

Univerza v Ljubljani
Fakulteta za elektrotehniko

Uroš Nahtigal

MERJENJE KAPACITIVNOSTI Z NAMENSKIM ČIPOM

Seminarska naloga

pri predmetu
Elektronska vezja

V Ljubljani, oktober 2009

1. UVOD

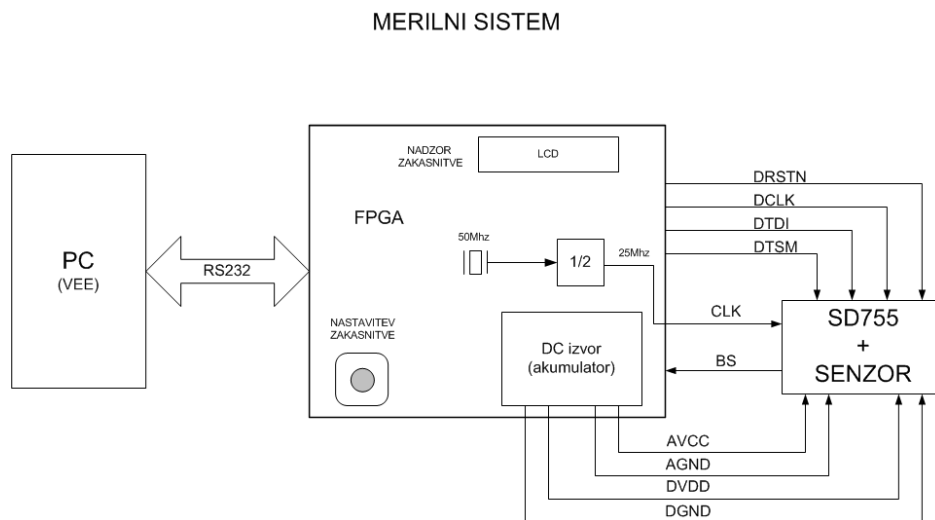
Namen seminarske naloge je bil izdelati merilni sistem, ki bo meril razliko kapacitivnosti s pomočjo namenskega ASIC-a (**application-specific integrated circuit**). Poleg ASIC-a je uporabljen FPGA čip Spartan 3e, ki skrbi za komunikacijo med osebnim računalnikom in ASIC-om ter pretvorbo rezultata v obliko, ki je primerna za prikaz na osebnem računalniku. Z namenom, da bi privarčeval čim več časa, sem uporabil Xilinx-ovo razvojno ploščo Spartan 3e starter kit, ki v veliki meri poenostavi programiranje FPGA čipa. Xilinx nudi programsko orodje za pisanje VHDL kode in programiranje Spartan 3e čipa.

To nalogo sem si izbral, ker sem tako dobil možnost, da se bolje spoznam z opisnim jezikom VHDL in nadgradim znanje, ki sem ga pridobil pri predmetu Integrirana vezja.

1.1 OPIS VEZJA

Pripravil sem merilni sistem kapacitivnosti oziroma boljše rečeno spremembo kapacitivnosti, ki jo merimo z namenskim čipom SD 755. Sistem ima dve možni uporabi. Prva je merjenje absolutne kapacitivnosti. Pogoji, je zadovoljivo natančen etalon, s pomočjo katerega lahko izračunamo merjenčevo kapacitivnost. Druga možnost uporabe je merjenje relativne spremembe kapacitivnosti, kar bi lahko s pridom uporabljali pri aplikacijah, ki uporabljajo senzorje na principu spreminjanja kapacitivnosti.

Če želimo izmeriti razliko kapacitivnosti je potrebno najprej pravilno nastaviti ASIC. Določiti območje v katerem pričakujemo razliko, da nam sistem slučajno ne zbeži v nasičenje. Rezultat prikazujemo na osebnem računalniku v programskem okolju VEE.



Slika 1: Merilni sistem

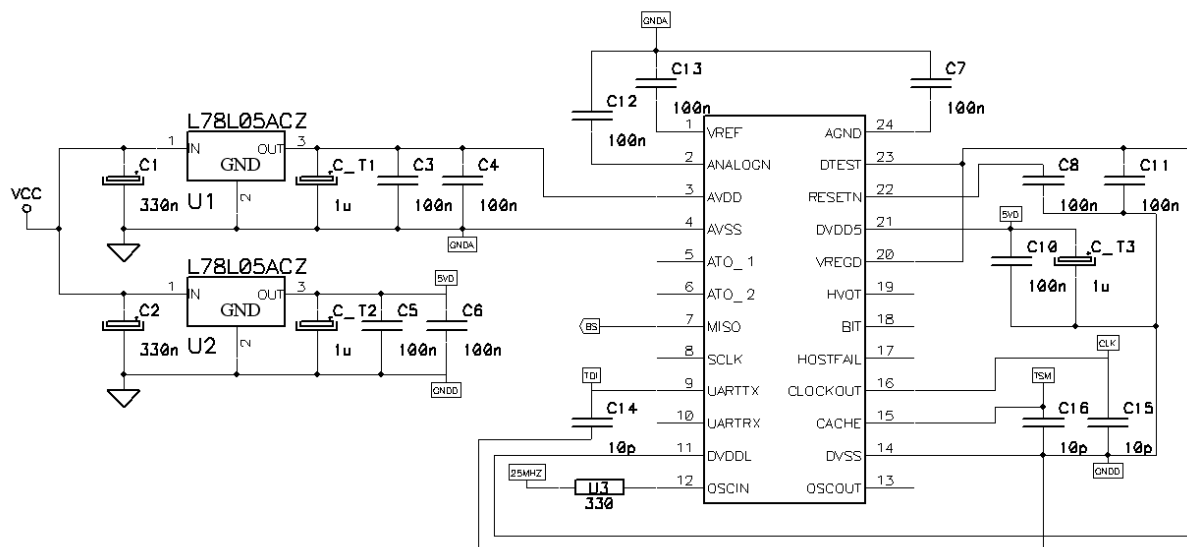
Sistem je sestavljen iz treh podsklopov (Slika 1): osebnega računalnika, FPGA modula in namenskega čipa ASIC. Z osebnim računalnikom spreminjamo nastavitve ASIC-a in prikazujemo izmerjene rezultate v programskem okolju VEE pro. FPGA modul služi za programiranje ASIC-a, obdelavo merilnih rezultatov, ki jih dobimo iz ASIC-a, in obdelan rezultat pošlje v osebni računalnik. ASIC meri razliko kapacitivnosti in preko digitalnega signala (bitstream) posreduje merilni rezultat vezju FPGA.

Meritev začnemo tako da priklopimo napajanje ASIC-u in FPGA plošči ter zaženemo VEE grafični vmesnik. Najprej je potrebno resetirati čip, s tem poskrbimo da se vsi notranji registri v čipu postavijo v začetna stanja nato sledi programiranje čipa. FPGA je bil že predhodno programiran, tako da začnemo dobivati rezultate takoj po programiranju ASIC-a. Merilni Rezultat prikazujemo na osebni računalniku. Hitrost zajemanja merilnih rezultatov s pomočjo FPGA je 150/s. Te merilne rezultate povprečimo, izračunavamo standardno deviacijo in jih shranimo na v poljubno tekstovno datoteko, kjer so pripravljeni na kasnejšo obdelavo.

2. OPIS DELOVANJA

Za čim manjši vpliv okolice na meritev sem uporabil 12 voltni akumulator. Prav tako sem ločil analogno in digitalno napajanje ter obe masi (Slika 1) ter na ta način poizkusil minimizirati vpliv digitalnega dela ASIC-a na občutljivi analogni del in tako povečati ločljivost sistema.

Napajalno vezje je sestavljeno iz dveh napetostnih regulatorjev in štirih gladilnih kondenzatorjev. Uporabljena sta bila 5 voltna regulatorja L78L05AC, dva 1 μ F elektrolitska kondenzatorja in dva 100nF "multilayer" kondenzatorja (Slika 2). FPGA razvojna plošča je napajana preko 5 voltnega AC/DC pretvornika.

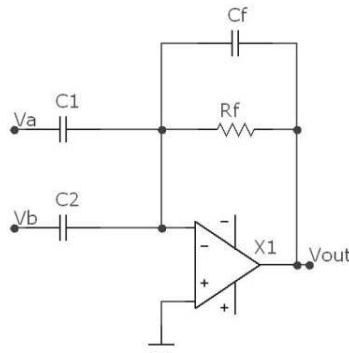


Slika 2: Napajalni del za ASIC

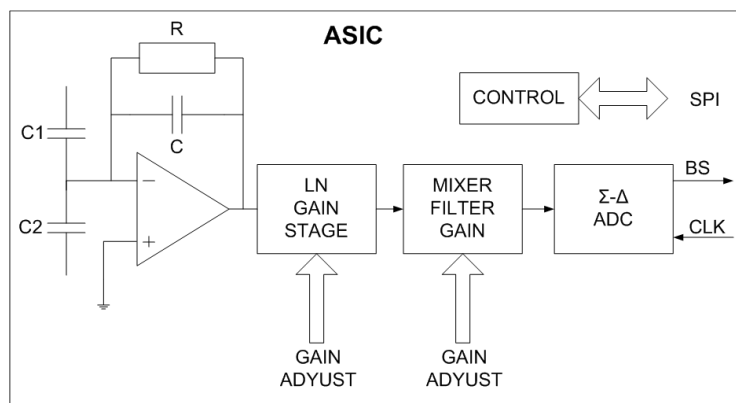
2.1 ASIC

Glavna naloga ASIC-a je primerjanje dveh kapacitivnosti. Vhodni analogni del deluje na preprostem principu (Slika 3). Vezje reši preprosta enačba $v_{out} = -\left(v_a * \frac{c_1}{c_f} + v_b * \frac{c_2}{c_f}\right) * \frac{s}{c_f * R_f + s}$, ki se ob predpostavki $v_b = -v_a$ spremeni v $v_{out} = -\frac{v_a * \Delta C}{c_f} * \frac{s}{c_f * R_f + s}$. Napetost v_{out} je pretvorjena v bitstream ki ga pošljemo v FPGA (Slika 4). Glavni spremenljivki, ki jih lahko programiramo sta amplitudi v_a in v_b ter ojačenje v_{out} pred Σ - Δ modulatorjem, ki

omogočata spreminjanje, območja meritve. Sistem omogoča tudi nastavljanje ojačenja v samem FPGA-ju, vendar je samo območje meritve določeno v ASIC-u.

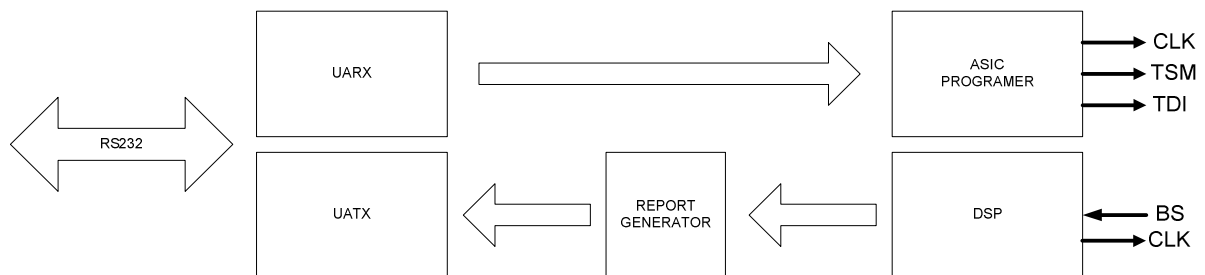


Slika 3: Vhodni analogni del ASIC-a



Slika 4: Blok shema ASICA

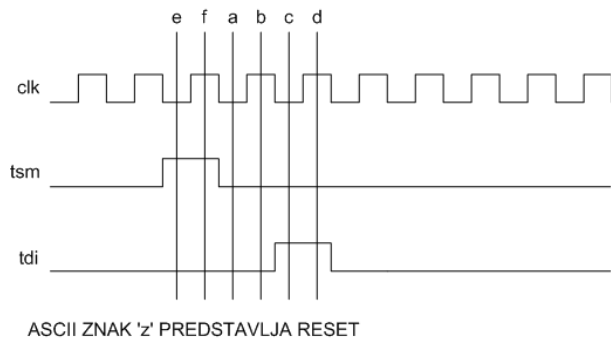
2.2 FPGA



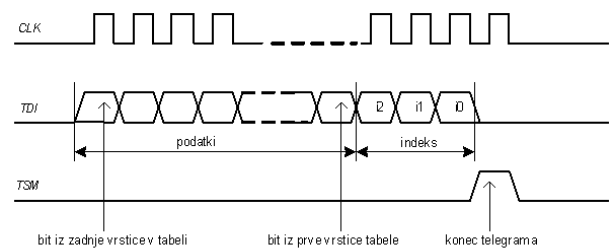
Slika 5: Blok shema FPGA modula

Vezje FPGA je razdeljeno v tri glavne skupine (Slika 5). Prva skrbi za programiranje ASIC-a kamor spadata UARX in ASIC programer. UARX skrbi za pravilen sprejem ukazov, ki jih pošljemo z osebnim računalnikom v FPGA. Komunikacija poteka preko serijskega protokola RS232. ASIC programer oziroma dekodek, služi za programiranje digitalnih signalov CLK, TMS in TDI bite glede na sprejeti ASCII znak.

Prva naloga FPGA modula je torej pravilno ASIC. ASIC programiramo tako, da mu pošljemo daljši niz logičnih '1' in '0', ki predstavlja verigo. Kakšna bo veriga, ki jo želimo poslati v ASIC, določajo ASCII znaki, ki jih pošljemo v FPGA s pomočjo osebnega računalnika. Te ASCII znake pa določajo nastavitve, ki jih želimo imeti v ASIC-u. Za programiranje imamo na voljo tri signale, ki so poimenovani CLK, TDI in TMS. Kaj predstavlja posamezen ASCII znak oziroma kako določa CLK, TMS in TDI vidimo na Slika 6. Za lažje sledenje nastavitvam je vsaka veriga shranjena, kot tekstovna datoteka na osebem računalniku, ki je ob zahtevi za programiranje prebrana in poslana v FPGA po serijskem "COM" portu. Zapisovanje posamezne verige se prične ko je signal TMS na '1', medtem, ko indeks verige določajo trije biti pred spremembo TMS signala na '1'. Podrobnejši opis pisanja verig je viden na Slika 7.

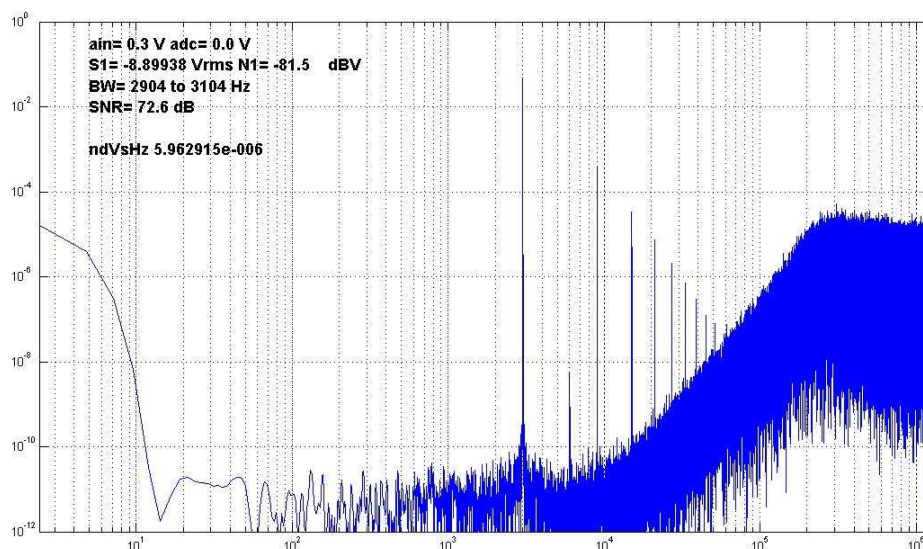


Slika 6: Interpretacija ASCII znakov



Slika 7: Zapisovanje verige

Naloga DSP-ja (**Digital signal processor**) pretvorba bitstream v format, ki ga je nato možno prikazovati s pomočjo PC-ja. Za razumevanje je potrebno najprej poznati spekter bitstreama (Slika 8), ki ga bomo procesirali v DSP-ju. Najbolj zanimiva je 3kHz komponenta.



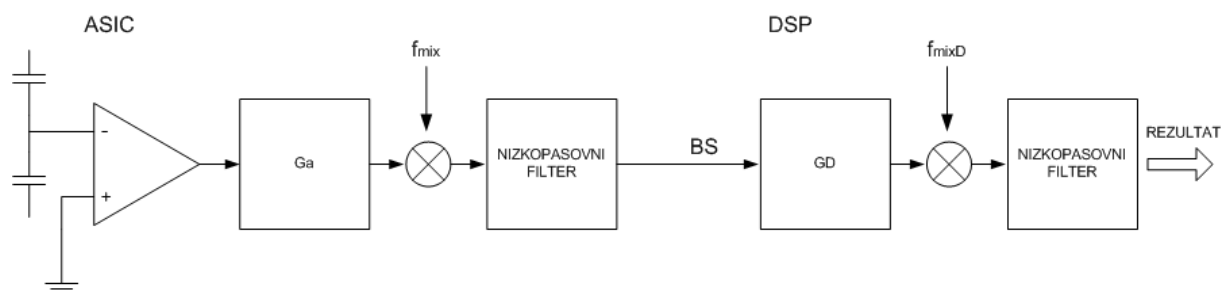
Slika 8: Spekter bitstream

Ta komponenta nosi informacijo o amplitudi V_{out} , ki odvisna od razlike kapacitivnosti in nastavljenega ojačenja ASIC-a. Celotno sliko bitstreama si moramo predstavljati na naslednji način. Napetost $V_{out} = A * \cos(\omega_s * t)$, ki jo dobim na izhodu analognega vhodnega dela (Slika 3), ojačimo z G_a in jo pomnožimo s $\cos(\omega_{mix} * t)$.

$$G_a * A * \cos(\omega_s * t) * \cos(\omega_{mix} * t) = \frac{1}{2} * G_a * A * [\cos((\omega_s - \omega_{mix}) * t) + \cos((\omega_s + \omega_{mix}) * t)]$$

Čemur sledi niskopasovni filter, ki izniči $\cos[(\omega_s + \omega_{mix}) * t]$ komponento, nefiltrirani del je glavna spektralna črta ki jo vidimo na Slika 8. V DSP-ju postopek še enkrat ponovimo vendar, v digitalni obliki, nakar dobimo spodnji rezultat. Celoten postopek ki smo ga opisali lahko vidimo na Slika 9.

$$\frac{1}{2} * G_a * A * [\cos((\omega_s - \omega_{mix}) * t)] * G_D * [\cos(\omega_{mixD} * t)] = \frac{1}{2} * \frac{1}{2} * G_a * A * G_D * [\cos(((\omega_s - \omega_{mix}) - \omega_{mixD}) * t)]$$



Slika 9: Princip delovanja sistema

Rezultat ki ga dobimo $N = \frac{1}{2} * \frac{1}{2} * G_a * A * G_D$ ob predpostavki, da smo pravilno nastavili frekvenco množenja (f_{mixD}), nam omogoča izračun razlike kapacitivnosti, ko upoštevamo $A = \frac{V_a * \Delta C}{C_f}$.

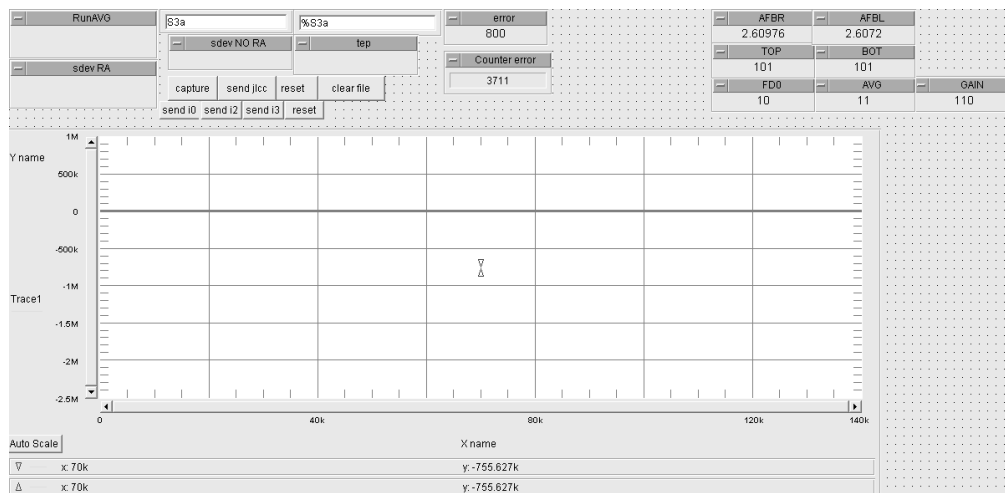
$$\Delta C = N * C_f / (G_a * G_D * A)$$

Zadnji korak je prenos rezultata v osebni računalnik. Najprej report generator pretvori rezultat N v heksadecimalni zapis, ki ga nato z desetimi ASCII znaki pošljemo preko UATX v osebni računalnik. Nastavitve serijskega porta na osebni računalniku so sledeče: BaudRate 19200, stop bit 2, Parity 0 in nobenega strojnega nadzora prenosa podatkov. Enake nastavitve moramo zagotoviti tudi v FPGA-ju, da zagotovimo pravi prenos podatkov.

2.3 PROGRAMSKO OKOLJE VEE

Pri prenosu rezultata od FPGA-ja do osebnega računalnika in za pošiljanju ukazov za ASIC smo uporabili pretvornik USB to serial UC-232A.

Slika 10 prikazuje grafični vmesnik, ki sem ga naredil s pomočjo VEE programskega jezika. Vmesnik prikazuje: rezultate, ki jih pošilja FPGA, izračun povprečne vrednosti, standardno deviacijo in ukaze za pošiljanje verig, reset čipa, ime tekstovne datoteke kamor shranjujemo rezultate in možnost brisanja te datoteke.



Slika 10: Grafični vmesnik v merilnem okolju VEE

3. ZAKLJUČEK

3.1 TEŽAVE

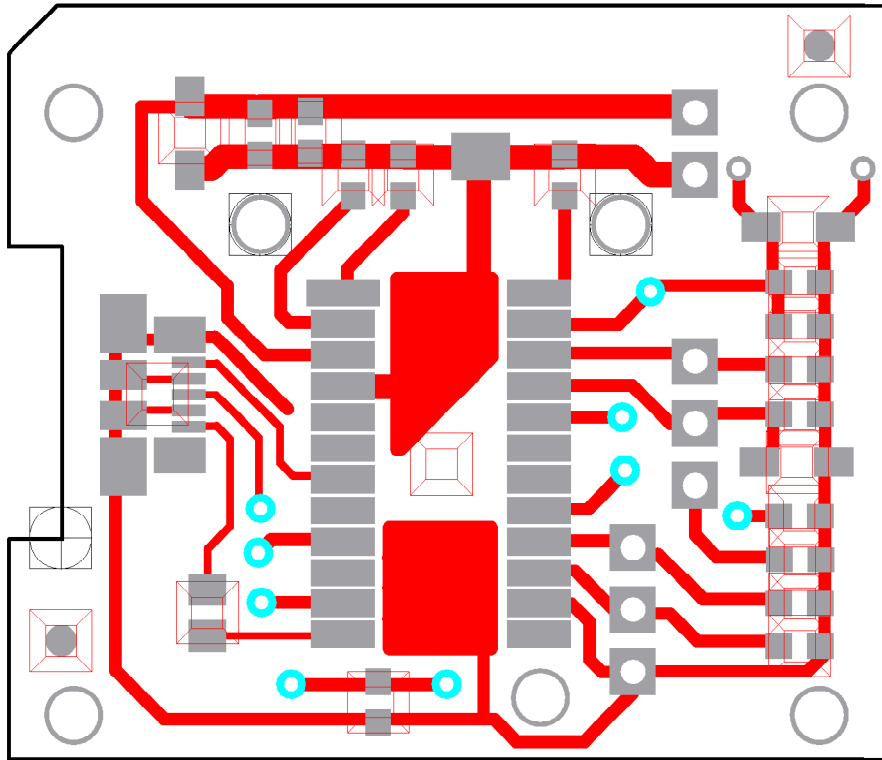
Od vsega me je najbolj presenetil čas, ki je pretekel od ideje do končnega izdelka in število problemov s katerimi sem se srečaval, ki so se kopičili iz dneva v dan. Eden izmed resnejših problemov je bil pravilen priklop mas, na kar na začetku sploh nisem bil pozoren in sem vse mase povezal na (-) sponko akumulatorja. Izkazalo se je da ob napačnem priklopu mas, vezje ni hotelo pravilno delovati, prihajalo je do napačnega vzorčenja bitstreama, napak v napetostnih nivojih in motečih tokov po masi. Najbolj problematični sta bili masi ASIC-a in FPGA-ja. Ena izmed boljših rešitev je bila ko je bilo središče za mase, kar masa od FPGA, ostale pa so povezane v obliki zvezde v to središče. Veliko časa mi je vzelo samo pisanje VHDL kode. Zelo problematičen je bil miselni preklon med programiranjem in opisovanjem vezja v VHDL-u. Drugi problem je da ob neupoštevanju pravil za pisanje v VHDL-u, lahko simulacije VHDL kode delujejo pravilno, medtem ko v realnosti prihaja do povsem nepričakovanih situacij.

3.2 MOŽNOST NADGRADNJE

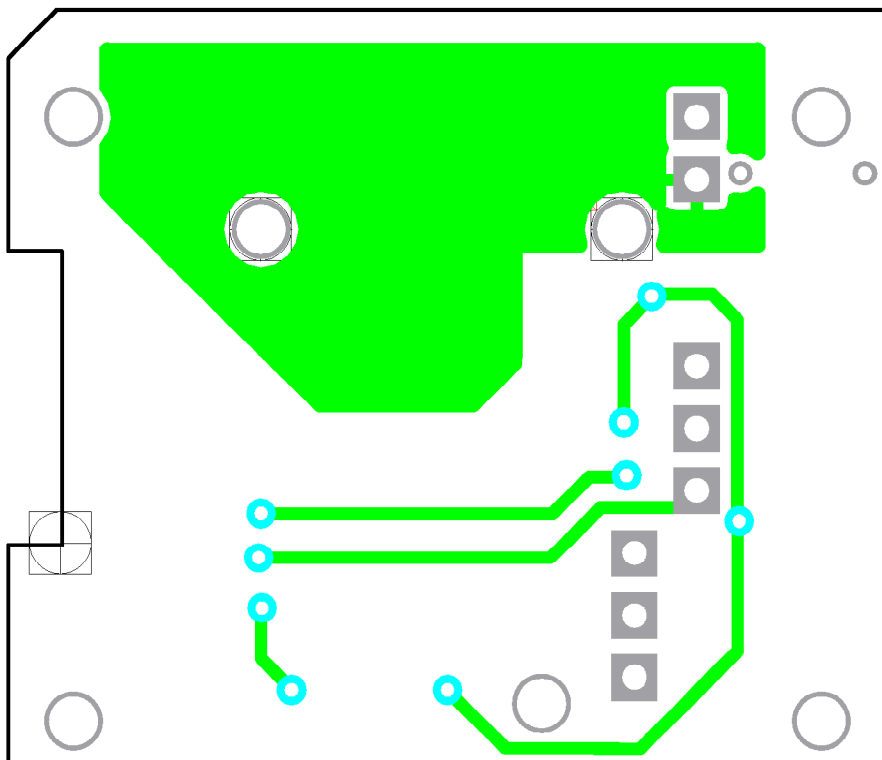
Da bi privarčevali še nekaj prostora bi lahko bila v naslednji verziji merilnega sistema ASIC in FPGA čip na skupnem PCB-ju. Z združitvijo ASIC-a in FPGA-ja, bi se tako znebili vseh nepotrebnih komponent, ki so prisotne na razvojni plošči in jima lahko zagotovili tudi skupno napajanje iz 12V akumulatorja. Korak naprej, bi pravzaprav bila izdelava novega namenskega čipa, ki bi ga sestavljal vhodni analogni del in digitalni del ter vsa potrebna periferija za prenos podatkov od čipa do osebnega računalnika.

VIRI

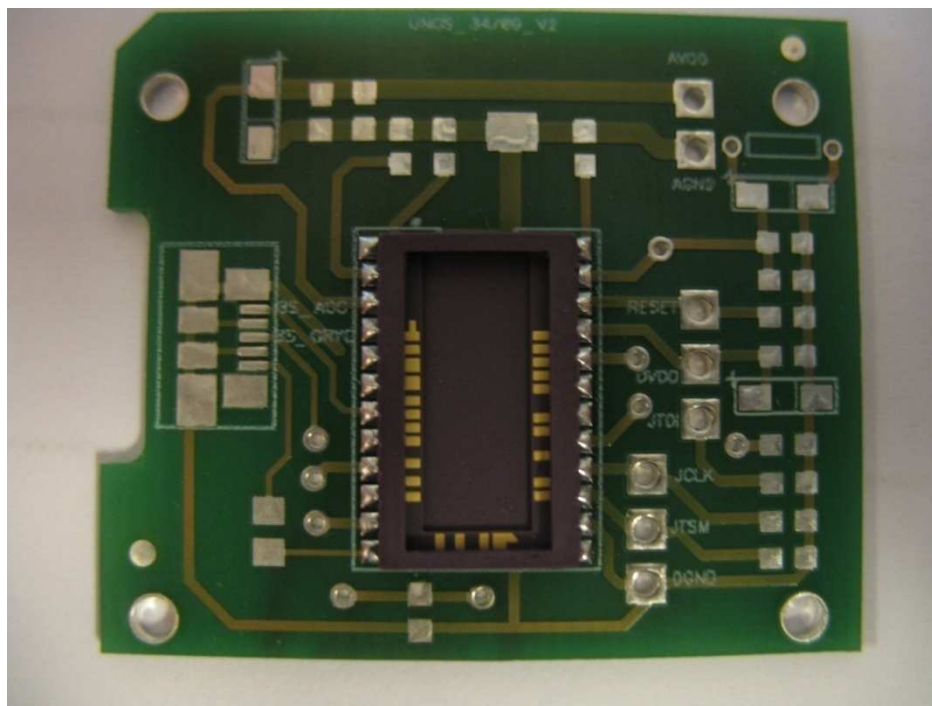
- [1] <http://uk.farnell.com/>
- [2] <http://www.opencores.org/>
- [3] <http://www.xilinx.com/support/#Spartan-3E>
- [4] http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf
- [4] <http://sl.wikipedia.org/wiki/ASCII>
- [5] Ben Cohen, Coding Styles and Methodologies
- [6] K.C. Chang, Digital systems Design with VHDL and Synthesis, 1999
- [7] S. Tomažič, S. Leonardis: Diskretni signali in sistemi, 2004
- [8] <http://www.bb-elec.com/product.asp?sku=UC232A>
- [9] Specifikacije za SD 755, LMFE



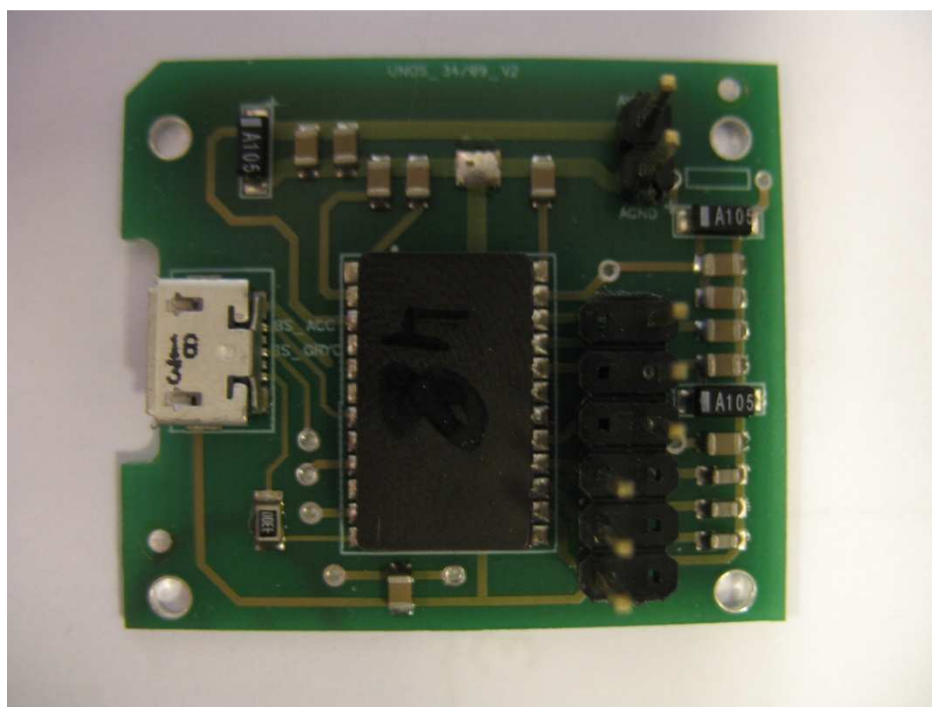
Slika 11: TOP tiskanine merilne ploščice



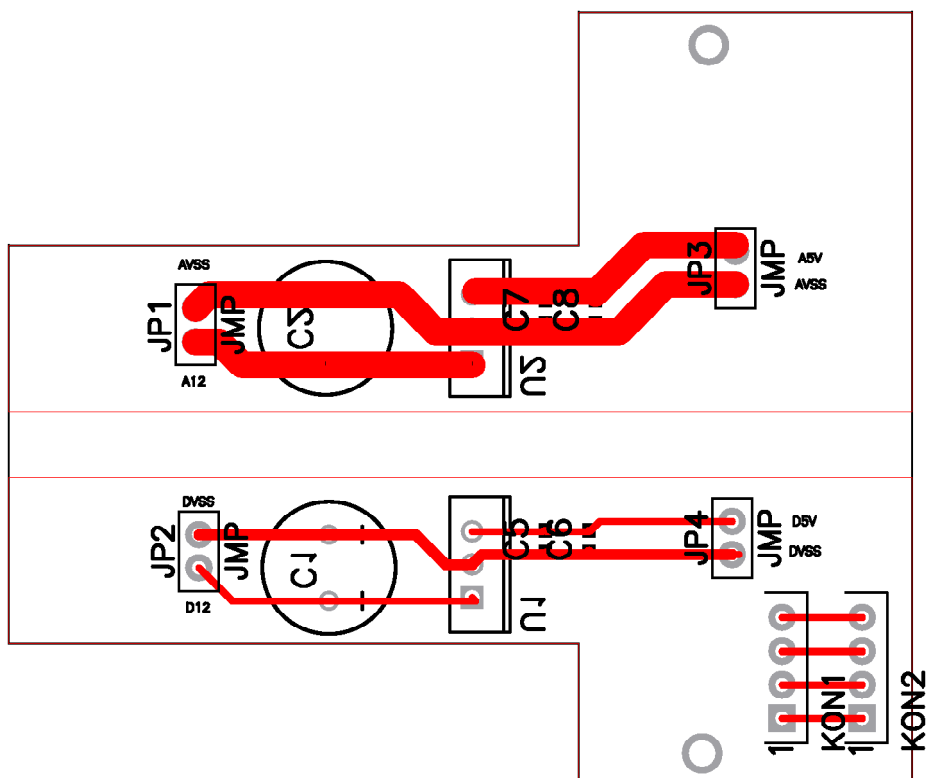
Slika 12: BOT tiskanine merilne ploščice



Slika 13: Realizirana merilna ploščica



Slika 14: Realizirana merilna ploščica z elementi



Slika 15: TOP napajalno vezje



Slika 16: Merilni sistem



Slika 17: Sestavne komponente merilnega sistema