

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1934

Akvizijski sustav iznad Nyquistove granice

Ivana Marinović

Zagreb, veljača 2020.

*Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

bla bla

SADRŽAJ

1. Uvod	1
2. Teorija sažimajućeg očitavanja	2
3. Sklopovska realizacija akvizicijskog sustava	4
3.1. Model sustava sažimajućeg očitavanja	4
3.2. Ispitivanje i rad kanala	6
3.2.1. Pretpojačalo	7
3.2.2. Digitalno-analogni pretvornik (DAP)	8
3.2.3. Integrator i sklop za prilagodbu signala	10
3.2.4. Analogno-digitalni pretvornik (ADP)	13
4. Razvoj sklopovske potpore na Zynq-7000 FPGA/ARM sustavu	14
4.1. AXI sučelje	15
4.1.1. AXI4-Lite sučelje	16
4.1.2. Implementacija <i>spi_ctrl</i> VHDL bloka	18
4.1.3. Implementacija <i>transfer_ctrl</i> VHDL bloka	22
5. Zaključak	25

1. Uvod

Uvod rada. Nakon uvoda dolaze poglavlja u kojima se obrađuje tema.

2. Teorija sažimajućeg očitavanja

Tradicionalni pristupi uzorkovanja signala i slika slijede Shannonov teorem koji kaže da frekvencija uzorkovanja mora biti bar dvostruko veća od najveće frekvencije prisutne u signalu koja se još naziva i Nyquistova frekvencija. Ovom principu podliježu gotovo svi sustavi prikupljanja podataka koji se koriste u audio i vizualnoj elektronici, medicinskim uređajima, komunikacijskim tehnologijama, itd. Analogno-digitalni pretvornici prevode fizičku informaciju, odnosno analogni signal u tok podataka omogućujući digitalnu obradu raznim programskim rješenjima. Standardni analogno-digitalni pretvornici jednoliko uzorkuju signale istom ili većom frekvencijom od Nyquistove granice.[intro] Zbog sve većih zahtjevi za sustavima visoke granične frekvencije i male potrošnje energije na tržištu se pojavljuju analogno-digitalni pretvornici koji su u granicama elektroničkih mogućnosti. Osim tog, postavljaju se zahtjevi za što boljim procesorima namjenjenima za digitalnu obradu signala (engl. *Digital signal processor, DSP*) i medijima za pohranu memorije. Postupak sažimajućeg očitavanja potpuno je drugačiji pristup u digitalnoj obradi signala koji omogućuje uzorkovanje signala i njegovu rekonstrukciju uz puno manji broj mjerenja u odnosu na klasično uzorkovanje. Sažimajuće očitavanje (engl. *compressive sensing*) nova je teorija u okviru teorija uzorkovanja signala, a pretpostavlja da signal može biti rekonstruiran s određenom statističkom točnošću iz puno manjeg broja uzoraka nego što Shannon-Nyquistov teorem zahtjeva.(jele) Signal koji se mjeri mora biti rijedak ili kompresibilan.

Rijetkost signala predstavlja činjenicu da je informacijski sadržaj vremenski kontinuiranog signala sažet kad je prikazan u odgovarajućoj transformacijskoj bazi, odnosno signal je K rijedak ako sadržava mali broj uzoraka ($K \ll N$) različitih od nule. Tada se signal može prikazati kao linearna kombinacija samo K vektora baze. $\square$

Rijetkost signala povezuje se sa svojstvom kompresibilnosti, odnosno svojstvo da signal u transformacijskoj bazi sadrži samo nekoliko značajnih koeficijenata, dok su ostali zanemarivo mali ili jednaki nuli. Kompresibilan signal moguće je

predstaviti sa samo K najznačajnijih koeficijenata, dok se ostali mogu odbaciti bez gubitka kvalitete. Sažeto očitavanje iskorištava činjenicu da mnogi prirodni signali zadovoljavaju uvjete rijetkosti i kompresibilnosti, odnosno mogu se prikazati sažetim zapisom u odgovarajućoj bazi. [intro]

Mjerenje u postupku sažetog očitavanja odnosi se na množenje signala s mjernom matricom. Za razliku od klasičnog uzorkovanja provodi se izravna kompresija nad signalom. Akvizicija signala provodi se u obliku

$$y = \varphi x \quad (2.1)$$

gdje je $x \in \mathbb{R}^N$ ili $\mathbb{C}^N$ rijedak ulazni signal duljine N , $\varphi \in \mathbb{R}^{M \times N}$ mjerna matrica dimenzija $M \times N$, a $y \in \mathbb{R}^M$ ili $\mathbb{C}^M$ mjerni vektor duljine M koji je puno manje duljine od ulaznog signala ($M \ll N$). Mjereni signal može biti rijedak samo u nekoj općenitoj bazi transformacije te se prethodna jednadžba može zapisati

$$x = \Psi s \quad (2.2)$$

$$y = \varphi x = \varphi \Psi s = \Theta s \quad (2.3)$$

gdje je $\Psi \in \mathbb{R}^{N \times N}$, $s \in \mathbb{R}^n$ ili $\mathbb{C}^n$ vektor koeficijenata s_i signala x u bazi Ψ , a $\Theta = \varphi \Psi$ je matrica sažetog očitavanja. Prema tome, signal bi se mogao rekonstruirati iz dobivenog mjernog vektora rješavanjem jednadžbe 2.1

$$x = y \varphi^{-1} \quad (2.4)$$

S obzirom da se radi o poddefiniranom sustavu linearnih jednadžbi koji ima beskonačno mnogo rješenja nameću se različiti algoritmi rješavanja ovakvog sustava. Uz mjerni vektor y i poznatu mjernu matricu ψ rekonstrukcijski problem svodi se na traženje signala x koji bi u potpunosti ili približno zadovoljio jednadžbu 2.1. [theory to app]

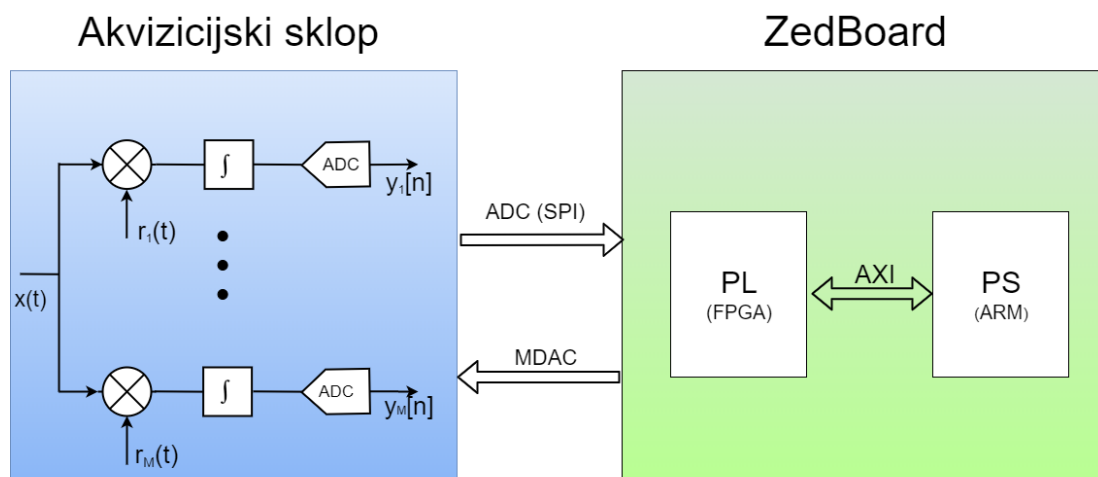
Jedno od predloženih rješenja je konveksna norma je ℓ_1 tako da se realizira rekonstrukcijski algoritam koji se temelji na minimizaciji ℓ_1 norme $\hat{s} = \arg \min_s \|s\|_1$. Rješenje dobiveno takvim postupkom će imati minimalnu sumu apsolutnih vrijednosti koeficijenata. Dokazano je da je uz postupak minimizacije ℓ_1 norme dovoljan broj mjerenja za stabilnu rekonstrukciju $M = O(K \log(N/K)) \ll N$. [karlo, donoho]

3. Sklopovska realizacija akvizicijskog sustava

Na temeljima teorijske podloge sažimajućeg očitavanja osmišljen je sustav koji radi akviziciju i rekonstrukciju signala puno manjom frekvencijom očitavanja nego što je prihvaćena u klasičnim mjerenjima signala. Radi se o hibridnom digitalno-analognom sustavu koji se sastoji od razvojnog sustava ZedBoard i akvizicijskog sklopa sažetog očitavanja. [Vito] U ovom poglavlju bit će opisan cjeloukupni model razvijenog sustava, a naglasak će biti u opisu i uhodavanju analognog sustava čija je zadaća akvizicija odnosno prikupljanje uzoraka mjerenog signala.

3.1. Model sustava sažimajućeg očitavanja

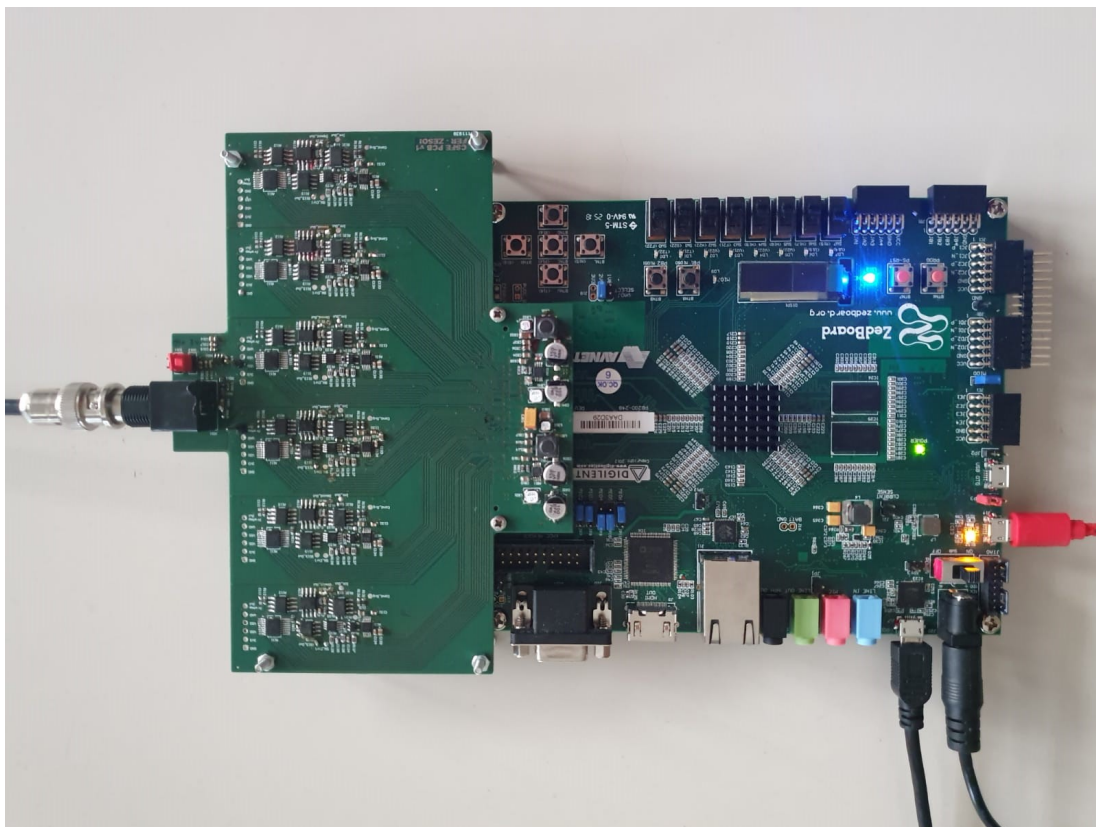
Na slici 3.1 prikazana je pojednostavljena blok shema cjelokupnog sustava za sažimajuće očitavanje koji je razvijen. Akvizicijski sklop mjeri ulazni signal i sažeto ga pohranjuje pomoću mjerne matrice. Sastoji se od šest identičnih kanala na koji se dovodi ulazni signal. Svaki od tih kanala radi množenje signala preko digitalno-analognog pretvornika, integraciju i konačno uzorkovanje analogno-digitalnim pretvornikom. Digitalno-analogni pretvornik množi ulazni signal s 8 bitnom vrijednosti koja predstavlja jedan član retka mjerne matrice φ iz jednadžbe 2.1. Množenje signala sa svih šest kanala istovremeno rezultira mjernim vektorom y . Rad jednog kanala i njegova sklopovska potpora bit će opisana detaljnije u potpoglavlju 3.2. Na drugoj strani sustava nalazi se razvojni sustav Zedboard. Veza između akvizicijskog sklopa i ZedBoarda ostvaruje se serijskim perifernim sučeljem (engl. *serial peripheral interface*, *SPI*) u slučaju analogno digitalnog pretvornika i paralelnim prijenosom prijenosom 8-bitnih podataka na ulaze digitalno-analognih pretvornika.



Slika 3.1: Blok shema sustava za sažimajuće očitavanje

Razvojni sustav ZedBoard temeljen je na Zynq-7000 ARM/FPGA SoC (engl. *System on Chip*) arhitekturi. Radi se o implementaciji integriranog sklopa u kojoj se na istoj silicijskoj pločici nalaze PS (engl. *Processing system*) i PL (engl. *Programmable Logic*). ZedBoard razvojni sustav istovremeno povezuje dvojezgreni procesorski sustav ARM Cortex-A9 i Xilinx programabilnu logiku temeljenu na FPGA (engl. *Field Programmable Gate Array*) tehnologiji u jedan čip. Kao što je već spomenuto uloga sustava sažimajućeg očitavanja je akvizicija ulaznog signala i njegova rekonstrukcija iz dobivenih uzoraka. Procesorski sustav (PS) zadužen je za rekonstrukciju signala, a dizajnom programabilne logike (PL) ostvarena je sklopovska potpora akvizicijskoj pločici i AXI (engl. *Advanced eXtensible Interface*) sučelje preko kojeg komuniciraju PS i PL dio ZedBoard razvojnog sustava. Na slici 3.2 prikazan je sustav za sažimajuće očitavanje u stvarnosti, odnosno šesterokanalni analogni akvizicijski sklop koji je preko FMC konektora povezan na ZedBoard. Ulazni signal dovodi se na BNC konektor.

Dizajn razvijenog akvizicijskog sustava implementiranog u PL dijelu Zedboarda izveden je u razvojnoj okolini Vivado Design Suite koristeći VHDL (engl. *Very High Speed Integrated Circuit Hardware Description Language*), formalni jezik za opis sklopovlja. Aplikacija koja izvodi rekonstrukcijski algoritam temeljen na statističkom sažimajućem očitavanju razvijena je u Xilinx SDK programskom okruženju, a izvodi se u PS dijelu Zedboarda.

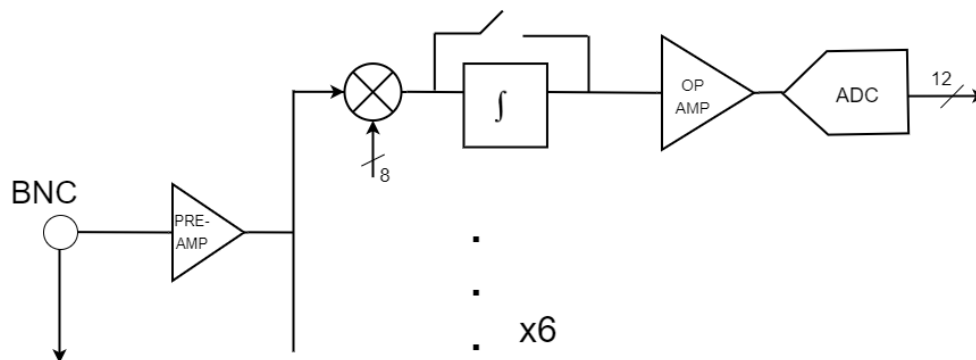


Slika 3.2: Šesterokanalni akvizicijski sklop priključen na ZedBoard razvojni sustav

3.2. Ispitivanje i rad kanala

Akvizicijski sklop za sažeto očitavanje signala koji se sastoji od šest kanala elektronički je dizajniran u sklopu diplomskog seminara *Akvizicijski sklop sustava sažimajućeg očitavanja*.¹ Prilikom izrade ovog diplomskog rada najprije se ispitao rad jednog kanala postepenim uhodavanjem svih elektroničkih komponenti. Nakon što se potvrdila ispravnost jednog kanala akvizicijski sklop bio je spreman za ostatak lemljenja. Za bolje razumijevanje u nastavku je detaljno objašnjen tok signala kroz komponente jednog kanala. S obzirom da je za funkcionalnost nekih integriranih elektroničkih sklopova na akvizicijskoj pločici potrebno upravljanje i generiranja raznih signala u kasnijim poglavljima detaljnije će se razraditi razvoj digitalnog dizajna u VHDL-u. Na slici 3.3 prikazana je pojednostavljena elektronička shema jednog od šest kanala.

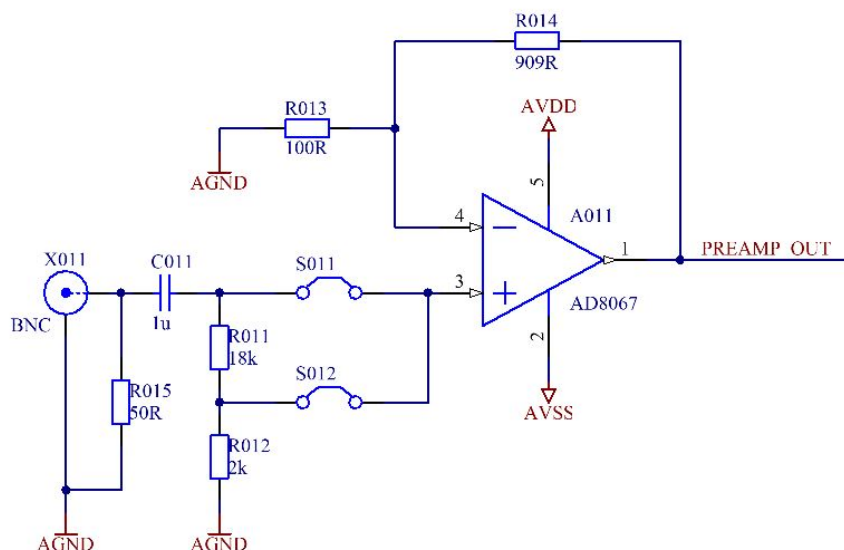
¹V. Papa, „Akvizicijski sklop sustava sažimajućeg očitavanja“, Diplomski seminar, FER 2019



Slika 3.3: Pojednostavljena elektronička shema jednog kanala akvizicijskog sklopa

3.2.1. Pretpojačalo

Ulazni signal se na akvizicijski sklop dovodi preko BNC konektora i prosljeđuje se na pretpojačalo AD8067² kojem se položajem kratkospojnika može mijenjati pojačanje. Na slici 3.4 prikazana je električna shema pretpojačala. Kratkospojnikom S011 odabire se slijedno pojačanje, a kratkospojnikom S012 faktor pojačanja namješta se na 10. Kondenzator C01 je postavljen za uklanjanje istosmjerne komponente signala kako ne bi dolazilo do pogreške u konačnom očitavanju rezultata.



Slika 3.4: Električna shema pretpojačala

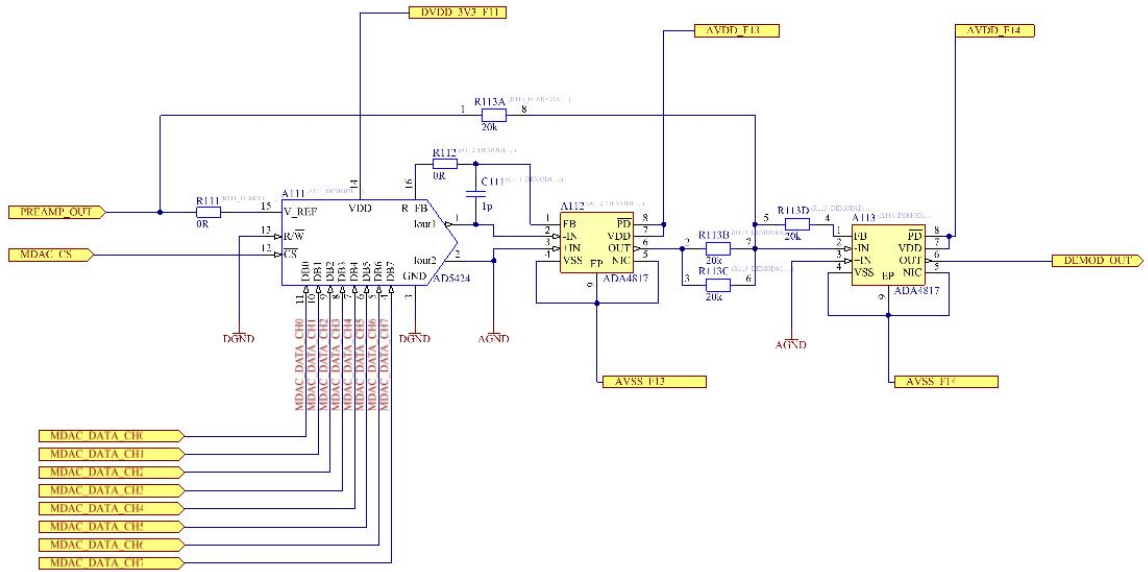
²Analog Devices, AD8067

3.2.2. Digitalno-analogni pretvornik (DAP)

Na početku svakog kanala nalazi se 8-bitni digitalno-analogni pretvornik AD5425 s paralelnim sučeljem. Za ostvarenje četverokvadratnog množenja prilikom realizacije sklopa potrebno je uključiti dodatna pojačala i otpornike kao što je to prikazano na slici 3.5. Prvo pojačalo ima ulogu pretvorbe signala s obzirom da je izlaz digitalno-analognog pretvornika strujnog oblika, a drugo pojačalo na svom izlazu daje signal koji je pojačan dva puta. [datasheet] Prijenosna funkcija ovako realiziranog digitalno-analognog pretvornika opisuje se

$$V_{OUT} = (V_{IN} * D/2^{n-1}) - V_{IN} \quad (3.1)$$

gdje D predstavlja frakcionalnu reprezentaciju digitalne riječi koja se dovodi na pinove DAP-a, a n označava njegovu razlučivost. U ovom radu koristi se 8-bitni DAP koji je zapravo realiziran kao programabilno pojačalo (engl. programmable-gain amplifier, PGA) te ima mogućnost četverokvadratnog množenja signala s 256 različitih razina u rasponu od -1 do 1.

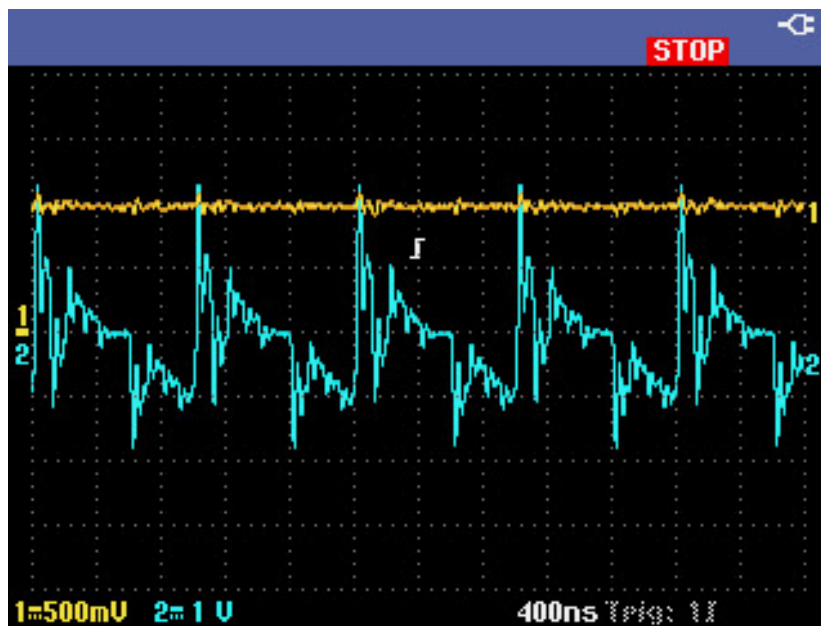


Slika 3.5: Digitalno-analogni pretvornik u funkciji PGA

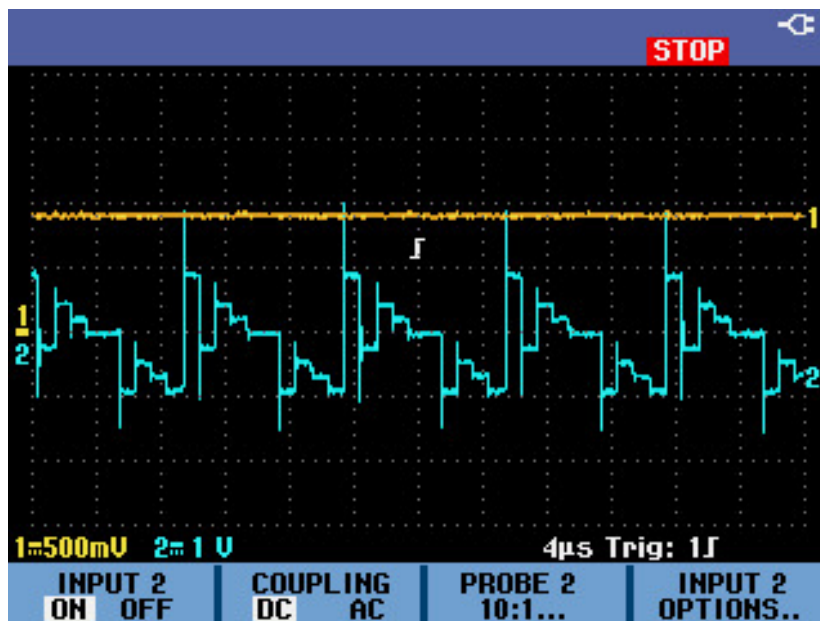
Frekvencija kojom DAP množi ulazni signal s danom 8-bitnom vrijednosti je deset puta veća od frekvencije uzorkovanja ADP-a koji se nalazi na kraju svakog kanala pa se za vrijeme perioda uzorkovanja ADP-a signal pomnoži s deset 8-bitnih vrijednosti. Sekvenca od deset različitih vrijednosti razlikuje se za svaki kanal, a periodički se ponavlja svakih $1\mu s$. Ovakvim pristupom ostvarena

je osnovna funkcija akvizicijskog sklopa, odnosno množenje ulaznog signala s mjernom matricom. Slijed od deset različitih vrijednosti u $1\mu s$ predstavlja jedan redak mjerne matrice iz čega slijedi da je realizirana mjerna matrica $\varphi \in \mathbb{R}^{6 \times 10}$.

FPGA dizajn implementiran u PL dijelu razvojnog sustava Zedboard omogućuje brzo mijenjanje 8-bitnih podataka koji se paralelno dovode na ulaze DAP-ova. [Lucija] Mjerna matrica unaprijed se definira prije svakog mjerenja, a vrijednosti se dohvaćaju iz PS dijela ZedBoard razvojnog sustava preko AXI sučelja koji će biti opisan detaljnije[]. Prilikom uhodavanja i provjeravanja ispravnosti DAP-a na ulaz je doveden DC signal amplitude 1 V. Na slici 3.6 prikazano je množenje ulaznog signala s deset različitih razina koje se ponavljaju svakih $1\mu s$, a na slici 3.7 prikazano je množenje ulaznog signala svakih $1\mu s$ unutar vremenskog okvira koji traje $10\mu s$. U prvom slučaju frekvencija rada DAP-a iznosi 10 Mhz, a u drugom 1 Mhz. Kao što je očekivano, uočena je velika razlika valnih oblika izlaznih signala ovisno o frekvenciji rada DAP-a. Nakon DAP-a izlazni signal se prosljeđuje na integrator koji bi trebao kompenzirati sve nepravilnosti.



Slika 3.6: Izlazni signal DAP-a, frekvencija rada 10 Mhz



Slika 3.7: Izlazni signal DAP-a, frekvencija rada 1 Mhz

3.2.3. Integrator i sklop za prilagodbu signala

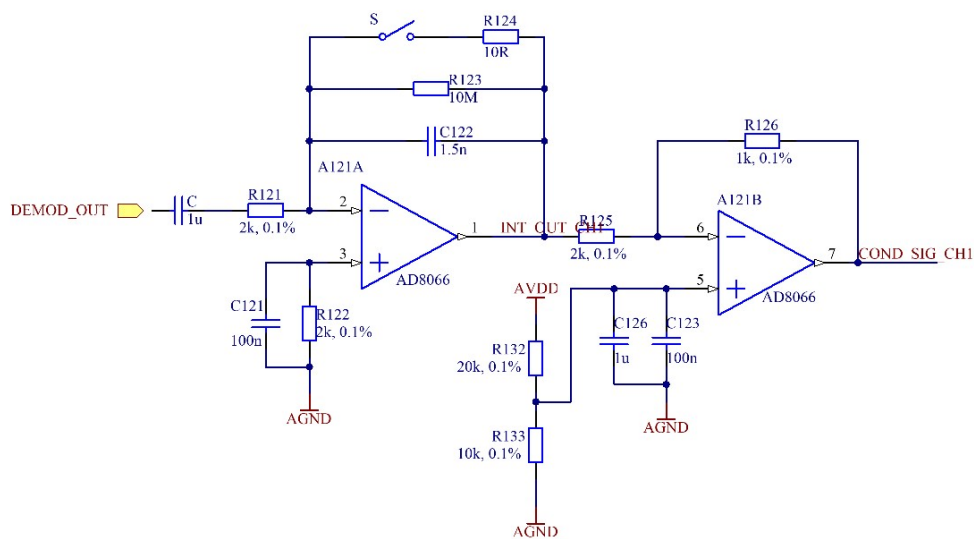
Digitalno-analogni pretvornici na akvizicijskom sklopu matrično množe ulazni signal s mjernom matricom i podatke proslijeđuju na integratore koji integriraju deset uzoraka s periodom $1\mu s$. Rezultat integracije jednog kanala je jedan član mjernog vektora y . Na slici 3.8 prikazana je električna shema integratora i sklopa za naponsku prilagodbu integriranog signala koji ulazni signal prilagođava rasponu napona pune skale ADP-a.

Resetiranje integratora realizirano je sklopkama ADG601³. Signal koji kontrolira uključivanje i isključivanje sklopki S prvo se dovodi na komparator TLV320⁴ radi smanjenja šuma same sklopke. [Vito] Generiranje kontrolnih signala koji upravljaju pražnjenjem integratora ostvareno je efektom „trčućeg svjetla“ u PL dijelu ZedBoard-a.[Lucija]

Integrirani signal INT_OUT prilagođava se rasponu napona pune skale [0-5 V] ADP-a preko pojačala A121B. Izlazni signal COND_SIG ima dvostruko manju amplitudu i superponiran DC offset od 2.5V u odnosu na integrirani signal INT_OUT. Time je omogućena analogno-digitalna pretvora pozitivnih i negativnih članova mjernog retka y .

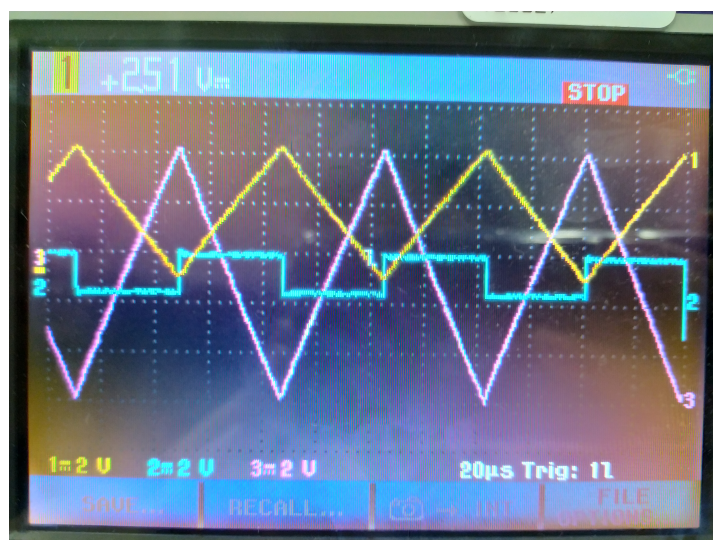
³Analog Devices, sklopka ADG601BRMZ7

⁴Analog Devices, komparator TLV3201AIDBVR



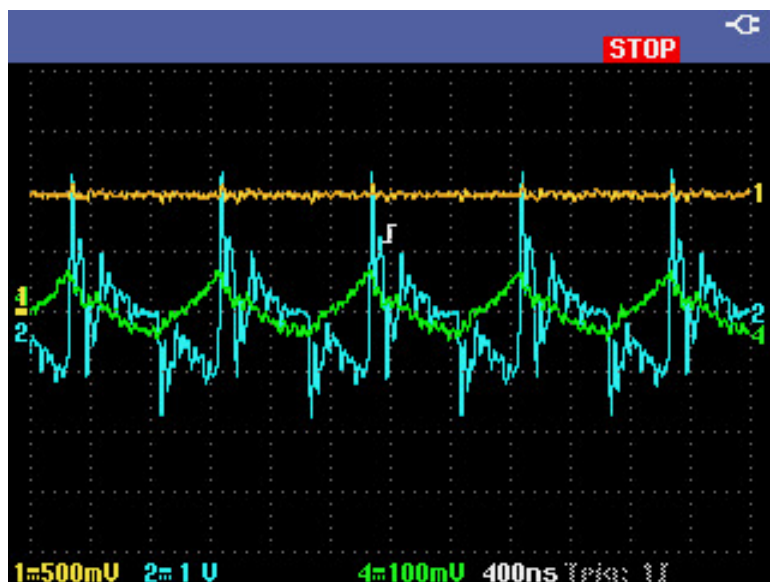
Slika 3.8: Integrator i sklop za prilagodbu signala

Na slici 3.9 prikazani su rezultati uhodavanja integratora i sklopa za naponsku prilagodbu integriranog signala na zaslonu digitalnog osciloskopa Tektronix THS3024. Radi jednostavnosti na ulaz je doveden pravokutni signal (Hz) koji je prikazan plavom bojom, žuti signal INT_OUT predstavlja izlaz iz integratora, a ljubičasti signal izlaz iz pojačala A121B, odnosno COND_SIG. Kao što je vidljivo na slici, izlazni signal sklopa za naponsku prilagodbu COND_SIG ima dvostruko manju amplitudu i DC pomak od 2.5 V. Detaljnijim proučavanjem signala i promjenom ulaznog signala potvrđena je ispravnost svih komponenti i željena funkcionalnost tog dijela sklopovlja.

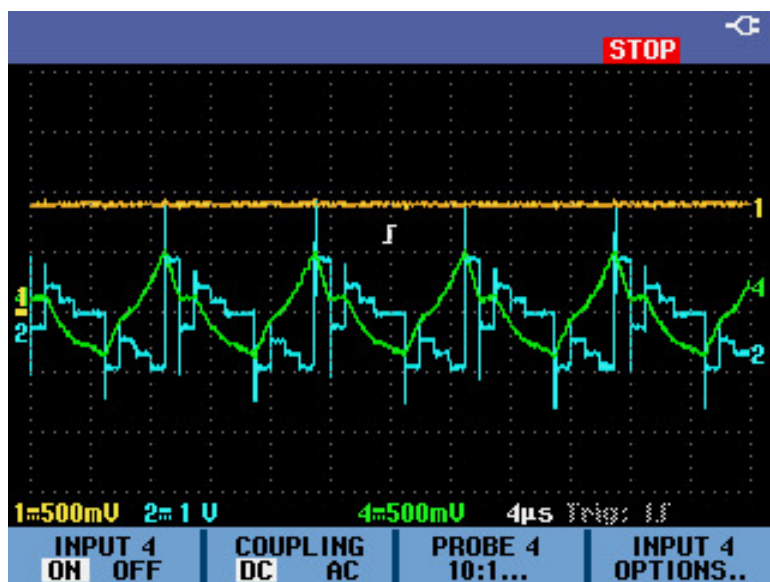


Slika 3.9: Ifmfmmf

Na slikama 3.10 i 3.11 prikazano je integriranje izlaznog signala DAP-a koji je dobiven množenjem ulaznog DC signala amplitude 1 V s valnim oblikom koji se ponavlja svakih $1\mu s$, odnosno svakih $10\mu s$. Ulazni signal DAP-a označen je žutom, izlazni signal plavom, a integrirani signal zelenom bojom.



Slika 3.10: Signal na izlazu DAP-a i integrirani signal, frekvencija rada DAP-a 10Mhz

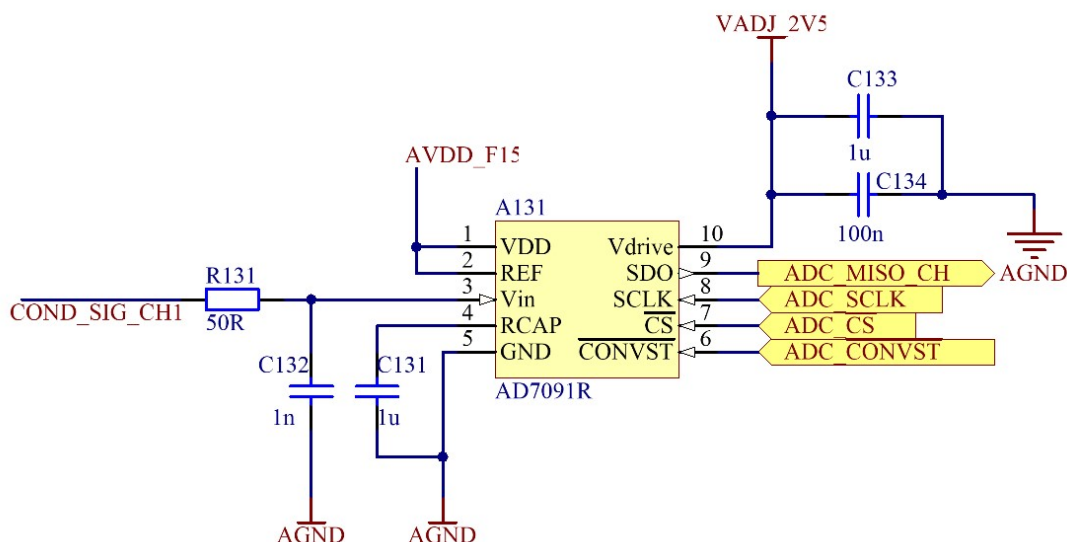


Slika 3.11: Signal na izlazu DAP-a i integrirani signal, frekvencija rada DAP-a 1 Mhz

3.2.4. Analogno-digitalni pretvornik (ADP)

Analogno-digitalni pretvornik AD7091R⁵ nalazi se na izlazu iz akvizicijskog sklopa te podatke na izlazu prosljeđuje ZedBoard sustavu za obradu podataka. Radi se o 12-bitnom sukcesivno aproksimativnom ADP-u koji radi s frekvencijom uzorkovanja 1 MHz. Izlaz analogno-digitalnog pretvornika ADC_MISO_CH prosljeđuje se PL dijelu ZedBoard-a preko serijskog perifernog sučelja (engl. *Serial Peripheral Interface, SPI*). Kao što je prikazano na slici 3.12 za pravilan rad ADP-a i ostvarenje SPI komunikacije potrebna su tri signala : ADC_SCLK, ADC_CS i ADC_CONVST.

Pretvorba podatka iz analognog u digitalni započinje na padajućem bridu signala ADC_CONVST. Na padajućem bridu signala ADC_CS započinje ispis podatka na vezu, odnosno tada je potrebno generirati signal ADC_SCLK koji predstavlja takt za SPI. [Lucija] Očitavanje podataka i privremena pohrana podataka vrši se razvijenim FPGA (engl. Field Programmable Gate Array) međusklopom koji koristi serijsko periferno sučelje (SPI, engl. Serial Peripheral Interface). Razvoj digitalnog dizajna takvog međusklopa bit će objašnjen u idućim poglavljima.

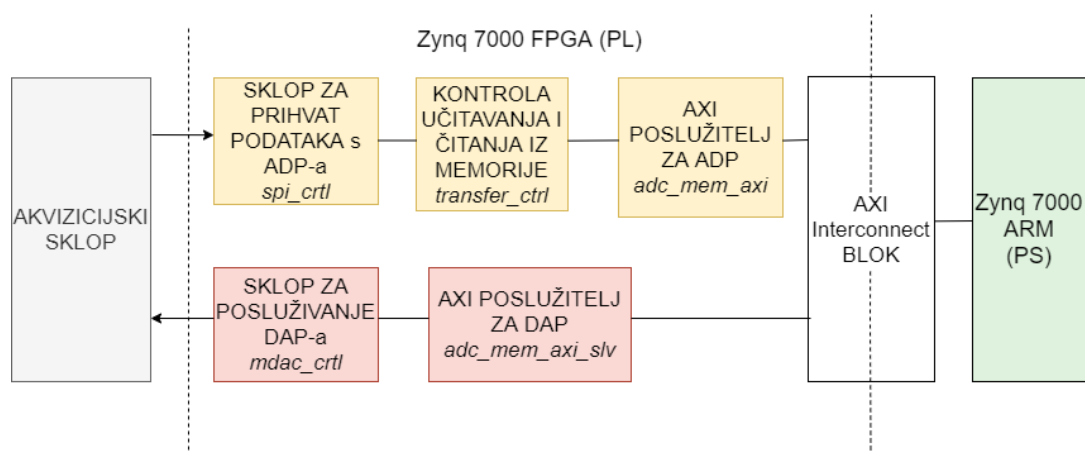


Slika 3.12: Električna shema analogno-digitalnog pretvornika

⁵Analog Devices, ADP AD7091R

4. Razvoj sklopovske potpore na Zynq-7000 FPGA/ARM sustavu

Izrada diplomskog rada zahtijevala je uhodavanje akvizicijske pločice i paralelan razvoj digitalnog dizajna za upravljanje tom istom pločicom. U ovom poglavlju naglasak će biti na sklopu realiziranom u PL dijelu Zedboard-a koji se nalazi u središtu cijelog sustava kao što je to prikazano na slici 4.1 pa s razlogom možemo govoriti o FPGA međusklopu. Osnovna uloga razvijenog sustava je osigurati akvizicijskom sklopu pristup mjernoj matrici te omogućiti paralelno očitavanje i pohranu izmjerenih podataka s njegovih serijskih izlaza. S obzirom da se mjerna matrica dohvaća iz PS-a, a izmjereni podaci prosljeđuju PS-u na kojem se izvršava daljnja rekonstrukcija potrebno je ostvariti komunikaciju između programabilne logike i procesorskog sustava. To je omogućeno AXI protokolom, a korišten je AXI4-Lite, jednostavna inačica AXI4 za komunikaciju. ZedBoard razvojna pločica pokazala se kao idealno rješenje za razvoj ovakvog sustava jer se FPGA i procesorski sustav nalaze na istom čipu pa je omogućena velika propusnost i velike brzine prijenosa podataka među njima.



Slika 4.1: FPGA komponente u kontekstu cijelog sustava

4.1. AXI sučelje

Na ZedBoardu PS i PL komuniciraju putem AXI (engl. *Advanced eXtensible Interface*) protokola koji omogućava siguran prijenos podataka iz procesorskog sustava prema programabilnoj logici, i obrnuto. AXI protokol je temeljen na AMBA sabirnici (engl. *Advanced Microcontroller Bus Architecture*) za ARM porodicu procesora posebno dizajniranu za visoke performanse i rad na visokim frekvencijama. [axi] AXI protokol je otvoreni standard i već godinama se koristi u industriji. Omogućuje spajanje više nadređenih sklopova (engl. Master) i više podređenih sklopova (engl. Slave) u jedan sustav. Posebno je optimiziran za FPGA implementaciju i koristi se za međusobno povezivanje sklopovlja u programabilnoj logici i sustavima na čipu. [H:B] Transakcije između AXI blokova odvijaju se preko pet AXI kanala :

- Adresni kanal za čitanje (engl. *Read Address Channel*)
- Podatkovni kanal za čitanje (engl. *Read Data Channel*)
- Adresni kanal za pisanje (engl. *Write Address Channel*)
- Podatkovni kanal za pisanje (engl. *Write Data Channel*)
- Kanal za potvrdu upisa (engl. *Write Response Channel*)

Kad nadređeni sklop želi upisati podatak u podređeni koriste se kanal za prijenos adrese na koju se podatak upisuje (engl. *Write Address Channel*), kanal za prijenos tog podatka (engl. *Write Data Channel*) i kanal za potvrdu primitka podatka (engl. *Write Response Channel*) preko kojeg podređeni sklop javlja nadređenom potvrdu upisa. Ako nadređeni sklop želi pročitati podatak iz podređenog, odnosno ako podređeni želi upisati podatak u nadređeni sklop koriste se kanal za prijenos adrese s koje se čita (engl. *Read Address Channel*) i kanal za slanje tog podatka (engl. *Read Data Channel*).

U sklopu diplomskog rada realiziran je sustav u kojem je nadređeni sklop, odnosno *Master*, procesorski sustav na ZedBoard-u, a za svaki kanal postoje dva podređena sklopa, odnosno dva *Slave-a* koja su razvijena u Vivado razvojnom okruženju. AXI Interconnect jezgra povezuje jedan *Master* s 12 *Slave* sklopova te se time ostvaruje sva komunikacija akvizicijske pločice i procesorskog sustava na ZedBoardu.

4.1.1. AXI4-Lite sučelje

AXI sučelje omogućava vezu nadređenog i podređenog sklopa pomoću 19 signala podijeljenih u već spomenutih pet kanala.

```
-- AXI4-Lite bus signals
awaddr    : in      std_logic_vector(31 downto 0);
awprot     : in      std_logic_vector( 2 downto 0);
awvalid    : in      std_logic;
awready    : out     std_logic;

wdata      : in      std_logic_vector(31 downto 0);
wstrb      : in      std_logic_vector( 3 downto 0);
wvalid     : in      std_logic;
wready     : out     std_logic;

bready     : in      std_logic;
bresp      : out     std_logic_vector( 1 downto 0);
bvalid     : out     std_logic;

araddr     : in      std_logic_vector(31 downto 0);
arprot     : in      std_logic_vector( 2 downto 0);
arvalid    : in      std_logic;
arready    : out     std_logic;

rready     : in      std_logic;
rdata      : out     std_logic_vector(31 downto 0);
rresp      : out     std_logic_vector( 1 downto 0);
rvalid     : out     std_logic;
```

Kanal za prijenos adrese na koju se upisuje ima 3 signala, a njihova imena imaju prefiks *aw*. Slovo *a* pokazuje da se radi o kanalu kojim se šalje adresa, a slovo *w* označava transakciju pisanja (engl. *write*). *Awaddr* je 32-bitni signal za prijenos adrese na koju se upisuje podatak. *Awready* je signal kojim *slave* javlja da je spreman primiti podatak tog kanala, odnosno adresu, a *awvalid* je signal kojim *master* javlja da je podatak spreman za slanje.

Nakon uspješnog slanja adrese, šalje se podatak kanalom za slanje podatka. Kanal ima 4 signala koji počinju sa slovom *w* što označava transakciju pisanja. *Wdata* je 32-bitni signal za slanje podatka. *Wstrb* je 4-bitni podatak koji označava koji bajtovi podatka predstavljaju korisnu informaciju koja se šalje. Na primjer, ukoliko je vrijednost podatka 0011 to znači da se gledaju zadnja dva bajta.

Treći kanal je kanal za potvrdu primitka podatka, označava se prefiksom *b*.

Signal *bresp* dvobitni je signal kojim *slave* javlja uspješnost primitka podatka.

Proces čitanja sličan je procesu pisanja. Prvo se koristi kanal za slanje adrese s koje se podatak čita. Kanal ima prefiks *ar* gdje *r* označava čitanje (engl. *read*). *Araddr* je signal veličine 32 bita preko kojeg se prenosi adresa.

U kanalu za slanje podatka podatak se šalje signalom *rdata* koji ima 32 bita. Dvobitni signal *rresp* služi za javljanje nadređenom sklopu je li transakcija uspješno završila. [LO]

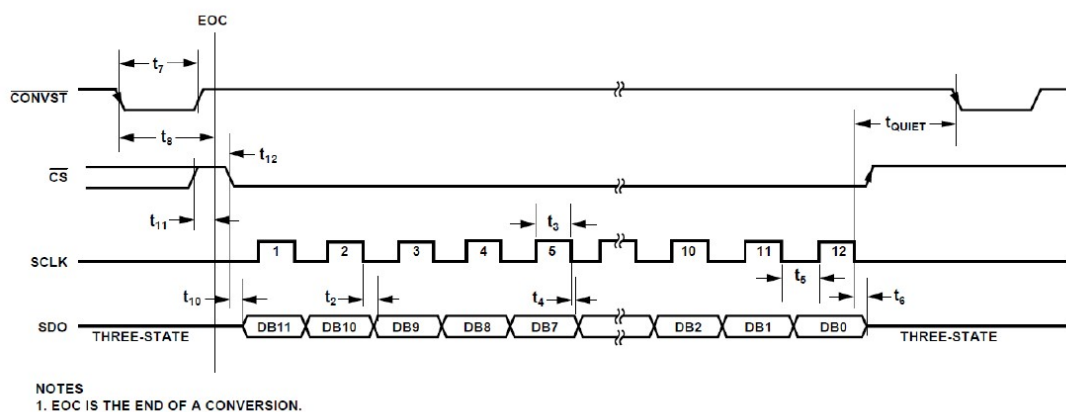
Važnu ulogu u svakom kanalu imaju upravljački signali za dvožično rukovanje (engl. *handshaking*), a to su signal (engl. *valid*) kojim pošiljalac potvrđuje da je podatak ili adresa na podatkovnom ili adresnom kanalu valjana i da primalac može obaviti prihvatanje adrese ili podatka, a drugi signal (engl. *ready*) koristi primalac kako bi označio da je spreman za primitak adrese ili podatka. Komunikacija se uspostavlja tek kad su oba signala aktivna. Prvo se šalju adrese memorijskih lokacija na kojima će se izvršavati neka od radnji, a onda tek podatak koji se šalje na prvi rastući brid takta nakon što oba signala postanu aktivni. [LO]

Implementacija AXI-Lite sučelja u VHDL-u je jako zahtjevna i treba se držati cijelog niza pravila koji su detaljno objašnjeni u Xilinx službenim dokumentima¹. Kako sama implementacija nije bila u fokusu ovog rada već je bilo potrebno prilagoditi standardno AXI-Lite sučelje za konkretne zadatke koje se odnose na akvizicijski sustav tako se i pristupilo opisivanju ovog rješenja.

¹Vivado AXI Reference Guide

4.1.2. Implementacija *spi_ctrl* VHDL bloka

Analogno-digitalni pretvornici koji se koriste na akvizicijskom sklopu kompatibilni su sa SPI, QSPI i MICROWIRE serijskim sučeljima. Zbog jednostavnosti, u ovom radu za komunikaciju s ADP-ovima implementirano je SPI sučelje. Točnije VHDL komponenta *spi_ctrl* je zadužena za ostvarivanje komunikacije i generiranje potrebnih signala prema ADP-u. Kad se ZedBoard isprogramira, VHDL komponenta *spi_ctrl* postaje pravi sklop realiziran u FPGA tehnologiji te u ovoj komunikaciji dobiva ulogu *master-a*, dok je ADP *slave*. Podaci koji stižu SPI sučeljem prenose se bit po bit, odnosno serijskom vezom te ih je potrebno akvizirati u riječi pravilnim redoslijedom. Broj bitova koje sadrži jedna riječ je 12, a stižu na svaki padajući brid signala SCLK koji dolazi iz *master* sklopa.

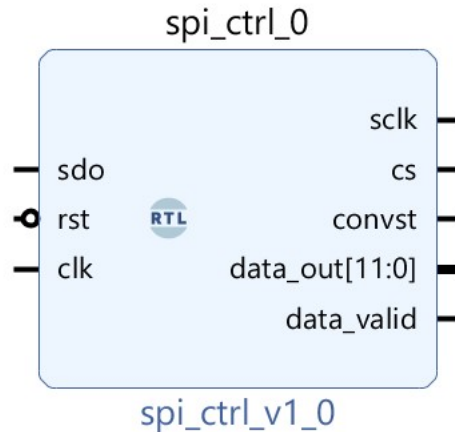


Slika 4.2: Vremenski dijagram ADP

Na slici 4.2 prikazan je vremenski dijagram preuzet iz službene dokumentacije AD pretvornika na kojem su prikazani signali kojima se upravlja AD pretvornikom. Prilikom dizajna *master* komponente trebalo je ispuniti sve zahtjeve i pravila koja se odnose na vremenska trajanja pojedinih signala. Padajući brid signala $\overline{CONVST}$ označava početak A/D pretvorbe te ga je potrebno postaviti u '1' prije kraja pretvorbe koji je na dijagramu označen s EOC(engl. *end of conversion*). Vrijeme trajanja konverzije je 650 ns kako piše u tehničkoj dokumentaciji ADP-a. Signal $\overline{CS}$ omogućuje vezu prema pojedinom ADP-u.

VHDL blok *spi_ctrl* ima 3 ulaza i 5 izlaza, a to je vidljivo i na slici ispod. Preko priključka *sdo* dovodi se izlazni signal ADP-a, odnosno niz bitova. *Clk* je standardni ulaz ako se želi ostvariti sinkron sklop, a *rst* se koristi u slučaju reseta cijelog sustava. Kao što je spomenuto ranije signali *sclk*, *cs* i *convst* upravljaju

radom ADP-a i oni se preko FMC konektora dovode do fizičkih pinova ADP-a.



Slika 4.3: Entitet *spi_ctrl* VHDL bloka

Kako bi se ulazima i izlazima mogli dodijeliti fizički priključci potrebno ih je definirati u *constraint* datoteci. Povezivanje komponenti i dodavanje priključaka slijedi nakon opisivanja pojedinih komponenti. Izlazni podatak *data_out* širine 12 bitova predstavlja izlaznu riječ, odnosno u ovom slučaju izmjerenu vrijednost mjernog vektora. *Data_valid* se postavlja u '1' kad je izlazna riječ ispravna za daljnje prosljeđivanje.

U tablici 4.1 dan je popis vremenskih konstanti koje su definirane u VHDL kodu. Cijela logika ovog sklopa ovisi o brojaču *count* koji povećava svoju vrijednost na svaki rastući brid takta *clk* sve dok ne dođe do kraja perioda, odnosno dok ne postane jednak konstanti *T*. Nakon cijelog perioda, *count* se vraća u '0' i proces kreće ispočetka. Ovisno o stanju brojača *count* generiraju se potrebni signali kako je prikazano u odsječku koda.

Tablica 4.1: Definirane konstante

Konstanta	Vrijednost	Vrijeme[ns]
<i>t7</i>	"1010"	100
<i>teoc</i>	"1000011"	670
<i>tclk_start</i>	"1000110"	700
<i>tclk_stop</i>	"1011110"	950
<i>T</i>	"1100100"	1000

```

if count = T then
    count <= (others => '0');
    convst_s <= '0';
    cs_s <= '1';
    sclk_s <= '0';
else
    count <= count +1;
if count = "00000000" then
    convst_s <= '0'; -- falling edge of convst starts an ADC
                        conversion
elsif count = t7 then
    convst_s <= '1'; -- convst must be held high after 100 ns
elsif count = teoc then
    cs_s <= '0';      -- cs must be low to enable communication

```

Na kraju svakog perioda uzorkovanja signali se postavljaju u inicijalne vrijednosti kako bi sve bilo spremno za novu pretvorbu. Konstanta T predstavlja 1 μ s što znači da će ADP očitati milijun uzoraka u jednoj sekundi (engl. *1 MSPS, 1 Mega Samples per second*). Zahtjev da signal *convst_s* mora minimalno 10 ns biti u niskoj razini realiziran je tako da kad brojač *count* postane jednak konstanti *t7* dolazi do promjene u '1'. Nakon što prođe vrijeme pretvorbe koje je definirano konstantom *teoc* mora se omogućiti veza prema samom ADP-u spuštanjem signala *cs_s* u nisku razinu. Isječak idućeg koda samo je nastavak prethodnog, ali su odvojeni radi preglednosti. Njime se ostvaruje generiranje takta *sclk* za ADP i akviziranje bitova u riječi, odnosno uzorke. Ova faza traje od *tclk_start* do *tclk_stop*, odnosno od 700-te ns do 940-te ns jednog perioda uzorkovanja. Svaki takt *sclk* traje 20 ns.

```

elsif count >= tclk_start and count < tclk_stop then
    if count_to_12 <= 12 then
        ready <= '0';
        if count(0) = '1' then
            sclk_s <= '0';
            count_to_12 <= count_to_12 +1;
        else
            sclk_s <= '1';
            if count > 70 then
                reg_data(12-count_to_12) <= sdo; --MSB bit
            end if;
        end if;
    end if;
end if;
else

```

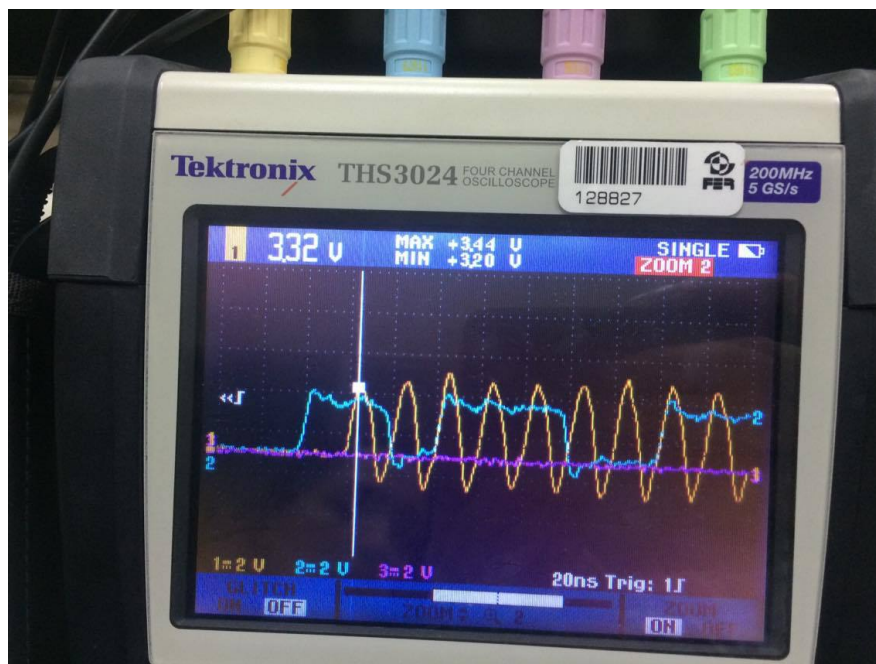
```

    reg_data(12-count_to_12)<= sdo; --LSB bit
    count_to_12 <= 0;
    sclk_s <= '0';
    ready <= '1';
end if;

```

Funkcionalnost ovog dijela koda oslanja se na brojač *count_to_12* koji se povećava na svaki drugi rastući brid takta a kad dosegne vrijednost 12 resetira se na 0. Brojač *count_to_12* zapravo se brine o redoslijedu postavljanja bitova u 12-bitne riječi. Kako je period takta ADP-a *sclk* dvostruko veći od perioda rada takta sustava za njegovo generiranje dovoljno je provjeravati najdonji bit brojača *count_to_12* i ovisno o tome postavljati '0' ili '1'. Podaci koji stižu SPI sučeljem prenose se bit po bit te se postavljaju na padajući brid takta *sclk*, a kako se ne radi o idealnom sklopu postoji vremensko kašnjenje nakon kojeg se bitovi stvarno postave na ulaz *sdo master* sklopa. Idealno, uzimanje uzorka s *sdo* priključka događa se u trenutku kad je prošlo pola perioda trajanja signala, odnosno na idućem rastućem bridu takta *sclk* kako je prikazano u isječku koda. Pritom se pazi da se vrijednost svakog bita zapisuje na odgovarajuće mjesto 12-bitnog registra koji će u konačnici sadržavati očitano riječ. Kad se upiše zadnji LSB (engl. *last significant bit*) bit podatak je spreman za slanje i postavlja se zastavica *ready*.

Generiranje signala *sclk* i *cs_s* te postavljanje bitova na priključak *sdo* snimljeno je digitalnim osciloskopom i prikazano na slici 4.4. Žuti signal predstavlja signal takta za ADP, *sclk*, plavi signal izlaz iz ADP-a, odnosno ulazni priključak *master* sklopa i ljubičasti signal *cs_s*. Posebno je vidljivo kašnjenje postavljanja bitova na *sdo* priključak u iznosu otprilike 10 ns što je ustvari pola perioda trajanja signala *sclk*. Ako ne bi uzeli u obzir kašnjenje, iz slike bi se moglo pogrešno zaključiti da se bitovi postavljaju na rastući brid takta *sclk*. Najviši MSB (engl. *most significant bit*) bit postavlja se već na padajući brid signala *cs_s* i trebao bi trajati sve do prvog padajućeg brida *sclk* kad se postavlja novi bit, ali zbog već spomenutog kašnjenja traje čak do drugog rastućeg brida *sclk* signala.



Slika 4.4: Uspostavljanje SPI komunikacije prema ADP-a

4.1.3. Implementacija *transfer_ctrl* VHDL bloka

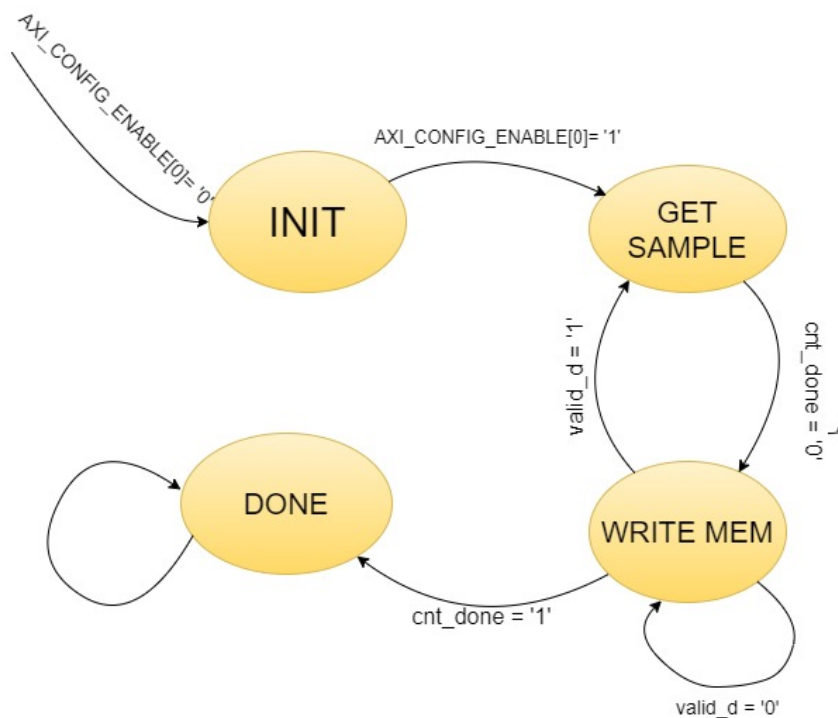
Nakon uhodavanja *master* sklopa koji komunicira s ADP-om i na svom izlazu daje ispravne 12-bitne vrijednosti prešlo se u realizaciju sklopa koji će spremati veću količinu ADC uzoraka u privremenu memoriju. Sklop treba pratiti stanje FPGA memorije u koju sprema ADC uzorke i ima zadatak signalizirati PS-u da je memorija puna. Tad procesorski sustav (PS) na ZedBoard-u kreće u fazu čitanja vrijednosti koje su zapisane u memoriju. Kao što je već prije istaknuto PL i PS dio komuniciraju preko standardnog AXI-Lite sučelja koje je potrebno nadograditi, odnosno dodati odgovarajuće registre i funkcije za obavljanje konkretnih zadataka. U ovom slučaju PL i PS dijele dva 32-bitna registra koji preko samojednog LSB bita ostvaruju potrebne signalizacije. Ovisno o registru *axi_config_enable* logički sklop *transfer_ctrl* započinje A/D pretvorbu, a preko registra *axi_status_done* logički sklop *transfer_ctrl* javlja PS-u da je memorija puna i da treba početi s čitanjem i daljnjom obradom. Dvopristupna memorija također se dijeli između PS i PL dijela budući da ju PL koristi za pisanje, a PS za čitanje. PL koristi fizičke adrese, a PS se koristi mapiranjem fizičkih adresa u virtualne adrese preko poziva funkcije *mmap*. U tablici 4.2 navedeni su imena registara i memorije te njihovi odmak od bazne adrese. Inače, dodjeljivanje fizičkih adresa AXI slave sklopova je vrlo jednostavno u alatu *Vivado Design Suite* i to će biti detaljno

objašnjeno prilikom povezivanja svih logičkih blokova.

Tablica 4.2: Nadogradnja AXI4-Lite sučelja

Naziv	Tip	Pomak od bazne adrese[dec]
<i>axi_mem</i>	Memorija	0
<i>axi_config_enable</i>	Konfiguracijski registar	16384
<i>axi_config_enable</i>	Statusni registar	16385

Upisivanje ADC uzoraka u dvoprístupnu memoriju *true_dpram_sclk* koja je preuzeta iz *Nove paradigme akvizicije podataka hibridnim sustavom*² realizirano je s automatom s konačnim brojem stanja (engl. *FSM, Final State Machine*). Sklop se nalazi u jednom od ova 4 stanja: INIT, GET_SAMPLE, WRITE_MEM, DONE. Na slici ispod prikazan je dijagram stanja i uvjeti prijelaza iz jednog stanja u drugo.



Slika 4.5: Uspostavljanje SPI komunikacije prema ADP-a

²A.Marinović, „Nove paradigme akvizicije podataka hibridnim sustavom“, Diplomski seminar, FER 2019

5. Zaključak

Zaključak.

Akvizijski sustav iznad Nyquistove granice

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.