

UNIVERZA V LJUBLJANI  
FAKULTETA ZA ELEKTROTEHNIKO

MAGISTRSKI ŠTUDIJ

MAGISTRSKO DELO

RAZVOJNI SISTEM ZA OBDELAVO VIDEO SIGNALA S  
PROGRAMIRLJIVIMI VEZJI

Franci Kopač, univ. dipl. inž. el.

Mentor: doc. dr. Andrej Trost, dipl. inž.

## Zahvala

---

Zahvaljujem se vsem, ki so pomagali, da je to magistrsko delo sploh nastalo:

- doc. dr. Andreju Trostu, za pomoč, nasvete in mentorstvo skozi celoten magistrski študij,
- prof. dr. Andreju Žemvi, za pomoč pri vseh mogočih organizacijskih zadevah,
- Jožetu Dediču, univ. dipl. ing. el., za številne nasvete, literaturo in tisoč ostalih stvari,
- mag. Matjažu Fincu, za pomoč pri oblikovanju člankov in načrtovanju modulov VHDL,
- Metki Golčman, za skrbno lektoriranje tega magistrskega dela, ogromno potrpljenja med njegovim nastajanjem in za mnogo lepih trenutkov.

Moji mami

# Kazalo

---

<b>1</b>	<b><u>SEZNAM UPORABLJENIH SIMBOLOV</u></b>	<b>1</b>
<b>2</b>	<b><u>POVZETEK</u></b>	<b>2</b>
<b>3</b>	<b><u>ABSTRACT</u></b>	<b>3</b>
<b>4</b>	<b><u>UVOD</u></b>	<b>4</b>
<b>4.1</b>	<b>PROBLEMATIKA</b>	<b>4</b>
<b>4.2</b>	<b>UPORABLJEN PRISTOP</b>	<b>6</b>
<b>5</b>	<b><u>OPIS STROJNE OPREME SISTEMA</u></b>	<b>8</b>
<b>5.1</b>	<b>UNIVERZALNI SISTEM ZA DIGITALNI ZAJEM SLIKE IN ZVOKA</b>	<b>9</b>
5.1.1	OSNOVNE ZAHTEVE ZA DEKODIRANJE DIGITALNEGA TOKA PO STANDARDU BT.656	10
5.1.2	PRETVORNIK VIDEO SIGNALA	10
5.1.3	PRETVORNIK ZVOČNEGA SIGNALA	13
5.1.4	KRMILNIK ZA NASTAVLJANJE PARAMETROV SISTEMA	15
5.1.5	NAČIN NASTAVLJANJA PARAMETROV PREK POVEZAVE RS-232	16
5.1.6	NAPAJALNI SKLOP	18
<b>5.2</b>	<b>SKLOP ZA OBDELAVO DIGITALNEGA VIDEO TOKA</b>	<b>19</b>
5.2.1	UČNI SISTEM SPARTAN3	19
5.2.2	POMNILNIŠKI RAZŠIRITVENI MODUL	20
5.2.3	POVEZOVANJE POMNILNIŠKEGA MODULA IN SISTEMA SPARTAN 3	22
<b>5.3</b>	<b>D/A PRETVORNIK ZA PRIKAZ SLIKE NA RAČUNALNIŠKEM MONITORJU</b>	<b>23</b>
<b>6</b>	<b><u>OPIS PROGRAMSKE OPREME ZA NASTAVLJANJE PARAMETROV A/D PRETVORBE, RAZVOJ MODULOV VHDL IN PROGRAMIRANJE VEZJA FPGA</u></b>	<b>24</b>
<b>6.1</b>	<b>NASTAVITVE AV ADC PLOŠČE</b>	<b>24</b>
6.1.1	OSNOVNE NASTAVITVE	24

6.1.2	DODATNE NASTAVITVE	26
6.1.3	DODATNE NASTAVITVE 2	27
6.1.4	DATOTEKA Z NASTAVITVAMI	27
6.1.5	EMULATOR SISTEMA	28
<b>6.2</b>	<b>PROGRAM ZA PROGRAMIRANJE IN NASTAVLJANJE VEZJA FPGA</b>	<b>28</b>
<b>6.3</b>	<b>RAZVOJNO OKOLJE XILINX ISE</b>	<b>29</b>
<b>7</b>	<b>OPIS ZGRADBE IN DELOVANJA MODULOV VHDL</b>	<b>32</b>
<b>7.1</b>	<b>SEZNAM MODULOV VHDL</b>	<b>32</b>
<b>7.2</b>	<b>MODUL ZA KRMILJENJE D/A PRETVORNIKA ZA ČRNO-BEL PRIKAZ NA RAČUNALNIŠKEM MONITORJU</b>	<b>33</b>
7.2.1	KRMILJENJE RAČUNALNIŠKEGA MONITORJA	33
7.2.2	ZGRADBA MODULA ZA PRIKAZ ČRNO-BELE SLIKE	36
<b>7.3</b>	<b>MODUL ZA 6-BITNI DOSTOP DO POMNILNIKA</b>	<b>37</b>
<b>7.4</b>	<b>MODUL ZA DEKODIRANJE ČRNO-BELEGA VIDEO SIGNALA</b>	<b>39</b>
7.4.1	STANDARD BT.656	39
7.4.2	ZGRADBA MODULA ZA DEKODIRANJE SIGNALA	42
<b>7.5</b>	<b>MODUL VHDL ZA BARVNI PRIKAZ SLIKE NA RAČUNALNIŠKEM MONITORJU</b>	<b>43</b>
<b>7.6</b>	<b>MODUL VHDL ZA UPRAVLJANJE POMNILNIŠKEGA MODULA Z RAZŠIRJENIM PODATKOVNIM VODILOM ZA DELO Z BARVNIM VIDEO SIGNALOM</b>	<b>44</b>
<b>7.7</b>	<b>MODUL VHDL ZA BARVNO DEKODIRANJE VIDEO SIGNALA</b>	<b>45</b>
7.7.1	DEKODIRANJE BARV IZ DIGITALNEGA TOKA PO STANDARU BT.656	45
7.7.2	PRETVORBA BARVNEGA PROSTORA YCrCb V BARVNI PROSTOR RGB	48
7.7.3	ZGRADBA MODULA VHDL ZA BARVNO DEKODIRANJE VIDEO SIGNALA	49
<b>7.8</b>	<b>MODUL VHDL ZA DEMONSTRACIJO OBDELAVE SLIKE</b>	<b>50</b>
<b>8</b>	<b>ZAKLJUČEK</b>	<b>52</b>
<b>9</b>	<b>LITERATURA</b>	<b>53</b>
<b>10</b>	<b>IZJAVA</b>	<b>54</b>
<b>11</b>	<b>PRILOGE</b>	<b>55</b>

## Seznam slik

---

Slika 1: Osnovna zgradba sistema _____	8
Slika 2: Zgradba sistema za zajem slike in zvoka _____	9
Slika 3: Blok shema integriranega vezja ADV7185 [1] _____	11
Slika 4: Priporočena vezava vezja ADV7185 [1] _____	12
Slika 5: Blok shema integriranega vezja AD1870 [3] _____	13
Slika 6: Potek izhodnih signalov vezja AD1870 [3] _____	14
Slika 7: Priporočena vezava vezja AD1870 [3] _____	14
Slika 8: Nastavljanje parametrov vezja ADV7185 s PC-jem prek krmilnika _____	15
Slika 9: Sistem za zajem slike in zvoka _____	18
Slika 10: Sistem Spartan 3 _____	20
Slika 11: Bralni cikel pomnilniških modulov AS7C34096 [14] _____	21
Slika 12: Zapisovalni cikel pomnilniških modulov AS7C34096 [14] _____	21
Slika 13: Razširitveni modul s pomnilnikom SRAM _____	22
Slika 14: Pomnilniški modul, nataknen na sistem Spartan 3 _____	22
Slika 15: D/A pretvornik za krmiljenje računalniškega monitorja _____	23
Slika 16: Glavno okno programa za nastavljanje sistema za zajem slike in zvoka _____	25
Slika 17: Vgrajeni opisi delovanja parametrov _____	25
Slika 18: Menijska vrstica, kjer lahko odpremo ostala okna _____	26
Slika 19: Okno Dodatne nastavitve _____	26
Slika 20: Okno Dodatne nastavitve 2 _____	27
Slika 21: Okno za datoteko z nastavitvami _____	27
Slika 22: Program za programiranje vezja FPGA _____	28
Slika 23: Razvojno okolje Xilinx ISE _____	29
Slika 24: Shematsko povezovanje modulov VHDL v okolju Xilinx ISE _____	30
Slika 25: Določitev vhodno/izhodnih povezav vezja _____	30
Slika 26: Potek načrtovanja sistema _____	31
Slika 27: Sinhronizacijski časi za vrstice _____	34
Slika 28: Potek navpične sinhronizacije _____	35
Slika 29: Zgradba modula za črno-beli prikaz slike _____	36
Slika 30: Zgradba modula za upravljanje pomnilnika _____	38
Slika 31: Potek stanj avtomata za dostop do pomnilnika _____	38
Slika 32: Zgradba slikovne vrstice v digitalnem toku BT.656 _____	41

Slika 33: Zgradba modula za dekodiranje video signala _____	43
Slika 34: Modul za prikaz barvne slike _____	43
Slika 35: Modul za upravljanje pomnilnika z razširjenimi vhodno-izhodnimi vodili _____	45
Slika 36: Rekonstrukcija barvne informacije s kopiranjem _____	46
Slika 37: Interpolacija barvne informacije iz dveh sosednjih slikovnih pik _____	46
Slika 38: Interpolacija barvne informacije iz štirih sosednjih slikovnih pik _____	47
Slika 39: Optimirana struktura vezja za pretvorbo iz barvnega prostora YCrCb v RGB [12] _____	48
Slika 40: Zgradba modula VHDL za barvno dekodiranje video signala _____	49
Slika 41: Zgradba demonstracijskega modula za obdelavo video signala _____	51
Slika 42: Zgradba sistema za prenos črno-belega video signala iz vhoda na izhod _____	51
Slika 43: Zgradba sistema z dodanim modulom za obdelavo video signala _____	51

## Seznam tabel

---

Tabela 1: Vrednosti parametra za ukaz za spreminjanje načina delovanja sistema _____	17
Tabela 2: Najpomembnejše karakteristike uporabljenih pomnilniških modulov [14] _____	21
Tabela 3: Sinhronizacijske vrednosti za vrstice _____	35
Tabela 4: Vsebina identifikacijskega podatka v sinhronizacijskem bloku _____	40
Tabela 5: Vrednosti podatkovnih bitov identifikacijskega podatka _____	40
Tabela 6: Razpored vrstic v slikovnem polju digitalnega toka po BT.656 _____	41

## Seznam enačb

---

Enačba 1: Povezava med vodoravno ločljivostjo in frekvenco pošiljanja podatkov .....	34
Enačba 2: Določanje vrednosti prvega kontrolnega bita [20] .....	41
Enačba 3: Določanje vrednosti drugega kontrolnega bita [20] .....	41
Enačba 4: Določanje vrednosti tretjega kontrolnega bita [20] .....	41
Enačba 5: Določanje vrednosti četrtega kontrolnega bita [20].....	41
Enačba 6: Enačba barvne interpolacije iz dveh sosednjih slikovnih pik.....	47
Enačba 7: Enačba barvne interpolacije iz štirih sosednjih pik .....	47
Enačba 8: Primer paralelnega filtra FIR za barvno rekonstrukcijo [13].....	47
Enačba 9: Pretvorba barvnega prostora YCrCb v RGB za 8-bitni digitalni tok [12].....	48
Enačba 10: Pretvorba barvnega prostora YCrCb v RGB za 10-bitni digitalni tok [12].....	48

# 1 Seznam uporabljenih simbolov

---

PC	–	Personal Computer
FPGA	–	Field Programable Gate Array
VHDL	–	Very High Speed Integrated Circuit (VHSIC) Hardware Description Language
PAL	–	Phase Alternating Line
NTSC	–	National Television Systems Committee
DIN	–	Deutsche Industrie Norm
RCA	–	Recording Corporation of America
BNC	–	Bayonet Nut Connector
DVD	–	Digital Versatile Disc
SCART	–	Syndicat des Constructeurs d'appareils Radio et de Télévision
FIFO	–	First In-First Out
USB	–	Universal Serial Bus
LED	–	Light Emitting Diode
EEPROM	–	Electrically Erasable Programmable Read Only Memory
CPLD	–	Complex Programmable Logic Device
SRAM	–	Static Random Access Memory
RGB	–	Red, Green, Blue
VGA	–	Video Graphics Array
DCM	–	Digital Clock Manager
FIR	–	Finite Impulse Response
RAM	–	Random Access Memory
ITU-R	–	International Telecommunication Union – Radiocommunication Sector
CCIR	–	Consultative Committee for International Radio communications
CMOS	–	Complementary Metal Oxide Semiconductor
RISC	–	Reduced Instruction Set Computing
A/D	–	Analogno/Digitalno
D/A	–	Digitalno/Analogno
HDTV	–	High Definition TeleVision
ASIC	–	Application Specific Integrated Circuit
NRE	–	Non-Recurring Engineering costs
I2C	–	Inter-Integrated Circuit
ASCII	–	American Standard Code for Information Interchange



## 2 Povzetek

---

Opisan je sistem za razvoj in preizkušanje tehnik za obdelavo video signala. Sestavlja ga več komponent: sistem z rekonfigurabilnim vezjem FPGA in programom za programiranje vezja FPGA, pomnilniški modul za sistem z vezjem FPGA, sistem za zajem zvočnega in video signala z vmesnikom za krmiljenje prek povezave RS-232 ter programom za nastavljanje parametrov z osebnim računalnikom v okolju Windows in izhodni D/A pretvornik za prikaz obdelanega video signala na standardnem računalniškem monitorju. Digitalni video signal iz video pretvornika je v standardnem zapisu BT.656, kar omogoča njegovo uporabo tudi z drugimi sistemi za obdelavo video signala. Poleg strojne in programske opreme za zajem video signala ter nastavitve pretvorbe so opisani tudi moduli VHDL za dekodiranje digitalnega toka po standardu BT.656, dostop do pomnilnika in prikaz video signala na računalniškem monitorju ter demonstracijski modul za prikaz preproste tehnike obdelave video signala. Opisan je tudi postopek njihovega načrtovanja. Podrobno so opisane tehnike za dekodiranje video signala v obliki BT.656, interpolacijo barvne informacije in pretvorbo barvnega prostora YCrCb, ki ga uporablja BT.656, v barvni prostor RGB, ki se uporablja za prikaz slike. Poleg tega je opisan še način prikaza slike na standardnem računalniškem monitorju in pa dostop do pomnilnika.

**Ključne besede:** FPGA, VHDL, obdelava video signala, BT.656, rekonfigurabilni sistemi

## 3 Abstract

---

We describe a system for testing and development of video signal processing techniques. It consists of several components: a reconfigurable FPGA-based system with programming software, a SRAM expansion card for the reconfigurable system, a system for analogue-to-digital conversion of video and audio with RS-232 configuration interface and Windows-based configuration software and, finally, an output D/A converter used to display the processed video signal on a standard computer monitor. The audio/video A/D converter outputs the video in a standard BT.656 digital stream, which makes it reusable for other similar video processing systems. The VHDL modules necessary to decode the BT.656 video stream, access the memory expansion cards and display the processed video are also described, along with a simple video processing demonstration module and the design process itself. BT.656 digital stream decoding techniques are discussed in detail, along with colour information interpolation, YCrCb to RGB colour space conversion, monitor driving and memory access techniques.

**Keywords:** FPGA, VHDL, video processing, BT.656, reconfigurable systems

## 4 Uvod

---

Digitalna obdelava video signalov je izredno obsežno raziskovalno območje z aplikacijami, ki jih srečujemo skoraj čisto povsod v našem vsakdanjem življenju (medicina, mediji, prometna tehnika, varovanje, robotika itd.). Zanja je značilno, da postavlja pred inženirja zelo hude tehnične zahteve. Te izhajajo iz ogromne količine obdelanih podatkov in, v večini primerov, zahteve, da obdelava poteka v realnem času.

### 4.1 Problematika

Da dobimo občutek o količini podatkov, s katerimi imamo opravka pri digitalni obdelavi video signala v realnem času, si oglejmo digitalni tok po standardu BT.656 [18]. Osnovni digitalni tok po standardu ima bitno širino 10 bitov, priteka pa s hitrostjo 27 MHz. To pomeni, da vsako sekundo prejmemo 270 Mb podatkov. Pri tem ne smemo pozabiti, da je potrebno te podatke še dekodirati, saj jih sestavljata dve polsliki in štiri zatemnitvene faze, sinhronizacijski bloki itd. Po dekodiranju dobimo barvno (24-bitno) sliko ločljivosti PAL (720 x 576 slikovnih pik), ki se osvežuje s hitrostjo 25 Hz. To pomeni 248,832 Mb slikovnih podatkov na sekundo. Če želimo obdelavo v realnem času (na vsako vhodno sliko oddamo izhodno sliko), moramo skupaj po vodilih procesorja vsako sekundo poslati najmanj pol gigabita podatkov. Če upoštevamo še dostop do podatkovnega in programskega pomnilnika, se ta številka hitro povzpne na več gigabitov na sekundo. Pri tem moramo upoštevati, da je bil televizijski sistem PAL zasnovan leta 1967 in da se trenutno standardizirajo televizijski sistemi, kot je HDTV, ki deluje z ločljivostjo 1920 x 1080 slikovnih pik (1,244 Gb/s). Verjetno najzahtevnejši trenutni komercialni sistem za obdelavo slike uporablja filmska industrija in zahteva ločljivost kar 2048 x 1536 (1,887 Gb/s) slikovnih pik.

Večnamenski procesorji tem količinam podatkov praviloma niso več kos. Ker njihova osnovna zgradba (von Neumannova arhitektura) zahteva, da vhodne podatke obdelamo zaporedoma, mora procesor zaradi izredne količine vhodnih podatkov delovati z izredno visoko frekvenco jedra (sodobni večnamenski procesorji imajo frekvenco jedra že dolgo višjo od 1 GHz, Intel® Pentium® Extreme Edition [17] deluje na primer s 3,2 GHz). To prinese s seboj izredno veliko porabo energije (omenjen procesor je načrtovan za termično moč 130 W in napajalni tok 135 A pri napetosti 1,3 V). Vsakršna prenosna uporaba takšnega procesorja je verjetno izključena. Druga težava takšnih sistemov je v tem, da njihov sistemski pomnilnik ne dohaja njihovega jedra, zato je njihovo sistemsko vodilo počasnejše od jedra (za omenjen

procesor deluje jedro z 3,2 GHz, sistemsko vodilo pa z 800 MHz, kar je kar štirikrat počasneje). To težavo načrtovalci procesorjev rešujejo s hitrim medpomnilnikom na procesorju, vseeno pa zaradi tega nikoli ne moremo do konca izkoristiti visoke hitrosti jedra. To sta dve najresnejši omejitvi, obstajajo pa še številne druge. Doslej velja, da pristop s standardnimi procesorji ne dosega zmogljivosti za obdelovanje video signalov v realnem času brez izgube ločljivosti, če pa jo dosega, porabi preveč moči. Ni ga na primer mogoče uporabiti v ročnih prenosnih napravah. V literaturi pravijo tej omejitvi von Neumannovo ozko grlo (von Neumann bottleneck) [1], izognemo pa se ji lahko izključno z drugačnim pristopom do obdelave podatkov.

Eden od možnih pristopov je paralelna obdelava podatkov z rekonfigurabilnimi sistemi. Tukaj postopke za obdelavo prejetih podatkov razdelimo na večje število manj kompleksnih vzporedno delujočih izvršilnih enot, ki lahko posledično delujejo z manjšo hitrostjo. Če je moral prej procesor za vsak prejet podatek izvršiti na primer 10 ukazov, tukaj pripravimo 10 vzporednih izvršilnih enot, ki isto delo opravijo v 1 ciklu. Ker so izvršilne enote zasnovane namensko, jih lahko običajno izvedemo tudi bolj učinkovito (optimiziramo), kar pomeni, da dosežemo še večjo hitrost delovanja in manjšo porabo.

Takšne sisteme so do pred kratkim izdelovali večinoma v obliki vezij ASIC, kjer je bila zgradba sistema določena z optičnimi maskami, uporabljenimi pri izdelavi integriranega vezja, in zato nespremenljiva. Takšna vezja se še vedno uporabljajo, če potrebujemo izredno zmogljivost za zelo veliko serijo izdelkov, hkrati pa ne potrebujemo prilagajanja lastnosti vezja. Slabosti teh vezij so izredno veliki nepovratni načrtovalski stroški (NRE), dolg čas načrtovanja in nespremenljivost.

Če ne potrebujemo najvišjih izvedljivih zmogljivosti ali če je serija izdelkov manjša, je veliko primernejša uporaba vezij FPGA. Ta vezja sicer v nekaterih pogledih ne dosegajo lastnosti vezij ASIC (manjša gostota logike, večje zakasnitve signalov, večja poraba), imajo pa bistveno prednost – možnost rekonfiguracije, saj omogočajo skoraj poljubno povezovanje splošnih digitalnih gradnikov v namenske digitalne sisteme. Sistem z vezjem FPGA torej združuje na videz nezdržljivi zahtevi – je prilagodljiv kot procesor, hkrati pa učinkovit skoraj tako kot namensko integrirano vezje ASIC. Zaradi tega je idealen za načrtovanje sistemov za obdelavo video signala, saj lahko na isti strojni opremi preizkusimo različne pristope in arhitekture, na koncu pa izberemo najustreznejšo in jo prilagodimo dani aplikaciji. Še več, z istim sistemom lahko ustrezemo več aplikacijam, saj je programiranje vezja FPGA hitro – nekaj sekund za celoten sistem in delček sekunde za podsisteme.

Uporaba vezij FPGA pa s seboj prinese tudi precej sprememb. Izvedba sistema s standardnim procesorjem pomeni, da je strojna oprema že določena, zato načrtujemo samo programsko opremo. Ko preidemo na vezja FPGA, se ponovno vrnemo na načrtovanje strojne opreme. To je od nekdaj zelo kompleksen proces, zato je bil razvit nov pristop z opisnim jezikom (VHDL, Verilog itd.). Načrtovanje zdaj poteka tako, da opišemo delovanje sistema (temu pravimo behavioralni opis) in pa osnovno zgradbo sistema (registre, vodila itd.), samo zgradbo logičnih vrat med posameznimi gradniki pa določi razvojno okolje iz opisa delovanja. Če smo prej med načrtovanjem napisali program v jeziku C++, zdaj načrtujemo zgradbo strojne opreme sistema v jeziku VHDL.

Načrtovanje sistemov z vezji FPGA pa ni vedno omejeno na načrtovanje strojnih modulov. Sodobnejša, večja vezja FPGA so zdaj dovolj obsežna, da so v njih vgrajeni fiksni procesorji s standardno arhitekturo (npr. PowerPC v vezjih Spartan 2 Pro proizvajalca Xilinx) ali pa uporabimo tako imenovane mehkojedrne (softcore) procesorje, ki so izvedeni s programirljivo logiko vezja FPGA (na primer sistem Altera Nios). Izkaže se namreč, da so procesorji zaradi svoje prilagodljivosti za nekatere naloge primernejši od neposredne strojne izvedbe, saj tako za časovno nekritične dele aplikacije ni potrebno porabljati sistemskih vrat, ki predstavljajo glavno merilo za stroške končnega sistema. Za takšne naloge (na primer uporabniški vmesnik, določanje parametrov itd.) še vedno napišemo programe, časovno kritične naloge pa izvedemo strojno. Takšnemu pristopu do načrtovanja pravimo sočasno načrtovanje programske in strojne opreme (hardware-software codesign) in je trenutno izredno aktivno področje raziskav.

## **4.2 Uporabljen pristop**

To magistrsko delo opisuje razvojni sistem za načrtovanje in preizkušanje aplikacij za digitalno obdelavo video signala. Namenjen je predvsem učenju in spoznavanju tehnik obdelave video signala, zato je njegova zgradba v največji meri poenostavljena in standardizirana. Sestavljajo ga trije glavni sklopi: sklop za pretvorbo najpogosteje uporabljenih analognih video zapisov v digitalni tok po standardu BT656, sklop za obdelavo digitalnega video signala in sklop za prikaz obdelanega video signala na standardnem računalniškem monitorju. Medtem ko je vhodni modul zasnovan tako, da je čim bolj univerzalen in lahko sprejme čim več zapisov, ki jih uporablja sodobna potrošniška elektronika, sta modula za obdelavo in prikaz digitalnega video toka po zmogljivostih omejena, da ceno celotnega sistema obdržimo na sprejemljivi ravni. Ker za učenje ni potrebno delovanje pri polni ločljivosti in hitrosti osveževanja, ampak nas bolj zanimajo načela

delovanja tehnik za obdelavo signala in njihova izvedba, to ne predstavlja večje omejitve, hkrati pa osebi, ki ta sistem uporablja, na zelo realen način pokaže neizprosne načrtovalske omejitve, ki jo bodo čakale pri načrtovanju podobnih sistemov.

Pomembno je poudariti, da smo sistem sestavili iz univerzalno uporabnih komponent, od katerih lahko prav vsako uporabimo tudi kot sestavni del drugih aplikacij. To je zelo pomembno, saj tako znižamo stroške sistema, hkrati pa olajšamo delo tistim, ki bi radi sistem prilagodili svojim zahtevam. Pri obdelavi video signalov pač velja, da je sistem, ki bo primeren za vse vrste aplikacij, predrag, zato ga je potrebno vedno prilagoditi specifikacijam.

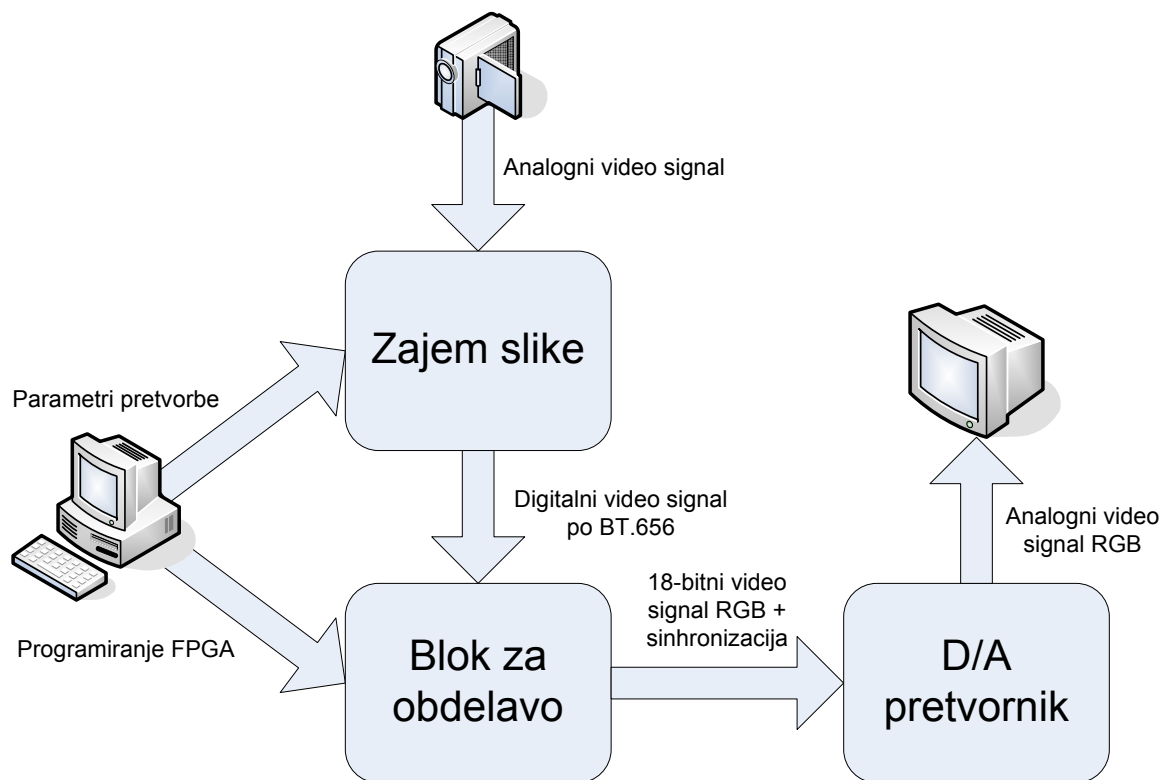
Srce modula za obdelavo digitalnega toka predstavlja rekonfigurabilno vezje FPGA, ki je ključno, da zagotovimo čimbolj široko uporabnost sistema. Z njim postane zgradba sistema prilagodljiva, tako da lahko vedno uporabimo arhitekturo, ki je najprimernejša za dano nalogo. Prav tako s tem prestopimo v drug svet načrtovanja. Klasično načrtovanje, ki obsega risanje shem in načrtovanje tiskanih vezij, zamenjamo z opisnim načrtovanjem v jeziku VHDL. Na isti plošči lahko tako enkrat poganjamo preprost procesor, drugič pa več vzporedno delujočih modulov za obdelavo slike. Takšna prilagodljivost ponovno zmanjša stroške, saj smo lahko kot modul za obdelavo slike uporabili že izdelan modul, ki se bo uporabljal pri učnem procesu, kar je prihranilo načrtovalsko delo in stroške.

Za ta projekt je bilo načrtovane in izdelane veliko strojne in programske opreme (sistem za zajem slike in zvoka, program za upravljanje tega sistema ter pomnilniški modul). Pomemben del tega projekta pa so tudi moduli VHDL, ki šele po programiranju v vezje FPGA dobijo svojo funkcijo. Ker je bil naš cilj izdelati sistem, ki bo primeren za učenje, so vsi ti moduli opremljeni z dokumentiranimi vmesniki, kar omogoča, da z njimi povežemo nove module in tako spreminjamo načine obdelave signala. Učenje bo tako lažje in manj zamudno, saj ne bo potrebno spoznavati celotnega ustroja sistema, ampak bo ta zreduciran na nekaj natančno opisanih modulov s preprosto razumljivimi vmesniki.

Sistem je bil načrtovan z namenom, da bi bil čimbolj poceni, da bi bile njegove komponente čimbolj univerzalno uporabne in da bi jih bilo preprosto uporabljati. Pri njegovem načrtovanju nismo želeli doseči vrhunskih zmogljivosti, ampak predvsem preprosto in hitro načrtovanje modulov VHDL za obdelavo video signala. Naštete lastnosti so s seboj prinesle strojne omejitve, ki pomenijo, da je omejena izhodna ločljivost tako po številu slikovnih pik kot tudi po barvni globini. Dosegljiva hitrost osveževanja je enaka vhodni, samo vezje FPGA pa je dovolj veliko tudi za precej zahtevne funkcije obdelave, tako da drugih bistvenih načrtovalskih omejitev ni.

## 5 Opis strojne opreme sistema

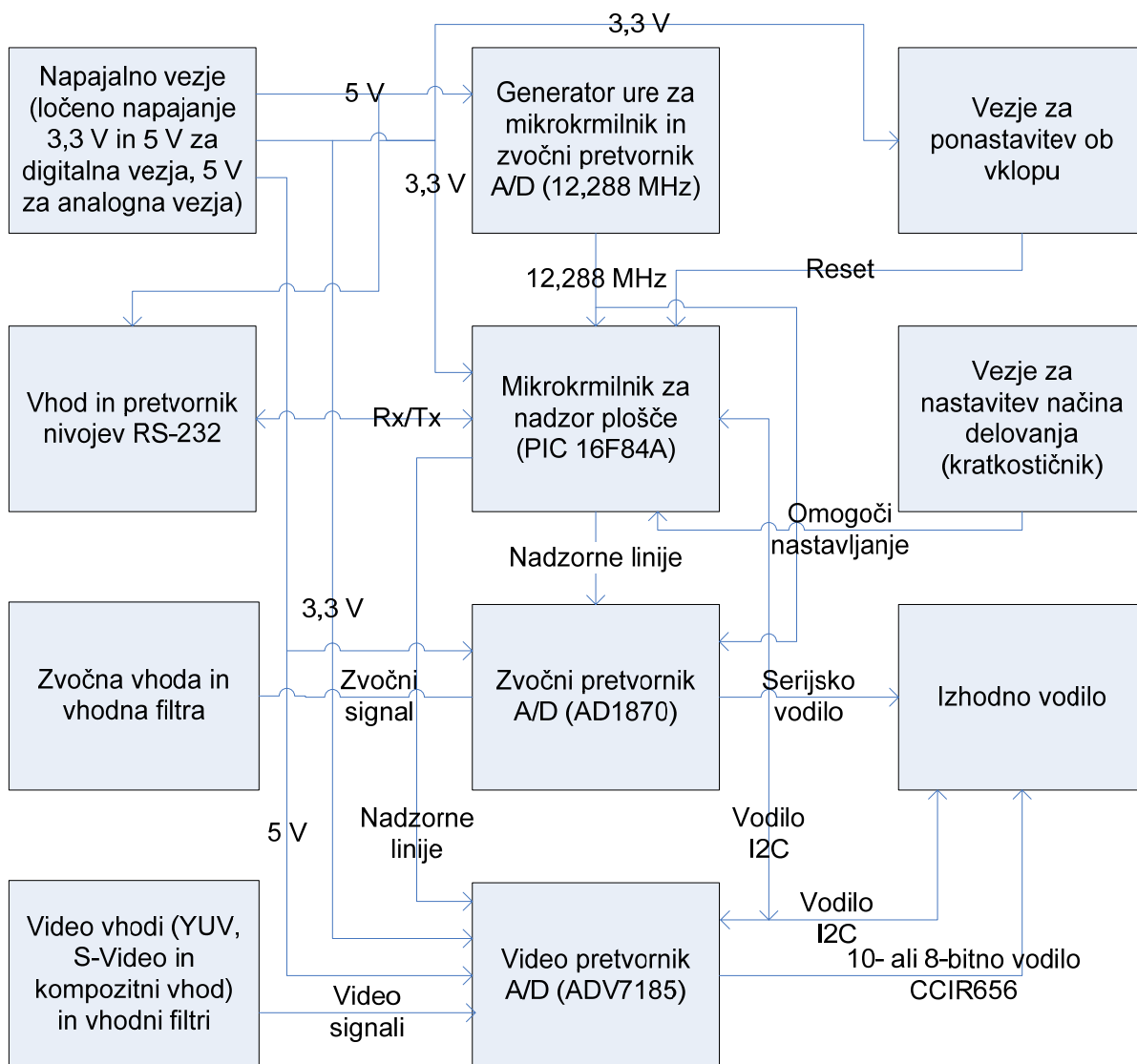
Sistem sestavljajo trije fizično ločeni glavni deli: vhodni A/D pretvornik, modul za obdelavo digitalnega toka in izhodni D/A pretvornik. Povezuje jih preprost mostič za priključitev, ki digitalne signale usmeri na ustrezno mesto na posameznih ploščah. Vsi trije moduli so opremljeni s standardnimi konektorji s po 40 priključki, za povezovanje med ploščami pa uporabljamo ploščate 40-žilne kable s standardnimi vtikači.



Slika 1: Osnovna zgradba sistema

## 5.1 Univerzalni sistem za digitalni zajem slike in zvoka

Pri načrtovanju vhodnega A/D pretvornika je bil naš cilj izdelati čim bolj vsestransko uporaben modul, ki bo zmožen prepoznavati čim več široko uporabljenih video zapisov in jih pretvoriti v digitalni tok po priporočilu BT.656 združenja ITU-R [18], ki predstavlja najpogosteje uporabljen digitalni zapis za video signal. Želeli smo tudi možnost nastavljanja parametrov pretvorbe s standardnim PC-jem in pa vgrajen A/D pretvornik za zvok.



Slika 2: Zgradba sistema za zajem slike in zvoka

Načrt povezave vseh sklopov sistema z vsemi priključki najdete v prilogi 1.



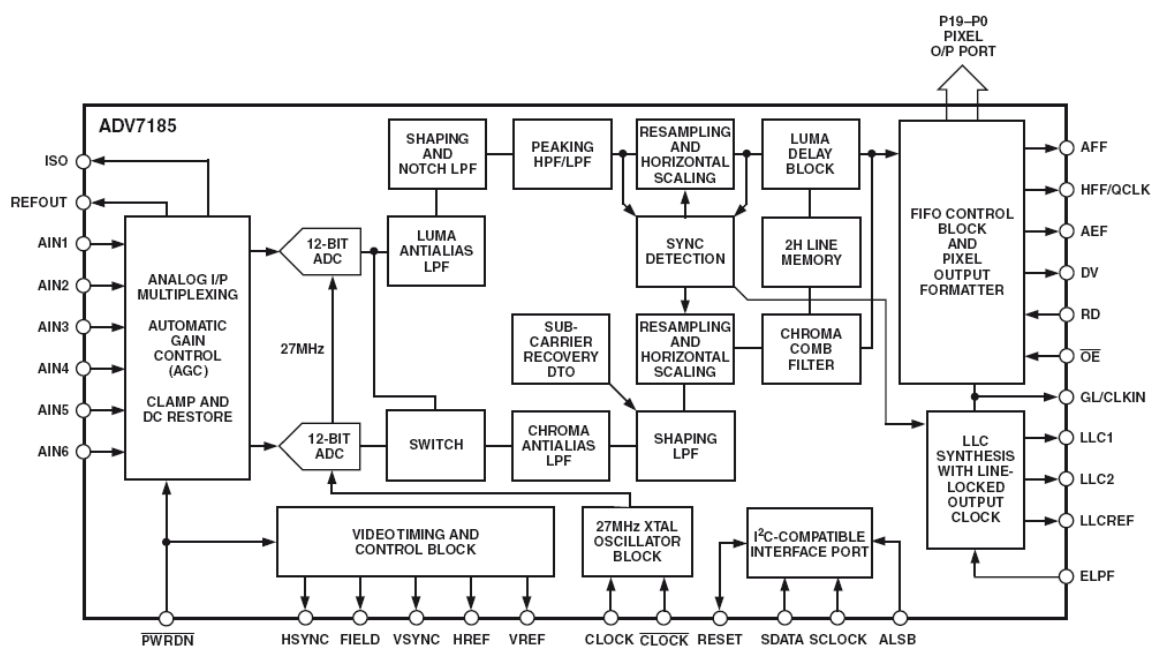
### **5.1.1 Osnovne zahteve za dekodiranje digitalnega toka po standardu BT.656**

Standard BT.656 predpisuje naslednje lastnosti prenosa digitalnega video signala pri uporabi sistema PAL:

- Hitrost prenosa znaša 27 MHz.
- Širina vodila lahko znaša 8 ali 10 bitov.
- Za vsako slikovno piko dobimo dva podatka, zato je frekvenca slikovnih pik enaka 13,5 MHz.
- Informacije o slikovnih pikah so v barvnem prostoru YCrCb.
- Za vsako slikovno piko je podan podatek o svetlosti, podatek o barvi pa je podan samo za vsako drugo slikovno piko (vzorčenje 4:2:2).
- Slika ima ločljivost (720 x 576) PAL in je razdeljena na dve polsliki po 720 x 288 slikovnih pik, prvo sestavljajo lihe, drugo pa sode vrstice.
- Med polslikami so polja za zatemnitev, ki ustrezajo zatemnitvam video signala PAL.
- Za sinhronizacijo skrbijo sinhronizacijska polja, ki uporabljajo rezervirane vrednosti \$FF in \$00. Vrednosti na vodilu so zato sicer omejene na obseg od \$01 do \$FE.

### **5.1.2 Pretvornik video signala**

Srce sistema predstavlja integrirano vezje ADV7185 [1], proizvajalca Analog Devices, ki zagotavlja A/D pretvorbo profesionalne kakovosti za izredno široko paleto video signalov. Sprejme praktično vse komercialne analogne slikovne standarde (PAL-(B/D/G/H/I/M/N), NTSC-(M/N)), izhodni podatki pa so vrste YCrCb (4:2:2 ali 4:1:1), ki se pošiljajo po standardu BT.656 (8 ali 10 bitov) ali pa paralelno (20 bitov). Vezje ima zelo visoko stopnjo integracije, tako da potrebuje le malo dodatnih komponent (vhodni filtri, blokirni kondenzatorji in kristal oziroma oscilator 27 MHz). Parametre pretvorbe nastavljamo po vodilu I2C s spreminjanjem ali branjem vrednosti krmilnih registrov, kar opravlja krmilnik PIC16F84, ki ga krmilimo z osebnim računalnikom prek povezave RS-232.



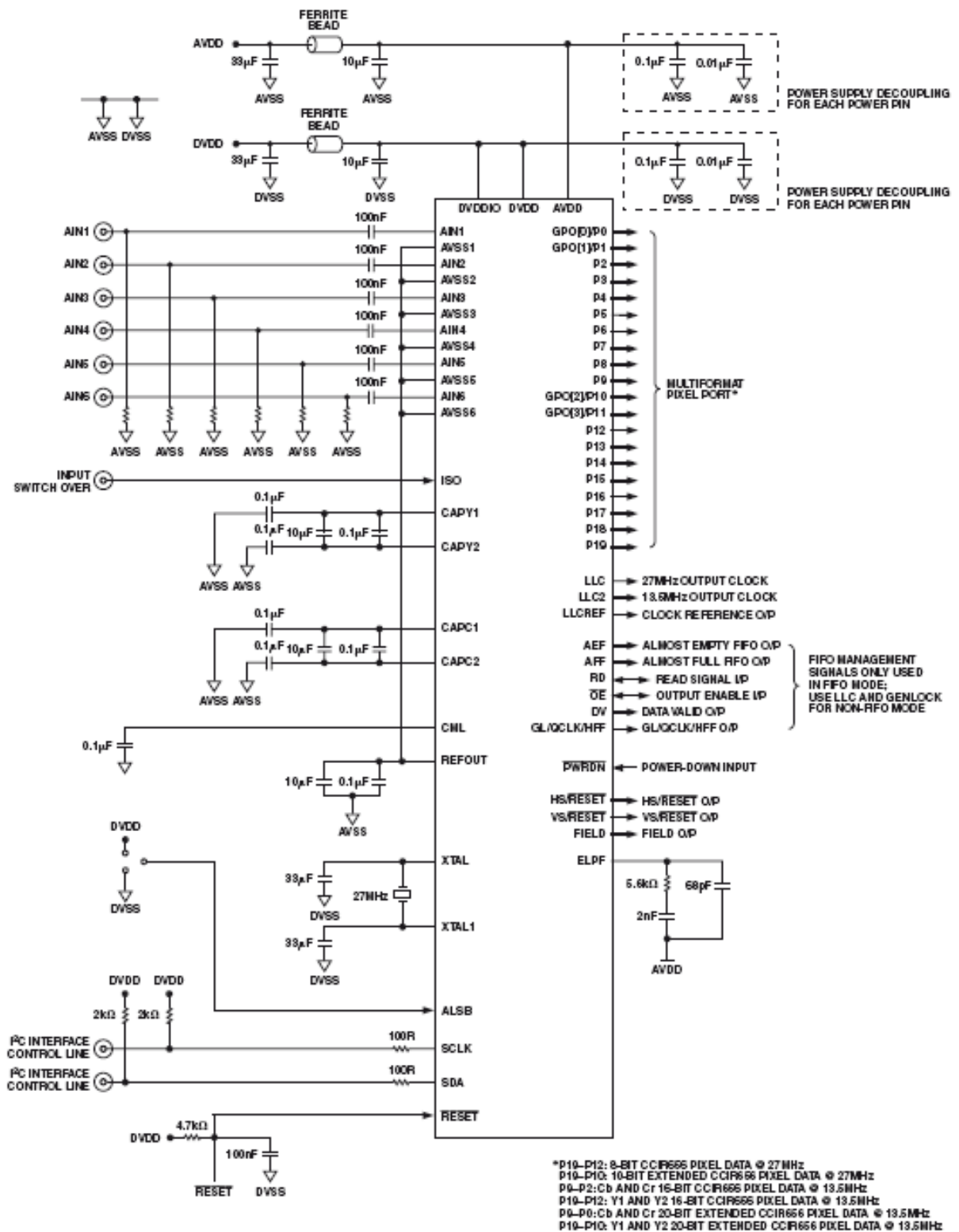
Slika 3: Blok shema integriranega vezja ADV7185 [1]

Ima šest video vhodov, ki jih vklopimo programsko, kar pomeni, da lahko na plošči pripravimo ustrezne priključke za vse slikovne zapise in da prilagodilniki skorajda niso potrebni. Na plošči najdemo tako naslednje video vhode: en priključek RCA za kompozitni video signal, en priključek DIN za signal S-Video in tri priključke BNC za signal vrste YUV. Ti vhodi pokrivajo 99 % predvidenih izvorov video signala, od videokamer, videorekorderjev, predvajalnikov DVD in televizorjev do grafičnih kartic z video izhodom. Edini pomembnejši priključek, ki ni na voljo, je priključek SCART, za katerega pa lahko kupimo prilagodilni kabel v vsaki tehnični trgovini.

Napajanje vezja ADV7185 zahteva dve napajalni napetosti: 3,3 V za digitalna vezja in 5 V za analogna vezja. Za to skrbi napajalni blok, opisan pozneje.

Izhodni digitalni signal ima lahko več različnih oblik, od 20-bitnega paralelnega izhoda z lastno uro ali vezjem FIFO in vhodom za zunanjo uro, do 8- ali 10-bitnega digitalnega toka po BT.656, vse izberemo z biti registra OCR prek I2C. Na voljo so tudi sinhronizacijski signali za vrstice, polslike in polja, izhodno vodilo pa je tristanjsko in ga je mogoče postaviti v stanje visoke impedance z vhodom OE ali z bitom TOD v registru OCR prek vodila I2C.

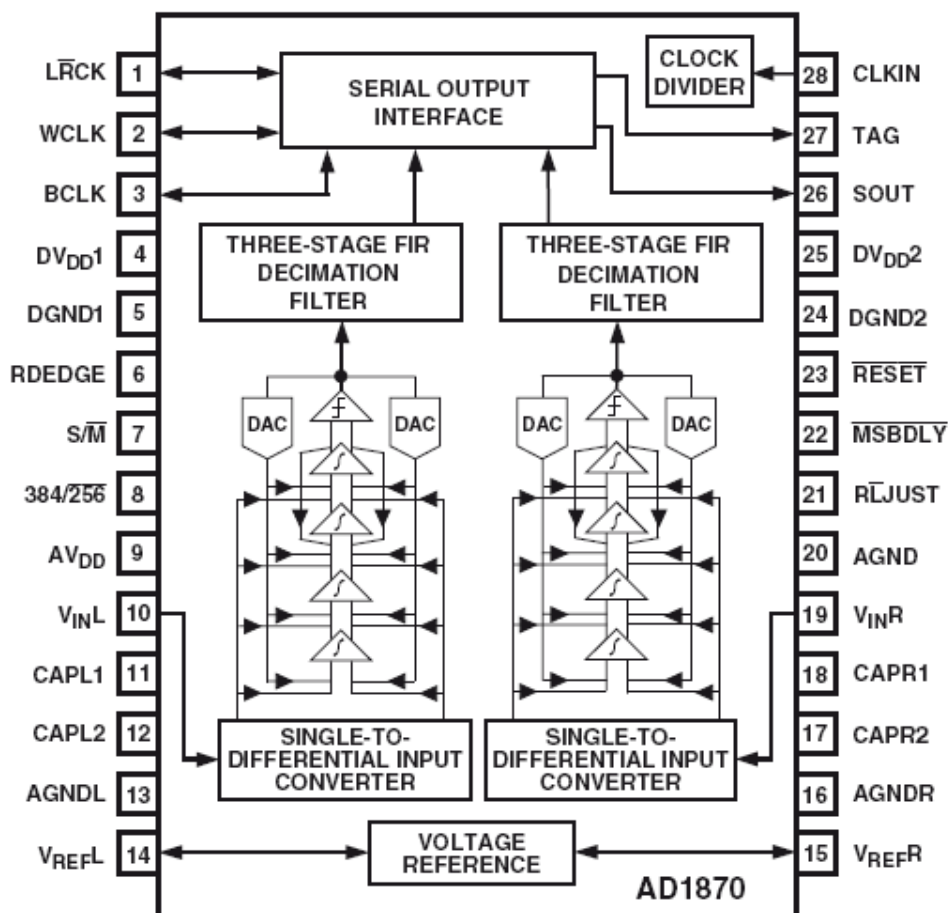
Natančen načrt priključitve in zunanjih komponent integriranega vezja ADV7185 najdete v prilogi 2.



Slika 4: Priporočena vezava vezja ADV7185 [1]

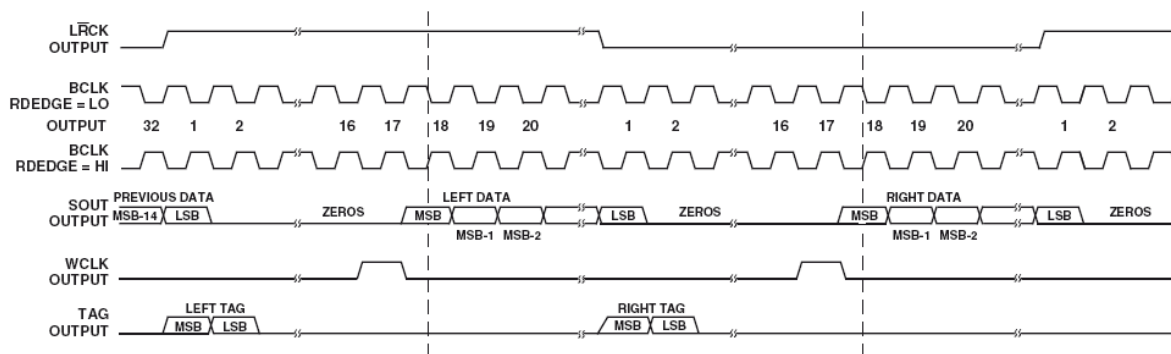
### 5.1.3 Pretvornik zvočnega signala

Pri nekaterih aplikacijah je dobrodošla tudi A/D pretvorba zvoka. Kljub temu da to ni bil naš osnovni cilj, smo sistemu dodali dvokanalni A/D pretvornik za zvočne signale, da bi bil sistem čim širše uporaben. Uporabili smo pretvornik AD1870 [3] proizvajalca Analog Devices, ki je dvojni 16-bitni  $\Sigma/\Delta$  A/D pretvornik četrtega reda s 64-kratnim prevzorčenjem.



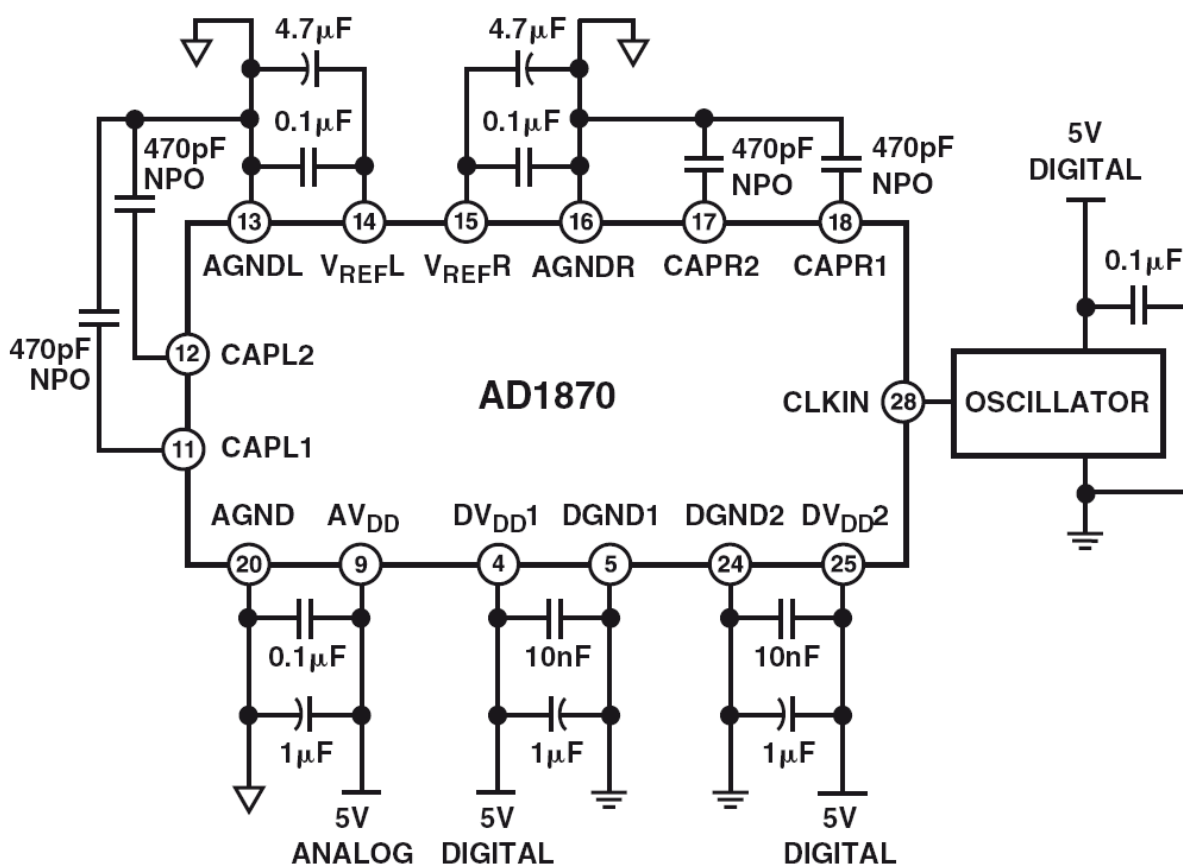
Slika 5: Blok shema integriranega vezja AD1870 [3]

Vezje potrebuje zelo malo zunanjih komponent, le vhodna RC filtra in filtra modulatorjev. Za delovanje potrebuje napetost 5 V, ki prihaja iz istega napajalnika kot za video A/D pretvornik. Odlikuje ga kakovostna pretvorba z malo šuma, presluha in popačenja. Za delovanje pri standardni frekvenci vzorčenja 44,1 kHz potrebuje vhodno uro 12,288 MHz, ki jo uporablja tudi mikrokrmilnik in jo dobimo s samostojnim oscilatorjem. Izhodno vodilo je serijsko, podatki pa se pošiljajo izmenoma za levi in desni kanal, s krmilnimi signali za vsak bit, besedo in zamenjavo kanala.



Slika 6: Potek izhodnih signalov vezja AD1870 [3]

Natančen načrt priključitve in dodatnih komponent zvočnega A/D pretvornika najdete v prilogi 3.



Slika 7: Priporočena vezava vezja AD1870 [3]

### 5.1.4 Krmilnik za nastavljanje parametrov sistema

Za nastavljanje parametrov video A/D pretvornika je na voljo vodilo I2C, ki pa ni najbolj primerno za neposredno povezavo z osebnim računalnikom. Primernejši vodili sta USB in RS-232, ki sta na voljo na veliki večini osebnih računalnikov. Zaradi tega smo se odločili, da sistemu dodamo mikrokrmilnik, ki služi kot posrednik med osebnim računalnikom in video A/D pretvornikom, tako da parametre, ki jih prejme prek vodila RS-232, prevede in pošlje v vezje ADV7185 po vodilu I2C.



Slika 8: Nastavljanje parametrov vezja ADV7185 s PC-jem prek krmilnika

Uporabili smo preprost mikrokrmilnik PIC16F84 [4], ki dobiva uro za delovanje iz zunanjega oscilatorja 12,288 MHz, napaja pa se z napetostjo 3,3 V, iz istega regulatorja (LM2973-3.3) kot vezje ADV7185. Za komunikacijo z osebnim računalnikom je potreben pretvornik nivojev po standardu RS-232, za kar smo uporabili integrirano vezje MAX232 [7], ki se skupaj z oscilatorjem napaja iz lastnega regulatorja z napetostjo 5 V. Za komunikacijo z vezjem ADV7185 sta potrebna le še dva upora, ki držita signala vodila I2C na visokem nivoju. Komunikacijo prek vodila RS-232 označuje utripanje zelene lučke LED, delovanje oscilatorja pa neprekinjeno vključena rdeča lučka LED.

Mikrokrmilnik poleg naštetega krmili še nekatere vhode obeh A/D pretvornikov in hrani nastavljene parametre v svojem pomnilniku EEPROM, tako da jih je po prvi določitvi mogoče naložiti tudi brez osebnega računalnika. Za to je namenjen kratkostičnik, ki vstavljen omogoči komunikacijo z osebnim računalnikom, odstranjen pa ob ponastavitvi sistema sproži hitro samodejno nastavitve registrov vezja ADV7185 na prej določene vrednosti. Za zanesljivo delovanje vezja je vgrajeno še ponastavitveno vezje MAX809T [8], ki ob vklopu in motnjah napajanja za določen čas vklopi ponastavitveni signal vezja.

Natančen načrt sklopa s krmilnikom, pretvornikom nivojev, oscilatorjem in ponastavitvenim vezjem najdete v prilogi 4. Program krmilnika (firmware) smo razvili v jeziku C [16], programirali pa smo ga prek vodila za serijsko programiranje. Je kratek in zasede le petino programskega pomnilnika, ki je na voljo.

### **5.1.5 Način nastavljanja parametrov prek povezave RS-232**

Če je kratkostičnik vstavljen v sistem, sistem ob vklopu pričakuje ukaze za nastavljanje parametrov, ki morajo v obliki štirih znakov ASCII priti prek povezave RS-232. Po pošiljanju vsakega ukaza se sistem odzove in tako potrdi sprejem ukaza. Ukaze lahko pošiljamo z običajnim terminalskim programom, ki je del vsakega operacijskega sistema, za lažje delo pa smo pripravili tudi poseben program, ki omogoča uporabniku bolj prijazno nastavljanje sistema.

#### **5.1.5.1 Parametri povezave RS-232**

Povezava RS-232 mora biti konfigurirana za hitrost 2400 b/s, 8 podatkovnih bitov, 1 stop bit in brez paritetnega bita. Ta precej nizka hitrost je bila izbrana zato, da je mogoče krmiljenje z najpreprostejšo mogočo strojno opremo. Za povezavo med krmilno napravo (npr. osebnim računalnikom) in sistemom za zajem slike in zvoka je prav tako dovolj trižilni kabel (Rx, Tx in ozemljitev), tako da lahko uporabimo najmanj pretvornikov nivojev RS-232.

#### **5.1.5.2 Ukaz za ponastavitev vezja**

Ukaz za ponastavitev vezja je velika črka "x" in je edini, ki ga sestavlja le en znak. Ta ukaz takoj vklopi ponastavitvene signale vseh integriranih vezij v sistemu, kar sistem vrne v stanje ob vklopu – kot da bi pritisnili tipko za ponastavitev na sistemu. Sistem ob tem vrne potrditev "OK".

#### **5.1.5.3 Ukaz za varčevanje z energijo**

Če hočemo sistem postaviti v stanje nizke porabe (v stanje nizke porabe postavimo vezji ADV7185 in AD1870, ostala vezja delujejo nemoteno), pošljemo sistemu ukaz "P0001". Vezje se nanj odzove z "OK".

#### **5.1.5.4 Ukaz za spreminjanje načina delovanja sistema**

Če hočemo spremeniti način delovanja sistema, pošljemo vezju ukaz "C00", potem pa še parameter, ki mu pove stanje signalov RDEDGE (aktivno stanje serijskega vodila vezja AD1870) in ISO (signal, ki sproži ponovno sinhronizacijo video signala). Parameter določimo po tabeli 1.

Stanje signala RDEDGE	Stanje signala ISO	Parameter
0	0	"00"
0	1	"01"
1	0	"02"
1	1	"03"

**Tabela 1: Vrednosti parametra za ukaz za spreminjanje načina delovanja sistema**

Sistem potem spremeni vrednost signalov v skladu s parametrom, shrani njihovo stanje v svoj EEPROM in vrne potrditev "OK".

### **5.1.5.5 Ukaz za nastavljanje registra vezja ADV7185**

Vezju ADV7185 nastavljamo parametre pretvorbe video signala s spreminjanjem vrednosti registrov, do katerih dostopamo prek vodila I2C. Če želimo nastaviti vrednost določenega registra na določeno vrednost, pošljemo ukaz "w", dvomestni šestnajstiški naslov registra in dvomestno šestnajstiško želeno vrednost registra. Naslovi in vrednosti registrov so natančno navedeni in opisani v dokumentaciji za vezje ADV7185.

Če želimo na primer nastaviti vhodni video signal na sponke YUV, z ustrezno obliko signala, moramo register na naslovu \$00 (Input Control) nastaviti na vrednost \$0A. To storimo z ukazom "w000A". Krmilnik potem prek vodila I2C nastavi vrednost registra v vezju ADV7185, preveri, ali je vpisovanje uspelo, shrani zapisano vrednost v svoj EEPROM in vrne potrditev "OK". V svojem pomnilniku EEPROM tako hrani zrcalno sliko registrov vezja ADV7185, ki jo uporabi, če je ob ponastavitvi vezja iz sistema odstranjen kratkostičnik. Takrat prepiše celotno vsebino svojega pomnilnika EEPROM v registre vezja ADV7185 in ga tako povrne v zadnje stanje pred ponastavitvijo sistema. Na ta način je sistem po prvi nastavitvi mogoče uporabljati brez računalnika.

### **5.1.5.6 Ukaz za branje registra vezja ADV7185**

Registre vezja ADV7185 je mogoče tudi brati. To lahko počnemo zato, da vidimo trenutno stanje vezja ali pa zato, da preverimo njegove nastavitve. Če hočemo prebrati stanje določenega registra, pošljemo sistemu ukaz "R00", potem pa dvomestni šestnajstiški naslov želenega registra. Krmilnik potem prebere register vezja ADV 7185 prek vodila I2C in vrne vrednost registra v dvomestni šestnajstiški obliki.

Če na primer želimo prebrati prej nastavljen register z naslovom \$00, pošljemo ukaz "R0000" in dobimo odziv "0A".



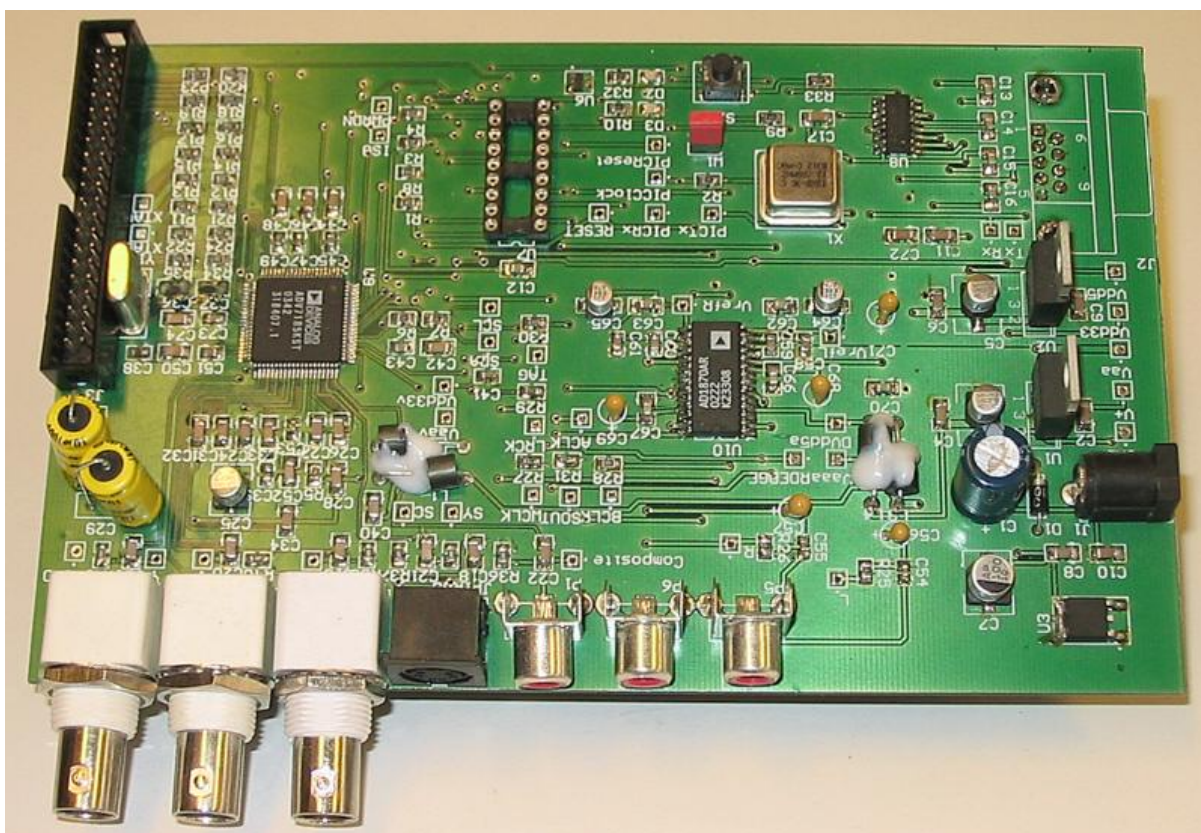
## 5.1.6 Napajalni sklop

Kadar združujemo vezja, ki delujejo z analognimi in digitalnimi signali, tako kot v našem sistemu, je pomemben del načrtovanja skrb za "čiste" napajalne napetosti. Stikalne motnje se namreč hitro prenesejo v analogni del vezja in motijo naš želeni signal. V takih primerih je napajalne napetosti dobro ločiti.

Naš sistem sicer potrebuje le dve napajalni napetosti, 3,3 V in 5 V, vendar smo se odločili, da bomo oscilator in pretvornik nivojev za RS-232 napajali iz ločenega regulatorja, saj bi drugače lahko motila občutljiva analogna vezja obeh A/D pretvornikov. Napajalno vezje tako sestavljajo trije regulatorji (LM1117 [5] in 2x LM2937 [6]), vhodna napetost lahko znaša od 6,5 do 12 V, pred napačno polariteto pa je napajalni blok zaščiten z diodo. Za zmanjšanje motenj, ki bi se prenašale po skupni negativni sponki napajanja, je celotno tiskano vezje v največji mogoči meri pokrito z bakrom, ki je vedno povezan z negativno sponko napajanja. Impedanca negativne sponke je tako zmanjšana na največjo mogočo mero, z njo pa tudi motnje, ki se prenašajo po njej.

Poraba vezja znaša tipično okoli 80 mA, izračunana največja mogoča poraba pa znaša okoli 200 mA. Za napajanje je torej dovolj običajen adapter za omrežno vtičnico.

Natančen načrt napajalnega sklopa najdete v prilogi 5.



Slika 9: Sistem za zajem slike in zvoka

## **5.2 Sklop za obdelavo digitalnega video toka**

Sklop za obdelavo digitalnega signala iz sistema za zajem slike in zvoka temelji na sistemu z vezjem FPGA Spartan 3, ki ga je zasnoval doc. dr. Andrej Trost, jaz pa sem mu dodal še zunanji pomnilniški modul.

### **5.2.1 Učni sistem Spartan3**

Učni sistem je v osnovi namenjen praktičnemu pouku načrtovanja vezij FPGA pri doc. dr. Andreju Trostu, ki ga je tudi zasnoval. Zaradi tega je izdelan tako, da je čimbolj preprost za uporabo. Sestavljajo ga vezje FPGA Xilinx Spartan 3 [9], vezje CPLD, vmesnik USB, oscilator in napajalnik.

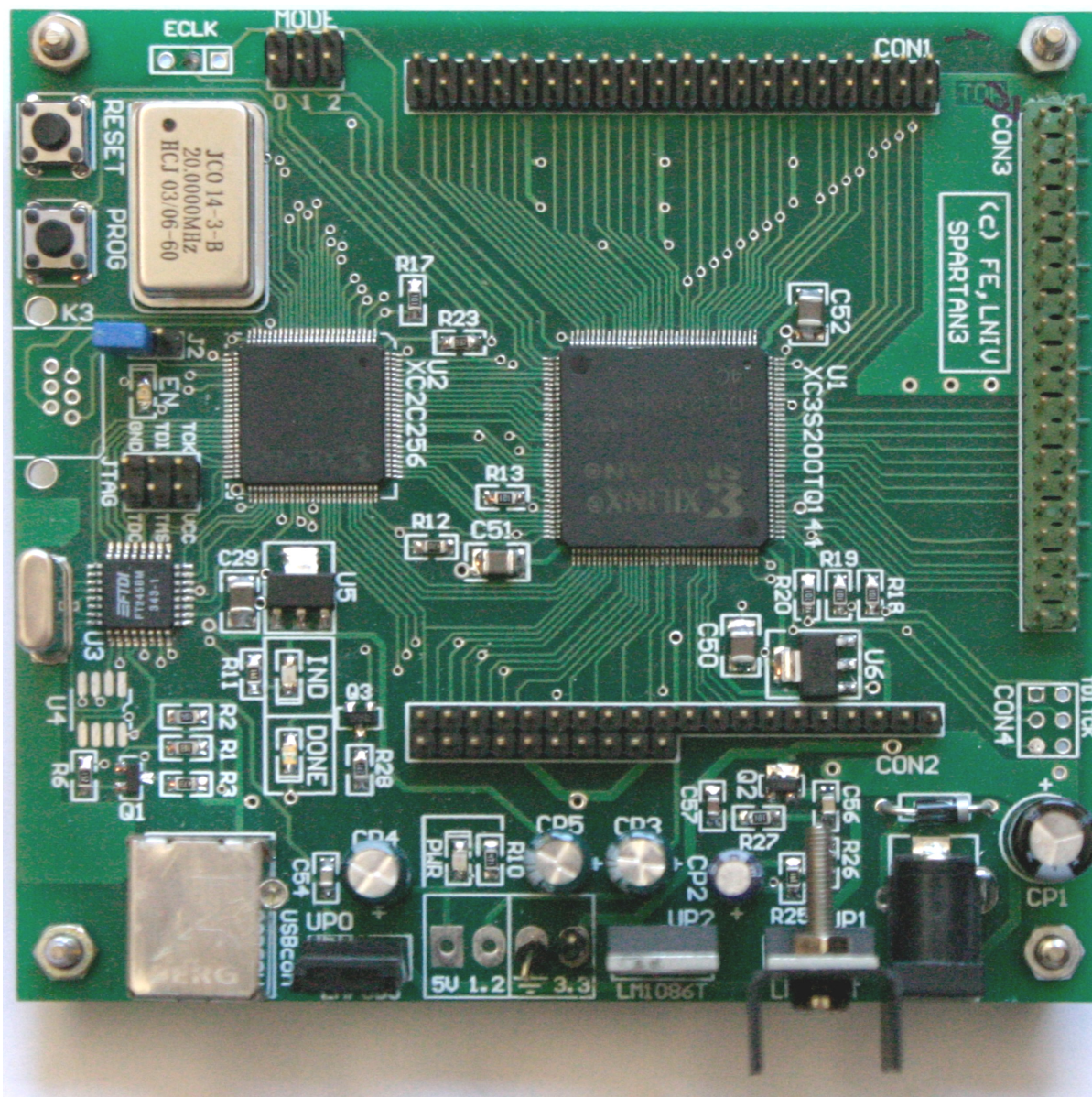
Vezje CPLD in vmesnik USB skrbita za programiranje vezja FPGA, v povezavi s programom za programiranje pa omogočata tudi pošiljanje podatkov na 8-bitno vodilo vezja FPGA in sprejem podatkov iz drugega 8-bitnega vodila vezja FPGA. Oscilator daje vezju FPGA stabilno uro 20 MHz, ki jo lahko z moduli DCM [10], vgrajenimi v vezje FPGA, pretvorimo v skoraj poljubno uro od 24 do 360 MHz, ki ima poleg tega še nadzorovano fazo po vsem vezju. Sistem ima tudi vhod za zunanjo uro.

Večina vhodno/izhodnih priključkov vezja FPGA je na voljo na treh konektorjih s po 40 priključki, od katerih sta dva namenjena kot prostor za razširitvene module, tretji pa za povezovanje sistema z zunanjimi napravami.

Programiranje vezja FPGA poteka z namenskim programom, ki sprejema datoteke *.bit* razvojnega okolja Xilinx ISE, prek vodila USB, vmesnika USB proizvajalca FTDI in vezja CPLD. Programiranje traja manj kot 5 sekund, vmesnik USB pa omogoča tudi z računalnikom krmiljeno spreminjanje osmih digitalnih vhodnih signalov na vezju FPGA, kar lahko uporabimo za nastavljanje med delovanjem.

Vse, kar rabimo vedeti za uspešno programiranje tega vezja, najdemo v načrtu sistema, ki ga najdete v prilogi 6, in pa v dokumentaciji podjetja Xilinx. Delo s sistemom je preprosto in hitro, posebej pomembno dejstvo pa je, da je za uporabljeno vezje FPGA XC3S200 na voljo brezplačno razvojno okolje Xilinx ISE.





Slika 10: Sistem Spartan 3

### 5.2.2 Pomnilniški razširitveni modul

Sistem Spartan 3 nima lastnega pomnilniškega modula, razen pomnilnika, ki je vgrajen v vezju FPGA. Tega pomnilnika za naše potrebe ni dovolj (ne zadostuje za shranjevanje ene slike pri predvideni ločljivosti). Zaradi tega smo se odločili, da številnim razširitvenim karticam, ki so že pripravljene za sistem, dodamo še eno.

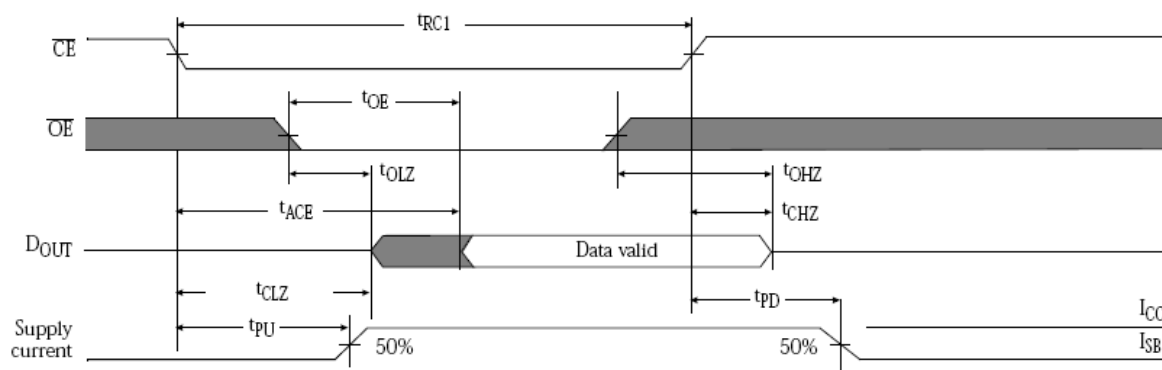
Ta modul vsebuje tri pomnilna vezja AS7C34096-12JC [14] vrste SRAM s po 512 KB pomnilnika in 8-bitnim podatkovnim vodilom. Toliko pomnilnika zadostuje za 2 sliki ločljivosti 512 x 512 slikovnih pik pri 24-bitnem zapisu barv, kar je za naše potrebe več kot dovolj. Ker je za razširitvene module na sistemu Spartan 3 predvidenih le 40 samostojnih

povezav (16 je skupnih z zunanjim priključkom, ki je zaseden v celoti), smo bili prisiljeni vsa tri pomnilna vezja povezati na isto naslovno in podatkovno vodilo, kar za trikrat zmanjša dosegljivo hitrost prenosa podatkov med pomnilnikom in sistemom. Pomanjkanje povezav je bilo celo tolikšno, da smo morali vhode za omogočanje vezij (CE) trajno priključiti na aktivno vrednost. To na delovanje sicer ne vpliva bistveno, po nepotrebnem pa se s tem za petkrat poveča poraba pomnilniških modulov, ko jih ne potrebujemo. Za komunikacijo s pomnilnikom nam ostane tako eno 19-bitno naslovno vodilo, eno 8-bitno podatkovno vodilo in dvakrat po trije signali za omogočanje vpisovanja in branja podatkov.

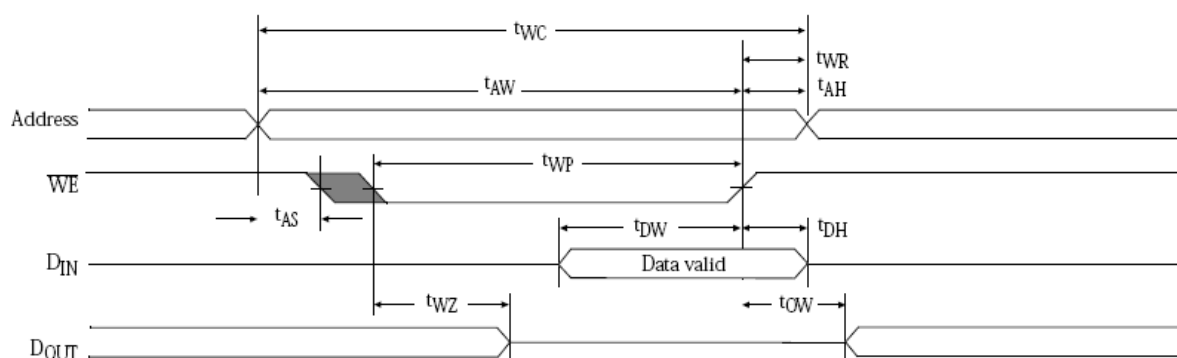
Hitrost dostopa do pomnilnika omejuje predvsem zakasnitev pri postavljanju naslova, saj mine od veljavnega naslova na vodilu do veljavnega podatka na izhodu 12 ns. Če upoštevamo še tipično hitrost postavljanja vodil v vezju FPGA, ki znaša okoli 8 ns, je očitno, da je hitrost branja fizikalno omejena na največ 50 MHz, pri realni implementaciji pa še na kakšen MHz manj (zaradi zakasnitev v samem vezju FPGA).

Najdaljši čas postavljanja naslovnega vodila	12 ns
Najdaljši čas postavljanja podatkovnega vodila	6 ns
Največja poraba toka	130 mA
Največja poraba toka med mirovanjem	20 mA

**Tabela 2: Najpomembnejše karakteristike uporabljenih pomnilniških modulov [14]**



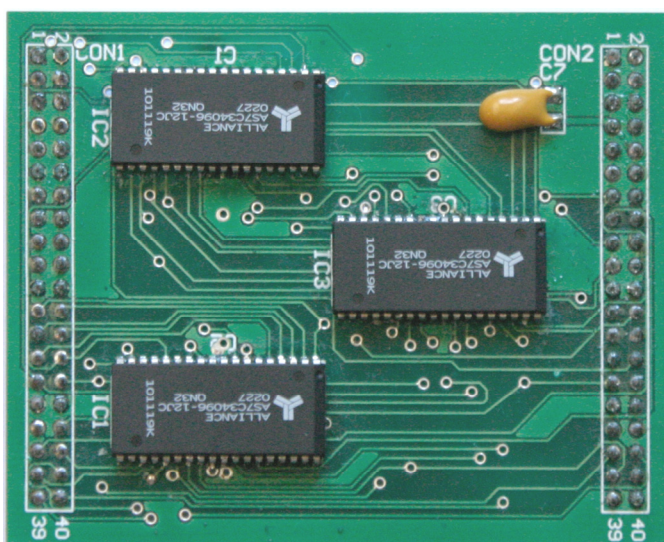
**Slika 11: Bralni cikel pomnilniških modulov AS7C34096 [14]**



**Slika 12: Zapisovalni cikel pomnilniških modulov AS7C34096 [14]**



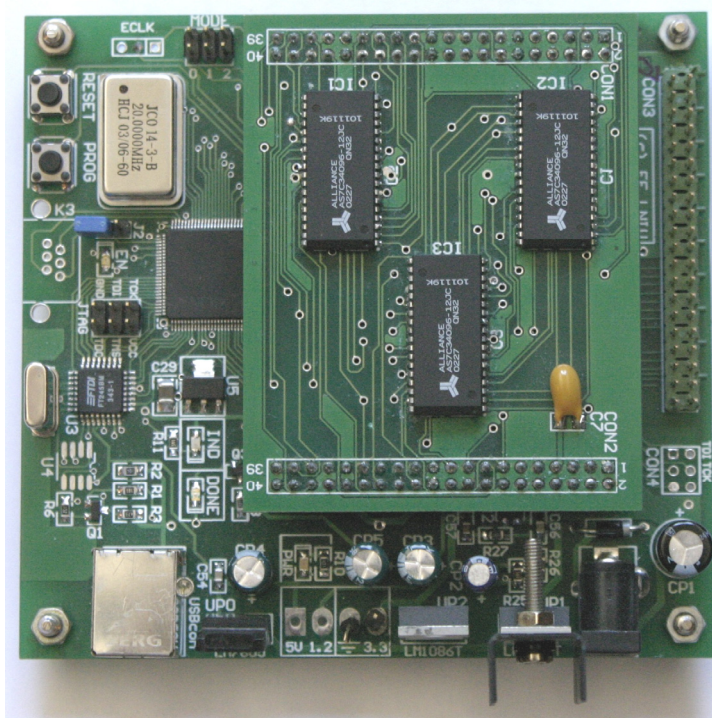
Ta del sistema je prav tako zasnovan povsem splošno in bo primeren za vse aplikacije, ki potrebujejo zunanji pomnilnik vrste SRAM. Njegov načrt najdete v prilogi 7.



Slika 13: Razširitveni modul s pomnilnikom SRAM

### 5.2.3 Povezovanje pomnilniškega modula in sistema Spartan 3

Pomnilniški modul je zasnovan tako, da ga natakne na razširitvena priključka sistema Spartan 3. Povezave med signali pomnilnika in vezja FPGA so opisane v obliki datoteke *.ucf* v razvojnem okolju Xilinx WebPack. Ustrezne nastavitve najdete že pripravljene v prilogi 8.



Slika 14: Pomnilniški modul, nataknen na sistem Spartan 3

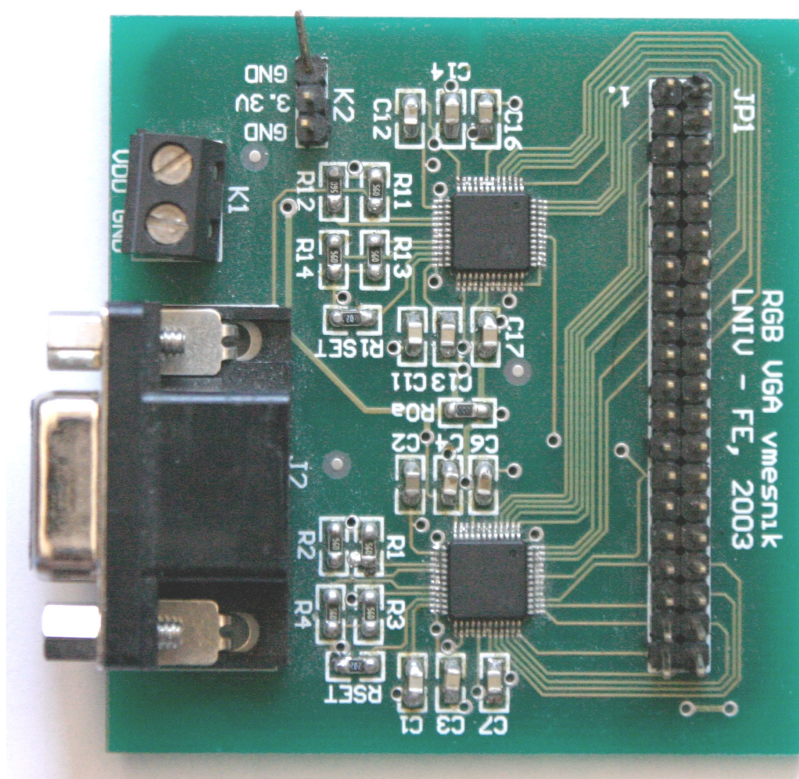
### 5.3 D/A pretvornik za prikaz slike na računalniškem monitorju

Kot zadnji del strojne opreme omenimo še D/A pretvornik za prikaz slike na računalniškem monitorju. Njegov avtor je Primož Šalič, dipl. inž., ki ga je sestavil kot del svoje diplomske naloge [19]. Pripravil je tudi modul VHDL za poganjanje pretvornika, ki pa za potrebe tega projekta ni bil primeren in ga nismo uporabili.

Sestavljata ga dva dvojna 8-bitna D/A pretvornika HI5628/6IN [15], ki pretvarjata signale za rdečo (R), zeleno (G) in modro (B) barvo. Sinhronizacijska signala sta speljana z vhodnega priključka neposredno na izhod in ju moramo poganjati s sistemom FPGA. S tem dobimo 5 signalov, ki so potrebni za upravljanje računalniškega monitorja. Povedati je potrebno, da gre pri tem modulu za gol D/A pretvornik in da je za vse sinhronizacijske čase, uro in signale RGB potrebno poskrbeti v vezju FPGA. Sami pretvorniki delujejo do frekvence 60 MHz, kar zadošča za ločljivost 800 x 600 slikovnih pik pri hitrosti osveževanja 85 Hz.

Ker na izhodnem priključku sistema Spartan 3 ni dovolj linij, da bi nanj lahko v celoti priključili sistem za zajem slike in D/A pretvornik (zmanjka 6 linij), v končni izvedbi poganjamo D/A pretvornik s po le 6 biti na barvni kanal. Kljub zmanjšanju ločljivosti ta rešitev še vedno daje za naše potrebe zadovoljive rezultate.

Načrt tega modula najdete v prilogi 9.



Slika 15: D/A pretvornik za krmiljenje računalniškega monitorja

## 6 Opis programske opreme za nastavljanje parametrov A/D pretvorbe, razvoj modulov VHDL in programiranje vezja FPGA

---

Za delo s sistemom za obdelavo video signala potrebujemo tudi programsko opremo, ki jo sestavljata dva namenska programa in razvojno okolje Xilinx ISE. Prvi je namenjen nastavljanju parametrov sistema za zajem slike in zvoka, drugi pa programiranju in krmiljenju vezja FPGA v sklopu za obdelavo podatkov. Z razvojnim okoljem Xilinx ISE načrtujemo in prevajamo module VHDL, ki omogočajo delovanje sistema.

### 6.1 Nastavitve AV ADC plošče

Ker je sistem za zajem slike in zvoka zasnovan izredno splošno, mu moramo pred uporabo določiti parametre delovanja. Predvsem moramo določiti uporabljen video vhod, uporabljen video standard in obliko izhodnega toka podatkov, nastaviti pa je mogoče še veliko drugih parametrov. V ta namen je vanj vgrajen krmilnik, ki nastavlja parametre sistema. Pošljemo jih s PC-ja prek povezave RS-232, ko pa so enkrat določeni, si krmilnik zapomni zadnje nastavitve in jih, če odstranimo kratkostičnik iz sistema, ob vklopu samodejno naloži v registre integriranega vezja ADV7185.

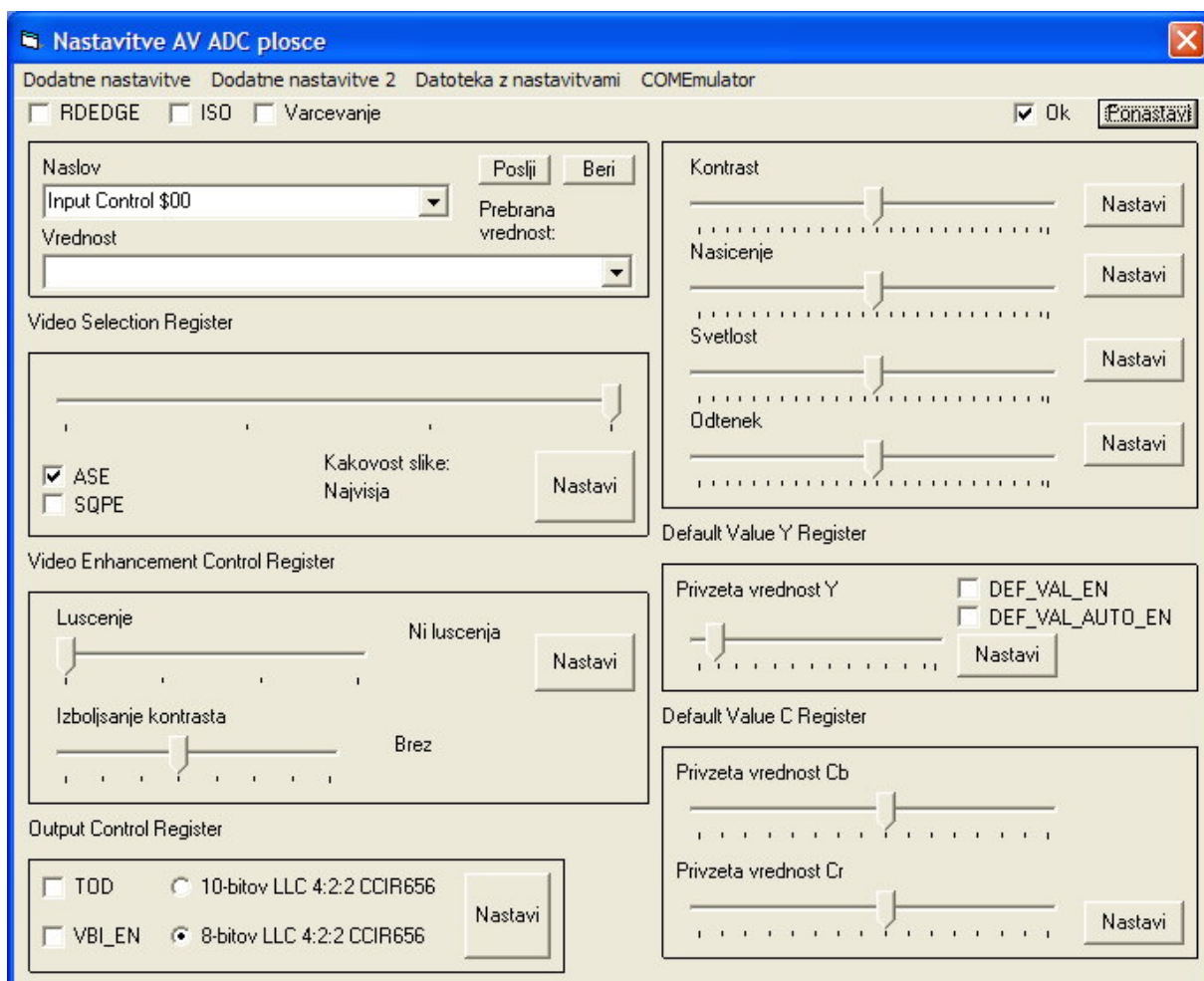
Nastavljanje lahko poteka prek terminalskega programa, s pomočjo ukazov, naštetih v opisu strojne opreme, za uporabniku prijaznejšo uporabo pa smo pripravili tudi program, ki nastavitve izračuna iz parametrov, ki mu jih pripravi uporabnik, in jih tudi pošlje. Program je bil izdelan v okolju Microsoft Visual Basic 6.0, sestavlja pa ga pet oken.

#### 6.1.1 Osnovne nastavitve

Okno z osnovnimi nastavitvami se odpre ob zagonu programa, na njem pa lahko nastavljamo registre iz bloka osnovnih nastavitvev (predvsem izvor, vrsto in kakovost video signala, velikokrat edine ob vklopu potrebne nastavitve), odpiramo ostala okna programa in ponastavimo sistem.

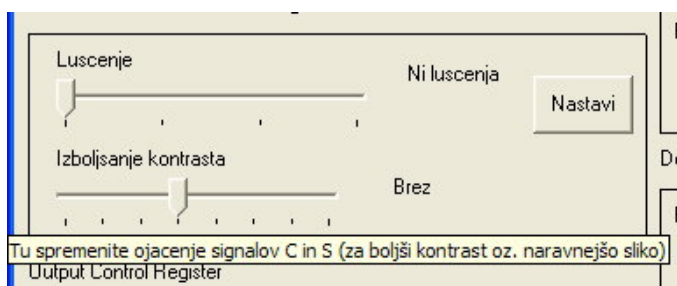
V vseh oknih velja, da so nastavitve razdeljene po okencih, ki ustrezajo enemu običajnemu ali razširjenemu registru, katerega ime, kot je navedeno v dokumentaciji vezja ADV7185, je napisano nad zgornjim levim vogalom okenca.

Registre nastavimo tako, da najprej določimo vrednosti parametrov z drsniki in gumbi za želeni register, potem pa pritisnemo gumb **Nastavi**. Ob tem program prek povezave RS-232 pošlje ustrezen ukaz. Če prejme potrditev o uspešni izvršitvi, se prikaže kljukica **Ok** v zgornjem desnem vogalu okna.



**Slika 16: Glavno okno programa za nastavljanje sistema za zajem slike in zvoka**

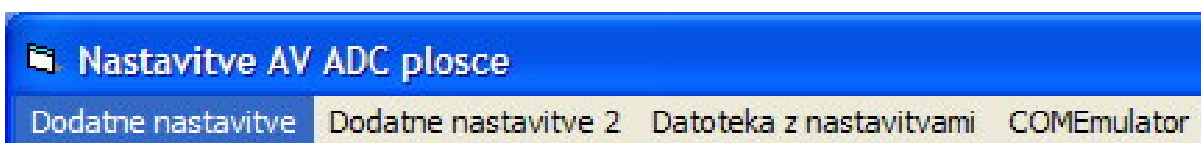
Parametri so označeni z imeni iz dokumentacije vezja ADV7185, kjer je tudi podrobno opisan njihov pomen. Da ni vedno potrebe po iskanju opisa parametrov, je za večino registrov in parametrov v vseh oknih programa vpisan opis delovanja, ki se pojavi, če kazalec miške nekaj sekund pustimo lebdeti nad ustreznim gumbom ali drsnikom.



**Slika 17: Vgrajeni opisi delovanja parametrov**



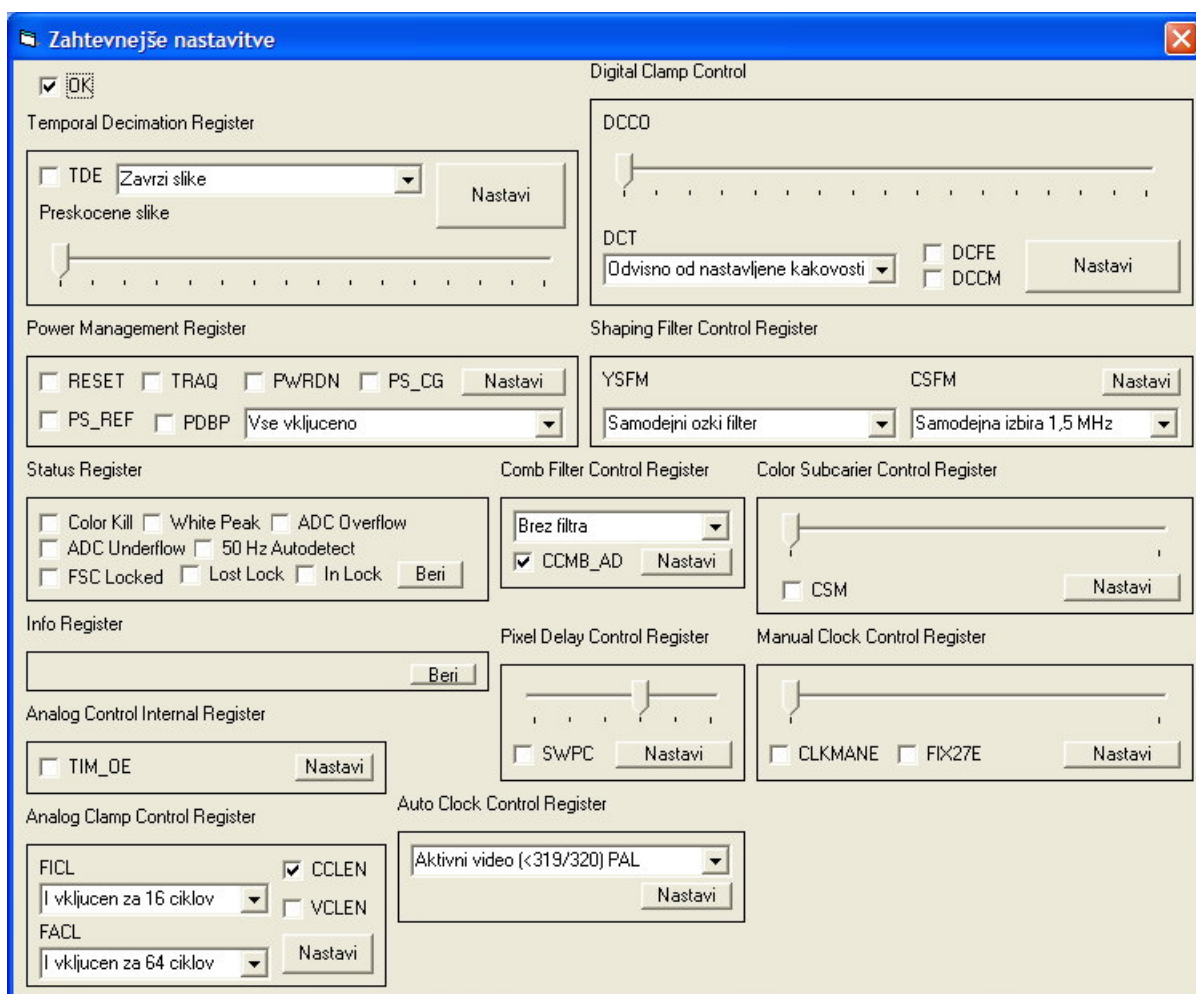
Ostala okna odpremo s klikom na menijsko vrstico.



Slika 18: Menijska vrstica, kjer lahko odpremo ostala okna

### 6.1.2 Dodatne nastavitve

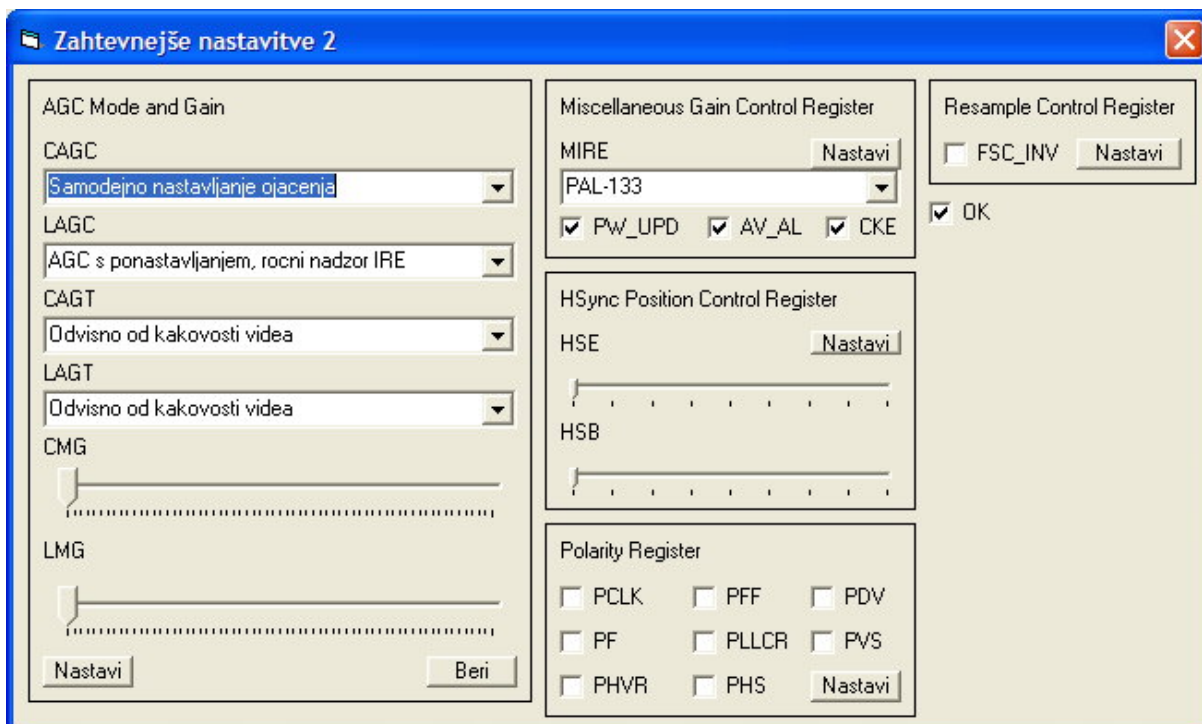
V oknu z dodatnimi nastavitvami lahko preverjamo stanje sistema (Status Register), kjer vidimo, ali je sistem ujel frekvenco nosilca barve in podobne informacije. Na voljo so še številne nastavitve, ki omogočajo prilagajanje pretvorbe video signala (nestandardnim) posebnostim izvora, in pa register, kjer lahko preberemo različico integriranega vezja ADV7185.



Slika 19: Okno Dodatne nastavitve

### 6.1.3 Dodatne nastavitve 2

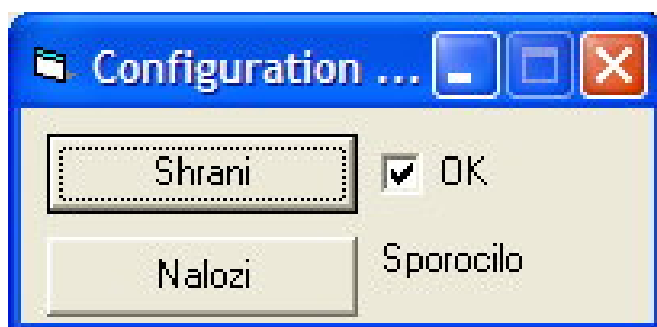
V tem oknu lahko nastavljamo različna ojačenja, polariteto izhodnih digitalnih video signalov in časovni položaj vodoravnega sinhronizacijskega impulza.



Slika 20: Okno Dodatne nastavitve 2

### 6.1.4 Datoteka z nastavitvami

To okno uporabljamo, ko so vse nastavitve vezja ADV7185 nastavljen po naših željah. Takrat kliknemo gumb **Shrani**, kar povzroči, da program prebere vrednosti vseh registrov vezja ADV7185 in jih shrani v datoteko *configfile.txt*. Ko želimo naslednjič drug sistem nastavitve na enake nastavitve, si lahko prihranimo zamudno nastavljanje posameznih registrov in preprosto pritisnemo **Naloži**. Program takrat iz datoteke *configfile.txt* prebere vse nastavitve in jih zapiše v registre vezja ADV7185. Ko je postopek končan, se prikaže kljukica.



Slika 21: Okno za datoteko z nastavitvami

### 6.1.5 Emulator sistema

Če odpremo to okno, se odpre le majhen kvadratak, ki označuje, da na serijskih vratih COM2 zdaj deluje emulator sistema. Emulator se obnaša kot sistem za zajem slike in zvoka, tako da ob sprejemu ostalih ukazov vrne potrditev, ob sprejemu ukaza za branje pa privzeto vrednost za brani register. Če ga želimo uporabljati, moramo kabel povezati med serijski vrati COM1 in COM2. Glavni namen tega emulatorja je demonstracija uporabe programa oziroma terminala ali pa preverjanje delovanja obeh serijskih vrat in kabla.

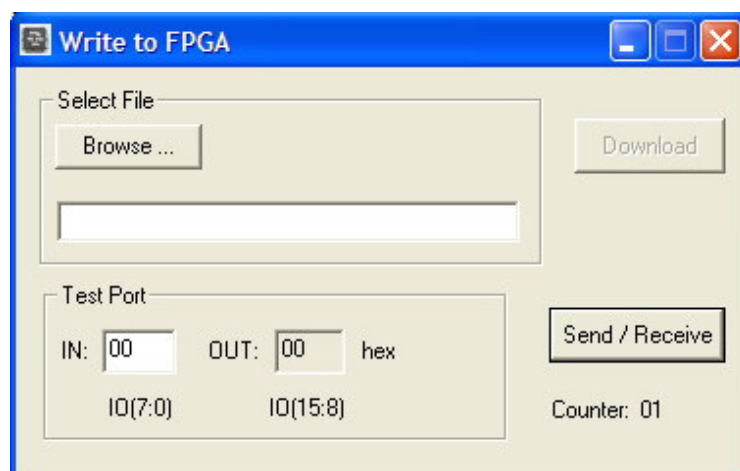
### 6.2 Program za programiranje in nastavljanje vezja FPGA

Ko napišemo in prevedemo modul VHDL, nam razvojno okolje Xilinx WebPack pripravi datoteko *.bit*, s katero je potrebno programirati vezje FPGA. To storimo s programom za programiranje in nastavljanje vezja FPGA, ki ga je pripravil Marko Mlakar [20], za uporabo s ploščo Spartan 3 pa ga je prilagodil doc. dr. Andrej Trost.

Program nudi še eno možnost – z njim lahko spreminjamo podatke na 8-bitnem vodilu, povezanim z vezjem FPGA, in beremo podatke z drugega 8-bitnega vodila, povezanega z vezjem FPGA. Na ta način lahko krmilimo module vezja FPGA brez dodatnih tipk, branje podatkov pa je uporabno za hitro preverjanje pravilnosti delovanja.

Delo s programom je preprosto – s klikom tipke **Browse** odpremo okno, v katerem poiščemo datoteko *.bit* za programiranje. Potem kliknemo tipko **Download**, ki izbrano datoteko naloži v sistem Spartan 3. Branje in pisanje na 8-bitni vodili izvedemo hkrati s tipko **Send/Receive**, pri tem se na prvo vodilo zapiše vrednost iz okenca **IN:**, v okencu **OUT:** pa se izpiše vrednost na drugem vodilu.

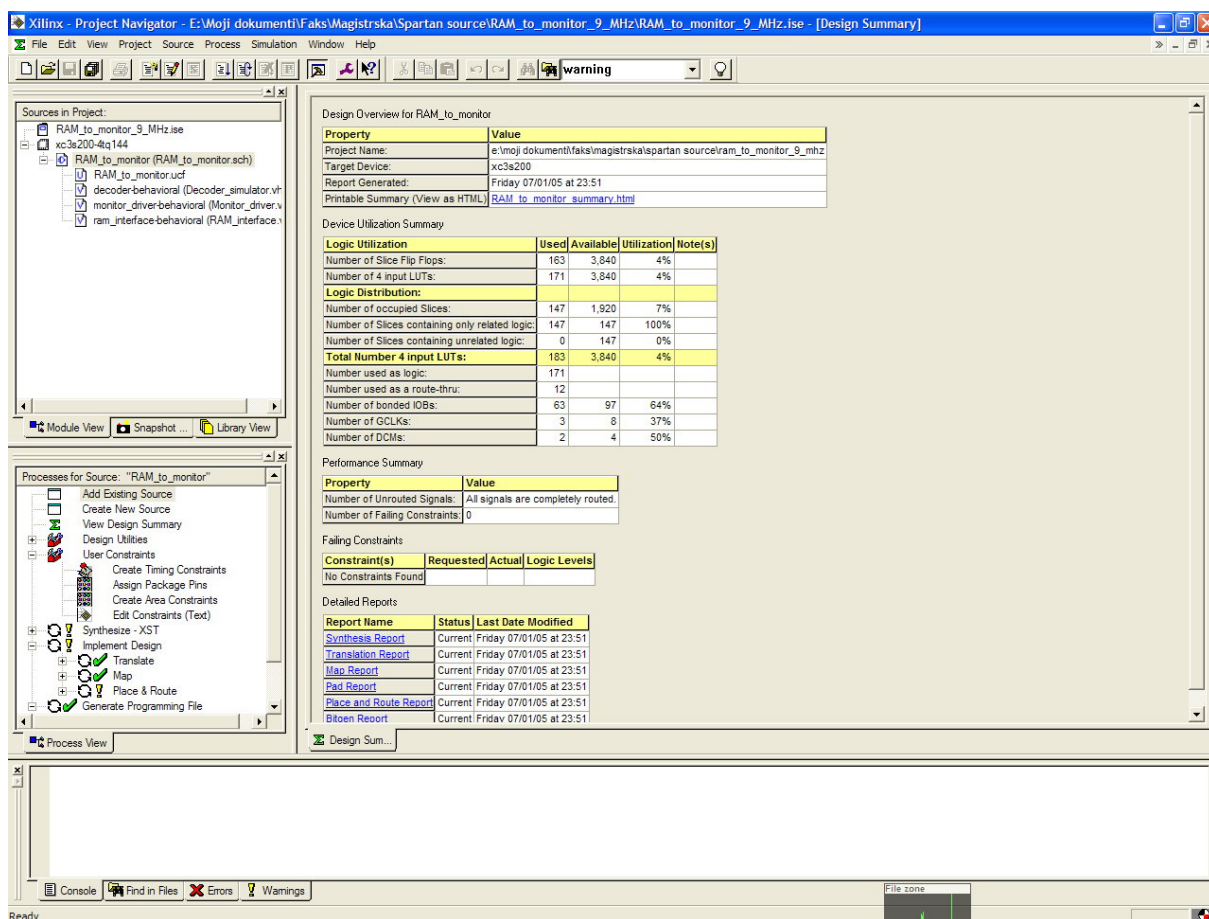
V prilogi 10 so podane definicije povezav vhodnega in izhodnega vodila z vezjem FPGA, kot so določene v datoteki *.ucf*.



Slika 22: Program za programiranje vezja FPGA

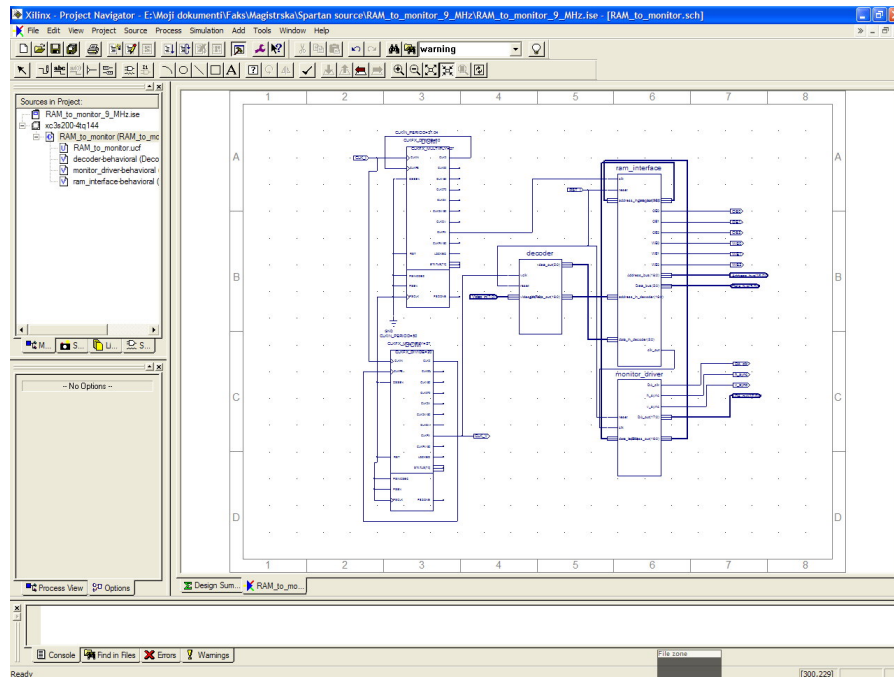
## 6.3 Razvojno okolje Xilinx ISE

Sistema ne moremo uporabljati brez razvojnega okolja za module VHDL, ki jih potrebujemo za programiranje vezja FPGA. Na tržišču je zelo veliko takšnih orodij, mi pa smo se odločili kar za orodje, ki ga ponuja proizvajalec vezja FPGA – podjetje Xilinx. Glavni razlog za to je, da gre za zelo zmogljivo orodje, ki je povrh tega tudi brezplačno, če ga uporabljamo z vezjem FPGA, ki je vgrajeno v naš rekonfigurabilni sistem. Če ga uporabljamo z zmogljivejšimi vezji, je potreben nakup licence.



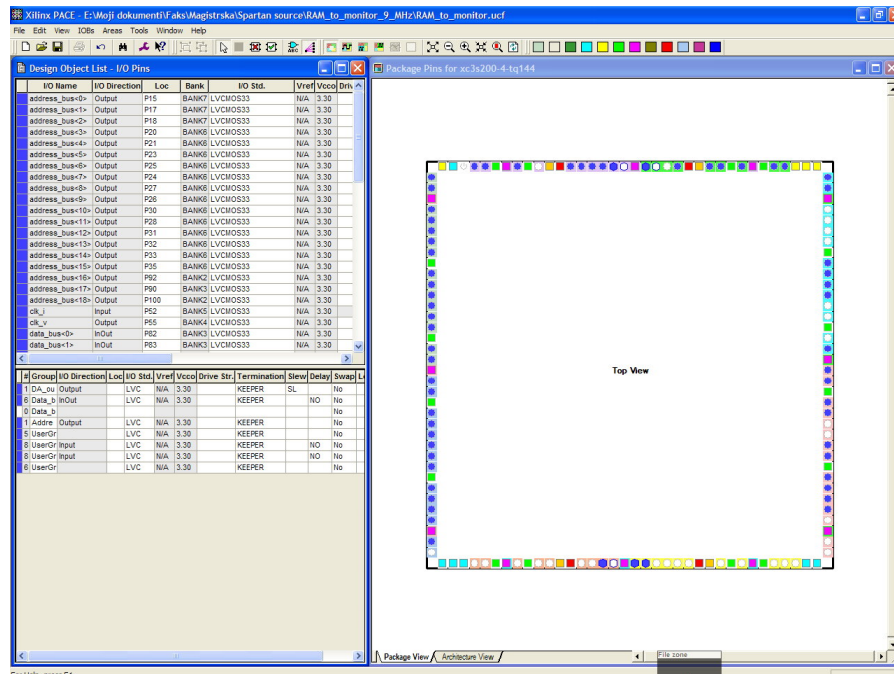
Slika 23: Razvojno okolje Xilinx ISE

Zelo uporabna funkcija tega okolja je možnost shematskega povezovanja modulov VHDL, ki je kot naročena za naš sistem. Uporabimo jo tako, da najprej napišemo posamezne module VHDL (recimo dekoder, pomnilniški modul in prikazni modul), zanje pripravimo shematske simbole, potem pa jih povežemo shematsko in dodamo še vse ostale potrebne komponente (vhodno-izhodne povezave, DCM itd.). Končanih modulov VHDL tako ni več potrebno spreminjati, spreminjanje arhitekture sistema pa je hitro, preprosto in preglednejše kot povezovanje modulov v jeziku VHDL. Še posebej priročno je vstavljanje modulov DCM in določanje vhodno/izhodnih signalov.



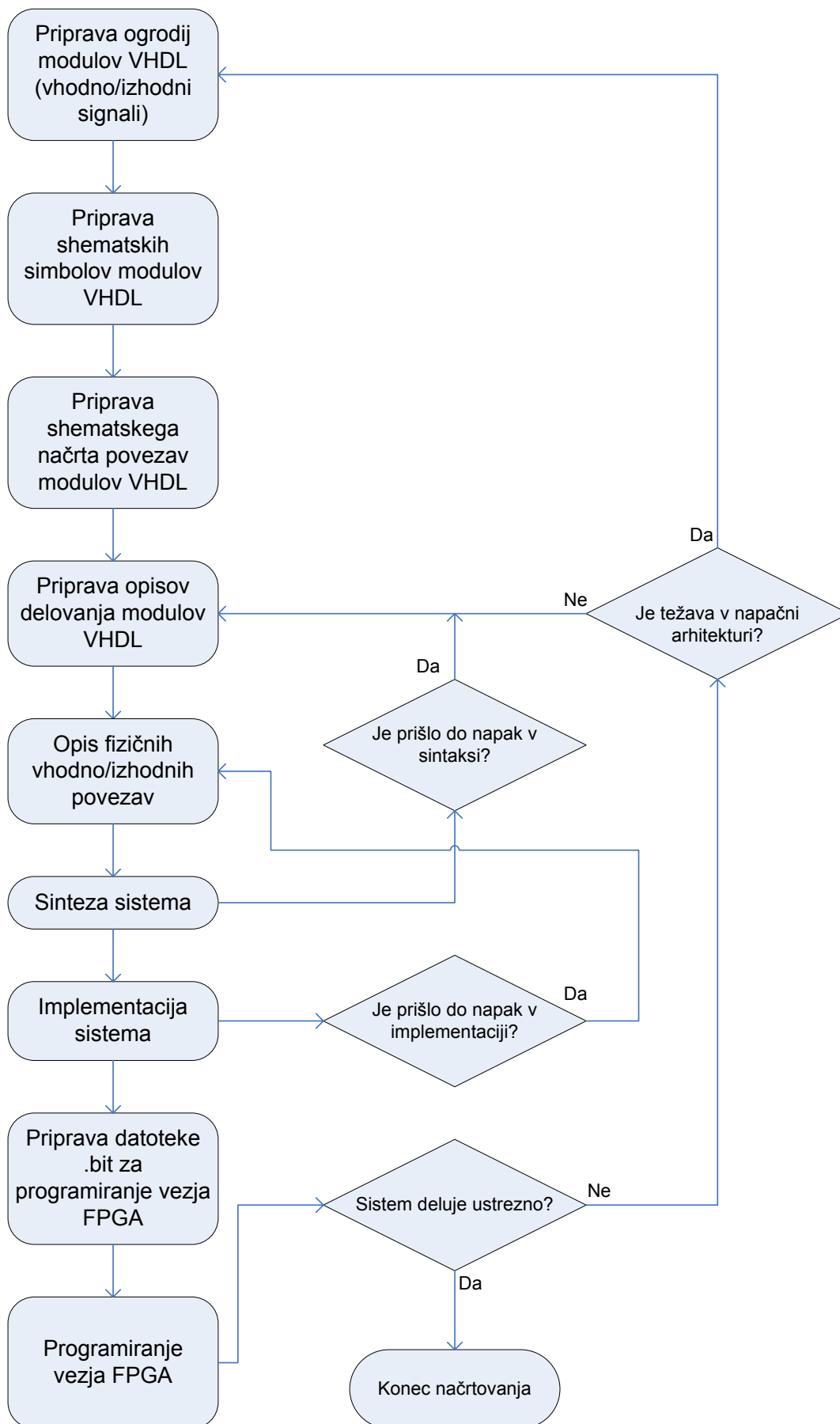
Slika 24: Shematsko povezovanje modulov VHDL v okolju Xilinx ISE

Ko tako sestavimo zeleno vezje, moramo določiti še vhodno/izhodne povezave, kar storimo z datoteko *.ucf* in programom Xilinx Pace. Vsaki povezavi določimo priključek vezja FPGA, njegov priključek na sistemu pa določimo s pomočjo načrta sistema Spartan 3 v prilogi 6.



Slika 25: Določitev vhodno/izhodnih povezav vezja

Na koncu izberemo še možnost **Generate Programming File**, ki vse datoteke prevede v datoteko *.bit*, s katero programiramo vezje FPGA.



Slika 26: Potek načrtovanja sistema

## 7 Opis zgradbe in delovanja modulov VHDL

---

V prejšnjih poglavjih smo opisali veliko strojne in programske opreme, ki je bila potrebna za sestavljanje sistema, ki je zahtevala tudi večino dela. Ne smemo pa pozabiti, da smo s tem šele pripravili ozadje za resnični namen našega sistema za obdelavo video signala. Vsa našeta strojna in programska oprema video signala namreč ne obdeluje, namenjena je le njegovi pretvorbi v digitalno obliko, pripravi sredstev, s katerimi bomo signal obdelovali, in prikazu obdelanega signala na monitorju. Za najosnovnejšo preslikovanje črno-belega slikovnega signala iz vhoda na računalniški monitor potrebujemo najmanj še tri module VHDL, ki bodo tvorili ogrodje našega projekta. Temu ogrodju bomo potem dodali več modulov, ki bodo omogočali tudi prikaz barvne slike in dejansko obdelavo video signala.

### 7.1 Seznam modulov VHDL

Za naš sistem smo pripravili 7 modulov VHDL, ki predstavljajo osnovo za uporabo sistema in razvijanje drugih modulov za obdelavo video signala:

- Modul za krmiljenje D/A pretvornika za črno-bel prikaz na računalniškem monitorju
- Modul za dostop do pomnilnika s strani prikazovalnega in dekodirnega modula za črno-bel video signal
- Modul za črno-belo dekodiranje 8-bitnega video digitalnega toka po standardu BT.656
- Modul za krmiljenje D/A pretvornika za barvni prikaz na računalniškem monitorju
- Modul za dostop do pomnilnika s strani prikazovalnega in dekodirnega modula za barvni video signal
- Modul za barvno dekodiranje 8-bitnega video digitalnega toka po standardu BT.656
- Modul za demonstracijo obdelave video signala – poudarjanje robov v sliki

## **7.2 Modul za krmiljenje D/A pretvornika za črno-bel prikaz na računalniškem monitorju**

Modul za prikaz slike skrbi za pravilno sinhronizacijo petih signalov, potrebnih za prikaz slike na računalniškem monitorju. To so signali barv (rdeča – *R*, zelena – *G* in modra – *B*) in pa vodoravni ter navpični sinhronizacijski signal *h\_sync* in *v\_sync*, ki prožita začetek risanja nove vrstice oziroma nove slike. Ta modul je časovno kritičen – če ne deluje ves čas z enako frekvenco, če je frekvenca prenizka ali če so sinhronizacijski časi nenatančni, monitor slike ne bo mogel prikazati.

### **7.2.1 Krmiljenje računalniškega monitorja**

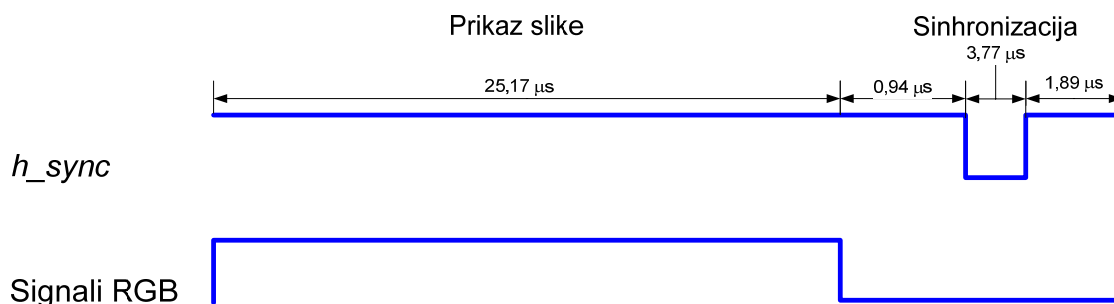
V osnovi bomo za prikaz uporabili standardni način prikaza VGA – 640 x 480 pik pri frekvenci osveževanja 60 Hz, ki ga podpirajo vsi standardni računalniški monitorji. V ta namen moramo najprej določiti potrebne čase sinhronizacije.



### 7.2.1.1 Vodoravna sinhronizacija

Risanje slikovne vrstice sestavljajo 4 faze [22]:

- Prikaz slike
- Zatemnitev pred sinhronizacijskim impulzom
- Vodoravni sinhronizacijski impulz
- Zatemnitev pred prikazom slike



**Slika 27: Sinhronizacijski časi za vrstice**

Vsaka od teh faz ima natančno določeno trajanje, ki se ga moramo držati, če želimo, da bo monitor ujel sinhronizacijo. Za prikaz slike imamo na voljo le prvo fazo, ki nam tako s svojim trajanjem ( $T_V$ ) določa povezavo med vodoravno ločljivostjo ( $R$ ) in hitrostjo pošiljanja podatkov ( $f_s$ ).

$$R = T_V \cdot f_s$$

#### Enačba 1: Povezava med vodoravno ločljivostjo in frekvenco pošiljanja podatkov

Pri opisu pomnilniškega modula smo že ugotovili, da je to najbolj ozko grlo našega sistema, saj ne dopušča frekvence branja in zapisovanja, ki bi bila kaj večja od 50 MHz. Na prvi pogled se zdi to veliko, vendar če želimo prikazovati barvno sliko, bomo tudi pri naši 6-bitni ločljivosti barvnih kanalov morali za vsako slikovno piko prebrati najmanj tri pomnilniške lokacije (najmanj zato, ker moramo v pomnilnik podatke tudi zapisovati). Frekvenca pretvorbe barvnih kanalov je torej navzgor omejena na največ 18 MHz, če pa želimo slikovne podatke z enako hitrostjo v pomnilnik tudi zapisovati, je hitrost pretvorbe omejena na slabih 9 MHz. Če vzamemo hitrost pošiljanja podatkov 9 MHz, smo našo vodoravno ločljivost tako omejili na približno 225 slikovnih pik.

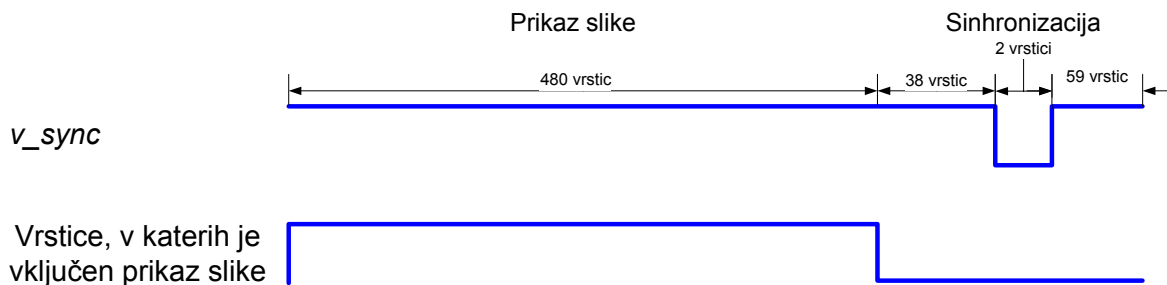
Določimo še sinhronizacijske čase, izražene v prehodih ure (9 MHz). Določimo jih enako kot ločljivost, z množenjem časa faze s frekvenco ure, potem pa jih zaokrožimo navzgor. Na koncu preverimo skupno trajanje vrstice, ki se mora čimbolj ujemati s prvotnim.

Faza	Trajanje ( $\mu\text{s}$ )	Število prehodov ure
Prikaz slike	25,17	225
Zatemnitev pred sinhronizacijo	0,94	8
Sinhronizacijski impulz	3,77	34
Zatemnitev pred prikazom slike	1,89	18
<b>Skupni čas vrstice (<math>\mu\text{s}</math>)</b>	<b>31,77</b>	<b>31,66</b>

Tabela 3: Sinhronizacijske vrednosti za vrstice

### 7.2.1.2 Navpična sinhronizacija

Ko določimo sinhronizacijske čase za vrstice, je čas, da določimo še sinhronizacijo celotne slike. Monitor v izbranem načinu prikazovanja prikazuje 480 vrstic, česar ne moremo spreminjati. Trajanje sinhronizacijskih faz tokrat merimo kar v vrsticah.



Slika 28: Potek navpične sinhronizacije

Naša ločljivost zdaj znaša 225 x 480 slikovnih pik. Da razmerje med širino in višino prikazanih predmetov ne bo preveč popačeno, bomo vsako drugo vrstico ponovili, kar pomeni, da bo naša končna ločljivost znašala 225 x 240 slikovnih pik, pri frekvenci osveževanja 60 Hz.

Še kratek komentar: izkaže se, da povprečen digitalni monitor le ni tako zelo izbirčen glede sinhronizacijskih časov. Pri enakem modulu sem lahko frekvenco ure spreminjal od 9 do 18 MHz, kar je preprosto spremenilo frekvenco osveževanja, ostala razmerja pa so ostala ista. Pomembna so torej le razmerja med trajanjem posameznih faz in pa omejitve za navpično in vodoravno osveževalno frekvenco.

## 7.2.2 Zgradba modula za prikaz črno-bele slike

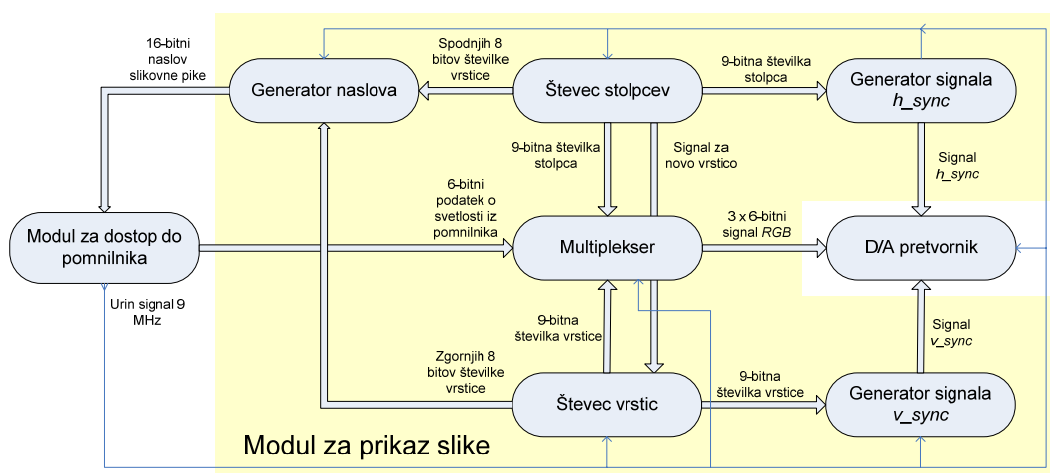
Modul za prikaz slike dobiva uro 9 MHz iz modula za dostop do pomnilnika, s katerim je povezan še z naslovnim in podatkovnim vodilom.

Za sinhronizacijo skrbita generatorja signalov  $h\_sync$  in  $v\_sync$ , ki krmilita sinhronizacijo glede na trenutno številko stolpca oziroma vrstice, ki ju dobita iz 9-bitnih števecov vrstic in stolpcev. Števec stolpcev poveča svojo vrednost za 1 ob vsakem prehodu ure in se postavi na ničlo ob koncu zadnje faze prikaza vrstice, števec vrstic pa poveča svojo vrednost ob koncu vsake vrstice. Na nič se postavi ob koncu zadnje faze prikaza slike.

Za 3 6-bitne signale  $R$ ,  $G$  in  $B$  skrbi multipleksor, ki daje na izhod trikrat ponovljeno 6-bitno vrednost (ta modul omogoča le črno-beli prikaz), ki jo je dobil iz modula za dostop do pomnilnika. Če vrednosti vrstice ali stolpca nista v obsegu, ko je dovoljen prikaz slike, postavi signale RGB na stanje 0 (zatemnitev).

Da dobimo iz modula za dostop do pomnilnika pravilno vrednost za svetlost slikovne pike, moramo na naslovno vodilo postaviti pravilen naslov. Naslove izračunavamo iz številke vrstice in stolpca. Spodnjih 8-bitov številke stolpca predstavlja spodnjih 8 bitov naslova, zgornjih 8-bitov številke vrstice pa predstavlja zgornjih 8 bitov naslova. Na ta način imamo v pomnilniku za prikaz slike zasedenih  $255 \times 255 = 65.535$  naslovov. Ker naša izhodna ločljivost znaša  $225 \times 240$  pik, je neuporabljenih 11.475 naslovov, kar znaša manj kot 20 % celote, za to ceno pa pridobimo izredno preprost (in hiter) način naslavljanja slike.

Ker za naslavljanje zanemarimo najmanj pomembni bit številke vrstice, bodo naslovi za vsako drugo vrstico enaki naslovom za prvo, kar pomeni, da bo enaka tudi narisana vrstica. Na ta način smo dosegli razpolovitev navpične ločljivosti, ki je potrebna zato, da se vsaj približno ohranijo razmerja med širino in višino na prikazani sliki.



Slika 29: Zgradba modula za črno-beli prikaz slike

### **7.3 Modul za 6-bitni dostop do pomnilnika**

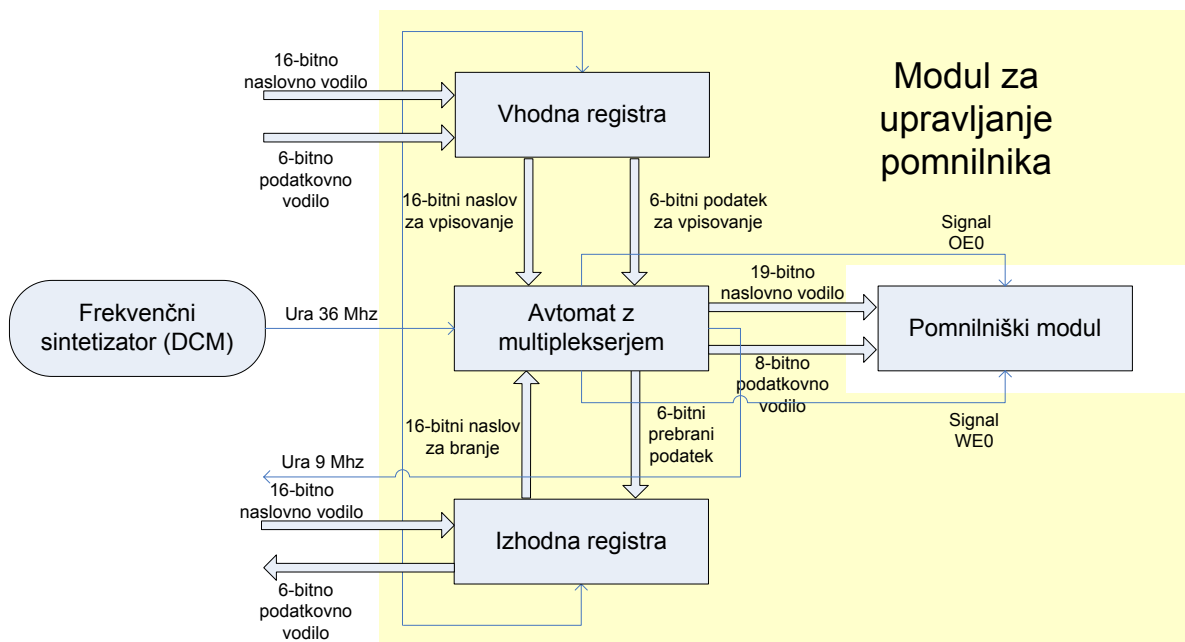
Za čimbolj preprosto načrtovanje je vedno priporočljivo ubrati modularni pristop, kjer si posamezne sklope naprave predstavljamo kot "črne škatle", ki imajo jasno definirane in čimbolj preproste vmesnike, podrobnosti njihovega delovanja pa so drugim delom sistema skrite. Lep primer za takšen način načrtovanja je naš modul za dostop do pomnilnika.

S sistemom je povezan s štirimi vodili in izhodom z uro (9 MHz) ter z vodili pomnilniškega modula. Ima naslovno vodilo za branje, podatkovno vodilo za branje, naslovno vodilo za vpisovanje in podatkovno vodilo za vpisovanje. Ob vsakem prehodu izhodne ure odda prebrani podatek iz naslova za branje, ki ga je prejel ob prejšnjem prehodu ure, in zapiše vhodno vrednost na naslov za vpisovanje.

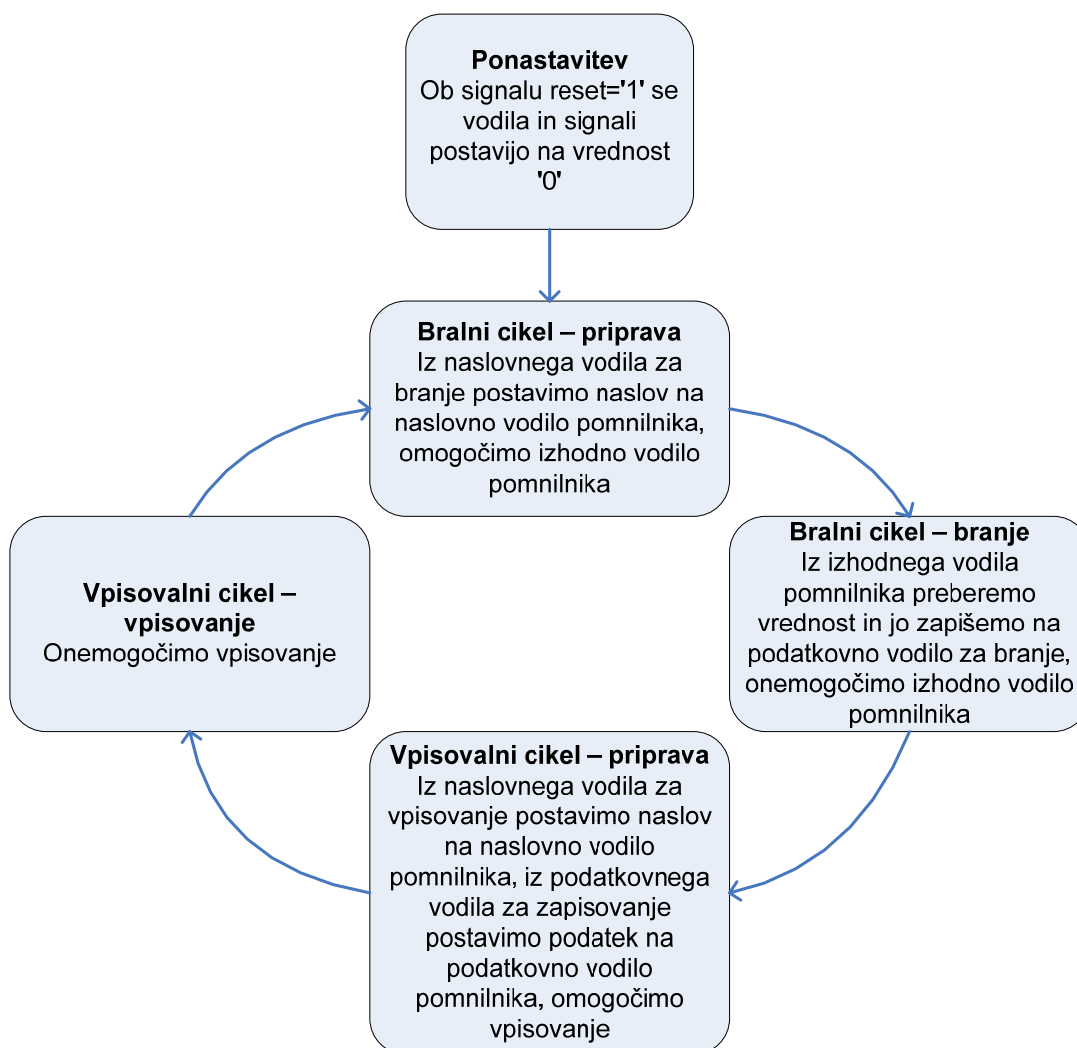
V enem ciklu izhodne ure preide skozi štiri stanja, zato ga poganjamo z uro frekvence 36 MHz, ki jo pridobimo iz enega od štirih frekvenčnih sintetizatorjev (DCM) v vezju FPGA Spartan 3. Kot vhodno uro za frekvenčni sintetizator smo vzeli izhod 20-MHz oscilatorja, ki je že vgrajen v sistemu Spartan 3. Izhodno uro 9 MHz ustvarimo tako, da dve zaporedni stanji držimo njen signal na visokem nivoju, dve pa na nizkem. S tem prihranimo en blok DCM.

S pomnilniškim modulom ga povezuje 19-bitno naslovno vodilo, 8-bitno podatkovno vodilo in po trije signali za omogočanje branja in pisanja iz posameznih integriranih vezij SRAM.

Ta blok deluje kot avtomat s štirimi stanji. Prvi dve sta namenjeni zapisovanju, drugi dve pa branju pomnilniške lokacije. V prvem stanju postavimo naslov iz vodila za naslov branja na naslovno vodilo pomnilniškega modula in vključimo signal za omogočanje branja iz prvega pomnilniškega modula. V drugem stanju preberemo podatke s podatkovnega vodila pomnilniškega modula, izključimo signal za omogočanje branja, vključimo signal izhodne ure in postavimo prebrane podatke na izhodno podatkovno vodilo. V tretjem stanju postavimo na naslovno vodilo pomnilniškega modula naslov iz vodila za naslov vpisovanja, na podatkovno vodilo pomnilniškega vodila postavimo podatke za vpisovanje in vključimo signal za vpisovanje. V četrtem stanju izključimo signal za vpisovanje in signal izhodne ure. Zdaj so v pomnilnik SRAM zapisani podatki na vhodnem vodilu, na izhodnem vodilu pa se nahaja zahtevana vsebina pomnilnika. Z naslednjim stanjem se postopek ponovi, tako da je modulom, priključenim na vhodna in izhodna vodila, na voljo ob vsakem ciklu izhodne ure.



Slika 30: Zgradba modula za upravljanje pomnilnika



Slika 31: Potek stanj avtomata za dostop do pomnilnika

## **7.4 Modul za dekodiranje črno-belega video signala**

Modul za dekodiranje črno-belega video signala sprejema 8-bitni video digitalni tok iz sistema za zajem slike in zvoka. Digitalni tok ima obliko v skladu s standardom BT.656, kar pomeni, da pritekajo podatki po 8-bitnem vodilu s frekvenco 27 MHz. Modul te podatke dekodira in pošlje podatke o svetlosti posameznih slikovnih pik na modul za nadzor pomnilnika, ki jih shrani v pomnilnik. Modul ima 8-bitno vhodno vodilo in vhod za uro iz sistema za zajem slike in zvoka, z modulom za nadzor pomnilnika pa komunicira prek 16-bitnega naslovnega in 6-bitnega podatkovnega vodila.

### **7.4.1 Standard BT.656**

Standard BT.656 [18] je nastal z namenom poenotenja digitalnih vmesnikov za video opremo. Predlagalo ga je združenje ITU-R (takrat še CCIR), trenutno pa je veljavna že njegova 4. različica. Obsega opis signalov na vodilu, specifikacije za oddajnik in sprejemnik na vodilu ter opis oblike digitalnega toka. Digitalni tok ima lahko tri oblike: 8-bitno paralelno, 10-bitno paralelno ali serijsko.

Naš sistem za zajem slike in zvoka podpira obe paralelni obliki, ne pa tudi serijske, poleg tega pa se ne držimo specifikacij za oddajnik in sprejemnik na vodilu. Sprejemnik in oddajnik v standardu sta namreč prilagojena za dolžine kablov, ki lahko znašajo do 200 m. Naš sistem mora podatke prenašati le kakih 10 cm daleč, zato je dovolj, da uporabimo običajne izhodne signale vezja ADV7185, ki preklapljajo med 0 in 3,3 V. Za ta projekt bomo uporabljali 8-bitno paralelno obliko digitalnega toka, saj preostali sistem ne zmore zagotoviti dovolj kakovostne obdelave slike (predvsem zaradi zmanjšane barvne ločljivosti na izhodu), da bi bilo smiselno izkoristiti kakovostnejšo 10-bitno obliko digitalnega toka, ki nam je sicer tudi na voljo.

#### **7.4.1.1 Opis 8-bitnega digitalnega toka po standardu BT.656**

Digitalno vodilo sestavlja 8-bitno podatkovno vodilo in ura s frekvenco 27 MHz. Na podatkovnem vodilu se ob vsaki pozitivni fronti ure (lahko tudi negativni – to nastavimo s parametrom PCLK v sistemu za zajem slike in zvoka) pojavi nov podatek. Podatki so v zapisu YCrCb 4:2:2, kar pomeni, da dobimo za vsako slikovno piko informacijo o svetlosti, informacijo o barvi pa le za vsako drugo slikovno piko.

Podatki si sledijo v zaporedju Cb1, Y1, Cr2, Y2, Cb3, Y3 itd. Par dveh podatkov (npr. Cb1, Y1) prestavlja eno slikovno piko, ki si tako sledijo s frekvenco 13,5 MHz. Ker dobimo

celotne informacije o barvi le za vsako drugo slikovno piko, moramo barve za posamezne slikovne pike pridobiti z interpolacijo iz najmanj dveh sosednjih slikovnih pik.

#### 7.4.1.2 Sinhronizacija digitalnega toka

Ob začetku in koncu vsake slikovne vrstice po vodilu prispe sinhronizacijski blok, ki ga sestavlja zaporedje štirih podatkov. Prvi trije so vedno enaki: \$FF, \$00, \$00, četrti (ID) pa nam pove, kateri sinhronizacijski blok je prispel. Sestavljajo ga en fiksni, trije podatkovni in štiri kontrolni biti.

Številka bita	Vsebina bita
7	Vedno 1
6	Bit za identifikacijo polslike (F)
5	Bit za identifikacijo zatemnitve (V)
4	Bit za identifikacijo začetka oz. konca vrstice (H)
3	Prvi kontrolni bit ( $P_0$ )
2	Drugi kontrolni bit ( $P_1$ )
1	Tretji kontrolni bit ( $P_2$ )
0	Četrti kontrolni bit ( $P_3$ )

Tabela 4: Vsebina identifikacijskega podatka v sinhronizacijskem bloku

Trije podatkovni biti nam povejo, katera polslika se pošilja (F), ali je vrstica zatemnjena (V) in ali blok označuje začetek ali konec (H) vrstice.

Bit	Vrednost	Pomen
<b>F</b>	0	Prva polslika
	1	Druga polslika
<b>V</b>	0	Normalna vrstica
	1	Zatemnjena vrstica
<b>H</b>	0	Začetek vrstice
	1	Konec vrstice

Tabela 5: Vrednosti podatkovnih bitov identifikacijskega podatka

Štirje kontrolni biti predstavljajo zaščito pred napakami v prenosu, določimo pa jih z naslednjimi formulami:

$$P_0 = V \oplus H$$

Enačba 2: Določanje vrednosti prvega kontrolnega bita [23]

$$P_1 = F \oplus H$$

Enačba 3: Določanje vrednosti drugega kontrolnega bita [23]

$$P_2 = F \oplus V$$

Enačba 4: Določanje vrednosti tretjega kontrolnega bita [23]

$$P_3 = F \oplus V \oplus H$$

Enačba 5: Določanje vrednosti četrtega kontrolnega bita [23]

### 7.4.1.3 Zgradba vrstic

Vrstice so sestavljene iz po dveh sinhronizacijskih blokov, bloka slikovnih podatkov in zatemnitvenega bloka na naslednji način:

\$FF \$00 \$00 ID	Cb Y Cr Y	\$FF \$00 \$00 ID	\$80 \$10 \$80 \$10
Sinhronizacija začetka vrstice, se ne ponovi	Slikovni podatki vrstice (720 slikovnih pik, 1440 podatkov)	Sinhronizacija konca vrstice, se ne ponovi	Zatemnitev (140 slikovnih pik, 280 podatkov)

Slika 32: Zgradba slikovne vrstice v digitalnem toku BT.656

### 7.4.1.4 Zgradba slikovnega polja

Vrstice celotne slike (720 x 576 slikovnih pik) so razdeljene na dve polsliki (720 x 288 slikovnih pik), od katerih prva polslika vsebuje lihe vrstice, druga pa sode. Celotno slikovno polje sestavljajo pod prejšnjo točko opisane vrstice, ki si sledijo po naslednjem ključu:

Številke vrstic	ID začetnega bloka	ID končnega bloka	Pomen vrstic
1–22	\$AB	\$B6	Prva zatemnitev prve polslike (22 vrstic)
23–310	\$80	\$9D	Prva polslika (288 vrstic)
311–312	\$AB	\$B6	Druga zatemnitev prve polslike (2 vrstici)
313–335	\$EC	\$F1	Prva zatemnitev druge polslike (22 vrstic)
336–623	\$C7	\$DA	Druga polslika (288 vrstic)
624–625	\$EC	\$F1	Druga zatemnitev druge polslike (2 vrstici)

Tabela 6: Razpored vrstic v slikovnem polju digitalnega toka po BT.656



## 7.4.2 Zgradba modula za dekodiranje signala

Osnovni modul za dekodiranje digitalnega toka sprejema vhodne podatke v osem 8-bitnih registrov, ki so povezani tako, da izhod prvega krmili vhod drugega, izhod drugega krmili vhod tretjega itd. – podobno kot pri shift registru, le da imamo tu opravka z celimi registri namesto posameznih bitov.

Zadnje štiri prejete podatke pošiljamo v modul za zaznavanje sinhronizacije, ki prepozna sinhronizacijske bloke za začetek vrstice, konec vrstice in začetek slikovnega polja. Tvori tri signale, ki krmilijo števec stolpcev in vrstic.

Števec stolpcev se poveča ob vsakih treh prejetih slikovnih pikah, to je 6 prejetih podatkov. To pomeni, da imamo vodoravno ločljivost enako tretjini vodoravne ločljivosti video signala PAL (720 slikovnih pik), kar znaša 240 slikovnih pik. Glede na to, da lahko na monitorju prikažemo le sliko širine 225 slikovnih pik, bo odrezanih 15 desnih stolpcev vhodne slike. Ob sprejemu signala za konec vrstice preklopi na vrednost 255, kjer ostane, dokler ne prejme signala za začetek vrstice, ko se ponastavi na vrednost 0.

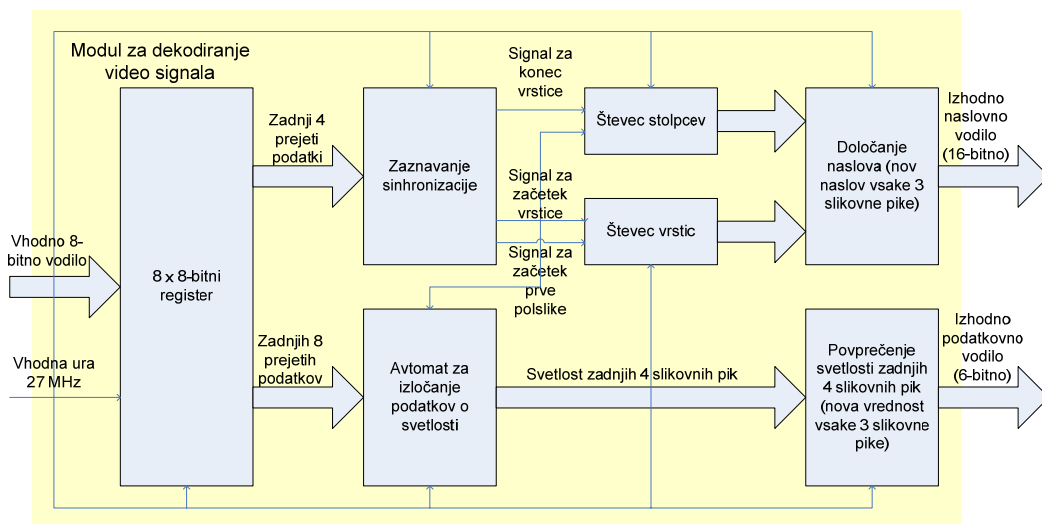
V tem trenutku se števec vrstic poveča za 1. Ko števec vrstic doseže vrednost 255, se ustavi. Števec vrstic se ponastavi, ko prejme signal za začetek slikovnega polja. Ker naša izhodna ločljivost znaša samo 240 vrstic, modul prebere le prvo polsliko, od katere potem ni mogoče prikazati zadnjih 48 vrstic.

Števca stolpcev in vrstic sta povezana z modulom za določanje naslova, ki enako kot pri modulu za prikaz preprosto združi po 8 bitov števec vrstic in stolpcev. Nastali naslov se pojavi na 16-bitnem izhodnem naslovnem vodilu.

Za dekodiranje podatkov o svetlosti je uporabljen avtomat z 8 stanji, ki se sinhronizira ob vsakem začetku vrstice, potem pa shranjuje podatke o svetlosti zadnjih 4 slikovnih pik v štiri registre, ki so povezani s povprečevalnikom. Svetlost slikovne pike je sorazmerna vrednosti Y v digitalnem toku, s to razliko, da lahko vrednost Y zasede le vrednosti od 16 do 240. Za naše potrebe bo dovolj, da vzamemo vrednost Y kot približek svetlosti slikovne pike in tako zmanjšamo kompleksnost dekoderja. S tem na kakovost bistveno ne vplivamo, še posebej če upoštevamo, da imamo za prikaz na voljo le 6 bitov. Vrednosti za barvo lahko zavržemo, saj jih za črno-bel prikaz ne potrebujemo.

Povprečevalnik sešteje svetlosti zadnjih 4 slikovnih pik in zavrže zadnje štiri bite rezultata, prvih 6 pa pošlje na 6-bitno izhodno podatkovno vodilo. Povprečenje zadnjih 4 vrednosti je izbrano zato, da lahko rezultat delimo s preprostim pomikom rezultata v desno. Vseeno se izhodni podatek obnovi vsake 3 slikovne pike, enako kot pri naslovnem vodilu.

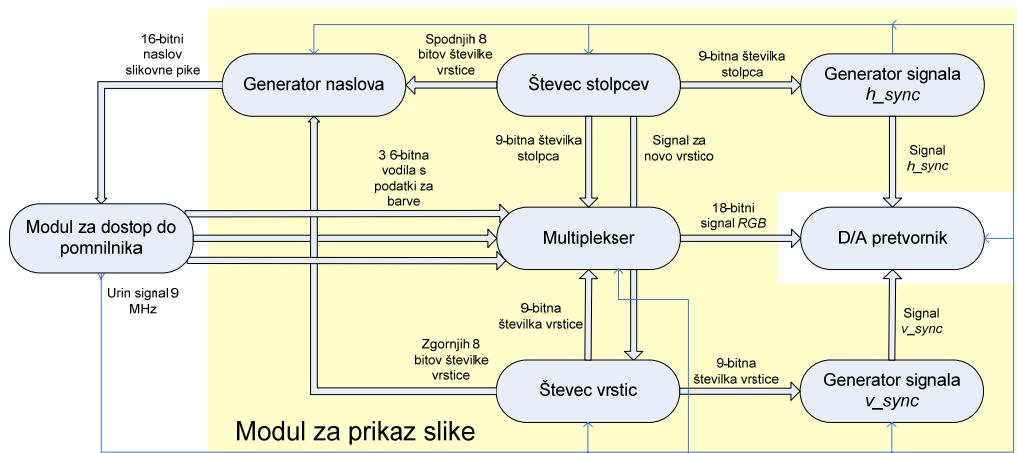
Frekvenca izhodnih podatkov je tako 4,5 MHz, kar je dvakrat manj kot pri modulu za dostop do pomnilnika, čeprav fazi gonilnih ur nista enaki. Ker fazi nista enaki, občasno modul za dostop do pomnilnika prevzame podatke v trenutku, ko se izhodno vodilo dekoderja še nastavlja. To je opazno kot pika, ki se ne ujema z okolico, vendar je hitrost osveževanja strani dovolj velika, razlika med frekvencami pa dovolj majhna, da te pike niso moteče.



Slika 33: Zgradba modula za dekodiranje video signala

## 7.5 Modul VHDL za barvni prikaz slike na računalniškem monitorju

Modul za prikaz barvne slike se od modula za prikaz črno bele slike bistveno razlikuje le v vhodnem podatkovnem vodilu. Namesto enega 6-bitnega vodila za svetlost ima 3 6-bitna vodila za rdečo, zeleno in modro barvo. Naslovno vodilo ostaja enako, enaki so tudi časi sinhronizacije in ura. Multiplekser več ne ponovi signala trikrat za vsak barvni kanal, ampak vsako barvo pošlje na svoj kanal pretvornika D/A.



Slika 34: Modul za prikaz barvne slike

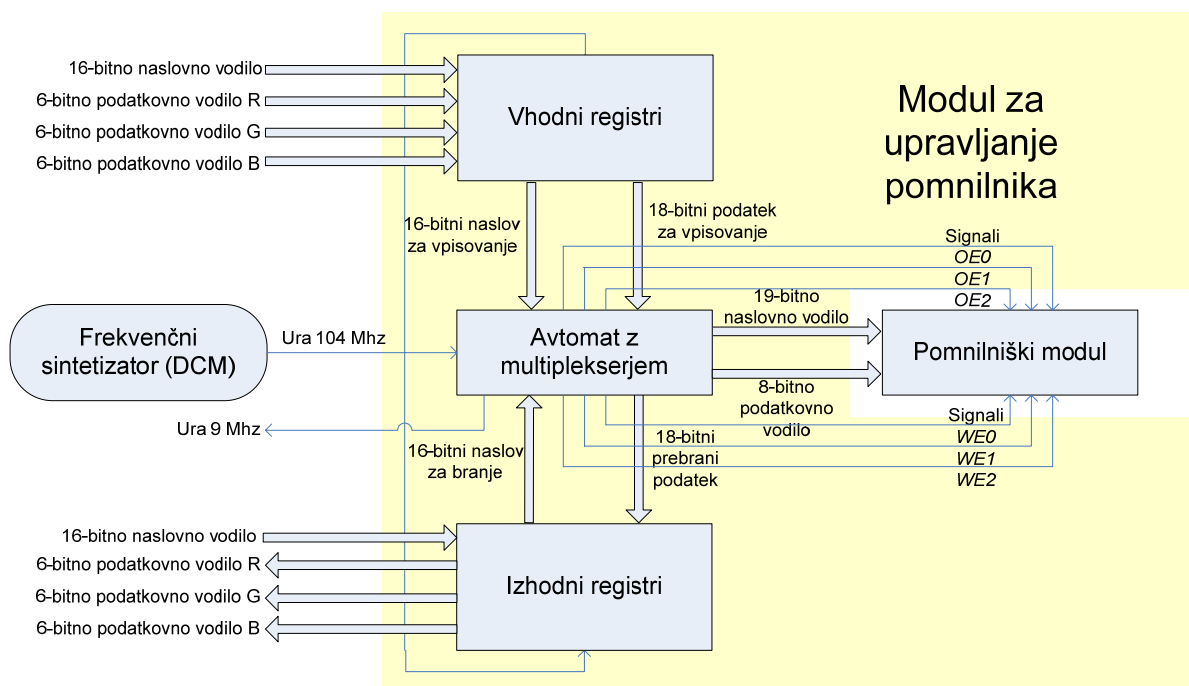
## **7.6 Modul VHDL za upravljanje pomnilniškega modula z razširjenim podatkovnim vodilom za delo z barvnim video signalom**

Osnovni modul VHDL za dostop do pomnilnika je omejen na eno 6-bitno vhodno in izhodno podatkovno vodilo. Za prikaz barvnega videa potrebujemo modul s po tremi vhodnimi in izhodnimi 6-bitnimi vodili.

Zaradi tega moramo v enem ciklu izhodne ure iz pomnilnika prebrati in zapisati po tri besede, kar pomeni, da mora imeti avtomat kar 12 stanj, vhodna ura pa mora imeti frekvenco kar 108 MHz. Glede na to, da traja cikel branja ene besede dve stanji, to pomeni, da je naš pomnilnik za okoli 8 % prek meje svoje deklarirane zmogljivosti. To deloma rešimo s tem, da naslovno vodilo nastavimo le enkrat za vseh šest bralnih oziroma pisalnih ciklov, posamezne besede pa potem izbiramo s signali za omogočanje branja oziroma vpisovanja, ki delujejo mnogo hitreje (6 ns namesto 12 ns). Ta pristop deluje, dokler ni potrebno zamenjati naslova na naslovnem vodilu, kar je potrebno storiti ob preklopi iz branja na vpisovanje in obratno. Deloma je mogoče težavo rešiti tako, da naslovno vodilo preklopimo eno stanje pred dejanskim nastopom stanja, katerega zadeva, vseeno pa je ta modul za upravljanje pomnilnika nestabilen in občasno prebere podatke iz napačnega naslova, medtem ko vpisovanje (vsaj za uporabljene pomnilniške module) deluje pravilno.

Za pravilno delovanje tega modula za dostop do pomnilnika bi potrebovali pomnilniške module, ki imajo čas postavitve naslovnega vodila krajši od 9 ns. Druga rešitev bi bila, da zmanjšamo hitrost branja podatkov, kar bi pomenilo zmanjšanje vodoravne ločljivosti celo pod 200 slikovnih pik, kar je verjetno prehuda omejitev. Uporabi zunanjšega pomnilniškega modula se žal ne moremo izogniti, saj je notranjega pomnilnika v vezju FPGA dovolj le za ca. 110 x 110 slikovnih pik, če sploh ne upoštevamo vseh registrov, ki so potrebni za delovanje drugih modulov. Zaenkrat se bomo torej morali sprijazniti z določeno mero nezanesljivosti pri branju pomnilnika, saj nam strojna oprema, ki je na voljo, druge rešitve ne dopušča.

Če bi imeli na voljo 16 vhodno-izhodnih signalov več, bi lahko drugače zasnovali pomnilniško kartico in do nje dostopali s 24-bitnim vodilom. Na ta način bi dosegli kar trikratno povečanje hitrosti prenosa podatkov iz pomnilnika in standardno ločljivost VGA, ki znaša 640 x 480 slikovnih pik. Žal pa bo ta rešitev morala počakati na zmogljivejšo različico sistema Spartan 3.



Slika 35: Modul za upravljanje pomnilnika z razširjenimi vhodno-izhodnimi vodili

## 7.7 Modul VHDL za barvno dekodiranje video signala

Razširitev ostalih modulov VHDL na delo z barvami je pomenila le skaliranje frekvence delovanja in širine vodil, pri barvnem dekodiranju video signala pa so potrebne globlje spremembe, saj smo za črno-belo dekodiranje informacije o barvi preprosto prezrli.

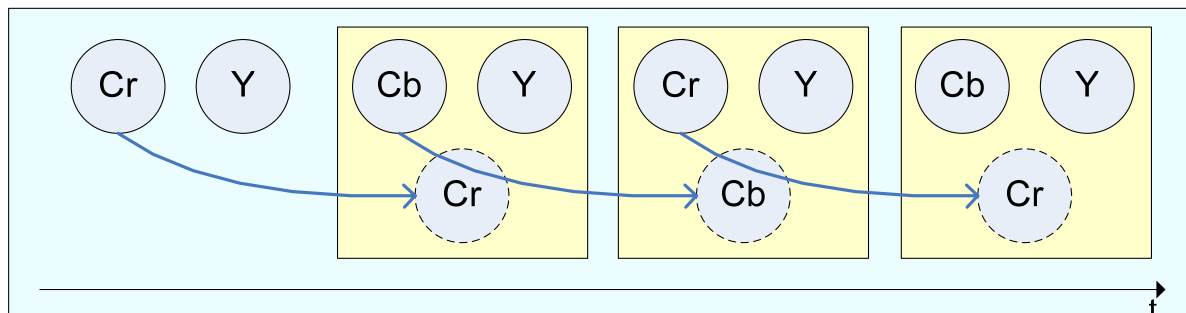
### 7.7.1 Dekodiranje barv iz digitalnega toka po standardu BT.656

Digitalni tok po standardu BT.656 prenaša informacije v zapisu YCrCb 4:2:2, kar pomeni, da dobimo za vsako slikovno piko celotno informacijo o njeni svetlosti in polovico informacij o njeni barvi. Vsako slikovno piko v digitalnem toku predstavlja par podatkov CbY ali CrY, za celotno informacijo o slikovni piki pa potrebujemo trojico YCrCb. Za uspešno dekodiranje moramo torej rekonstruirati manjkajoče informacije o barvi.

Naslednji korak je pretvorba barvnega prostora YCrCb v barvni prostor RGB, ki je potreben za prikaz slike na zaslonu

### 7.7.1.1 Rekonstrukcija barvnih informacij s kopiranjem

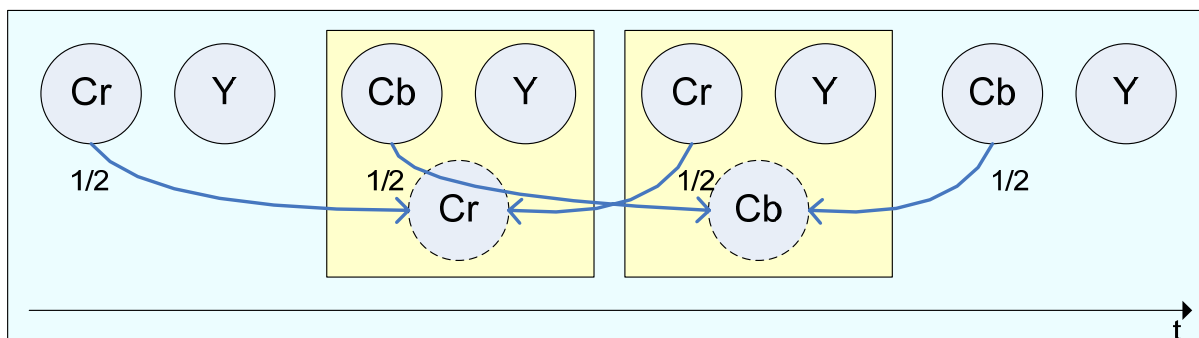
Najpreprostejši način rekonstrukcije barvnih informacij je, da jih preprosto prekopiramo iz prejšnjih slikovnih pik. Slabost tega načina rekonstrukcije je, da barvna informacija v tem primeru zaostaja za informacijo o svetlosti, kar povzroči nenaraven prikaz ostrih robov v sliki.



Slika 36: Rekonstrukcija barvne informacije s kopiranjem

### 7.7.1.2 Rekonstrukcija barvnih informacij z interpolacijo

Od kopiranja je boljša interpolacija, saj tukaj barvno informacijo računamo iz slikovnih pik pred in za želeno slikovno piko. Barvna informacija se zato časovno ujema s svetlostjo. Za to plačamo z večjo kompleksnostjo rekonstrukcije, saj moramo za vsako slikovno piko izračunati uteženo vsoto najmanj dveh podatkov, kar zahteva uporabo množilnikov. Najpogosteje se uporabljata interpolacija iz dveh ali štirih sosednjih slikovnih pik. Slikovne pike so utežene tako, da imajo največji vpliv tiste, ki so časovno najmanj oddaljene od tiste, za katero izračunavamo barvno informacijo.

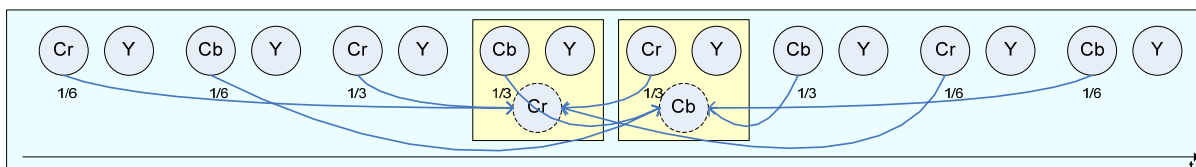


Slika 37: Interpolacija barvne informacije iz dveh sosednjih slikovnih pik

$$Cr_0 = \frac{1}{2}Cr_{-1} + \frac{1}{2}Cr_1$$

$$Cb_0 = \frac{1}{2}Cb_{-1} + \frac{1}{2}Cb_1$$

**Enačba 6: Enačba barvne interpolacije iz dveh sosednjih slikovnih pik**



**Slika 38: Interpolacija barvne informacije iz štirih sosednjih slikovnih pik**

$$Cr_0 = \frac{1}{6}Cr_{-2} + \frac{1}{3}Cr_{-1} + \frac{1}{3}Cr_1 + \frac{1}{6}Cr_2$$

$$Cb_0 = \frac{1}{6}Cb_{-2} + \frac{1}{3}Cb_{-1} + \frac{1}{3}Cb_1 + \frac{1}{6}Cb_2$$

**Enačba 7: Enačba barvne interpolacije iz štirih sosednjih pik**

### 7.7.1.3 Rekonstrukcija barvnih informacij s filtrom FIR

Če želimo zares visoko kakovost rekonstrukcije, interpolacijo razširimo na še več slikovnih pik. Takrat začnemo govoriti o paralelnem filtru FIR.

$$Cr_0 = (160 \cdot (Cr_{-1} + Cr_1) - 48 \cdot (Cr_{-3} + Cr_3) + 24 \cdot (Cr_{-5} + Cr_5) - 12 \cdot (Cr_{-7} + Cr_7) + 6 \cdot (Cr_{-9} + Cr_9) - 42 \cdot (Cr_{-11} + Cr_{11})) \cdot \frac{1}{256}$$

$$Cb_0 = (160 \cdot (Cb_{-1} + Cb_1) - 48 \cdot (Cb_{-3} + Cb_3) + 24 \cdot (Cb_{-5} + Cb_5) - 12 \cdot (Cb_{-7} + Cb_7) + 6 \cdot (Cb_{-9} + Cb_9) - 42 \cdot (Cb_{-11} + Cb_{11})) \cdot \frac{1}{256}$$

**Enačba 8: Primer paralelnega filtra FIR za barvno rekonstrukcijo [13]**

## 7.7.2 Pretvorba barvnega prostora YCrCb v barvni prostor RGB

Po rekonstrukciji barvnih informacij dobimo celotne informacije o slikovnih pikah v barvnem prostoru YCrCb. V večini aplikacij, tudi naši, se za končni prikaz slike uporablja barvni prostor RGB. Dekoder mora torej poleg interpolacije poskrbeti tudi za pretvorbo v barvni prostor RGB. Pretvorba poteka z matričnim množenjem in je odvisna od bitne širine digitalnega toka.

Izkaže se, da lahko izkoristimo simetrijo pretvorbe in namesto 7 uporabimo le 5 množilnikov.

$$R = 1,164 \cdot (Y - 16) + 1,596 \cdot (Cr - 128)$$

$$G = 1,164 \cdot (Y - 16) - 0,813 \cdot (Cr - 128) - 0,392 \cdot (Cb - 128)$$

$$B = 1,164 \cdot (Y - 16) + 2,017 \cdot (Cb - 128)$$

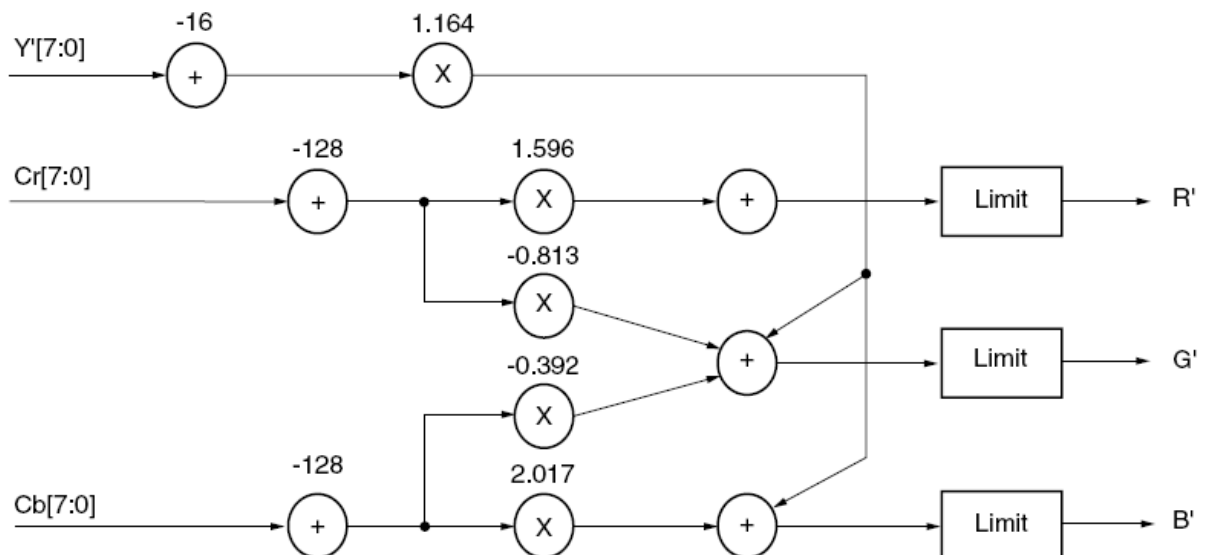
Enačba 9: Pretvorba barvnega prostora YCrCb v RGB za 8-bitni digitalni tok [12]

$$R = 1,164 \cdot (Y - 64) + 1,596 \cdot (Cr - 512)$$

$$G = 1,164 \cdot (Y - 64) - 0,813 \cdot (Cr - 512) - 0,392 \cdot (Cb - 512)$$

$$B = 1,164 \cdot (Y - 64) + 2,017 \cdot (Cb - 512)$$

Enačba 10: Pretvorba barvnega prostora YCrCb v RGB za 10-bitni digitalni tok [12]



Slika 39: Optimirana struktura vezja za pretvorbo iz barvnega prostora YCrCb v RGB [12]

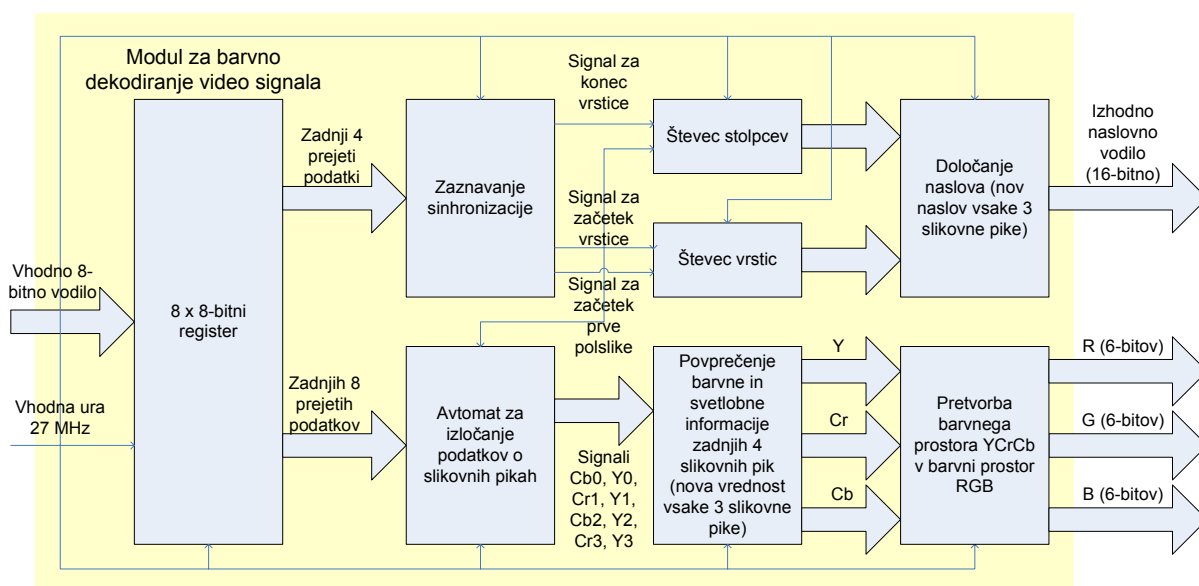
### 7.7.3 Zgradba modula VHDL za barvno dekodiranje video signala

Modul za barvno dekodiranje video signala uporablja enake podsestave za sinhronizacijo in določanje naslova kot črno-beli modul, bistvena sprememba je v avtomatu za izločanje podatkov o slikovnih pikah in modulu za povprečenje.

Avtomat za izločanje podatkov o slikovnih pikah zdaj pošilja povprečevalniku vseh osem zadnjih prejetih podatkov, razvrščene po registrih C0, Y0, C1, Y0, C2, Y2, C3, Y3, poleg tega pa še signal, ki pove, kateri registri C vsebujejo informacijo Cr, kateri pa Cb. Ta signal je potreben, ker se povprečenje dogaja vsake tri slikovne pike, kar pomeni, da so ob vsakem povprečenju vrednosti Cr in Cb zamenjane.

Povprečevalnik še vedno povpreči podatke os svetlosti za zadnje 4 slikovne pike, tokrat pa povpreči tudi barvne informacije in jih zapiše v izhodna registra Cb in Cr glede na stanje signala vsebine registrov C. Na ta način dobimo na izhodu tri 8-bitne signale Y, Cb in Cr.

Zdaj je na vrsti modul za pretvorbo barvnega prostora. Sestavlja ga 6 seštevalnikov in pet množilnikov, pri čemer smo uporabili vgrajene 18-bitne množilnike vezja FPGA Spartan 3. Izhodni signali RGB se potem omejijo na maksimalno vrednost 255, da ne pride do prelivov, ki bi lahko nastali zaradi računskih napak, na izhodna vodila pa je poslanih le zgornjih 6 bitov.



Slika 40: Zgradba modula VHDL za barvno dekodiranje video signala



## **7.8 Modul VHDL za demonstracijo obdelave slike**

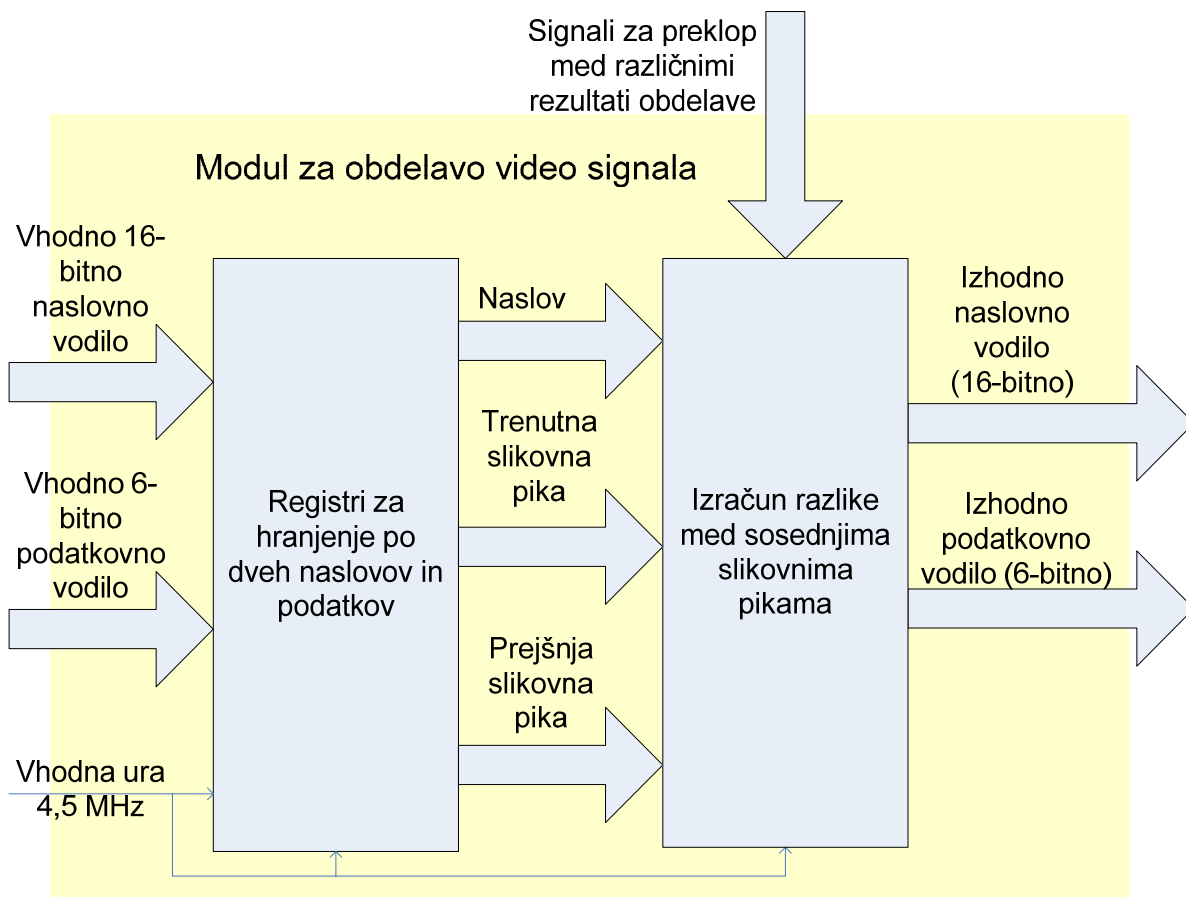
Eden temeljnih namenov našega sistema je preučevanje tehnik za obdelavo slike. V ta namen smo pripravili preprost modul, ki je zamišljen kot vzorec za načrtovanje podobnih modulov za obdelavo slike. Načrtovan je tako, da ga vstavimo med dekodirni in pomnilniški modul za črno-bel video signal, uro pa dobiva iz dekodirnega modula. Spreminjanje ostalih modulov ni potrebno, kar hkrati tudi potrjuje uporabnost naše modularne zasnove. Operacije, ki jih izvaja, lahko izbiramo z digitalnimi vhodi, ki jih lahko krmilimo prek vmesnika USB.

Za ostali sistem je videti kot medpomnilnik vrste FIFO, saj sprejete slikovne pike obdela in pošlje pod istim naslovom naprej v pomnilnik.

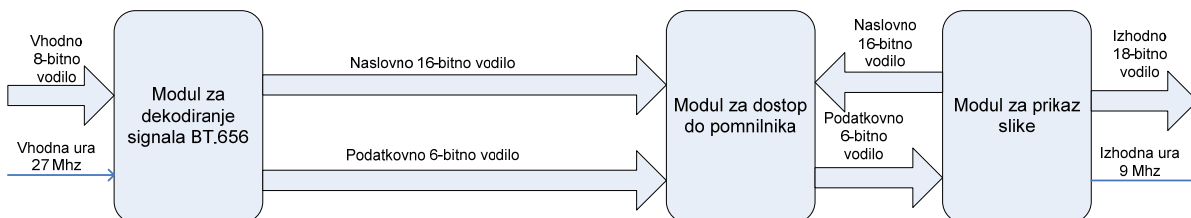
Sama obdelava slike je zelo preprosta, saj preprosto gledamo razliko v svetlosti zaporednih slikovnih pik. Z vhodnimi signali, ki jih pošiljamo prek vodila USB, lahko potem določimo kateri podatki naj se odpošljejo v pomnilnik. Ena možnost je, da pošljemo nespremenjen podatek o svetlosti – prikaže se nespremenjena vhodna slika. Druga možnost je, da pošiljamo v pomnilnik samo povečan podatek o razliki med sosednjima slikovnima pikama, kar nam približno pokaže položaj in ostrino robov v sliki. Tretja možnost je, da pošljemo na izhod seštevek podatka o svetlosti in razlike med sosednjima pikama. Tako dobimo na zaslonu vhodno sliko, ki ima poudarjene robove.

Ta modul služi le kot demonstracija, ki kaže princip umeščanja modulov za obdelavo signala v sistem. Če želimo pripraviti modul za zahtevnejšo obdelavo, moramo v modulu za obdelavo shraniti večji del prikazane slike. Za to lahko uporabimo blokovni pomnilnik RAM, ki je na voljo v vezju FPGA Spartan 3 in ga je dovolj za lokalne tehnike obdelave (2D konvolucija itd.), kjer za obdelavo ne potrebujemo shranjevanja celotne slike.

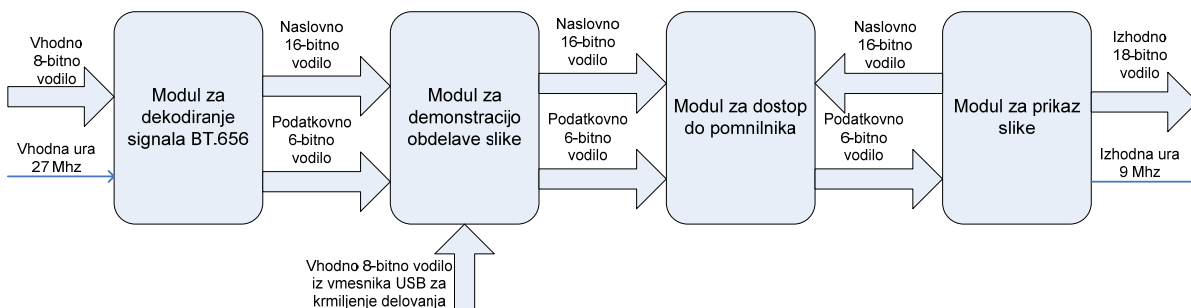
Pri uporabljeni ločljivosti lahko v blokovni pomnilnik RAM shranimo do okoli 160 vrstic črno-bele slike oziroma okoli 50 vrstic barvne slike. Za zahtevnejše tehnike obdelave slike, kjer moramo imeti shranjeno celotno sliko, bi morali modulu za upravljanje pomnilnika dodati še en komplet vhodno-izhodnih vodil, kar bi ga upočasnilo še za polovico in tako še bolj zmanjšalo že tako omejeno ločljivost. Zaključimo lahko torej, da nam strojna oprema takšne obdelave slike praktično ne dopušča.



Slika 41: Zgradba demonstracijskega modula za obdelavo video signala



Slika 42: Zgradba sistema za prenos črno-belega video signala iz vhoda na izhod



Slika 43: Zgradba sistema z dodanim modulom za obdelavo video signala

## 8 Zaključek

---

Ob razvoju opisanega sistema sem dobil veliko praktičnih izkušenj. Podobno kot moji kolegi pri svojih projektih sem imel največ težav s hitrostjo dostopa do pomnilnika. To je glavno ozko grlo pri večini projektov z rekonfigurabilnimi vezji in na koncu tisti člen, ki velikokrat drastično omeji končno zmogljivost sistema. Pri našem sistemu je omejil izhodno ločljivost prikaza slike na računalniškem monitorju, kateri sem potem seveda prilagodili ločljivost celotnega sistema.

Naslednja velika omejitev je omejitev števila vhodno/izhodnih signalov vezja FPGA, ki so bili na voljo na priključkih sistema. Zaradi te omejitve smo morali izhodno ločljivost barv omejiti na 18 (3 x po 6) bitov. Prav tako smo morali uporabiti ožja vodila za dostop do pomnilnika RAM, kar je za faktor 3 (za vsako barvno slikovno piko potrebujemo branje treh lokacij pomnilnika RAM) zmanjšalo hitrost dostopa do pomnilnika, hkrati pa povečalo porabo sistema, saj so pomnilniška vezja vedno vključena.

Na drugi strani pa je sistem, če upoštevamo omejeno ločljivost, še dokaj neizkoriščen. Končna poraba logičnih modulov vezja FPGA v sistemu za demonstracijo obdelave video signala ne presega 25 %. Obdelava dekodiranega video signala poteka s hitrostjo 4,5 MHz, kar je daleč pod dosegljivo hitrostjo. Porabili nismo tudi nič blokovnega pomnilnika RAM. Obdelava signala je torej lahko bistveno bolj kompleksna od prikazane.

Spregovorimo še o zmogljivosti sistema glede na njegovo porabo ceno. Celotni sistem je v majhni seriji mogoče izdelati za okoli 250 €, njegove posamezne komponente pa so zasnovane univerzalno in bodo uporabne tudi za druge projekte, kar stroške še dodatno zmanjša. Za doseženo prilagodljivost in zmogljivost ta cena ni prevelika.

Tudi poraba sistema ni pretirana, saj za celotni sistem med delovanjem znaša okoli 500 mA oz. 3 W pri napajalni napetosti 6 V. V primerjavi s porabo računalniškega monitorja je to zanemarljivo.

Naš končni zaključek je, da je projekt uspel in da je izdelani sistem kljub omejitvam primerno orodje za učenje tehnik obdelave video signala. Na tržišču je sicer še mnogo drugih podobnih rešitev, ki pa večinoma ciljajo le posamezne člene v verigi pretvorbe video signala. Celovitost in univerzalna uporabnost sistema sta njegovi pomembni prednosti, zato upamo, da bo čim več uporabnikom pomagal pri odkrivanju skrivnosti načrtovanja tovrstnih sistemov.

## 9 Literatura

---

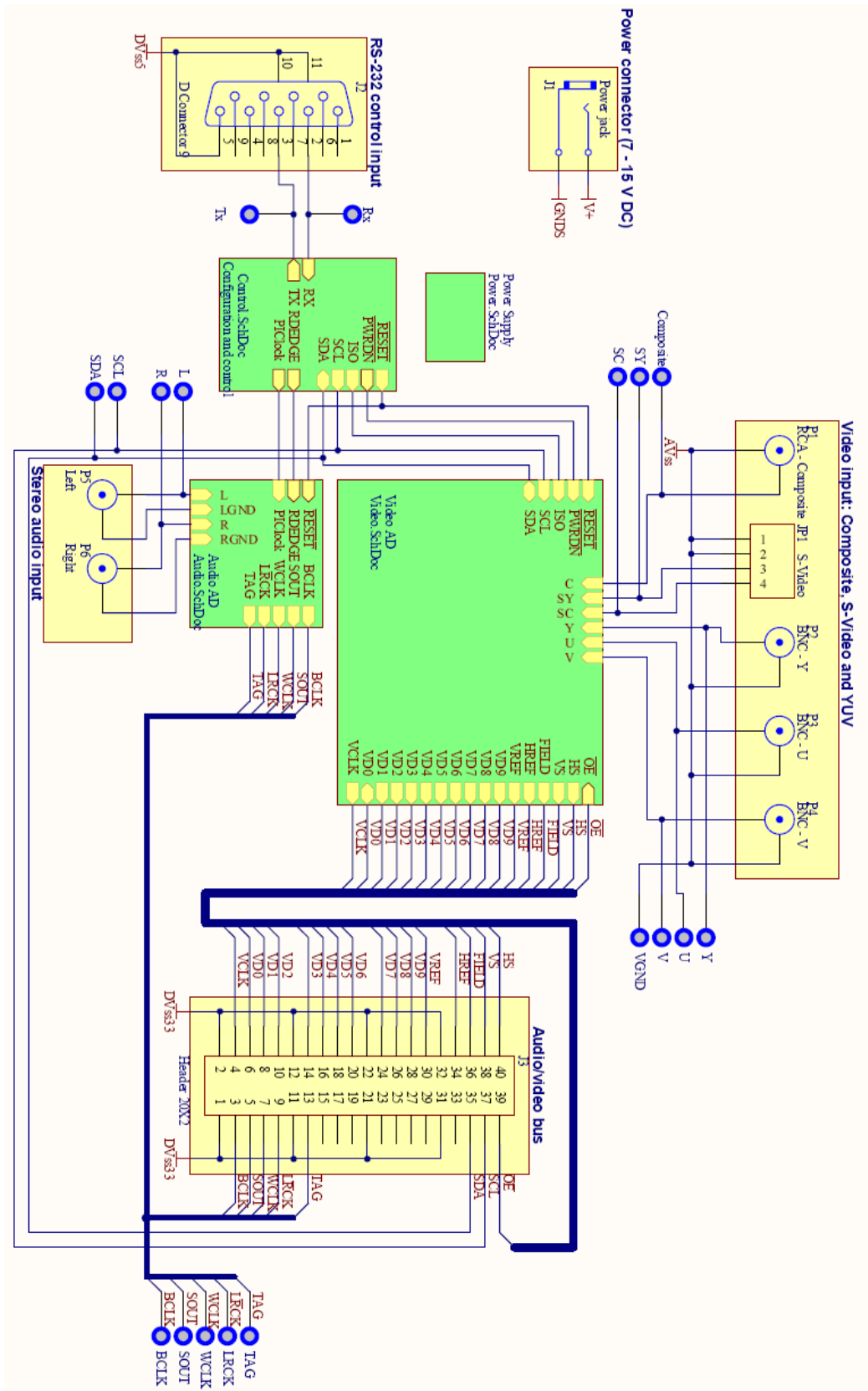
- [1] Reiner Hartenstein, "The digital divide of computing", Proceedings of the first conference on computing frontiers, Ischia, Italy, 357–362, 2004.
- [2] Analog Devices, "ADV7185 - Professional NTSC/PAL Video Decoder with 10-bit CCIR656 output", Dokumentacija, [www.analog.com](http://www.analog.com), 2002.
- [3] Analog Devices, "AD1870 – Single Supply 16-Bit  $\Sigma$  -  $\Delta$  Stereo ADC", dokumentacija, [www.analog.com](http://www.analog.com), 2002.
- [4] Microchip Technology, "PIC16F84A Data Sheet", dokumentacija, [www.microchip.com](http://www.microchip.com), 2003.
- [5] National Semiconductor, "LM1117 800mA Low-Dropout Linear Regulator", dokumentacija, [www.national.com](http://www.national.com), 1999.
- [6] National Semiconductor, "LM2937, 500 mA Low Dropout Regulator", dokumentacija, [www.national.com](http://www.national.com), 2000.
- [7] Maxim, "MAX220-MAX249 +5V-Powered, Multichannel RS-232 Drivers/Receivers", dokumentacija, [www.maxim-ic.com](http://www.maxim-ic.com), 2003.
- [8] Maxim, "MAX803L/M/R/S/T/Z, MAX809J/L/M/R/S/T/Z, MAX810L/M/R/S/T/Z 3-Pin Microprocessor Reset Circuits", dokumentacija, [www.maxim-ic.com](http://www.maxim-ic.com), 2001.
- [9] Xilinx, "Spartan-3 FPGA Family: Complete Data Sheet", dokumentacija, [www.xilinx.com](http://www.xilinx.com), 2005.
- [10] Xilinx, "Using Digital Clock Managers (DCMs) in Spartan-3 FPGAs", dokumentacija, [www.xilinx.com](http://www.xilinx.com), 2003.
- [11] Xilinx, "Digital Video & Image Processing – Xilinx Solutions for the Broadcast Chain", predstavitev, [www.xilinx.com](http://www.xilinx.com), 2004.
- [12] Xilinx, "Color Space Converter: Y'CrCb to R'G'B'", dokumentacija, [www.xilinx.com](http://www.xilinx.com), 2005.
- [13] Xilinx, "Digital Component Video Conversion 4:2:2 to 4:4:4", dokumentacija, [www.xilinx.com](http://www.xilinx.com), 2001.
- [14] Alliance Semiconductor, "AS7C4096, AS7C34096, 5V/3.3V 512K  $\times$  8 CMOS SRAM", dokumentacija, [www.alsc.com](http://www.alsc.com), 2002.
- [15] Intersil Corporation, "HI5628 8-Bit, 165/125/60MSPS, Dual High Speed CMOS D/A Converter", dokumentacija, [www.intersil.com](http://www.intersil.com), 1999.
- [16] HI-TECH Software, "PICC Manual", dokumentacija, [www.htsoft.com](http://www.htsoft.com), 2002.
- [17] Intel, "Intel® Pentium® Processor Extreme Edition 840", dokumentacija, [www.intel.com](http://www.intel.com), 2005.
- [18] ITU, "ITU-R Recommendation 656", standard, Recommendations of the CCIR, Volume XI - Part 1, Stran 105 do 117, 1990.
- [19] Primož Šalič, "Generator slike za računalniški monitor", diplomsko delo, Fakulteta za elektrotehniko Univerze v Ljubljani, 2003.
- [20] Marko Mlakar, "Konfiguriranje FPGA vezij preko USB vmesnika", diplomsko delo, Fakulteta za elektrotehniko Univerze v Ljubljani, 2005.
- [21] Wikipedia, "PAL", članek v spletni enciklopediji, <http://en.wikipedia.org/wiki/PAL>, 2005.
- [22] Tomi Engdahl, "VGA timing information", spletna stran, [http://www.epanorama.net/documents/pc/vga\\_timing.html](http://www.epanorama.net/documents/pc/vga_timing.html), 2002.
- [23] Spacewire UK, "A Brief Introduction to Digital Video", spletna stran, [http://www.spacewire.co.uk/video\\_standard.html](http://www.spacewire.co.uk/video_standard.html), 2005.

## 10 Izjava

---

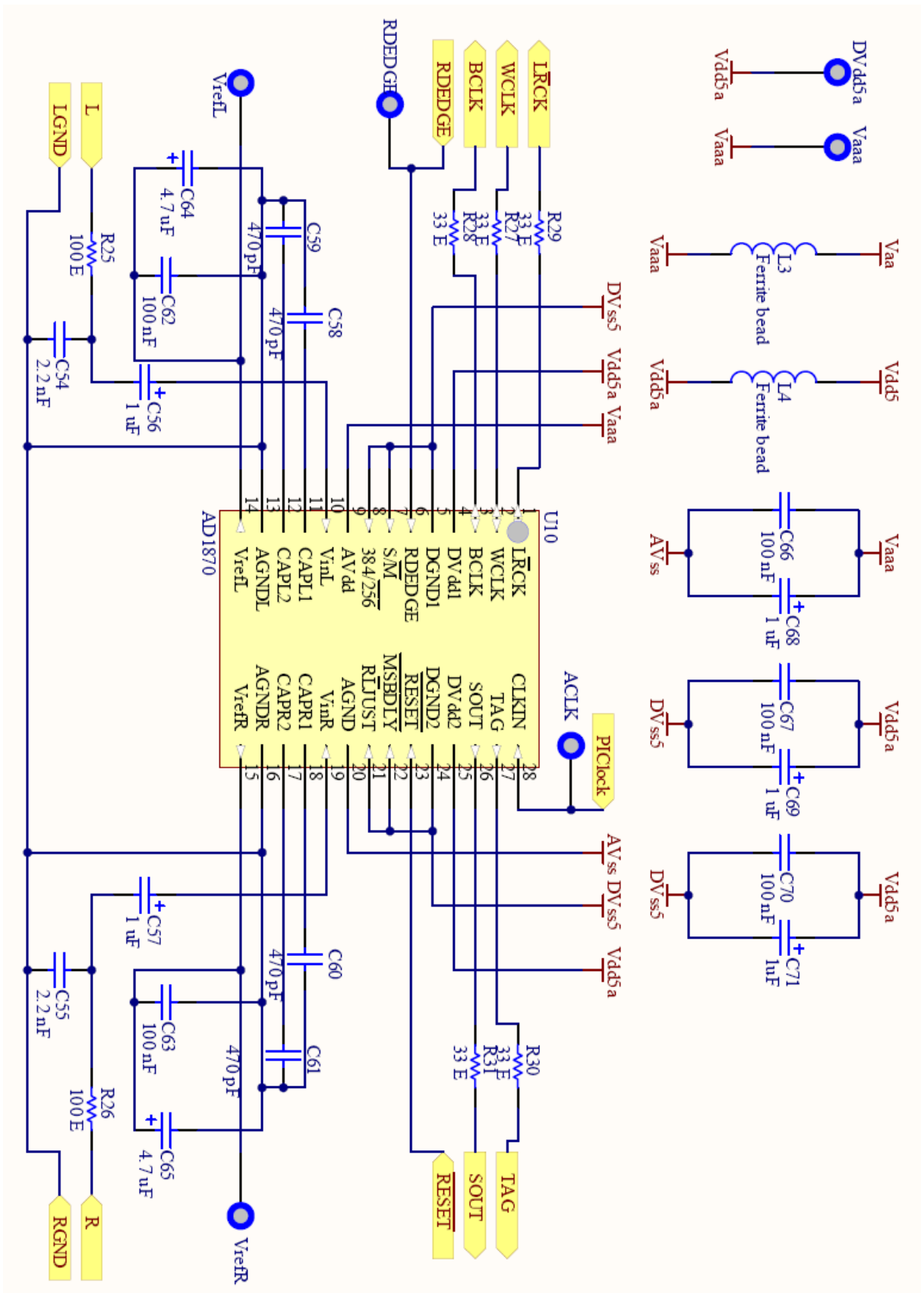
Izjavljam, da sem magistrsko delo izdelal samostojno pod vodstvom mentorja **doc. dr. Andreja Trosta**. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

# 11 Priloge



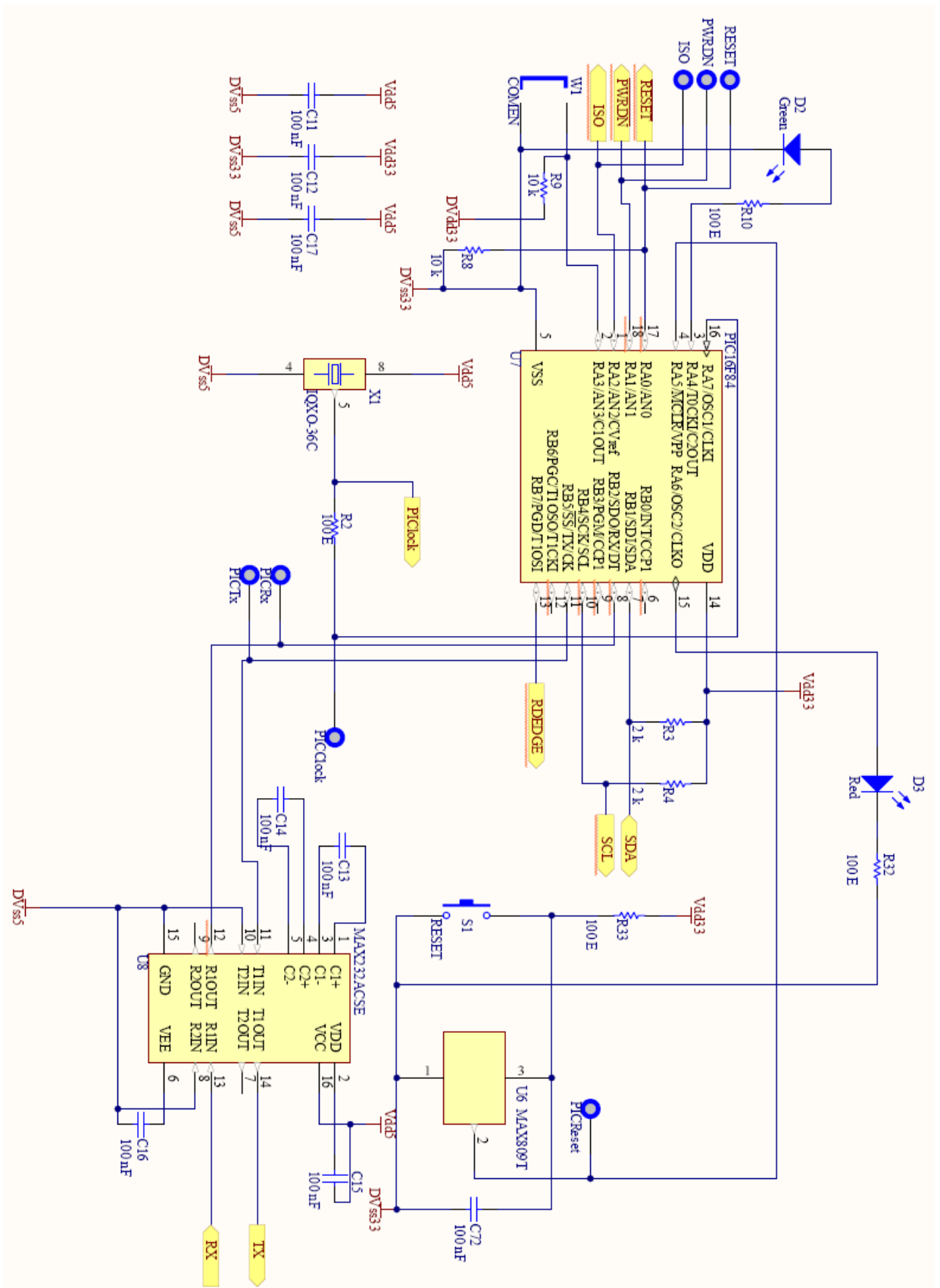
Priloga 1: Povezave med sklopi sistema in priključki



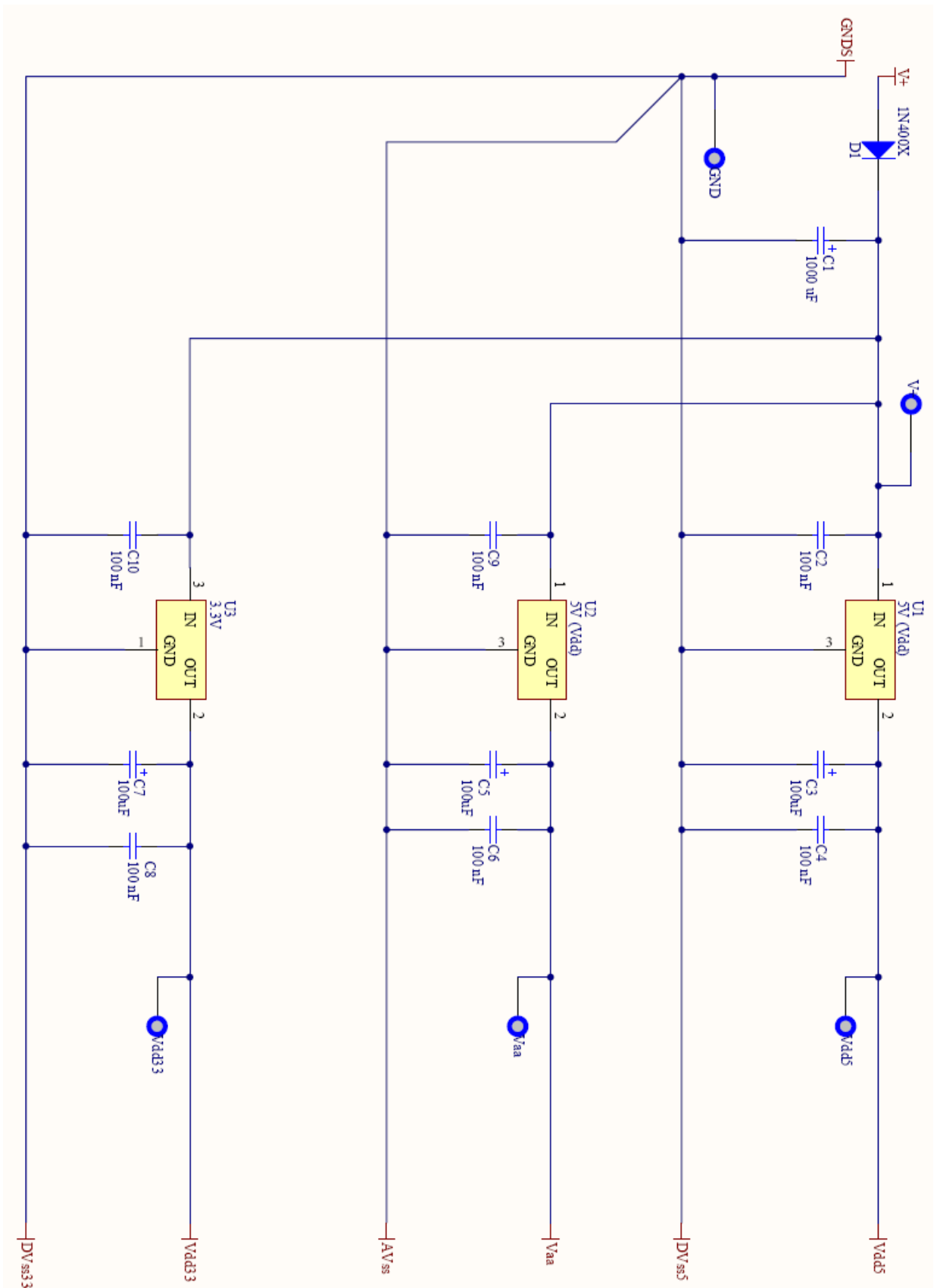


Priloga 3: Shema vezja AD1870, priključkov in zunanjih komponent

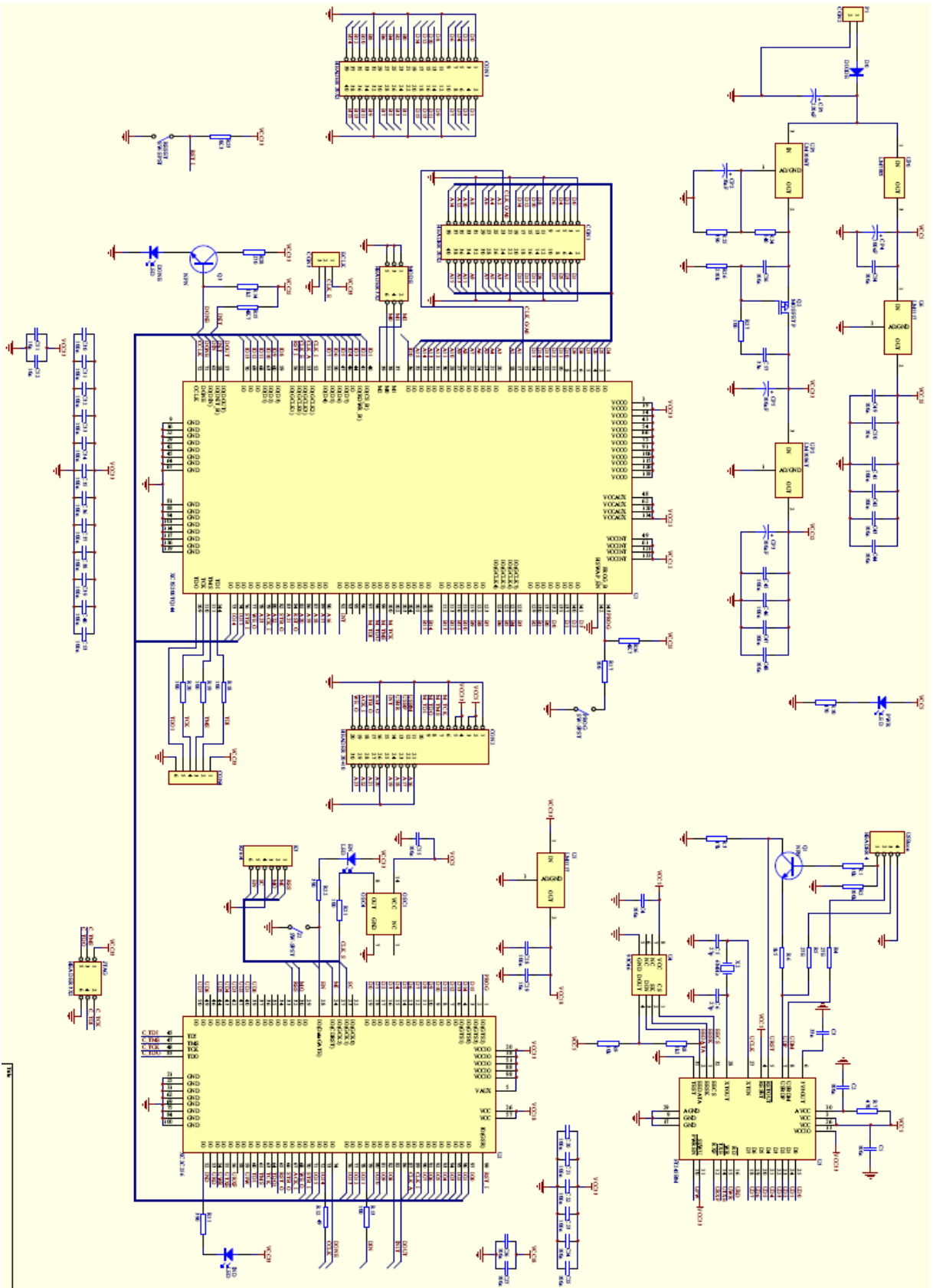




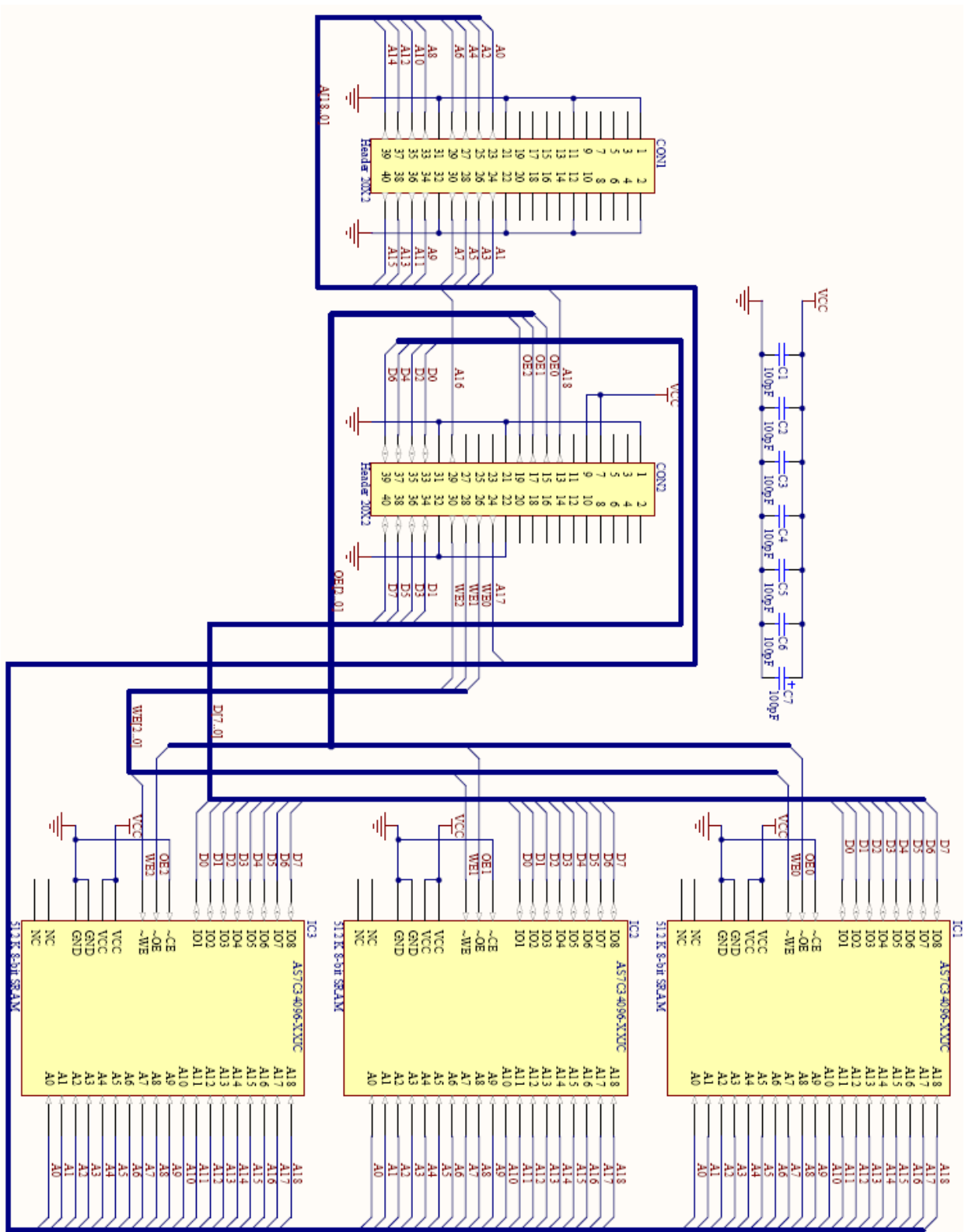
Priloga 4: Shema krmilnika, pretvornika nivojev, ponastavitvenega vezja in oscilatorja



**Priloga 5: Načrt napajalnega sklopa**



Priloga 6: Načrt sistema Spartan 3 (avtor doc. dr. Andrej Trost)



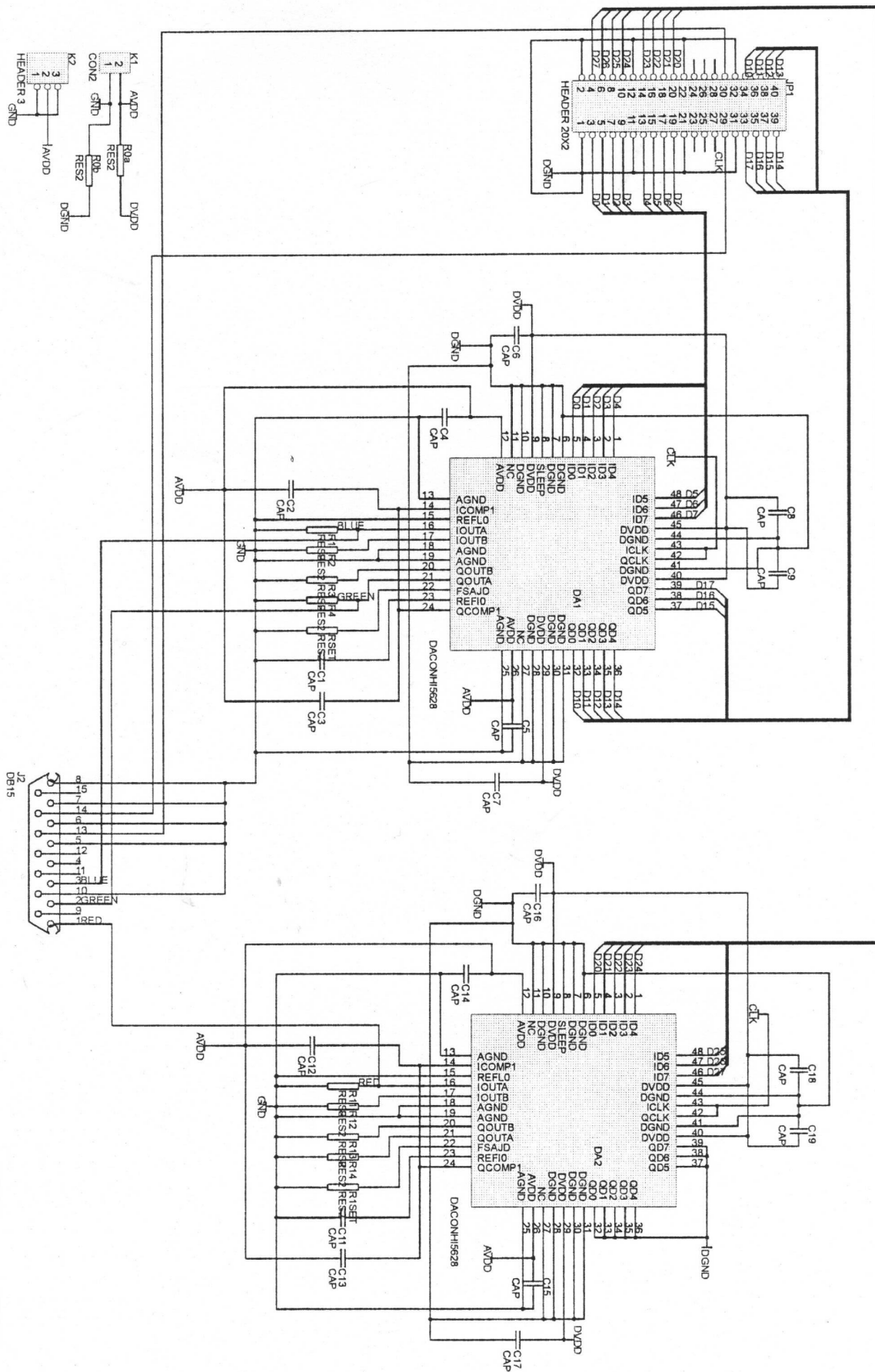
Priloga 7: Načrt pomnilniškega razširitvenega modula

```

NET "Address_bus<0>" LOC = "P15" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<10>" LOC = "P28" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<11>" LOC = "P30" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<12>" LOC = "P31" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<13>" LOC = "P32" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<14>" LOC = "P33" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<15>" LOC = "P35" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<16>" LOC = "P92" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<17>" LOC = "P90" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<18>" LOC = "P100" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<1>" LOC = "P17" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<2>" LOC = "P18" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<3>" LOC = "P20" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<4>" LOC = "P21" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<5>" LOC = "P23" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<6>" LOC = "P24" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<7>" LOC = "P25" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<8>" LOC = "P26" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Address_bus<9>" LOC = "P27" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<0>" LOC = "P84" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<1>" LOC = "P85" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<2>" LOC = "P82" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<3>" LOC = "P83" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<4>" LOC = "P79" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<5>" LOC = "P80" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<6>" LOC = "P77" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "Data_bus<7>" LOC = "P78" | IOSTANDARD = LVCMOS33 | DRIVE = 24 ;
NET "OE0" LOC = "P99" | IOSTANDARD = LVCMOS33 | DRIVE = 2 ;
NET "OE1" LOC = "P98" | IOSTANDARD = LVCMOS33 | DRIVE = 2 ;
NET "OE2" LOC = "P97" | IOSTANDARD = LVCMOS33 | DRIVE = 2 ;
NET "WE0" LOC = "P89" | IOSTANDARD = LVCMOS33 | DRIVE = 2 ;
NET "WE1" LOC = "P87" | IOSTANDARD = LVCMOS33 | DRIVE = 2 ;
NET "WE2" LOC = "P86" | IOSTANDARD = LVCMOS33 | DRIVE = 2 ;

```

**Priloga 8: Definicija priključitve pomnilnika na vezje FPGA v obliki *.ucf***



Priloga 9: Načrt D/A pretvornika za krmiljenje računalniškega monitorja (avtor Primož Šalič)

```
NET "out_data<0>"      LOC = "p59"  ;
NET "out_data<1>"      LOC = "p60"  ;
NET "out_data<2>"      LOC = "p63"  ;
NET "out_data<3>"      LOC = "p68"  ;
NET "out_data<4>"      LOC = "p69"  ;
NET "out_data<5>"      LOC = "p70"  ;
NET "out_data<6>"      LOC = "p73"  ;
NET "out_data<7>"      LOC = "p74"  ;
NET "in_data<0>"       LOC = "p36"  ;
NET "in_data<1>"       LOC = "p40"  ;
NET "in_data<2>"       LOC = "p41"  ;
NET "in_data<3>"       LOC = "p44"  ;
NET "in_data<4>"       LOC = "p46"  ;
NET "in_data<5>"       LOC = "p47"  ;
NET "in_data<6>"       LOC = "p50"  ;
NET "in_data<7>"       LOC = "p51"  ;
```

**Priloga 10: Definicije povezav za podatkovni vodili, ki jih lahko beremo s programom za programiranje vezja FPGA, v obliki *.ucf***