TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Veiko Rütter 214250IACB

KIIPLABORI KAAMERAMOODULI EDASIARENDUS

Bakalaureusetöö

Juhendaja: Kaiser Pärnamets
Magister
Lektor
Kaasjuhendaja: Ants Koel
Doktor
Dotsent

Tallinn 2022

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Veiko Rütter

16.05.2022

Annotatsioon

Käesoleva bakalaureusetöö eesmärk on arendada tarkvara ONsemi PYTHON300 sensoril põhineva kaameramooduli draiveri (*driveri*) jaoks. Riistvara oli eelnevalt valmis disainitud ning koostatud, kuid testimata. Riistvara koosneb pistikutega sidusplaadist, PicoZed 7020 töötlusmoodulist ning kaamera kandemoodulist. Bakalaureusetöö ülesannete hulka kuulub riistvara testimine, Linuxi operatsioonisüsteemi koostamine ja installeerimine töötlusmoodulile, VHDL (*VHSIC Hardware Description Language*) disain kaamera ja töötlusmooduli sidumiseks ning tarkvara kirjutamine töötlusmoodulile programmeerimiskeeles C.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 21 leheküljel, 7 peatükki, 15 joonist, 2 tabelit.

Abstract

Camera Module Development for LoC Application

The aim of this bachelor's thesis is to develop a software for a camera module, which is based on the ONsemi PYTHON300 sensor. The hardware was designed and assembled previously, but it is not tested. The hardware consists of a breakout module with connectors, PicoZed 7020 processing module and camera sensor carrier module. Bachelor's thesis tasks include hardware testing, compiling and installing a Linux operating system on a processing module, VHDL (*VHSIC Hardware Description Language*) design for connecting the camera sensor and the processing module, and writing the software in the programming language C for the processing module.

The thesis is in Estonian language and contains 21 pages of text, 7 chapters, 15 figures, 2 tables.

Lühendite ja mõistete sõnastik

ARM	Advanced RISC Machine
AXI	Advanced Extensible Interface
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
eMMC	Embedded MultiMediaCard
FIFO	First In First Out
FIT	Flattened Image Tree
FPGA	Field-Programmable Gate Array
GPIO	General Purpose Input Output
HTTP	Hypertext Transfer Protocol
IRQ	Interrupt Request
LAN	Local Area Network
LoC	Lab-on-a-chip
LVDS	Low-Voltage Differential Signaling
MMC	MultiMediaCard
PCB	Printed Circuit Board
PL	Programmable Logic
PLL	Phase Locked Loop
PNG	Portable Network Graphics
PS	Processing System
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROI	Region of Interest
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Antud bakalaureusetöö on üks osa kiiplabori LoC (*Lab-on-a-chip*) voolutsütomeetria süsteemist. Käesoleval hetkel on olemas tööstuslikke voolutsütomeetria seadmeid, mis on mõeldud töötamiseks töötasapinnal ning need sõltuvad välistest töötlussüsteemidest [1] [2] . Kuid tänapäeva tehnoloogia võimaldaks ehitada kaasaskantavaid seadmeid, mis suudaksid tulemusi analüüsida koheselt. Kiiplabor on seade mis sisaldab endas mitmeid analüüsimise funktsioone väga väikesel alal, tavaliselt ruutmillimeetril [3] .

Arendus põhineb magistritöö alustel eelnevalt valmistatud riistvaral [4] . Oluline aspekt süsteemi seisukohast on, et kiirus millega kaamera sensorilt kaadreid salvestatakse, oleks ligikaudu 1000 kaadrit sekundis. Lisaks peab saama lihtsate vahenditega kaamera sensorit konfigureerida, kuna süsteemi on vaja järgnevates projektides vastavalt katsetulemustele häälestada.

1.1 Ülesandepüstitus

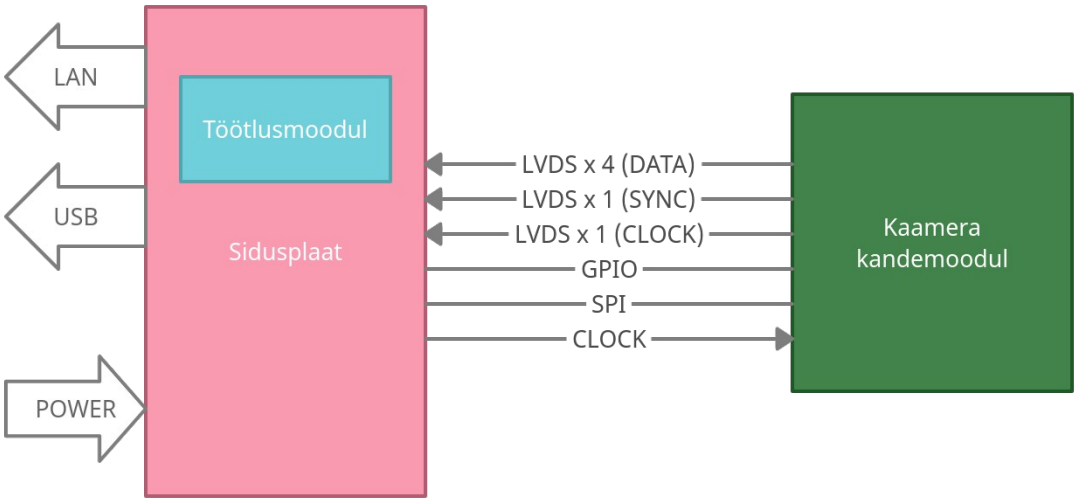
Leida ja parandada trükkplaatide vead. Arendada või integreerida ning testida sardtarkvara (*firmware*) edasi täisfunktsionaalse kaameramooduli valmistamise suunas. Alamülesanneteks on seejuures Linuxi tuuma kohandamine, liideste töölepanek (LAN (*Local Area Network*), GPIO (*General Purpose Inout Output*), SPI (*Serial Peripheral Interface*), LVDS (*Low-Voltage Differential Signaling*)), andmevahetuse planeerimine ja testimine pildisensori, RAM (*Random Access Memory*), FPGA (*Field-Programmable Gate Array*), ja mikroprotsessori vahel.

Arenduse vahenditeks sai valitud:

- Xilinx Vivado v2019.1 (64-bit) [5] . Projektis kastutatava süsteemikiibi jaoks ametlikult pakutav arenduskeskond.
- arm-linux-gnueabi-gcc (Debian 11.2.0-18) 11.2.0 [6] . Vabavarana saadaval C kompilaator, mis võimaldab ristkompileerimist erinevate arhitektuuride vahel.

2 Riistvara ehitus

Riistvara koosneb kolmest komponendist - pistikutega sidusplaadist, PicoZed 7020 töötlusmoodulist ning kaamera kandemoodulist. Riistvara plokkskeemist annab ülevaate Joonis 1.



Joonis 1. Riistvara plokkskeem peamiste ühendustega sidusplaadi ja kaamera kandemooduli vahel.

Kaamera kandemoodul ühendub kahe lintkaabliga sidusplaadi külge ning on viimase kaudu omakorda ühendatud töötlusmooduli FPGA külge. Lisaks on lahendatud kaamera kandemooduli PCB (*Printed Circuit Board*) peal ka sensori toitelahendus.

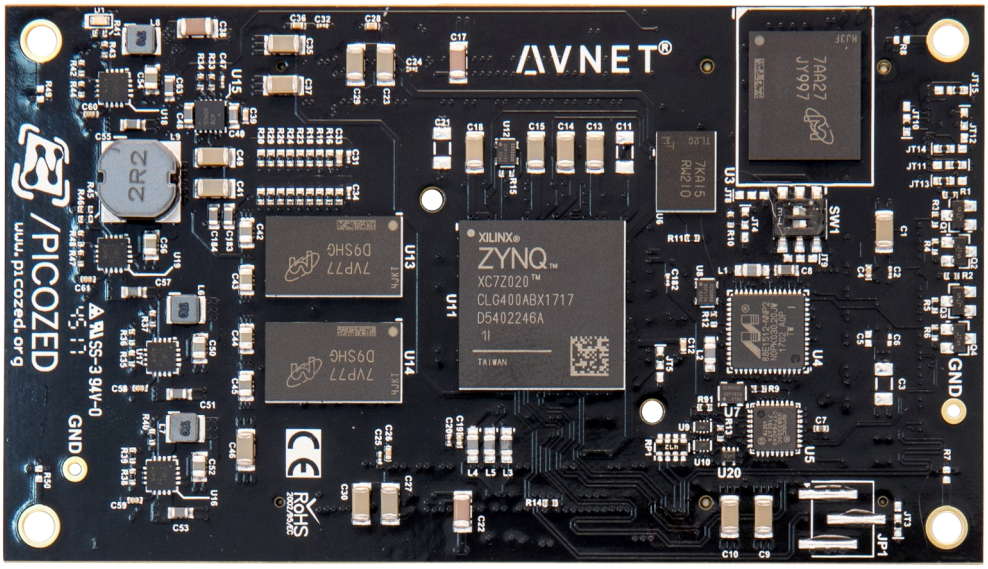
2.1 Sidusplaat

Sidusplaat sisaldab endas töötlusmooduli toitelahendust ning erinevaid sisend-väljund liideseid nagu USB (*Universal Serial Bus*), *Ethernet*, GPIO. Sidusplaati on vaja eeskätt PicoZed 7020 mooduli hoidmiseks ning kaamera kandemooduli ühendamiseks. Sidusplaat oli eelnevalt disainitud ning kokku joodetud, kuid testimata [4] .

2.2 Töötlusmoodul

Töötlusmoodul PicoZed 7020 on Avnet poolt disainitud moodul, mis sisaldab Xilinx Zynq-7000 süsteemikiipi (XC7Z020-1CLG400), 1 GB RAM ning 8 GB eMMC (*Embedded MultiMediaCard*) [7] . PicoZed 7020 on kujutatud Joonisel 2.

Zynq-7000 on süsteemikiip mis koosneb ARM (*Advanced RISC Machine*) Cortex-A9 mikroprotsessorist PS (*Processing System*) ning integreeritud FPGA integraallülitusest PL (*Programmable Logic*) [8] . Ühendus PS ja PL vahel baseerub AXI (*Advanced Extensible Interface*) lahendusel.



Joonis 2. PicoZed 7020 [7] .

2.3 Kaamera kandemoodul

Kaamera kandemoodul sisaldab ONsemi PYTHON300 sensorit [9] . Sensor on joodetud trükkplaadi peale, mis ühtlasi on mõeldud ka objektiivi kinnitamiseks.

Sensor vajab enda tööks taktsignaali kiirusega 72 MHz ning sensorit konfigureeritakse SPI liidese kaudu. Sensor edastab kaadreid üle kuue LVDS signaalipaari, millest üks on taktsignaal, üks on sünkronisatsiooni kanal ning ülejäänud neli on andmekanalid.

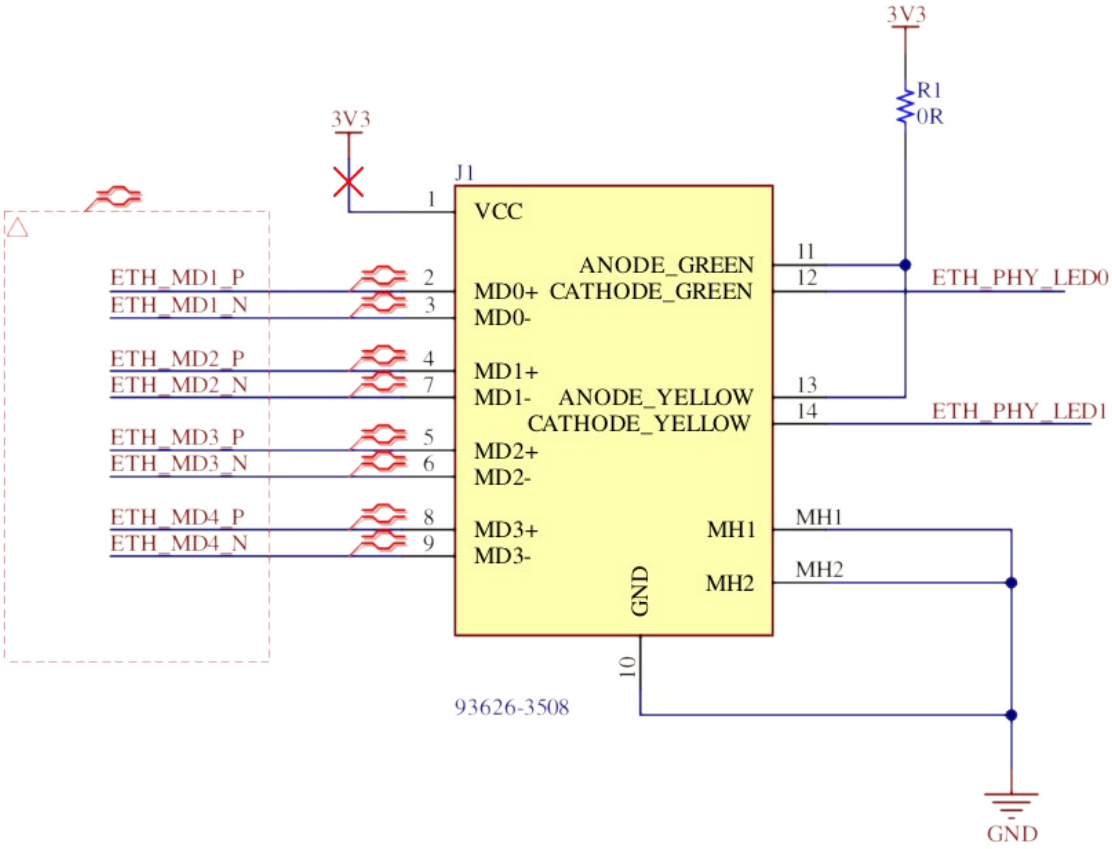
3 Riistvara testimine

Toimiv riistvara on eelduseks edasistele arendustele. Kuna riistvara oli testimata, siis esimene ülesanne kujutas endast kõikide erinevate komponentide töötamise veendumises. Peale elektriskeemi põhjalikku uurimist sai kontrollitud, et riistvara erinevate toiteradade vahel ei esineks lühiseid. Kõigepealt ühendati toiteploki külge sidusmoodul ilma tööstusmoodulita ning ilma kaamera kandemoodulita. Seejärel veenduti kõikide erinevate toiteradade pingetes. Järgmine samm oli sidusplaadi toitmine koos PicoZed 7020 töötlusmooduliga ning kõige lõpuks koos kaamera kandemooduliga. Järgnevalt on välja toodud avastatud vead ning võimalikud lahendused nende parandamiseks.

3.1 LAN link

Sidusplaadile on disainitud RJ45 pistik, mis on ühendatud töötlusmooduli LAN signaalidega, võimaldamaks 1 Gbit/s ühenduskiirust. Kuid ühendades RJ45 pistiku võrku, ei tuvastanud Linuxi draiver linki. RJ45 pistik on Molex 93626-3508 [10] . Molexi andmelehel on näidatud, et viik number 1 peaks olema toites, kuid PicoZed 7020 skeemi näidetes on jäetud vastav viik ühendamata.

Lahendusena sai ettevaatlikult läbi lõigatud RJ45 viik number 1. Pärast seda tuli LAN link üles, kuid 1 Gbit/s kiirust ei tuvastatud, jäi ainult 100 Mbit/s. Kogu ülejäänud süsteemi edasiseks arendamiseks sellest esialgu ka piisab. Läbi lõigatud viiku kirjeldab Joonis 3.



Joonis 3. RJ45 pistik ning läbi lõigatud viik.

3.2 Kaamera kandemooduli 1.8 V toide

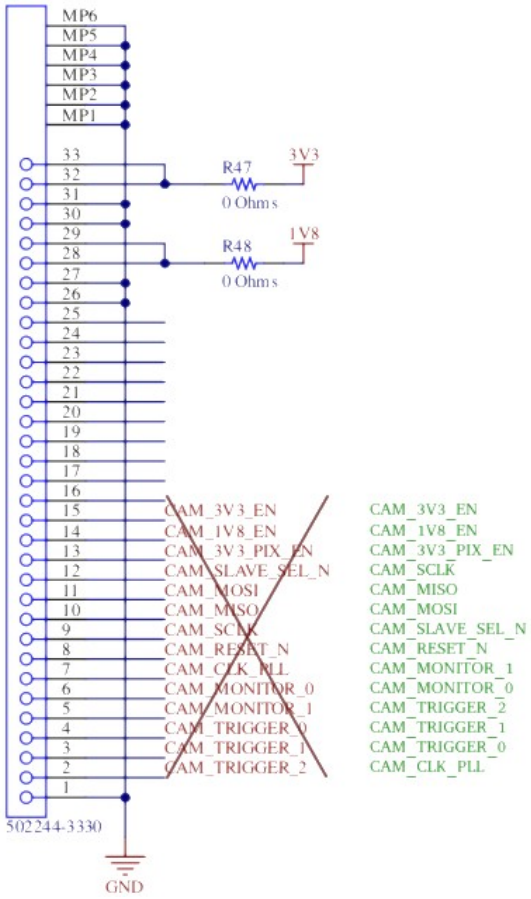
Multimeetriga testides selgus, et kaamera kandemooduli 1.8 V toiterada on lühises. Kuna elektriskeemil ja PCB disainil lühist ei olnud, oli järgmine oletus vigases jootes. Ükshaaval 1.8 V toiteraja küljes olevaid komponente eemaldades saigi leitud selline komponent, mille all oli liiga palju joodist ja see omakorda tekitas lühise. Tegemist oli erinevate passiivsete komponentidega, korpuse suurusega 0603. Trükkplaadi jooteväljad olid liiga lähestikku.

Komponentide korrektsel tagasi jootmisel kadus ka antud viga.

3.3 Kaamera kandemooduli elektriskeem

Kuna esialgne kommunikatsioon kaamera kandemooduliga ei õnnestunud, siis uurimisel selgus, et kaamera kandemooduli elektriskeemi ja PCB lintkaablite pistikute viigud olid erinevad.

Multimeetriga sai viigud üle mõõdetud ning koostatud korrigeeritud lintkaabli elektriskeem, mis on kirjeldatud Joonisel 4.



Joonis 4. Korrigeeritud lintkaabli ühendusskeem.

3.4 Kaamera kandemooduli objektiivi kinnitusaugud

Kaamera kandemooduli PCB küljes on kaks auku, mille kaudu kinnitatakse poltidega objektiiv. Kuid need augud ei vasta objektiivi küljes olevatele aukudele. Augud on üksteisele liiga lähedal. Samuti jooksevad trükkplaadil olevad rajad objektiivi alt läbi, mistõttu võib tekkida lühise oht, kuna objektiiv on metallist. Uusi auke ei ole võimalik puurida, kuna vastavates asukohtades trükkplaadil asuvad komponendid ja rajad. Sama situatsioon on ka sel juhul, kui keerata objektiivi 90 kraadi.

Probleemi saab lahendada, kui disainida uus trükkplaat. Kuid testimise eesmärgil sai hetkel ajutise lahendusena kinnitatud objektiiv liimiga.

4 Linuxi seadistamine

Kuigi Zynq-7000 võimaldab laadida tarkvara ka ilma operatsioonisüsteemita, on siiski mõistlikum kasutada operatsioonisüsteemi, milles on juba implementeeritud võrgu protokollid ning riistvara draiverid. Üheks sobivaks operatsioonisüsteem selleks on Linux, mis sai valitud antud projekti jaoks [11] . Põhjus Linuxi valimiseks oli eelkõige selles, et kõik antud protsessori jaoks vajalikud draiverid on Linuxi tuumas olemas. Lisaks on Linuxi tuum avatud lähtekoodiga ning vajadusel oleks võimalik draivereid muuta või uusi lisada. Linuxi laadimise saab jagada neljaks etapiks:

- Riistvarasse sisse ehitatud alglaadur laeb välkmälust teise taseme alglaaduri ning käivitab selle.
- Teise taseme alglaadur leiab välkmälust Linuxi tuuma, *devicetree* ning *ramdiski*. Laeb need muutmällu ning käivitab Linuxi tuuma. Teise taseme alglaaduriks on U-Boot [12] .
- Linuxi tuum käivitub ning konfigureerib vastavalt *devicetrees* kirjeldatud draiverid. Lisaks haagib tuum omale juurfailisüsteemiks *ramdiski* ning käivitab sealt init nimelise programmi.
- Init nimeline programm on Linuxi kõige esimene käivitatud programm, ning kõik järgmised programmid on initi poolt käivitatud. Esimesed initi poolt käivitatud programmid seadistavad süsteemi ning pärast seda on operatsioonisüsteem laetud.

Uurides Avnet poolt pakutud U-Booti selgus, et U-Boot üritab laadida välkmälu juurkataloogis asuvat „image.ub” nimelist faili. Antud fail peab olema FIT (*Flattened Image Tree*) formaadis. FIT formaadis olev fail sisaldab endas nii Linuxi tuuma, *devicetreed* kui ka *ramdiski* ning seda faili on väga lihtne ise koostada. Selline lahendus sobis ideaalselt kuna võimaldas väga kiiret süsteemi testimist. Järgnevalt on välja toodud koostatud Linuxi distributsiooni olulisemad osad.

4.1 Linuxi tuum

Linuxi tuum tuleb konfigureerida ja kompileerida ARM arhitektuuri jaoks. Konfigureerimisel tuleb ära märkida Zynq-7000 jaoks vajaminevad draiverid, eeskätt UART (*Universal Asynchronous Receiver-Transmitter*), *Ethernet*, MMC (*MultiMediaCard*) ja FPGA framework. Viimane võimaldab FPGA konfiguratsiooni muuta Linuxi operatsioonisüsteemis [13] .

4.2 *Devicetree*

Devicetree kirjeldab, kuidas riistvara on protsessoriga ühendatud. Kui draiverid on tuuma sisse kompileeritud, siis tuleb nad ka *devicetree* failis ära kirjeldada, andes draiveritele vajadusel mäluaadressid ja IRQ (*Interrupt Request*) numbrid [14] . Põhjus nende parameetrite manuaalsel seadistamisel on selles, et suurem osa sardsüsteemide lisaseadmeid ei ole automaatselt tuvastatavad.

4.3 *Ramdisk*

Ramdisk on virtuaalne failisüsteem, mida hoitakse muutmälus. Käesoleva lahenduse jaoks on koostatud minimaalne failisüsteem, mis sisaldab init programmi, busybox nimelist programmi ning sensori juhtimiseks loodud programmi [15] .

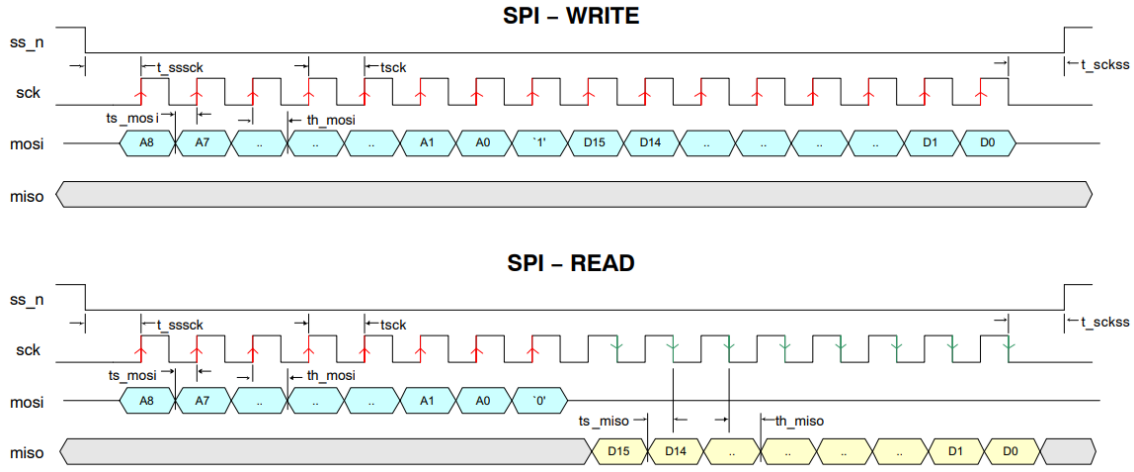
Lisaks antakse Linuxi tuumale käivitamisel info, et süsteemis saada olev muutmälu on 128 MB, kuna ülejäänud 896 MB on vaja jätta kaamera andmevoo talletamiseks.

5 Sensori andmevahetus

ONsemi PYTHON300 sensorit juhitakse SPI liidese kaudu ning kaadrite andmevahetus toimub LVDS liidese kaudu. Kuna aga sensor on ühendatud süsteemikiibi FPGA külge, siis ei pääse talle ilma FPGA konfigureerimiseta mikroprotsessorist ligi.

5.1 Sensori juhtimine

Esialgse lahendusena sai proovitud Xilinxi poolt valmis disainitud SPI plokki, mis ühendati mikroprotsessori külge AXI kaudu [16] . Antud lahendus ei töötanud korrektselt, sest Xilinxi SPI plokk võimaldas andmeid vahetada 8 biti kaupa. Sensori käsud aga sisaldavad 26 bitti. Sensori SPI protokolli kirjeldab Joonis 5.

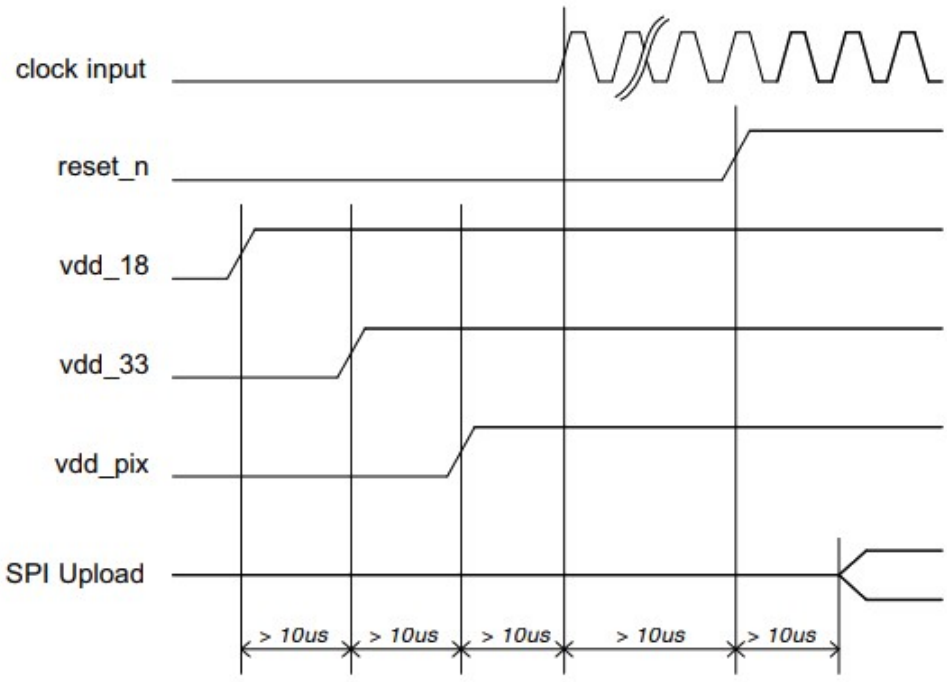


Joonis 5. Sensori SPI protokoll.

Lahendusena said SPI signaalid ühendatud läbi FPGA mikroprotsessori GPIO signaalidega, ning kogu SPI protokoll tarkvaras implementeeritud. Sensori juhtimine on ühekordne tegevus, seega ei ole oluline antud lahenduse kiirus.

Lisaks SPI protokollile vajab sensor ka kindlaks määratud järjekorras toidete sisse lülitamisi ning taktsignaali. Kaamera kandemooduli trükkplaadil on kolm toite muundurit ning nende juhtsignaalid on ühendatud FPGA külge. Samuti on ühendatud

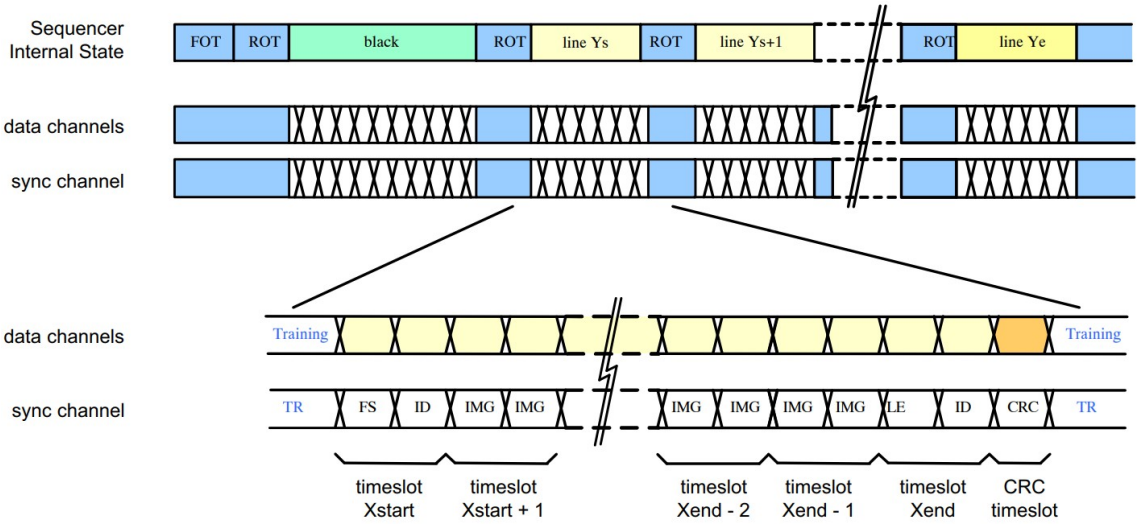
FPGA külge sensori /RESET signaal. Kõik need signaalid said läbi FPGA mikroprotsessori GPIO signaalidega ühendatud, et oleks võimalik neid tarkvaraliselt juhtida. Taktsageduse genereerib FPGA, kasutades Xilinxi poolt pakutud „Clocking Wizard" plokki, millel omakorda juhtsisendi signaal on ühendatud mikroprotsessori GPIO signaaliga [17] . Joonis 6 kirjeldab sensori sisse lülitamist.



Joonis 6. Sensori sisse lülitamine.

5.2 Andmevahetus

Kaadrite andmed ONsemi PYTHON300 sensorist saadetakse töötlusmooduli FPGA osale LVDS signaalipaaride kaudu. FPGA ülesanne on kaadrite andmed vastu võtta, ning anda need edasi mikroprotsessorile. Levinud lahendus sellistel juhtudel on DMA (*Direct Memory Access*) kasutus. Võttes kasutusele DMA, suudab FPGA selle abil andmeid otse mikroprotsessori muutmällu kirjutada. Lisaks peab FPGA analüüsima sünkronisatsiooni kanalit, et aru saada millal kaader algab ja lõpeb. Sünkronisatsiooni ning andmekanalite taktsignaal saadetakse samuti LVDS signaalipaari kaudu. Sensor võtab sisendiks FPGA poolt genereeritud taktsignaali ja genereerib sellest omakorda 5x kõrgema sagedusega signaali PLL (*Phase Locked Loop*) abil ning kasutab saadud signaali LVDS taktsignaaliks. Ülevaate üle LVDS signaalipaaride edastatavatest andmetest annab Joonis 7.



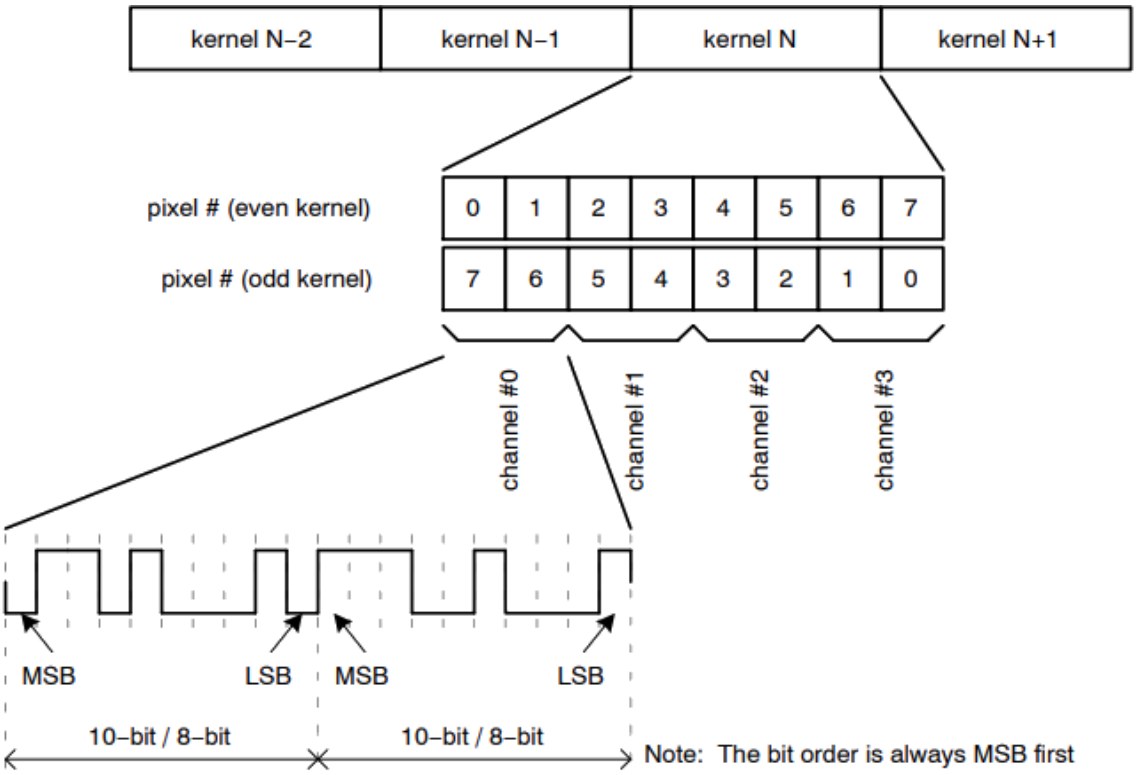
Joonis 7. Andmete edastus üle LVDS signaalide.

Sünkronisatsiooni kanal kasutab erinevate väärtustega andmesõnu, mille abil saab kindlaks teha, mis tüüpi andmed parasjagu andmekanalites on. Eriti tähtis on sünkronisatsiooni kanalis saadetav *training pattern*. Selle abil leitakse andmesõnade algused bitijadast. Tabel 1 annab ülevaate erinevatest sünkronisatsiooni andmesõnadest.

Tabel 1. Sünkronisatsiooni kanali andmesõnad.

Nimi	**Väärtus**	**Tähendus**
TR	0x3A6	*Training pattern*. Selle järgi leitakse bitijadast andmesõnade algused.
BL	0x015	Must piksel. Andmeid kasutab sensor sisemiselt, et korrigeerida filtrite parameetreid.
FS	0x2AA	Kaadri algus.
FE	0x32A	Kaadri lõpp.
LS	0x0AA	Horisontaalse pikslirea algus kaadris.
LE	0x12A	Horisontaalse pikslirea lõpp kaadris.
IMG	0x035	Piksli andmed.
CRC	0x059	CRC (*Cyclic Redundancy Check*) väärtus. Iga horisontaalse pikslirea kohta arvutatakse CRC väärtus.

Lisaks sünkronisatsiooni andmesõnadele on vaja veel jooksvalt kaadri pikslite asukohta arvutada. Olenevalt pikslite asukohast, saadetakse neid erinevatel LVDS kanalitel. Piksleid loetakse sensoritest nn. tuumade (*kernel*) kaupa. Üks tuum koosneb kaheksast pikslist. Ning vaheldumisi järjestatakse pikslid andmekanalitesse vastupidises järjekorras eelmisele. Joonis 8 kirjeldab pikslite asukohta andmekanalites.



Joonis 8. LVDS pikslite formaat.

Avnet pakub enda tarkvara repositooriumis ONsemi sensorite dekodeerimise plokki FPGA jaoks [18] . Tegu on käesolevaks hetkeks üsna vana tarkvaraga, kuid selle kasutuselevõtt tundus alguses mõistliku lähenemisena. Antud plokk võtab sisse kõik sensori LVDS kanalid ning genereerib analüüsitud andmetest video väljundi. Saadud video väljund on vaja muuta AXI andmevooks kasutades Xilinxi poolt pakutavat „Video In To AXI4-Stream” plokki [19] . Ning antud andmevoo saab mikroprotsessori muutmällu kirjutada „AXI Video Direct Memory Access” ploki abil [20] .

Sellel lahendusel oli aga mitmeid puudusi:

- Avnet näitekood oli tehtud ONsemi PYTHON1300 kaamera jaoks. Kuid see näitekood ei suutnud PYTHON300 kaameraga, millel on väiksem pildiresolutsioon, korrektset kaadrit edastada. Osa kaadrist kadus ära.
- Avnet sensori dekodeerimise plokil ei ole mitte mingisugust dokumentatsiooni, et saaks uurida milles viga on.
- Testimise käigus selgus, et Avnet sensori dekodeerimise plokk vajab enda tööks LVDS taktsignaalist vähemalt neli korda kõrgemat taktsignaali. Kuid käesoleva

riistvara puhul ei ole see võimalik kui tahta sensori andmete edastuskiirus viia maksimumini. PicoZed 7020 võimaldab LVDS signaalidel maksimaalset kiirust 950 Mbps ning kaamera suudab saata kiirusel 720 Mbps. Neli korda kõrgem kiirus kaamera poolt võimaldatavast kiirusest oleks 2880 Mbps.

- „AXI Video Direct Memory Access" plokk võimaldab ainult 32 kaadrit muutmällu kirjutada. Vaja oleks aga lahendust, mis võimaldaks kirjutada muutmällu kaadreid kogu vaba muutmälu ulatuses.

Xilinx pakub omalt poolt plokki nimega „SelectIO Wizard" [21] . Kasutades seda plokki, on võimalik vastu võtta ja kokku pakkida LVDS signaalide andmeid, kasutades taktsignaalina samuti LVDS signaali. Alternatiivse lahendusena Avnet sensori dekoodrile sai otsustatud proovida „SelectIO Wizard" plokki. „SelectIO Wizard" ploki kasutamisega aga ei ole enam võimalik FPGA poolt sünkronisatsiooni kanalit analüüsida. Seega tuli kirjutada kõik kaamera andmed mikroprotsessori muutmällu, kaasa arvatud sünkronisatsiooni andmed, ning lasta sünkronisatsiooni andmeid analüüsida hoopis tarkvaral. „SelectIO Wizard" plokil sai valitud pakkimise meetod 5/6 ning väljundiks 32 bitine andmesiin. See tähendab, et ühte 32 bitisesse andmesõnasse paigutatakse kuuest taktsignaali taktist saadud viis LVDS kanalit, kokku 30 bitti. Viis LVDS kanalit selletõttu, et lisaks neljale andmekanalile on vaja ka sünkronisatsioonikanali andmeid. Kuna tegemist ei ole enam video andmetega ning ei ole teada kaadrite alguseid ega lõppe, siis sobib muutmällu kirjutamiseks tavaline DMA [22] . Tavaline DMA tuleb panna Scatter-Gather režiimi, mis võimaldab kasutada muutmälus asuvat kirjeldust andmete salvestamise asukoha ja suuruse kohta. Kuivõrd LVDS taktsagedus on erinev DMA ploki taktsagedusest, siis tuleb nende vahele disainida veel FIFO (*First In First* Out) plokk, erinevate taktsageduste võimalustega nii sisendis kui ka väljundis [23] . Antud lahendus töötab maksimaalsel kiirusel riistvara seisukohast, kuid sellevõrra peab tarkvara palju rohkem tööd tegema. Kuna aga ei ole vaja pidevat andmevoogu, vaid just lühiajalist kiirete kaadrite hulka, siis antud projektis sobib selline lahendus. Tarkvara kasutamisel võetakse vaba muutmälu ulatuses maksimaalsel kiirusel andmeid kaamera sensorist ning hiljem analüüsitakse neid andmeid ja leitakse erinevad kaadrid.

Eelised sellel lahendusel Avnet sensori dekodeerimise ploki ees on järgmised:

- Kaamera tüüp ei ole riistvara seisukohast enam oluline, kogu sünkronisatsiooni töö teeb ära tarkvara.
- Ei ole vaja kasutada ilma dokumentatsioonita süsteemi osasid.
- Sensori andmeid saab muutmällu kirjutada maksimaalse kiirusega.
- Tavalisel DMA plokil ei ole andmemahu piirangut muutmällu kirjutamisel.

6 Tarkvara

Kaadrite saamiseks sensorilt on vaja sensor sisse lülitada, sensor seadistada ning DMA seadistada. Selle jaoks sai kirjutatud kaks programmi, mis töötavad Linuxi kasutajarežiimis, kuid nõuavad siiski administraatori õigusi. Üks programmidest on mõeldud sensori seadistamise jaoks ja teine sünkronisatsiooni andmete analüüsimiseks ning kaadrite leidmiseks.

6.1 Sensori juhttarkvara

Juhttarkvara põhiülesanne on ONsemi PYTHON300 sensori tööparameetrite seadistamine. Lisaks saab sellega sensorit nii sisse kui ka välja lülitada ning konfigureerida DMA plokki FPGA sees. Tarkvara on disainitud selliselt, et kõiki toiminguid saab teha üksteisest sõltumatult, andes vajalikud käsud ja parameetrid programmile käsureale kaasa. Selline lähenemine jätab võimaluse edasiste katsetuste läbiviimisel jooksvalt muuta vajalikke väärtusi sensoris. Kõige mugavam on kasutada *shelli* skripte. Näiteks võib sensoril ära määrata ROI (*Regions Of Interest*) ala, ning edastada väiksemaid kaadreid suurematel kiirustel. Programm tuleb käivitada administraatori õigustega kuna oma töös peab programm vastavalt käsule saama Linuxis ligi paljudele ressurssidele millele tavakasutajal vaikimisi ei ole õigusi:

- Füüsilisele muutmälule, mida Linux ise ei kasuta.
- GPIO signaalidele.
- Virtuaalse muutmälu kaudu AXI küljes olevale DMA ploki registritele.

Programm kasutab GPIO signaale, et teha tarkvaraline SPI liides sensori juhtimiseks. Lisaks üksikutele käskudele on programmi lisatud veel mugavamaks kasutamiseks mõningad käsud, mis teevad kas mitu erinevat toimingut korraga või seadistavad sensori registrid varem valmis defineeritud väärtustega. Ülevaate käskudest annab Tabel 2. Näide skriptist asub Joonisel 9.

Tabel 2. Juhtprogrammi parameetrite kirjeldused.

Käsk	**Lisaparameetrid**	**Tähendus**	**Näide**
fpga	bitstream faili-nimi	Konfigureerib FPGA vastava bitstreamiga	python300 fpga main.bin
off	-	Sensori välja lülitamine	python300 off
on	-	Sensori sisse lülitamine	python300 on
read	aadress	Sensori registri lugemine	python300 read 0
write	aadress, väärtus	Sensori registri kirjutamine	python300 write 144 0x0009
list	aadress	Kuvab sensori registrite väärtused	python300 list 0
pll	-	Ootab kuni sensori PLL on lukustunud	python300 pll
start	-	Käivitab sensori vaike-väärtustega	python300 start
dmasg_start	-	Käivitab DMA Scatter-Gather režiimis	python300 dmasg_start
dmasg_stop	-	Seiskab DMA	python300 dmasg_stop

```
#!/bin/sh

python300 off
python300 fpga /lib/firmware/fpga.bin
python300 on
python300 start
python300 dmasg_start
python300 dmasg_stop
```

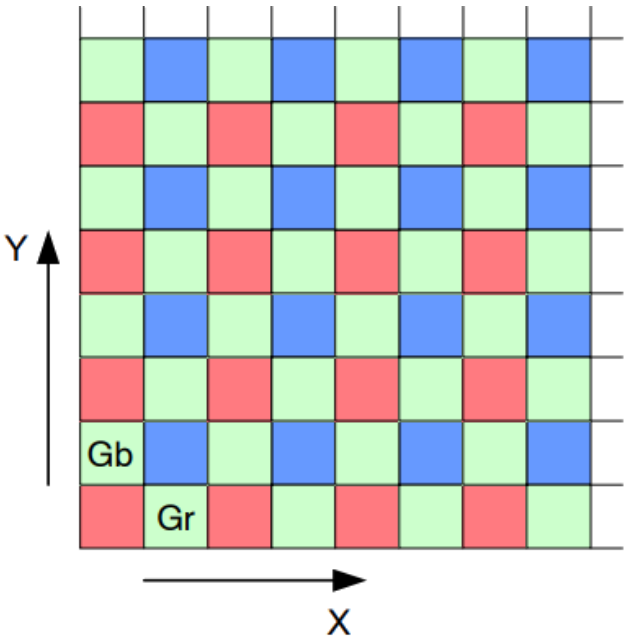
Joonis 9. Kaamera sensori seadistamise skript mis kasutab loodud tarkvara.

6.2 Pildiserver

Kuna vähemalt esialgu on sensorilt vaja kaadreid edasiseks seadme katsetamiseks, siis sai välja arendatud lahendus, kus kaamera pilti saab vaadata veebilehitsejaga. Programm pannakse süsteemis *daemon* režiimis taustale käima. Tarkvara kasutab oma töös *libmicrohttpd* ning *libpng* teeke [24] [25] . Lahendus tekitab HTTP (*Hypertext Transfer Protocol*) serveri, mis kuulab TCP (*Transmission Control Protocol*) pordil 80 päringuid veebilehitsejast [26] [27] . Kui veebilehitseja küsib faili nimega „image.png”, siis

hakatakse muutmälust otsima sünkronisatsiooni andmesõnu ja tuvastatakse esimese muutmälus oleva kaadri asukoht. Vastavalt edasistele sünkronisatsiooni andmesõnade juhistele leitakse kõik vajalikud andmeosad, et koostada terviklik kaader, ning seejärel kasutatakse *libpng* teeki, et genereerida muutmällu PNG (*Portable Network Graphics*) formaadis pilt, mis omakorda edastatakse veebilehitsejale.

Oluline on siinjuures see, et saadud pilt ei ole lõpliku värvilahendusega, vaid tegu on monokroomse pildiga, milles pikslid kajastavad sensori maatriksi iga elemendi analoogväärtust. Pikslid on PNG formaadis värvilisele pildile kodeeritud rohelise värvina. Ülejäänud värvide heleduste väärtused puuduvad, seega on veebilehitsejas näha erinevates rohelistes toonides monokroomne pilt. Värviline pilt on võimalik saavutatada, kui sensori maatriksi elementide analoogväärtusi ning värvifiltrit arvesse võttes teisendada individuaalseid piksleid vastavatesse värvilistesse toonidesse. Konkreetsel sensoril on kasutusel Bayeri filter [28] . Bayeri filter on mosaiik värvidest ruudukujulisel võrgul, kus rohelisi filtreid on punaste ja siniste filtritega võrreldes kaks korda rohkem. Bayeri filtri mosaiik on kujutatud Joonisel 10. Kuivõrd lõpptulemusena on võimalik saada värviline pilt, siis on ka PNG formaadiks valitud RGB toega formaat, et oleks võimalik tulevikus minimaalsete muudatustega ka värvilist pilti edastada. Antud töös edastatakse siiski monokroomset pilti, kuna sensori maatriksi muutmata algandmed on edasiste katsetuste jaoks olulisemad.

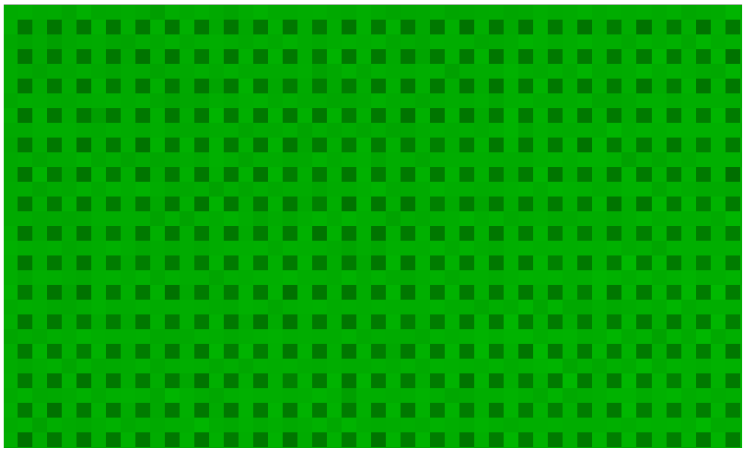


Joonis 10. Bayeri filtri mosaiik.

Joonisel 11 on PNG formaadis pilt, mis on koostatud töötlemata kaamera sensori andmetest.

Joonis 11. Töötlemata kaamera sensori pilt.

Joonisel 12 on kujutatud sama pilt suurendatuna. Pildil on näha heledatest ja tumedatest pikselitest moodustunud muster, mis kajastab Bayeri filtri mosaiiki.



Joonis 12. Suurendatud kaamera sensori pilt.

Kontrollimaks, kas antud pildi andmed on kaamera sensorist korrektselt saadud, sai tehtud skript programmis Matlab [29] . Joonisel 13 on näidatud skript, mis kasutab Matlabi funktsiooni *demosaic* [30] . Kõigepealt eraldati värvilisest pildist ainult roheline värv ning saadi edasiseks töötlemiseks monokroomne pilt. Joonisel 14 on kujutatud saadud pilt halltoonides. Saadud andmetest koostati Bayeri filtri abiga värviline pilt, mis on kujutatud Joonisel 15.

```
image = imread('image.png');
green = image(:,:,2);
imshow(green);
bayer = demosaic(green, 'bggr');
imshow(bayer);
imwrite(bayer, 'bayer.png');
```

Joonis 13. Programmis Matlab tehtud skript Bayeri filtriga arvutamiseks.

Joonis 14. Töötlemata kaamera sensori pilt halltoonides.

Joonis 15. Monokroomsest pildist Bayeri filtri abiga koostatud värviline pilt.

7 Kokkuvõte

Kiiplabori üheks süsteemi osaks on kaamera sensor, mis võimaldaks salvestada kaadreid muutmällu kiirusel 1000 kaadrit sekundis, et analüüsida proove. Antud lõputöö on osa kaamera sensori süsteemi loomisest.

Lõputöö raames testiti ning parandati olemas olevat riistvara. Uuriti ning katsetati erinevaid võimalusi, kuidas kaamera sensorit juhtida ning kuidas kaamera sensorilt andmeid töötlusmooduli muutmällu saada. Lõpptulemusena arendati välja VHDL disain FPGA jaoks, et siduda kaamera sensorit töötlusmooduliga. Koostati Linuxi distributsiooni fail FIT formaadis. Kirjutati tarkvara nii kaamera sensori juhtimiseks kui ka piltide serveerimiseks üle kohtvõrgu. Arendatud süsteemiga on võimalik minna järgnevate katsetuste juurde ning liikuda edasi täisfunktsionaalse kiiplabori suunas.

Kasutatud kirjandus

[1] BD Accuri™ C6 Flow Cytometer Instrument Manual. Loetud aadressil: https://bif.wisc.edu/wp-content/uploads/sites/389/2017/11/bd_accuri_c6_flow_cytometer_instrument_manual.pdf Kasutatud: 03.05.2022

[2] Attune™ NxT Acoustic Focusing Cytometer. Loetud aadressil: https://assets.thermofisher.com/TFS-Assets/LSG/manuals/100024235_AttuneNxT_HW_UG.pdf Kasutatud: 03.05.2022

[3] Lab on a Chip. Loetud aadressil: https://taltech.ee/en/labonchip Kasutatud: 03.05.2022

[4] Ersoy Erkan, Development of a Camera Prototype for a Flow Cytometry System, Loetud aadressil: https://digikogu.taltech.ee/et/Item/9c9664ac-d8f6-4250-b2f9-658f1729fc44 Kasutatud: 03.05.2022

[5] Xilinx, Vivado 2019.1 - Installation and Licensing. Loetud aadressil: https://www.xilinx.com/support/documentation-navigation/design-hubs/2019-1/dh0013-vivado-installation-and-licensing-hub.html Kasutatud: 12.05.2022

[6] GCC, the GNU Compiler Collection. Loetud aadressil: https://gcc.gnu.org/ Kasutatud: 12.05.2022

[7] Avnet, PicoZed. Loetud aadressil: https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/picozed/ Kasutatud: 03.05.2022

[8] Xilinx, Zynq-7000. Loetud aadress: https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html Kasutatud: 03.05.2022

[9] ONsemi, PYTHON300: CMOS Image Sensor, 0.3 MP (VGA), Global Shutter. Loetud aadressil: https://www.onsemi.com/products/sensors/image-sensors/python300 Kasutatud: 03.05.2022

[10] Molex, MXMag Gigabit Ethernet 8 Core Non-PoE Single-Port RJ45 Connector. Loetud aadressil: https://www.molex.com/molex/products/part-detail/modular_jacks_plug/0936263508 Kasutatud: 03.05.2022

[11] The Linux Kernel Archives. Loetud aadressil: https://www.kernel.org/ Kasutatud: 03.05.2022

[12] Das U-Boot - the Universal Boot Loader. Loetud aadressil: https://www.denx.de/wiki/U-Boot Kasutatud: 03.05.2022

[13] FPGA Manager. Loetud aadressil: https://www.kernel.org/doc/html/v4.19/driver-api/fpga/fpga-mgr.html Kasutatud: 03.05.2022

[14] Device Tree Reference. Loetud aadressil: https://elinux.org/Device_Tree_Reference Kasutatud: 12.05.2022

[15] Busybox. Loetud aadressil: https://busybox.net/ Kasutatud: 15.05.2022

[16] Xilinx, AXI Quad SPI. Loetud aadressil: https://www.xilinx.com/products/intellectual-property/axi_quadspi.html Kasutatud: 03.05.2022

[17] Xilinx, Clocking Wizard. Loetud aadressil: https://www.xilinx.com/products/intellectual-property/clocking_wizard.html Kasutatud: 15.05.2022

[18] Avnet HDL Reference Designs. Loetud aadressil: https://github.com/Avnet/hdl Kasutatud: 03.05.2022

[19] Xilinx, Video In to AXI4-Stream. Loetud aadressil: https://www.xilinx.com/products/intellectual-property/video_in_to_axi4_stream.html Kasutatud: 03.05.2022

[20] Xilinx, AXI Video DMA, Loetud aadressil: https://www.xilinx.com/products/intellectual-property/axi_video_dma.html Kasutatud: 03.05.2022

[21] Xilinx, SelectIO Interface Wizard. Loetud aadressil: https://www.xilinx.com/products/intellectual-property/selectio_interface_wizard.html Kasutatud: 03.05.2022

[22] Xilinx, AXI DMA Controller, Loetud aadressil: https://www.xilinx.com/products/intellectual-property/axi_dma.html Kasutatud: 03.05.2022

[23] Xilinx, AXI Streaming FIFO. Loetud aadressil: https://www.xilinx.com/products/intellectual-property/axi_fifo.html Kasutatud: 03.05.2022

[24] GNU Libmicrohttpd. Loetud aadressil: https://www.gnu.org/software/libmicrohttpd/ Kasutatud: 03.05.2022

[25] Libpng. Loetud aadressil: http://www.libpng.org/pub/png/libpng.html Kasutatud: 03.05.2022

[26] Lauri Võsandi, Video capture with VDMA. Loetud aadressil: https://lauri.xn--vsandi-pxa.com/hdl/zynq/xilinx-video-capture.html Kasutatud: 09.05.2022

[27] Lauri Võsandi, AXI Direct Memory Access Loetud aadressil: https://lauri.xn--vsandi-pxa.com/hdl/zynq/xilinx-dma.html Kasutatud: 09.05.2022

[28] Introduction to Bayer Filters. Loetud aadressil: https://www.arrow.com/en/research-and-events/articles/introduction-to-bayer-filters Kasutatud: 06.05.2022

[29] Matlab. Loetud aadressil: https://www.mathworks.com/products/matlab.html Kasutatud: 12.05.2022

[30] Convert Bayer pattern encoded image to truecolor image. Loetud aadressil: https://www.mathworks.com/help/images/ref/demosaic.html Kasutatud: 12.05.2022

Lisa 1– Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Veiko Rütter

1 Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Kiiplabori kaameramooduli edasiarendus", mille juhendajad on Kaiser Pärnamets ja Ants Koel

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2 Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3 Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

16.05.2022

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2– Vivados arendatud disain töötlusmooduli FPGA jaoks