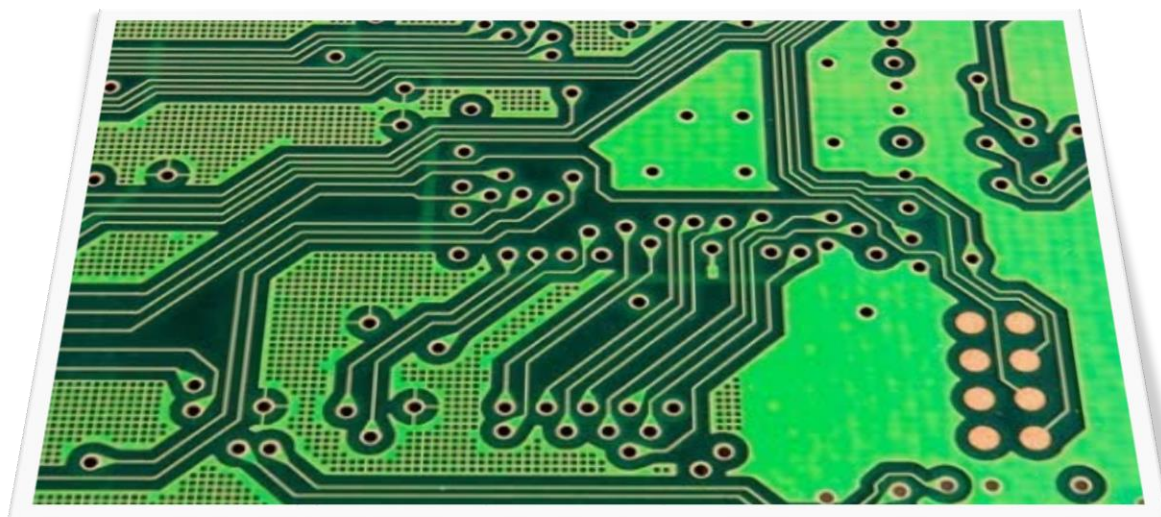


Univerza v Ljubljani
Fakulteta za elektrotehniko



Logični Analizator

POROČILO LABORATORIJSKIH VAJ

Digitalni elektronski sistemi

Timotej Derstvenšek | Fakulteta za elektrotehniko | 31. maj 2016

Vsebina

Uvod	1
Shema vezja.....	2
CPLD in konektorji	3
Napajanje vezja	5
VGA.....	6
Logična sonda	7
Tiskano vezje	8
Razporejanje elementov in povezovanje.....	8
Seznam elementov	9
Vmesnik v vezju CPLD	10
Blokovna shema in delovanje.....	10
Simulacija.....	11
Zaključek	12
Viri.....	12

Kazalo slik

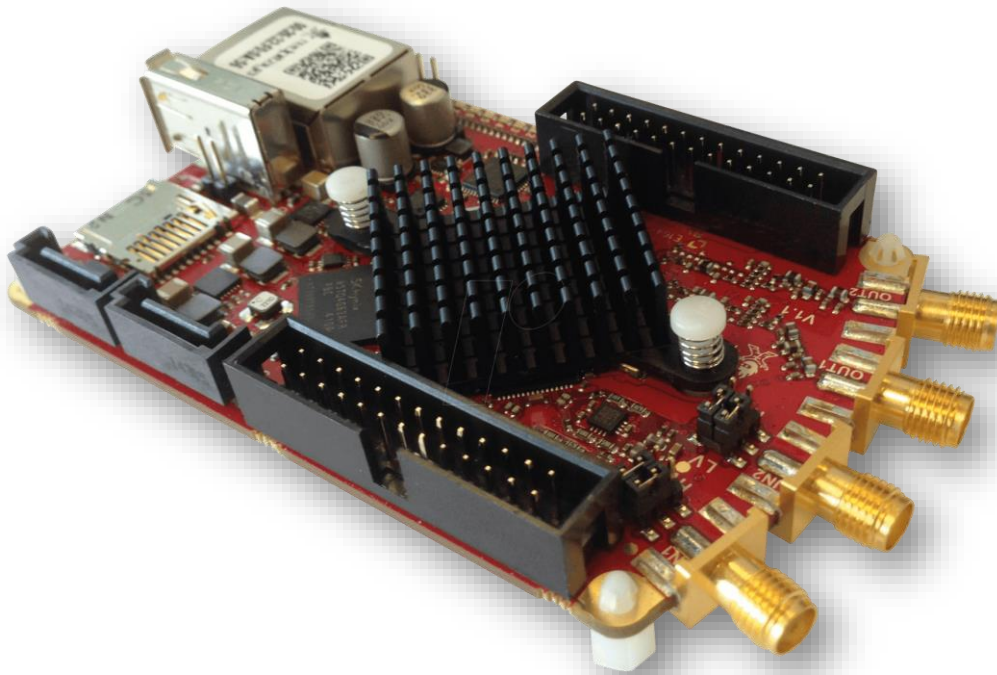
Slika 1: Red Pitaya	2
Slika 2: razširitveni konektorji.....	3
Slika 3: konektor E1 Slika 4: konektor KI	4
Slika 5: konektor E2, napetostni regulator in kratkospojnik, Slika 6: Napajanje vezja CPLD	5
Slika 7: D/A pretvornik, Slika 8: VGA	6
Slika 9: Logična sonda, Slika 10: Konektor za programiranje.....	7
Slika 11: Razmak med E1 in E2, Slika 12: Končna postavitev.....	8
Slika 13: Končno tiskano vezje.....	9
Slika 14: Blokovna shema, Slika 15: Simulacija.....	11

Uvod

Pri predmetu Digitalni elektronski sistemi smo se pri laboratorijskih vajah lotili projekta, da bi ustvarili razširitveni modul za že obstoječe vezje Red Pitayo¹, s katerim bi realizirali

¹ <http://redpitaya.com/>

logični analizator. S pomočjo projekta smo se naučili osnov izdelave vezij, programiranja v VHDL jeziku, ter uporabe programskih paketov kot je Vivado² in Eagle³. Sodeloval sem v arhitekturni skupini, kjer smo se podrobneje spoznali z načrtovanjem, strukturiranjem in oblikovanjem vezja.



Slika 1: Red Pitaya

Shema vezja

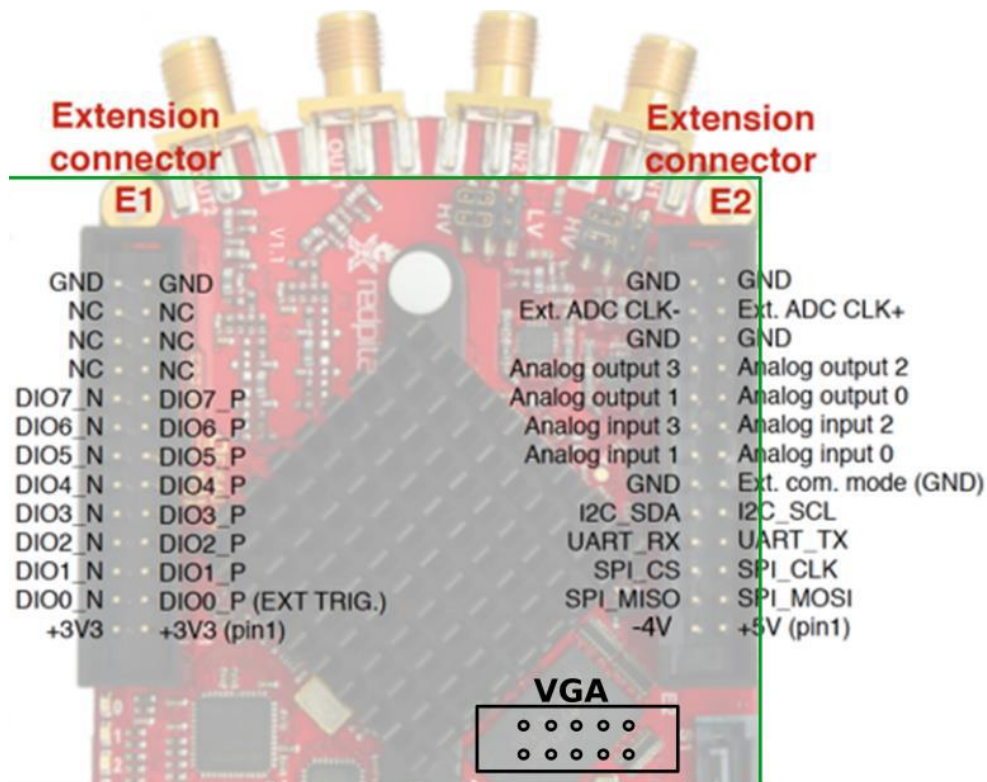
Vsaka izdelava vezja se vedno prične s shemo. Za izdelavo te, smo uporabljali programsko orodje Eagle, ki je bil naša izbira, zaradi dostopnosti ter enostavnejše uporabe.

² <http://www.xilinx.com/products/design-tools/vivado.html>

³ <http://www.cadsoftusa.com/eagle-pcb-design-software/about-eagle/>

CPLD in konektorji

Najprej smo v shemo vključili CPLD z oznako *xc9572xl-tq100*, za katerega je bilo potrebno prenesti⁴ novo knjižnico, saj ga standardne knjižnice ne vsebujejo. Nato smo dodali konektorje E1 (FE13-2), E2 (FE13-1) in logične vhode KI (MA17-2), preko katerih je razširitveni modul povezan z Red Pitayo, kot je prikazano na sliki 2.

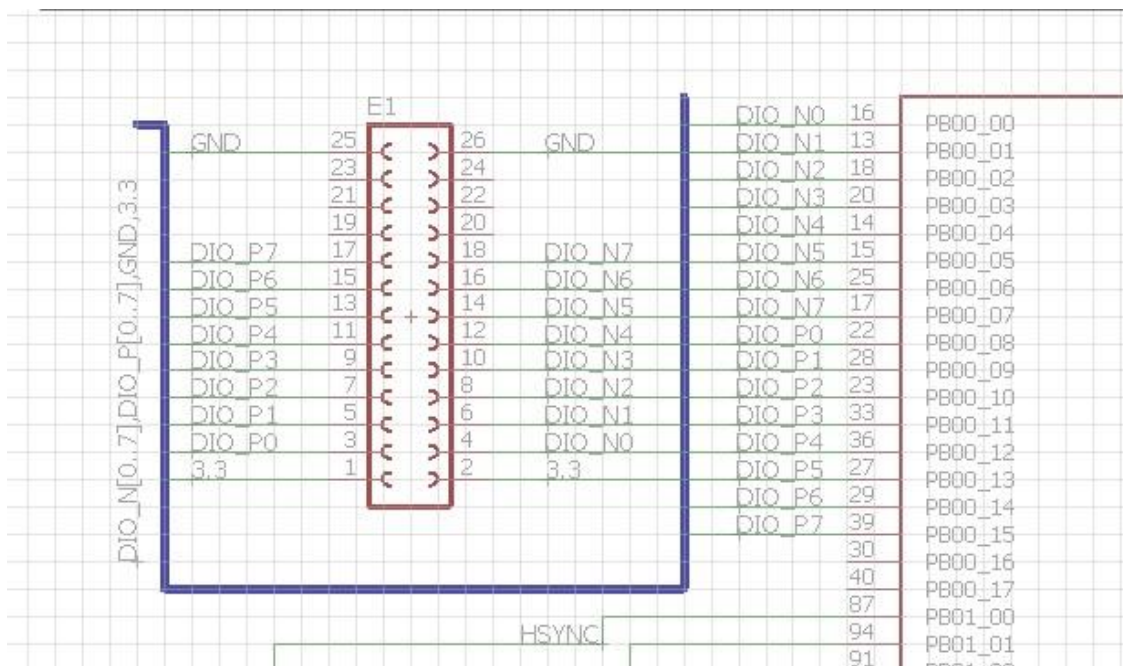


Slika 2: razširitveni konektorji

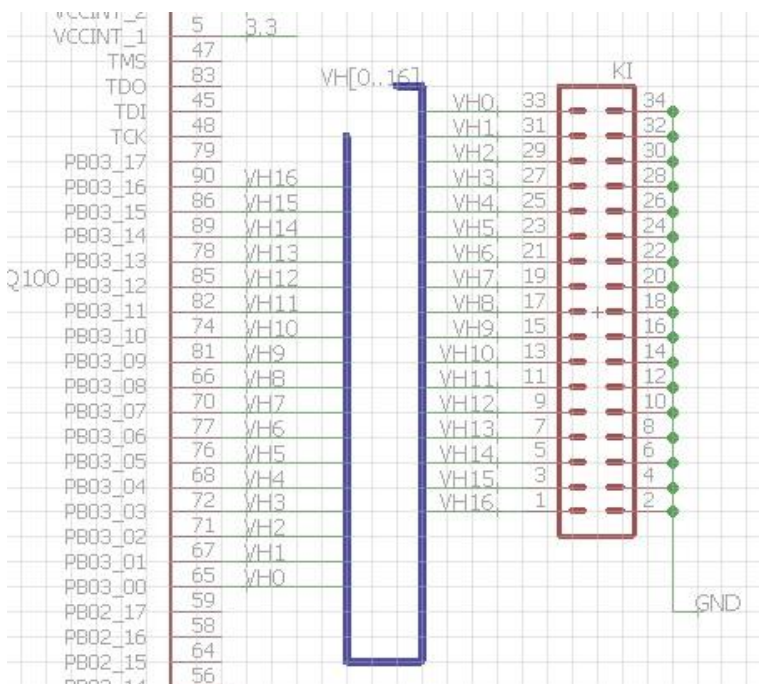
Razširitveni konektor E1 (extension connector) povezuje logične signale (DIO_N in DIO_P) ter 3.3V napajanje, medtem ko razširitveni konektor E2, povezuje z razširitvenim modulom le 5V napajanje, kajti ostalih signalov ne potrebujemo.

Za konektor E1 sem si izbral bazo 00 (PB00) CPLD-ja. Logični signali (DIO_N) si sledijo od PB00_00 do PB00_07, signali (DIO_P) pa od PB00_08 do PB00_15, kot prikazuje shema spodaj. Za boljšo preglednost so povezave narejene s pomočjo vodila (bus).

⁴ <https://github.com/FrankBuss/c64pla/blob/master/eagle/xc9572xl-tq100.lbr>



Slika 3: konektor E1

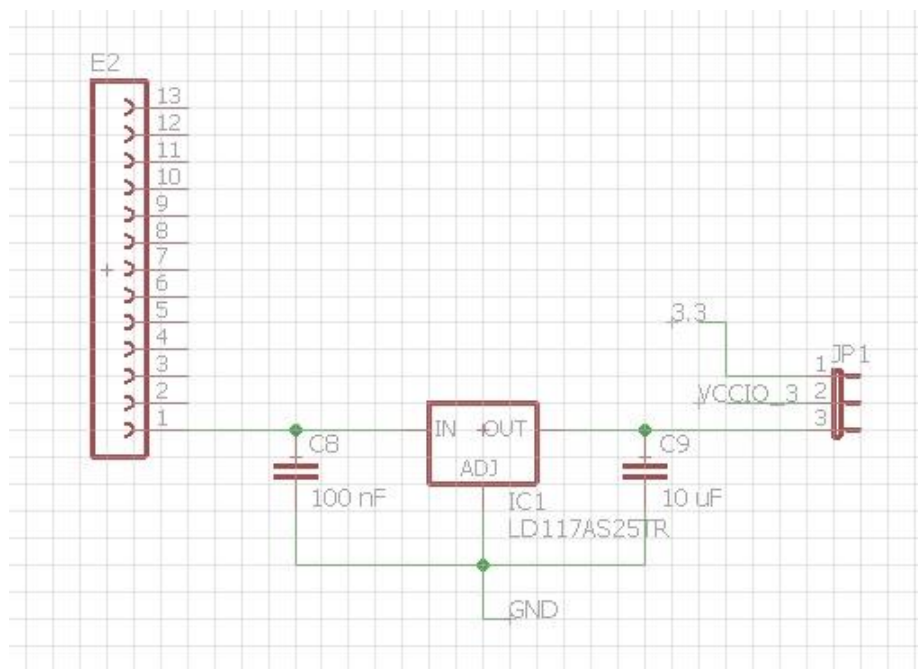


Slika 4: konektor KI

Logične vhode KI (MA17-2) smo povezali na signale CPLD banke 03 (PB03) in sicer na način, da so ena vrsta signalov logični vhodi (VH0-VH16), druga vrsta pa masa (GND). Pri povezovanju tega konektorja pa smo že morali razmišljati tudi o zaporedju povezovanja, da bi se na tiskanem vezju povezave čim manj križale, torej je orientacija zelo pomembna.

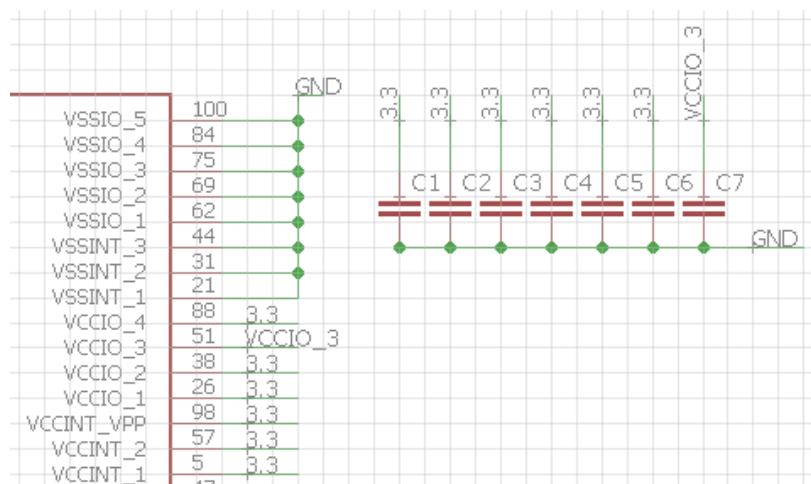
Napajanje vezja

Za izbiro željene napajalne napetosti logičnih vhodov LVCMOS (3.3V ali 2.5V), smo v vezje dodali napetostni 2.5V linearni regulator LM1117 (LM117AS25TR) in kratkospojnik (JP2E). Napetostni regulator dobi s priključka konektorja E2, 5V napajanje in ima na vhodu kondenzator (C-EUC1210) z vrednostjo 100 nF na izhodu pa z vrednostjo 10 μ F, ta skrbita za bolj stabilno napajanje. Izhod regulatorja je nato povezan s kratkospojnikom na katerem sta napajani 3.3V in alternativno (VCCIO_3).



Slika 5: konektor E2, napetostni regulator in kratkospojnik

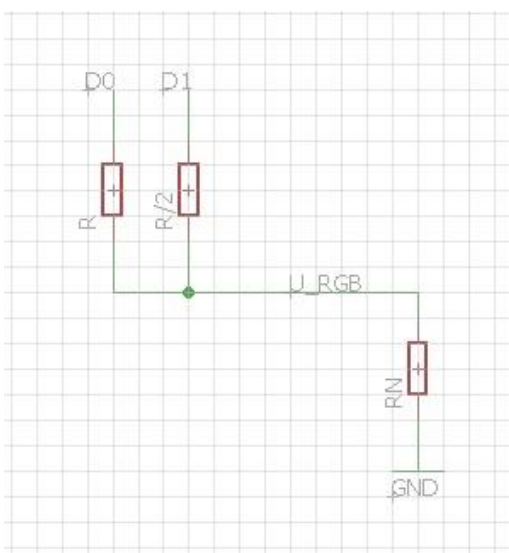
Vse napajalne priključke vezja CPLD (VCC) smo povezali na 3.3V napajanje s konektorja E1 priključke (VSS) pa na maso (GND), razen priključka VCCIO_3, na katerem je alternativno napajanje. Dodali smo še kondenzatorje (C-EUC0603) za vse (VCC) priključke.



Slika 6: Napajanje vezja CPLD

VGA

Na vezje smo dodali 2x5 pinski konektor (MA05-2), to je analogni VGA priključek za računalniški monitor, ta prenaša signale za rdečo, zeleno in modro barvo (RGB). Poleg barv pa še HSYNC (horizontal synchronisation) ter VSYNC (vertical synchronisation). Barvni signali so omejeni z 0,7V-nim maksimalnim napetostnim nivojem, zato so ključne pravilne vrednosti uporov (R-EU_M0805), ki tvorijo za posamezne barve 2-bitni D/A



Slika 7: D/A pretvornik

pretvornik. V našem primeru je logična enica 3.3V, za izračun uporov upoštevamo, da je vsota tokov I0 in I1 enaka toku ki teče skozi upor Rn. Sledi sledeča formula:

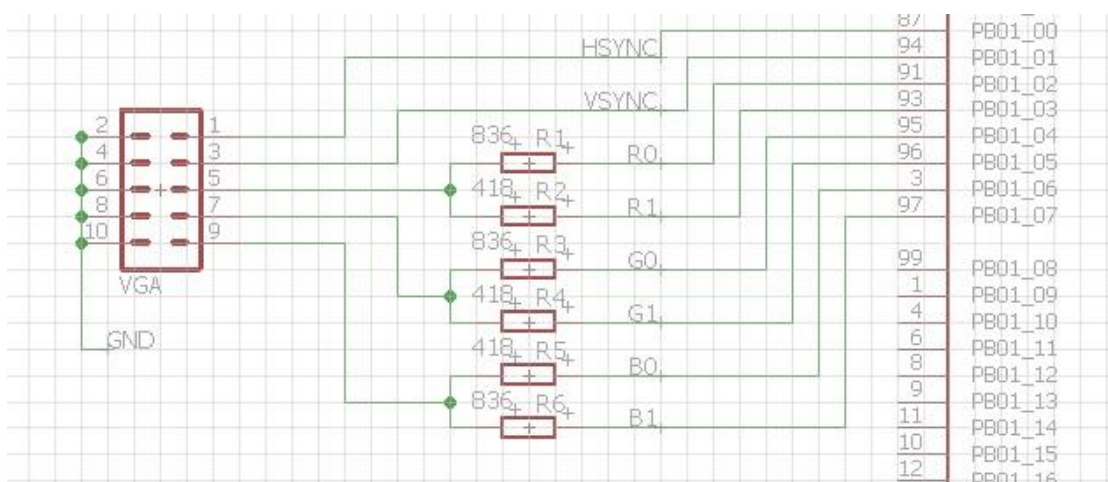
$$\frac{U_{rgb}}{R_n} = D_0 * \frac{U - U_{rgb}}{R} + D_1 * \frac{U - U_{rgb}}{R/2},$$

kjer je U napajanje (3.3V), Rn upornost sponk posameznih barv proti masi (75Ω), Urgb napetost barvnih signalov (0.7V) in D1 D0 dvobitni podatek (11 za izračun). Iz formule izrazimo iskano upornost R, vstavimo številke in dobimo:

$$R = \frac{(2.6V + 2 * 2.6V) * 75\Omega}{0.7V} = 835,7\Omega$$

→ 840Ω (zaokroženo)

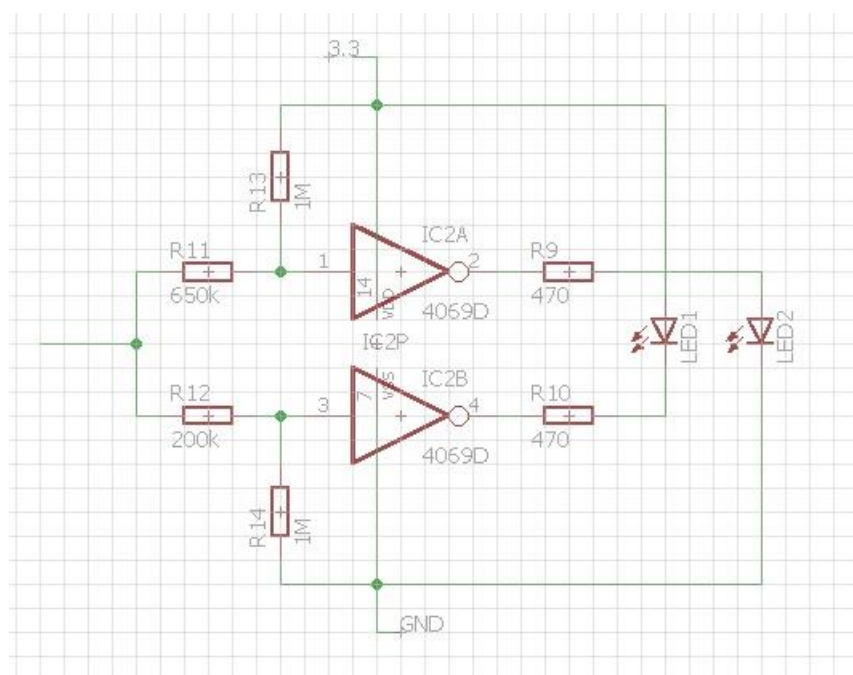
Upornosti zaokrožimo navzgor saj omejujejo največji tok. Vrednosti uporov ostalih signalov pa se manjšajo z $\frac{1}{2^n}$, v našem primeru imamo samo dvobitni podatek, torej so 420Ω (zaokroženo). Na CPLD sem si izbral za povezave banko 01 (PB01).



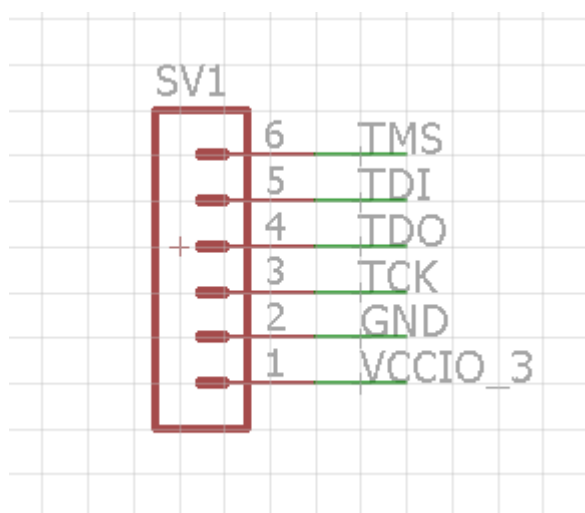
Slika 8: VGA

Logična sonda

Nato smo shemo dopolnili še s preprosto logično sondo, ki zazna logično ničlo ali logično enico. Uporabili smo hex negator (*4069*). Če je na vходу visoko stanje, skozi prvo diodo (LED1) steče tok in ta zasveti, (LED2) pa zaradi toka v zaporno smer ne. Ob nizkem stanju se zgodi ravno obratno. V plavajočem stanju ne sveti nobena od LED diod, torej dobili smo delujočo logično sondo. Pomagali smo si s spletno stranjo: <http://www.edn.com/design/test-and-measurement/4391393/Single-hexinverter-IC-makes-four-test-gadgets>, svojo izvedbo pa sem predstavil v zaključku kot nadgradnji.



Slika 9: Logična sonda



Povsem na koncu dodamo še konektor za programiranje CPLD-ja, s signali (TCK) test clock, (TMS) test mode select, (TDI) test data in, ter (TDO) test data out. Ko je shema končana, zaženemo še program za preverjanje (ERC), ki preveri če so vzpostavljene vse povezave in če prihaja do kakšnih nehotenih napak.

Slika 10: Konektor za programiranje

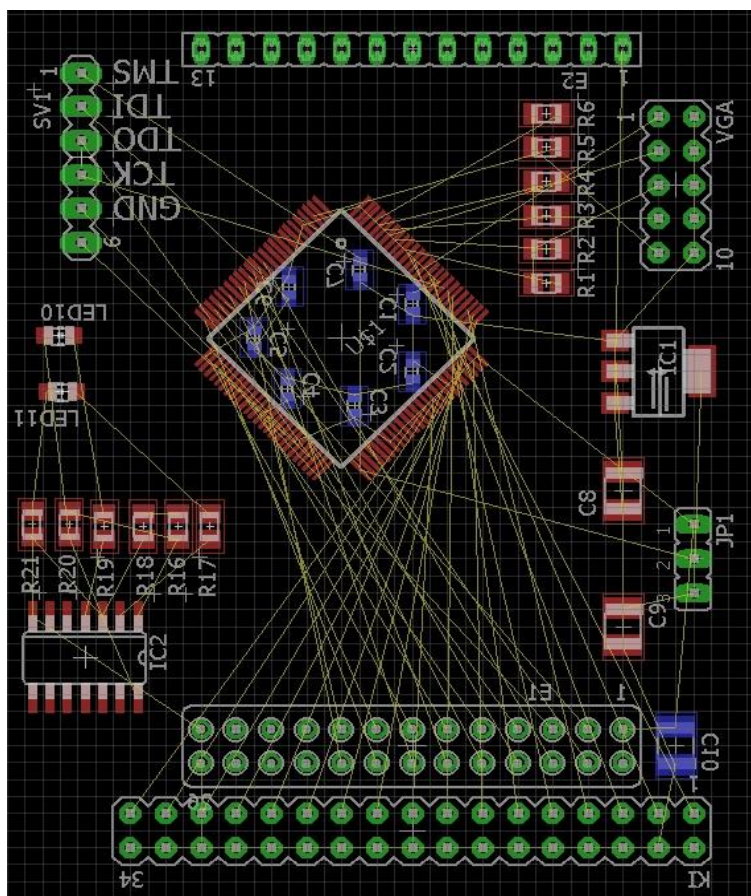
Tiskano vezje

Razporejanje elementov in povezovanje

Po končani shemi, ki vsebuje vse elemente in je brez napak, se lotimo razporejanja elementov na tiskano vezje. Najbolj pomembna je razporeditev konektorjev E1 in E2, katerih zadnji pini morajo biti narazen točno 2100 mils, kajti le tako bo ta razširitveni modul mogoče povezati z Red Pitayo. Pazimo še na orientacijo konektorja E1, zaradi vrstnega reda priključkov ter strani vezja.



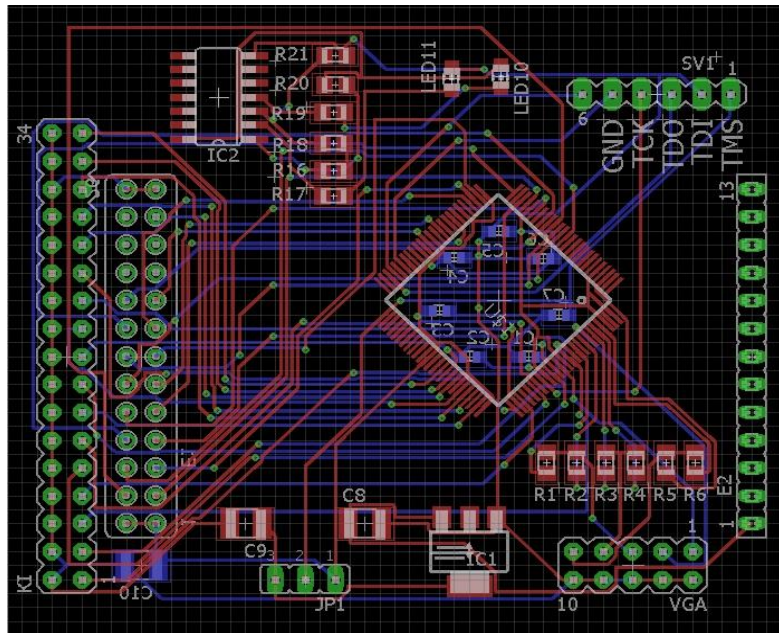
Slika 11: Razmak med E1 in E2



Slika 12: Končna postavitve

Pri postavitvi poskušamo upoštevati, da so konektorji na robu vezja, za lažje priklapljanje, da so elementi v zadostnem razmaku in da kondenzatorje postavimo v bližino napajanj in mas tiskanih vezij. Za pravila povezovanja smo uporabili minimalno širino povezav (8 mil), minimalni odmik (8 mil, od roba 4 mil), minimalni premer spajkalnega otoka-pad ali skožnika-via (0,6 mm) ter potek povezav le pod kotom, ki je večkratnik 45°. Pravila je določilo podjetje za izdelavo vezij Lingva.

Povezali smo le napajanje saj smo bili zelo omejeni s časom, zato sem za ostale signale uporabil autorouter, ki je sicer povezal vse povezave, vendar je vse prej kot idealen, zato se ga običajno izogibamo. Po izvedbi CAM process dobimo gerber datoteke⁵ (.gko, .gts, .gtl, .gto, .gbs, .gbo, .gbl), ki so potrebne za različne korake izdelave.



Slika 13: Končno tiskano vezje

Seznam elementov

OPIS:	ELEMENT:	OZNAKA:	KOLIČINA:
CPLD	xc9572xl-tq100	U\$1	1x
konektor	FE13-2	E1	1x
konektor	FE13-1	E2	1x
konektor	MA17-2	KI	1x
Napetostni regulator	LM117AS25TR	IC1	1x
Kratkospojnik	JP2E	JP1	1x
konektor	MA05-2	VGA	1x
Kondenzatorji (10 μ F)	C-EUC1210	C9, C10	2x
Kondenzator (100 nF)	C-EUC1210	C8	1x
Kondenzatorji	C-EUC0603	C1, ..., C7	7x
Upori (836 Ω)	R-EU_M0805	R1, R3, R5	3x
Upori (418 Ω)	R-EU_M0805	R2, R4, R6	3x
Upori	R-EU_M0805	R16, ..., R21	6x
LED diodi	CHIP-LED0805	LED10, LED11	2x
Konektor za programiranje	MA06-1	SV1	1x

⁵ <http://www.gerber-viewer.com/>

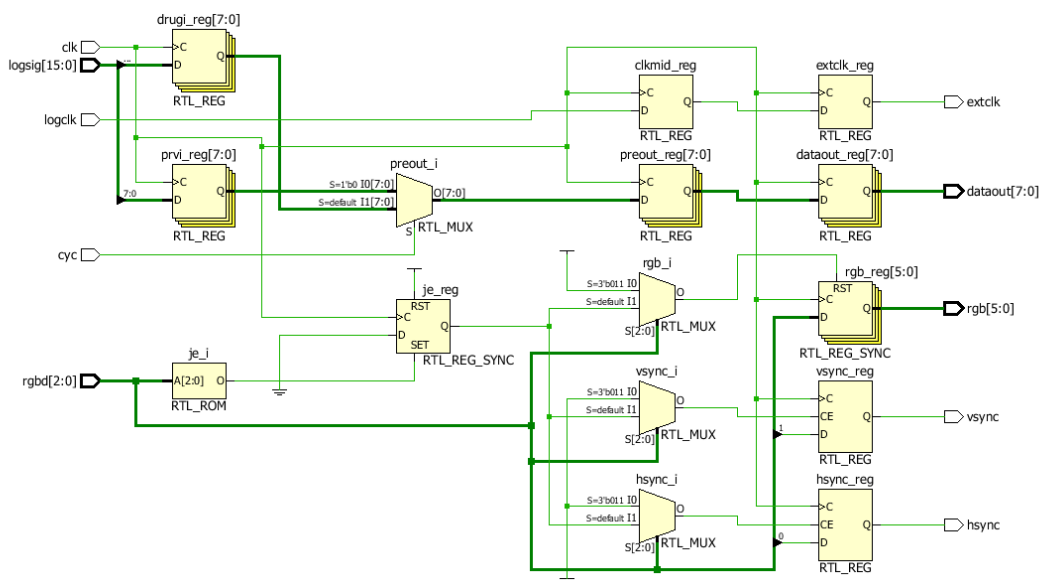
Vmesnik v vezju CPLD

V vezju CPLD je vmesnik za prenos 16-bitnih logičnih vhodov in vezje za dekodiranje VGA. Za programiranje CPLD smo uporabili programski paket Vivado, ki se izvaja v VHDL jeziku, ter nam med drugim omogoča simulacijo.

Blokovna shema in delovanje

Na vhodu imamo 16 logičnih signalov (logsig) in uro (logclk), ki jih želimo posredovati Red Pitayi. Želimo, da vmesnik deluje sinhrono zato vse nadzorujemo z najhitrejšo uro (clk), sledi da vse zapakiramo v p1:process. Za 16 podatkov potrebujemo 16 vhodov, katerih Red Pitaya nima, zato podatke multipleksiramo, torej razdelimo na 2x8 podatkov in pošljamo ločeno. Ob vrednosti '0' signala cyc se prenese prvih 8 bitov (prvi), ob vrednosti '1' pa zadnjih 8 (drugi). Na izhodu dobimo podatke (dataout) in uro (extclk), katera ima še umesno stanje (clkmid). Za dekodiranje VGA ustvarimo p2:process, kjer se podatki prenašajo ob uri (clk). Za upravljanje zaslona potrebujemo signale barv (rgb) ter sinhronizacijska signala (hsync in vsync), te dekodiramo tako, da ustvarimo nov signal (rgbd). Ob vrednosti "011" tega signala, je izhod za barve enak "0000000" ob tem pa si to stanje zapomnimo še s signalom (je), kajti ob naslednjem ciklu barvni izhod ostane enak, ob tem pa hsync postane enake vrednosti kot prvi, vsync pa drugi bit rgbd. Ob drugih vrednostih je izhod rgb možno izračunati po formuli iz 84. in 85. vrstice. Ko v programu Vivado poženemo možnost (open elaborated design) dobimo blokovno shemo s slike 14 na naslednji strani.

```
94
53 p1:process(clk)
54 begin
55     if rising_edge(clk) then
56         clkmid <= logclk ;
57         extclk <= clkmid ;
58
59         prvi <= logsig (7 downto 0);
60         drugi <= logsig (15 downto 8);
61
62         if (cyc = '0') then
63             preout <= prvi;
64         else
65             preout <= drugi;
66         end if;
67         dataout <= preout;
68     end if;
69 end process;
70
71 p2:process(clk)
72 begin
73     if rising_edge(clk) then
74         if (rgbd = "011") then
75             rgb <= "000000";
76             je <= '1';
77         else
78             if (je = '1') then
79                 rgb <= "000000";
80                 hsync <= rgbd(0);
81                 vsync <= rgbd(1);
82                 je <= '0';
83             else
84                 rgb <= rgbd(2) & rgbd(2) & rgbd(1)
85                 & rgbd(1) & rgbd(0) & rgbd(0);
86             end if;
87             je <= '0';
88         end if;
89     end if;
90 end process;
91 end Behavioral;
```



Slika 14: Blokovna shema

Simulacija

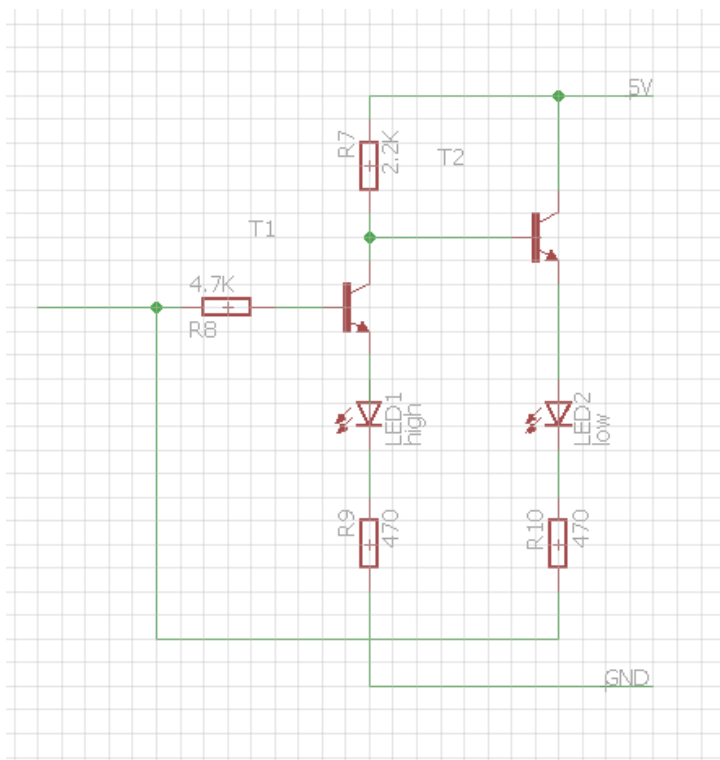
Delovanje je zelo lepo razvidno iz simulacije, ki nam ga omogoča programsko orodje. Najprej nastavimo vhode na željene vrednosti, v mojem primeru je clk s periodo 50ns, logclk s periodo 100ns, cyc s periodo 150ns in 16-bini vhodni signali (logsig) vrednosti '0402' šestnajstiško. Vrednost rrgb sem spreminjal med '3' in '6' šestnajstiško. Ob kazalcu obarvanem rumeno vidimo prehod rrgb s '3' na '6', nato sledi sprememba signalov vsync na vrednost '1' ($\text{vsync} \leq \text{rrgb}(1)$) kot željeno, hsync ostane na '0' ($\text{hsync} \leq \text{rrgb}(0)$) in po enem ciklu se spremeni tudi rgb z vrednosti '00000000' na tisto, ki je določena s formulo.



Slika 15: Simulacija

Zaključek

Po zaključenih laboratorijskih vajah in s tem tudi projektom, znamo uporabiti različna programska okolja, naučili smo se različne vrste dela in postopke pri izdelavi vezij, dobili smo večji vpogled v tehnologije uporabljene za izdelavo, testiranje in oblikovanje. Predstavljen nam je bil celoten potek od ideje do izvedbe projekta. Projekt bi sam označil kot uspešen in sem zelo zadovoljen s spoznanji in znanjem, ki sem ga pridobil. Logični analizator bi morda lahko nadgradili z uporabo DVI povezave z monitorjem, namesto VGA, ob večji zalogi časa bi seveda lahko optimizirali tiskano vezje z boljšimi povezavami in da bi naše vezje čim manj sevalo, bilo čimbolj združljivo, torej prestalo EMC test (electromagnetic compatibility). Prav tako pa bi lahko logično sondo realizirali brez uporabe negatorja, na primer z dvema tranzistorjema kot prikazuje shema spodaj (ob visokem nivoju je odprt prvi ob nizkem pa drugi tranzistor).



Viri

- <http://lniv.fe.uni-lj.si/des.html>
- <http://redpitaya.com/>
- <http://www.edn.com/design/test-and-measurement/4391393/Single-hexinverter-IC-makes-four-test-gadgets>
- <http://michaelleonard.com/how-to-design-the-perfect-pcb-part1/>
- <http://users.volja.net/ipavlic/AD%20pretvorniki.pdf>
- http://lrtme.fe.uni-lj.si/lrtme/slo/UNIVSS/meri_pret/seminar%202011/DA%20PRETVORBA.pdf
- <http://www.xilinx.com/>
- <http://www.cadsoftusa.com/training-service/>