inicijalizacija obuhvaća paljenje kamere, pokretanje odgovarajućih *deserializera* i na posljetku konfiguraciju registara odabranih kamera.



Dijagram 4.1. Prikaz koraka koje je potrebno napraviti kako bi se kamera pripremila za snimanje.

Drugi problem koji je bilo potrebno riješiti je pokretanje i kontrola odgovarajućih VDMA upravljača. Po jedan VDMA upravljač se uključuje za svaku aktivnu kameru. Osnovni princip funkcioniranja VDMA-e je da primi 10 okvira videa s kamere i zapisuje ih na RAM ploče na određeno memorijsko mjesto rezervirano za to. Postupak se ponavlja dok sve kamere ne pošalju 10 okvira. Kada je prijenos prvih 10 okvira sa svake kamere završen, započinje prijenos idućih 10 okvira sa svake kamere, ali na drugo memorijsko mjestu u RAM-u ploče. Istovremeno, CDMA šalje okvire zapisane na prvom memorijskom mjestu u RAM-u ploče na RAM računala preko PCIe sučelja. Kako je prva memorijska zona sada prazna, idući okviri se opet mogu zapisivati u nju, a CDMA kreće sa slanjem okvira iz druge memorijske zone. Cijeli proces se ponavlja sve dok traje snimanje. Prikaz koraka koji se izvršavaju kada korisnik pokrene snimanje sa kamerom nalazi se na **dijagramu 4.2.**



Dijagram 4.2. Prikaz izvršenih koraka pri pokretanju ili zaustavljanju snimanja.

4.1. Inicijalizacija kamere

Hardverski je izvedeno da se neke kamere pale u parovima, odnosno na taj način je izveden dovod napajanja na portove. Parovi kamera su: kamera 1 i 2, kamera 3 i 4, kamera 5 i 6. Kamere 7, 8 i 9 su napajane zasebno. Prvi korak u inicijalizaciji kamere je dovođenje napona do porta na koji je spojena. Ovaj postupak prikazan je u **programskom pseudo-kodu 4.1.**

```
Funkcija paljenje_kamere (primljeni parametar: broj porta)
{
      ako je kamera na portu već upaljena
      {
            nastavi s programom;
      }
      ako kamera na portu nije upaljena
      {
            dovedi napajanje na port;
      }
}
```

Programski pseudo-kod 4.1. Funkcija za paljenje kamere.

Nakon uspješnog paljenja kamere potrebno je upaliti i konfigurirati sve odgovarajuće *deserializere* kako bi bili spremni za primanje serijski poslaganih podataka, njihovo slaganje u paralelne tokove i prosljeđivanje prema VDMA upravljačima. Kako ne bi došlo do konflikta na I^2C sabirnici i kako bi signal došao od *serializera* kamere do određenog *deserializera* uvodi se tzv. maskiranje adresa na sabirnici. Svakom *deserializeru* je dodijeljena vlastita maska i na taj način se uspješno razlikuju i izbjegava se eventualno preklapanje adresa. Funkcija za inicijalizaciju *deserializera* je prikazana u **programskom pseudo-kodu 4.2.** Ona se poziva za svaki *deserializer* koji je potrebno inicijalizirati.

```
Funkcija deserializer_init (primljeni parametar: broj porta)
{
      pokreni deserializer za odgovarajući port()
      {
            postavi masku;
            upiši naredbu za paljenje u za to predviđeni registar;
      }
      spoji deserializer sa procesorom;
      konfiguriraj deserializer za priljeni port;
}
```

Programski pseudo-kod 4.2. Funkcija za paljenje kamere.

4.2. Konfiguracija kamere

Kameru je potrebno konfigurirati kako bi se dobilo željno ponašanje za vrijeme snimanja. Na primjer, postavlja se željena rezolucija, ciljani broj okvira po sekundi, uključivanje testnog uzorka, itd. Preko I^2C protokola kameri se šalju adresa registra i vrijednost koja se želi upisati u taj registar. Kamera ima velik broj registara i ova operacija može trajati relativno dugo. Funkcija za konfiguraciju kamere prima adresu registra i vrijednost koju treba upisati u taj registar i poziva se sve dok se ne upišu vrijednosti u svaki zadani registar. Prikaz pseudo koda funkcije se nalazi u **programskom pseudo-kodu 4.3.**

```
Funkcija konfiguracija_kamere (primljeni parametri: adresa registra,
vrijednost)
{
      pokušaj upisati vrijednost u registar;
      ako je vrijednost uspješno upisana
      {
            nastavi sa programom
      }
      ako vrijednost nije uspješno upisana
      {
            prekini program;
            obavjesti korisnika o pogrešci;
      }
}
```

programski pseudo-kod 4.3. Funkcija za konfiguraciju kamere.

4.3. Pokretanje i zaustavljanje snimanja

Pokretanje i zaustavljanje snimanja se izvršava vrlo jednostavno. Samo je potrebno kameri poslati naredbu za pokretanje odnosno zaustavljanje snimanja. Registri koji su zaslužni za kontrolu snimanja su 0x3012 za senzor OV10640 i 0x0100 za senzor OV10635. Upisivanjem vrijednosti 0x01 u navedene registre kamere započinju snimanje. Upisivanjem vrijednosti 0x00 kamere zaustavljaju snimanje. Funkcije za pokretanje i zaustavljanje snimanja na kameri prikazane su **programskim pseudo-kodom 4.4.**

```
Funkcija pokreni_snimanje_na_kameri (ulazni parameter: broj porta)
{
      Spoji deserializer sa procesorom;
      pokušaj upisati vrijednost 0x01 u registar 0x3012 ili 0x0100 u
kameru;
      ako je vrijednost uspješno upisana
      {
            nastavi sa programom
      }
      ako vrijednost nije uspješno upisana
      {
            prekini program;
            obavjesti korisnika o pogrešci;
      }
}

Funkcija zaustavi_snimanje_na_kameri (ulazni parameter: broj porta)
{
      Spoji deserializer sa procesorom;
      pokušaj upisati vrijednost 0x00 u registar 0x3012 ili 0x0100 u
kameru;
      ako je vrijednost uspješno upisana
      {
            nastavi sa programom
      }
      ako vrijednost nije uspješno upisana
      {
            prekini program;
            obavjesti korisnika o pogrešci;
      }
}
```

programski pseudo-kod 4.4. Funkcije za pokretanje i zaustavljanje snimanja na kameri.

4.4. Upravljanje VDMA upravljačima

Nakon što kamere započnu snimanje, pokreće se nit (engl. *thread*) koja obrađuje video podatke i nit koja upravlja VDMA upravljačima. Uključuje se VDMA upravljač za svaku aktivnu kameru. Podaci sa kamere se spremaju u jedan međuspremnik koji se nakon popunjavanja počinje prebacivat na RAM računala preko PCIe sabirnice. Dok se taj međuspremnik prebacuje, VDMA-e pune drugi međuspremnik s novim podacima iz senzora kamera. Kada se drugi spremnik popuni, prvi je već prebačen i spreman je za primanje novih podataka. Postupak se ponavlja dok traje snimanje. Korišteni su prekidi programa (engl. *interrupt*) i povratne funkcije (engl. *Callback function*) kako bi se ostvarila komunikacija između

niti. Prikaz funkcija koje se pozivaju prilikom pokretanja i zaustavljanju snimanja nalazi se u **programskom pseudo-kodu 4.5.**

```
Funkcija pokreni_snimanje (Ulazni parameter: broj porta)
{
      Pokreni kameru na primljenom portu;
      Omogući VDMA interruptove;
      Pokreni VDMA-u za primljeni port;
      Nakon prijenosa određenog broja okvira pozovi callback
      funkciju za VDMA zapisivanje;

      Ako nije došlo po pogreške
      {
            Zapiši okvire;
      }
      Ako je došlo po pogreške
      {
            Obavijesti korisnika o pogrešci;
      }
}

Funkcija zaustavi_snimanje (Ulazni parameter: broj porta)
{
      Zaustavi VDMA-u;
      Zaustavi kameru na primljenom portu;
      Onemogući VDMA interruptove;
}
```

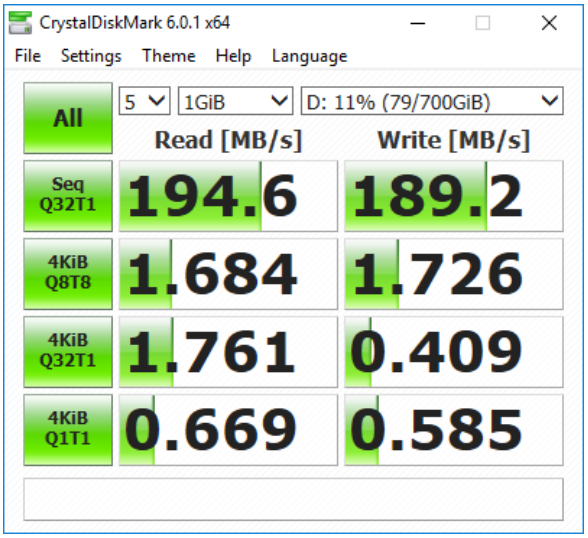
Programski pseudo-kod 4.5. Funkcije za pokretanje i zaustavljanje snimanja koje koriste VDMA-e za spremanje podataka u memoriju.

5. REZULTATI TESTIRANJA

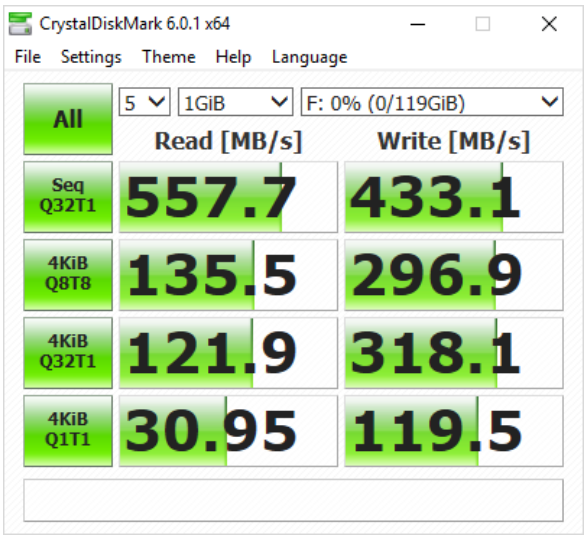
U ovom poglavlju su izneseni rezultati testiranja implementiranog rješenja. Nakon rješavanja bilo kakvog problema važno je napraviti testiranja kako bi se dobio uvid u hardverska ili softverska ograničenja, performanse i eventualne greške u implementiranom kodu.

5.1. Maksimalni broj kamera

Nakon osposobljavanja jedne kamere i jednog VDMA upravljača, postepeno je testirano snimanje s više kamera odjednom. Ispostavilo se da se javljaju određena hardverska ograničenja prilikom zapisivanja video datoteke na tvrdi disk računala. Testiranjem je utvrđeno da je maksimalni broj kamera koji može istovremeno snimati i zapisivati video datoteku na tvrdi disk računala puno manji od 9. Brzina zapisivanja tvrdog diska testirana je programom *CrystalDiskMark* [15]. Kao što se može vidjeti na **slici 5.1.** mehanički tvrdi disk koji je korišten ima brzinu zapisivanja od otprilike 190 *MB/s*. Na **slici 5.2.** mogu se vidjeti performanse korištenog *SSD* (engl. *Solid State Drive* – vrsta diska sa ugrađenom *flash* memorijom) diska. Njegova brzina zapisivanja iznosi oko 433 *MB/s.*



Slika 5.1. Brzine mehaničkog diska.



Slika 5.2. Brzine SSD diska

Količina podataka koja se prenosi u jednoj sekundi videa za proizvoljni broj kamera opisana je formulom (5-1).

$$Brzina\ prijenosa = rezolucija \cdot broj\ bajtova\ po\ elemntu\ slike \cdot broj\ okvira\ po\ sekundi \cdot broj\ kamera\ [B/s] \tag{5-1}$$

Implementirana je podrška za rezolucije 1280x1080 za senzor OV10640 i 1280x800 za senzor OV10635. Za 1 kameru, uvrštavanjem u formulu (5-1), dobivamo potrebnu brzinu prijenosa podataka od približno 79 *MB/s* za OV10640 i 59 *MB/s* za OV10635. Smanjivanjem rezolucije povećava se broj kamera koje istodobno mogu snimati video sadržaj i zapisivati ga. Pri tome se značajno smanjuje kvalitete slike, što nije pogodno za ADAS algoritme jer moraju biti što precizniji i točniji. Iz tog razloga napravljena je podrška samo za maksimalne rezolucije koje kamere mogu podržati.

$$Brzina\ prijenosa\ (OV10640) = 1280 \cdot 1080 \cdot 2 \cdot 30 \cdot 1 = 82944000\ B/s \approx 79\ MB/s$$

$$Brzina\ prijenosa\ (OV10635) = 1280 \cdot 800 \cdot 2 \cdot 30 \cdot 1 = 61440000\ B/s \approx 59\ MB/s$$

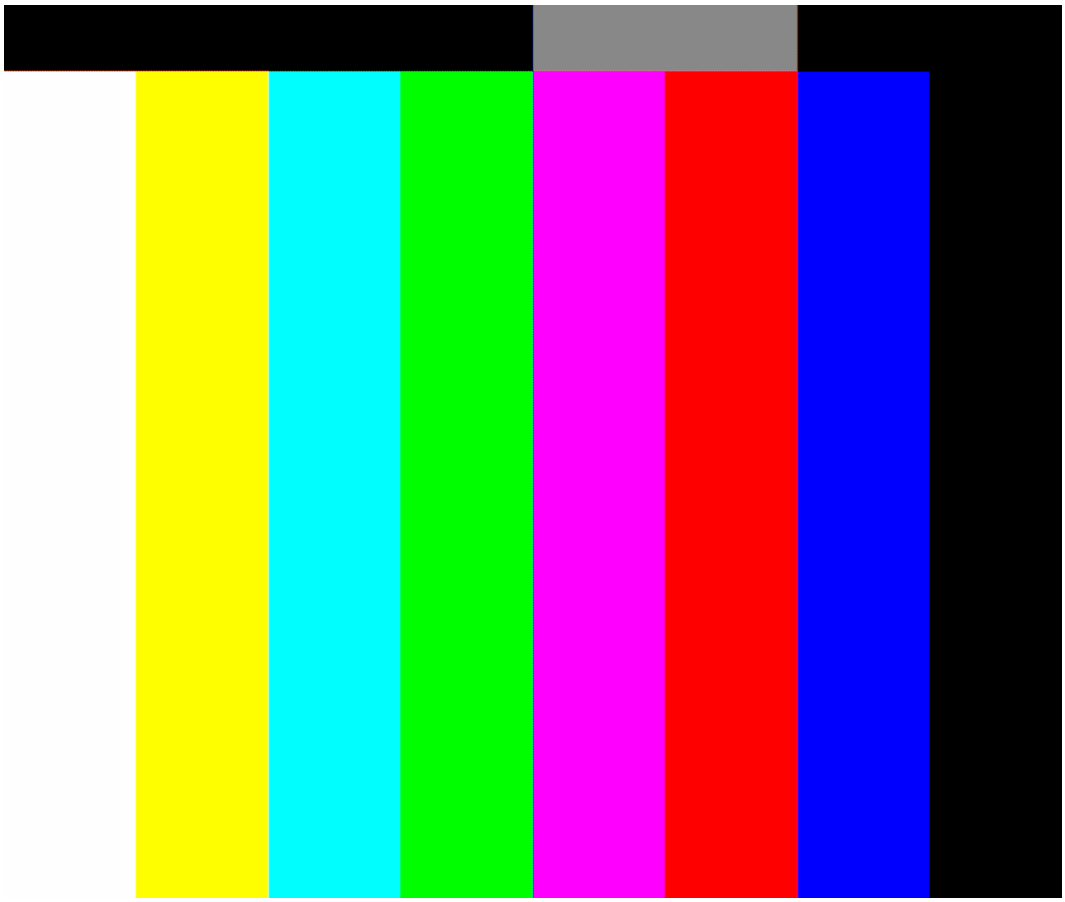
Kako prosječna brzina zapisivanja mehaničkog tvrdog diska iznosi oko 190 *MB/s*, maksimalni broj kamera koji se može koristiti bez ikakvih problema na takvom disku je 2 ako se

koristi senzor OV10640, ili 3 ako se koristi senzor OV10635. Brzina zapisivanja SSD diska je oko 433 *MB/s* što je dovoljno za snimanje sa 5 OV1040 kamera ili 7 OV10635 kamera u isto vrijeme. Moguće je pokrenuti snimanje i sa većim brojem kamera, ali postoji velika mogućnost da dođe do gubitaka okvira, raznih artefakata na slici ili nekih drugih nepredviđenih ponašanja. Na **slici 5.3.** prikazan je jedan okvir koji je uspješno snimljen kamerom.



Slika 5.3. Jedan okvir videa snimljenog kamerom.

Pošto ni jedan prototip AMV Grabber ploče koji je bio na raspolaganju za testiranje nije imao funkcionalno svih 9 portova, testiranja s 9 kamera su napravljena pomoću FPGA generatora okvira. Generator okvira u osnovi emulira kamere i proizvodi testni uzorak koji se preko VDMA upravljača šalje prema RAM memoriji ploče. Primjer jednog testnog okvira može se vidjeti na **slici 5.4.**



Slika 5.4. Prikaz testnog okvira generiranog sa FPGA generatorom okvira.

Teoretska maksimalna brzina prijenosa podataka DDR3 upravljača je otprilike 4.2 *GB/s*. Može se zaključiti da RAM nije usko grlo i da bez poteškoća može raditi sa maksimalnim brojem kamera. Brzina prijenosa podataka preko PCIe sabirnice je otprilike 2 *GB/s* i također ne predstavlja usko grlo jer se na računalo prenosi maksimalno 711 *MB/s* ako radi svih 9 kamera. Količina RAM memorije (1 *GB*) je također dostatna za spremanje svih okvira.

5.2. Prijedlog za povećanje maksimalnog broja kamera

Brzina zapisivanja podataka na mehanički tvrdi disk predstavlja najveću prepreku pri korištenju predviđenog broja kamera. Jedan od mogućih načina za povećavanje broja kamera koje mogu snimati i zapisivati video datoteke istovremeno je korištenje SSD diskova koji se spajaju na računalo putem PCIe sučelja [16]. Takvi diskovi u pravilu imaju znatno veće brzine od „običnih“ SSD diskova koji se spajaju putem *SATA III* (engl. *Serial Advanced Technology Attachment Third Generation*) sučelja [17]. Teoretski, PCIe SSD diskovi bi bez problema mogli pružiti dovoljnu brzinu pisanja da podrže zapisivanje video datoteka sa svih 9 kamera istovremeno. Najveća mana takvih diskova je izrazito visoka cijena i relativno mali kapacitet.

6. ZAKLJUČAK

Ovim radom uspješno je razvijena podrška za kontrolu 2 vrste kamera na platformi za pomoć pri razvoju i testiranju ADAS algoritama. Kontrola kamere obuhvaća konfiguraciju senzora kamere, detekciju neuspješne konfiguracije, kontrolu odgovarajućih VDMA upravljača te pokretanje i zaustavljanje snimanja. Podržane su kamere koje koriste OV10635 i OV10640 senzore. Platforma se temelji na Zynq XC7Z030 procesorskoj jedinici. Teoretski se podržava snimanje i zapisivanje podataka s 9 kamera istovremeno, ali testiranja su pokazala da u praksi simultano može raditi najviše 5 kamera (na testnoj konfiguraciji). U radu je predloženo rješenje hardverskog ograničenja koje se javilo prilikom implementacije. Projekt je razvijan u *Xilinx SDK* i *Xilinx Vivado Design Suite* razvojnim okruženjima.

Ovaj diplomski rad riješio je samo jednu problematiku ukupnog projekta. Cijeli projekt je rađen u timu i svaki član tima je imao zadatak implementirati određen podsustav uređaja. Konačni proizvod je u potpunosti funkcionalan, ali postoje neka odstupanja od prvobitnih planova zbog hardverskih ograničenja ili nedostatka vremena za implementaciju. Razvijeni uređaj bi se u budućnosti mogao koristiti za ubrzavanje procesa izrade i testiranja ADAS algoritama. Kako se ADAS tehnologija sve više razvija i postaje popularnija, potreba za ovakvom vrstom uređaja će također sve više rasti. Auto industrija će se nastaviti rapidno razvijati i gotovo je sigurno kako će kompleksnost ADAS algoritama rasti do te mjere da će omogućiti sigurnu autonomnu vožnju u ne tako dalekoj budućnosti.

LITERATURA

[1] „ ZMP for Autonomous Driving ADAS Technology Change the movement of people and goods in practical use“. [Na internetu]. Dostupno na: https://www.zmp.co.jp/en/knowledge/adas_dev. [Pristupljeno: 04-ruj-2018].

[2] „Proto board enables testing of active safety systems“. [Na internetu]. Dostupno na: https://automotive.electronicspecifier.com/design/proto-board-enables-testing-of-active-safety-systems. [Pristupljeno: 14-ruj-2018].

[3] „ARCCORE - News“. [Na internetu]. Dostupno na: https://www.arccore.com/news/arccore-introduces-new-arctic-fusion-prototype-board-with-potential-for-adas-development-2016-06-02. [Pristupljeno: 14-ruj-2018].

[4] „RT-RK - Automotive“. [Na internetu]. Dostupno na: http://www.rt-rk.com/services/automotive. [Pristupljeno: 14-ruj-2018].

[5] M. Krbanjević, I. Rešetar, i V. Škobić, „Univerzalna platforma za ispitivanje uređaja za mašinsku vizuelnu percepciju okoline u sistemima za pomoć u vožnji i autonomno kretanje vozila“. [Na internetu]. Dostupno na: https://www.etran.rs/common/pages/proceedings/ETRAN2017/RT/IcETRAN2017_paper_RT1_4.pdf. [Pristupljeno: 11-ruj-2018].

[6] „Valdez and Becker - 2015 - Understanding the I2C Bus.pdf“. .

[7] „Get Connected: SerDes demystified - Analog Wire - Blogs - TI E2E Community“. [Na internetu]. Dostupno na: https://e2e.ti.com/blogs_/b/analogwire/archive/2014/05/08/get-connected-serdes-demystified. [Pristupljeno: 06-ruj-2018].

[8] „OmniVision | A leading developer of advanced digital imaging solutions.“ [Na internetu]. Dostupno na: https://www.ovt.com/sensors/OV10640. [Pristupljeno: 11-ruj-2018].

[9] „OmniVision | A leading developer of advanced digital imaging solutions.“ [Na internetu]. Dostupno na: https://www.ovt.com/sensors/OV10635. [Pristupljeno: 11-ruj-2018].

[10] „OmniVision | A leading developer of advanced digital imaging solutions.“ [Na internetu]. Dostupno na: https://www.ovt.com/. [Pristupljeno: 11-ruj-2018].

[11] „AXI Video DMA (AXI VDMA)“. [Na internetu]. Dostupno na: https://www.xilinx.com/products/intellectual-property/axi_video_dma.html#documentation. [Pristupljeno: 10-ruj-2018].

[12] „AXI Video Direct Memory Access v6.3 - LogiCORE IP Product Guide“. [Na internetu]. Dostupno na: https://www.xilinx.com/support/documentation/ip_documentation/axi_vdma/v6_3/pg020_axi_vdma.pdf. [Pristupljeno: 11-ruj-2018].

[13] „Xilinx Software Development Kit (XSDK)“. [Na internetu]. Dostupno na: https://www.xilinx.com/products/design-tools/embedded-software/sdk.html. [Pristupljeno: 10-ruj-2018].

[14] „Vivado Design Suite“. [Na internetu]. Dostupno na: https://www.xilinx.com/products/design-tools/vivado.html. [Pristupljeno: 10-ruj-2018].

[15] „CrystalDiskMark“, *Crystal Dew World [en]*. [Na internetu]. Dostupno na: https://crystalmark.info/en/software/crystaldiskmark/. [Pristupljeno: 10-ruj-2018].

[16] „PCIe vs. SATA: Which Type of SSD Is Best for You?“, *MakeUseOf*. [Na internetu]. Dostupno na: https://www.makeuseof.com/tag/pcie-vs-sata-type-ssd-best/. [Pristupljeno: 10-ruj-2018].

[17] „What is Serial ATA (Serial Advanced Technology Attachment or SATA)? - Definition from WhatIs.com“, *SearchStorage*. [Na internetu]. Dostupno na: https://searchstorage.techtarget.com/definition/Serial-ATA. [Pristupljeno: 10-ruj-2018].

ABSTRACT

This thesis is based on enabling and controlling camera modules of the dedicated embedded device (AMV Grabber) which was developed in the RT-RK research and development institute. CPU of the embedded device is based on the Xilinx SoC architecture. Software was written in C programming language and it deals with implementation of Xilinx IP DMA blocks such as VDMA. The FPGA design of the board relies on these blocks. Software modules for controlling cameras and VDMA are written in *Xilinx SDK*. They include powering specific cameras on and off, configuring cameras, turning on specific VDMA controllers, starting and stopping video streams from cameras. Theoretically, AMV Grabber board can control up to 9 cameras simultaneously, but there are some hardware limitations in practical application which are also described in this paper. Developed board can greatly speed up process of testing and implementation ADAS algorithms.

Keywords: AMV Grabber**,** ADAS, VDMA, embedded systems, C programming language, camera, Xilinx, FPGA

SAŽETAK

Ovaj je rad temeljen na omogućavanju i kontroli kamera namjenskog ugradbenog uređaja (AMV Grabber) koji je razvijen u institutu za istraživanje i razvoj RT-RK. Procesorska jedinica ugradbenog uređaja temelji se na Xilinx SoC arhitekturi. Softver je napisan u C programskom jeziku i bavi se implementacijom Xilinx IP DMA blokova kao što je VDMA. FPGA dizajn ploče oslanja se na ove blokove. Softverski moduli za upravljanje kamerama i VDMA su napisani u *Xilinx SDK* razvojnom okruženju. Oni uključuju uključivanje i isključivanje određenih kamera, njihovu konfiguraciju, uključivanje određenih VDMA upravljača, pokretanje i zaustavljanje snimanja. Teoretski, AMV Grabber ploča može kontrolirati do 9 kamera istovremeno, ali postoje neka hardverska ograničenja u praktičnoj primjeni koja su također opisana u ovom radu. Ovaj ugradbeni uređaj može uvelike ubrzati proces razvijanja i testiranja ADAS algoritama.

Ključne riječi: AMV Grabber, ADAS, VDMA, ugradbeni sustavi, C programski jezik, kamera, Xilinx, FPGA

ŽIVOTOPIS

Dominik Mitrović rođen je u Slavonskom Brodu 17.04.1993. gdje provodi većinu svog života. Pohađao je osnovnu školu „Ivan Goran Kovačić“ koju prolazi s odličnim uspjehom. 2008. godine upisuje opću gimnaziju „Matija Mesić“ u Slavonskom Brodu koju 2012. koju također završava s odličnim uspjehom. Iste godine upisuje preddiplomski studij elektrotehnike na Elektrotehničkom fakultetu sveučilišta Josipa Jurja Strossmayera u Osijeku. Na drugoj godini se opredjeljuje za smjer „Komunikacije i informatika“. Nastavlja diplomski studij 2016. na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, odabire izborni blok Mrežne tehnologije. 2017. godine postaje stipendist Instituta RT-RK u Osijeku gdje obavlja praksu i radi na diplomskom radu.

Ima odlično poznavanje engleskog jezika te je informatički pismen. Zna se koristiti velikim brojem programerskih razvojnih alata i programa. Od programskih jezika zna C i Javu, a snalazi se i sa Pythonom, HTML-om, CSS-om, JavaScript-om i C#-om. Vrlo je snalažljiv u radnom okruženju.