



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA MIKRORAČUNARSKU
ELEKTRONIKU



Dizajn i verifikacija IP jezgra za skremblovanje audio signala
Diplomska teza (bečelor)

Kandidat

Kosta Ivančević, EE 8 / 2013

Mentor

Prof. dr Rastislav Struharik

Jul 2017



UNIVERZITET U NOVOM SADU • FAKULTET TEHNIČKIH NAUKA
21000 HOBI CAD, Trg Dositeja Obradovića 6

KLJUČNA DOKUMENTACIJSKA INFORMACIJA


Redni broj, RBR :		
Identifikacioni broj, IBR :		
Tip dokumentacije, TD :	Monografska publikacija	
Tip zapisa, TZ :	Tekstualni štampani primerak	
Vrsta rada, VR :	Diplomski rad	
Autor, AU :	Kosta Ivančević	
Mentor, MN :	Dr Struharik Rastislav, red. prof.	
Naslov rada, NR :	Dizajn i verifikacija IP jezgra za skremblovanje audio signala	
Jezik publikacije, JP :	Srpski	
Jezik izvoda, JL :	Srpski	
Zemlja publikovanja, ZP :	Srbija	
Uže geografsko područje, UGP :	AP Vojvodina	
Godina, GO :	2017.	
Izdavač, IZ :	Autorski reprint	
Mesto i adresa, MA :	Fakultet tehničkih nauka, 21000 Novi Sad, Trg Dositeja Obradovića 6	
Fizički opis rada, FO : (poglavlja/strana/citata/tabela/slika/grafika /priloga)	(7/47/0/1/48/0/0)	
Naučna oblast, NO :	Elektrotehnika	
Naučna disciplina, ND :	Embedded sistemi i algoritmi	
Predmetna odrednica/Ključne reči, PO :	Dizajn, verifikacija, skremblovanje	
UDK		
Čuva se, ČU :	Biblioteka Fakulteta tehničkih nauka, Trg Dositeja Obradovića 6	
Važna napomena, VN :		
Izvod, IZ :	U ovom radu su predstavljeni dizajn i verifikacija IP jezgra za obradu audio signala. Izuzev deljano opisanog dizajna i verifikacije, projektovan je audio sistem za fizičku demonstraciju rada IP jezgra.	
Datum prihvatanja teme, DP :		
Datum odbrane, DO :		
Članovi komisije, KO :		
Predsednik:	dr Vranjković Vuk	Potpis mentora
Član:	msc Rakanović Damjan	
Član, mentor:	dr Struharik Rastislav	



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	Monographic publication		
Type of record, TR :	Textual Printed Material		
Contents code, CC :	Bachelor Thesis		
Author, AU :	Kosta Ivančević		
Mentor, MN :	Ph.D. Struharik Rastislav, full professor		
Title, TI :	Design and Verification of IP Core for Audio Signal Scrambling.		
Language of text, LT :	Serbian		
Language of abstract, LA :	English		
Country of publication, CP :	Serbia		
Locality of publication, LP :	AP Vojvodina		
Publication year, PY :	2017.		
Publisher, PB :	Author's reprint		
Publication place, PP :	Faculty of technical sciences, 21000 Novi Sad, Trg Dositeja Obradovića 6		
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	(7/47/0/1/48/0/0)		
Scientific field, SF :	Electrical Engineering		
Scientific discipline, SD :	Embedded systems and algorithms		
Subject/Key words, S/KW :	Design , Verification, Scrambling		
UC			
Holding data, HD :	The library of Faculty of technical sciences, Trg Dositeja Obradovića 6, Novi Sad		
Note, N :			
Abstract, AB :	This Bachelor Thesis deals with the design and verification of IP core for audio signal scrambling. In addition, the audio system for practical demonstration is also designed.		
Accepted by the Scientific Board on, ASB :			
Defended on, DE :			
Defended Board, DB :	President:	Ph.D. Vranjković Vuk	Menthor's sign
	Member:	MSc. Rakanović Damjan	
	Member, Mentor:	Ph.D. Struharik Rastislav	

	UNIVERZITET U NOVOM SADU • FAKULTET TEHNIČKIH	Број:
	NAUKA	
	21000 NOVI SAD, Trg Dositeja Obradovića 6	Датум:
	ZADATAK ZA DIPLOMSKI RAD	

STUDIJSKI PROGRAM:	Energetika, elektronika i telekomunikacije
UKOVODILAC STUDIJSKOG PROGRAMA:	Dr Milan Sečujski

Student:	Kosta Ivančević	Broj indeksa:	EE 08/2013
Oblast:	Embedded sistemi i algoritmi		
Mentor:	Dr Struharik Rastislav, red. prof.		

NA OSNOVU PODNETE PRIJAVE, PRILOŽENE DOKUMENTACIJE I ODREDBI STATUTA FAKULTETA IZDAJE SE ZADATAK ZA DIPLOMSKI RAD, SA SLEDEĆIM ELEMENTIMA:

- problem – tema rada;
- način rešavanja problema i način praktične provere rezultata rada, ako je takva provera neophodna;
- literatura;

NASLOV DIPLOMSKOG RADA:

Dizajn i verifikacija IP jezgra za skremblovanje audio signala

TEKST ZADATKA:

U ovom radu potrebno je:

- Projektovati IP jezgro za skremblovanje audio signala
- Verifikovati funkcionalnost IP jezgra za skremblovanje audio signala
- Projektovati system za demonstraciju rada prethodno dizajniranog IP jezgra

Rukovodilac studijskog programa:	Mentor rada:

Primerak za: ☐ – Studenta; ☐ – Studentsku službu fakulteta

Sadržaj

Zahvalnica	5
Sadržaj diplomskog rada	6
Uvod	7
2.1 Enkripcija i skremblovanje signala	7
2.2 Problem	8
2.3 Predlog rešenja	9
2.4 Permutacije nizova	9
2.5 Furijeova transformacija signala	10
Dizajn IP jezgra za skremblovanje audio signala	15
3.1 Interfejs IP jezgra za skremblovanje audio signala	16
3.2 Režimi rada data scramblera	18
3.3 IP jezgro za implementaciju Furijeove transformacije	21
3.4 Data scrambler	24
3.5 Reset generator	25
3.6 Output FIFO bafer	25
3.7 Rezultati sinteze	26
Verifikacija IP jezgra za obradu audio signala	27
4.1 Stimulus	30
4.2 Checker	31
4.3 Scoreboard	32
4.4 "Tape-out" kriterijum	32
4.5 Rezultati verifikacije	33
4.6 Hardverasko testiranje dizajna	34
Projektovanje audio sistema i demonstracija rada IP jezgra za skremblovanje audio signala	35
5.1 Blok dizajn i opis rada audio sistema	35
5.2 Opis pojedinačnih komponenata audio sistema	37
5.3 Razvoj softvera, grafičkog interfejsa i korišćenje audio sistema	44
Zaključak	46
Literatura	46

Zahvalnica

Ovom prilikom, autor želi da izrazi zahvalnost svom mentoru dr Struharik Rastislavu na lepoj saradnji tokom izrade ovog diplomskog rada. Tok samog projekta je sproveden kroz niz konsultacija sa mentorom koje su imale pozitivan uticaj na krajnji rezultat i u velikoj meri doprinele kvalitetu ovog diplomskog rada. Takođe, autor se zahvaljuje na sugestiji ovako izazovne teme diplomskog rada. Izuzev doprinosa na konkretnom radu, mentor je, kroz niz predmeta koje predaje na smeru Mikroračunarska elektronika, uspeo da prenese adekvatna znanja i, što je još bitnije, prenese osnove inženjerske intuicije koja je dobra polazna osnova za dalji inženjerski napredak autora.

Takođe, autor se zahvaljuje prof. dr Vranjković Vuku na pomoći i korisnim savetima i usmeravanjima koje je pružio tokom izrade ovog rada, doprinoseći njegovoj praktičnoj primeni i implementaciji u realnim sistemima.

Na kraju, autor se zahvaljuje nastavnom osoblju Fakulteta tehničkih nauka, koji predaju na usmerenju Mikroračunarska elektronika u okviru smera Energetika, elektronika i telekomunikacije, i koji su imali pozitivan uticaj na akademski razvoj autora.

Sadržaj diplomskog rada

U okviru ovog diplomskog rada biće izložen proces implementacije IP jezgra za skremblovanje audio signala. Na ideju o realizovanoj temi, autor je došao kroz niz konsultacija sa mentorom. Ovakav izbor teme omogućava da autor stekne teorijska i praktična znanja u vezi sa obradom audio signala, kao i da demonstrira stečena znanja iz oblasti digitalnog dizajna, verifikacije, dizajna na sistemskom nivou, kao i razvoja softvera za dati embedded sistem. Pozitivni aspekt izbora teme je činjenica da je autor kroz izradu ovog diplomskog rada obuhvatio znanja iz više različitih predmeta koji se izučavaju na usmerenju „Embedded sistemi i algoritmi“ u okviru smera „Mikroračunarska elektronika“. Enkapsulacija stečenih znanja tokom studija u jedan projekat je jedan od osnovnih razloga za izbor teme.

Obzirom da je izrada ovog diplomskog rada proces koji obuhvata više različitih etapa, u daljem tekstu će biti izložena tematika svake od pojedinačnih etapa, kao i njihove međusobne veze.

U drugom poglavlju je data definicija problema, predlog mogućeg rešenja, kao i kratka matematička osnova na koju se oslanja predloženo rešenje.

U trećem poglavlju je izložen proces dizajna IP jezgra za skremblovanje audio signala korišćenjem HDL jezika¹. U okviru ove glave je dat strukturni opis IP jezgra pri čemu je detaljno opisana svaka od strukturnih komponenata, kao i njihovo međusobno povezivanje. Takođe, izneta je i procena hardverskih resursa neophodnih za implementaciju samog jezgra, kao i opis komunikacionih interfejsa za povezivanje IP jezgra u njegovo sistemsko okruženje.

Četvrto poglavlje je posvećeno procesu funkcionalne verifikacije samog jezgra. Uzimajući u obzir da se primena datog IP jezgra svodi na obradu signala sa izuzetno redukovanom upravljačkom logikom, prilikom verifikacionog procesa nije korišćena neka od standardnih verifikacionih metodologija npr. UVM². U ovom odeljku, osim samog procesa verifikacije, izneto je detaljno obrazloženje izabrane verifikacione metodologije.

Peta glava je posvećena dizajnu sistema za demonstraciju rada IP jezgra na razvojnoj ploči Zedboard, kompanije Xilinx. Detaljno su opisane sve komponente ovog sistema i dat kratak opis međusobnih veza i zavisnosti. U ovoj temi je dat kratak osvrt na razvijeni softver i korisnički interfejs.

Šesta glava je posvećena sumiranju implementacije rešenja zadatog problema. Takođe, dat je osvrt na otvorene probleme i moguća poboljšanja samog dizajna, što otvara mogućnost daljeg rada na ovoj temi u budućnosti.

¹ HDL – hardware design language je jezik koji se koristi za opis ponašanja i strukture digitalnih elektronskih kola.

² UVM – Universal Verification Methodology je jedan od trenutnih industrijskih trendova u domenu funkcionalne verifikacije.

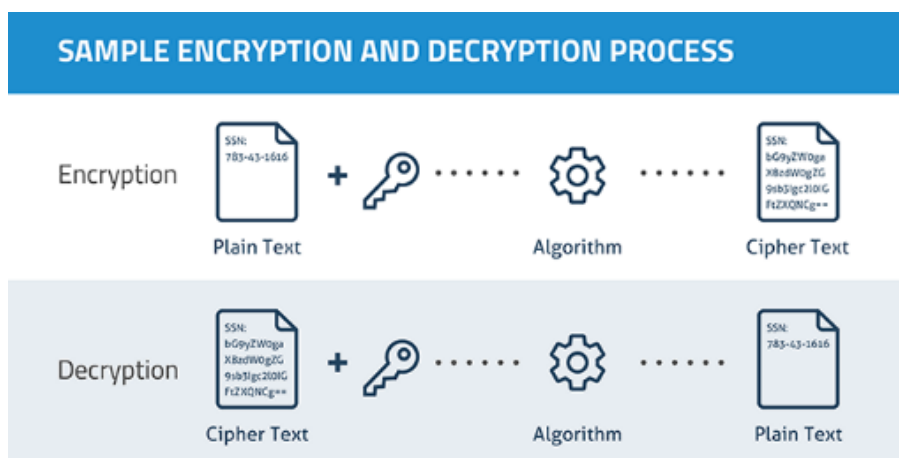
Uvod

U ovom poglavlju biće izložena problematika koja se obrađuje, predlog mogućeg rešenja, kao i matematička osnova na koju se čitalac kasnije može referencirati.

2.1 Enkripcija i skremblovanje signala

Enkripcija je kriptografski proces kojim se sadržaj poruke menja tako, da samo "privilegovani" korisnici imaju pristup informacijama. Privilegovanost se, u konkretnom slučaju, odnosi na posedovanje ključa za enkripciju ili poznavanje algoritma po kojem je ona rađena. Privilegovani korisnici su svi korisnici kojima je poruka namenjena. Ovo znači da je poruka dostupna svima, ali samo korisnici kojima je ona namenjena imaju mogućnost da je tumače i ekstrahuju korisne informacije. Enkripcija se po pravilu radi na predajnoj strani. Na prijemnoj strani se vrši dekripcija, odnosno inverzan postupak dobijanja korisnih informacija iz enkriptovanog sadržaja. Smisao enkripcije je zaštita podataka tokom prenosa.

Nemoguće je obezbediti sistem u kojem će poruku primiti samo korisnik kome je ona namenjena, pa se polazi od činjenice da poruku neće nabaviti samo korisnik kome je namenjena, već i niz drugih korisnika. Zbog toga se sadržaj poruke menja enkripcijom, tako da ova poruka suštinski ima informacioni sadržaj samo korisnicima kojima je namenjena. Ovim se osigurava pouzdan prenos i zaštićen sadržaj poruka.

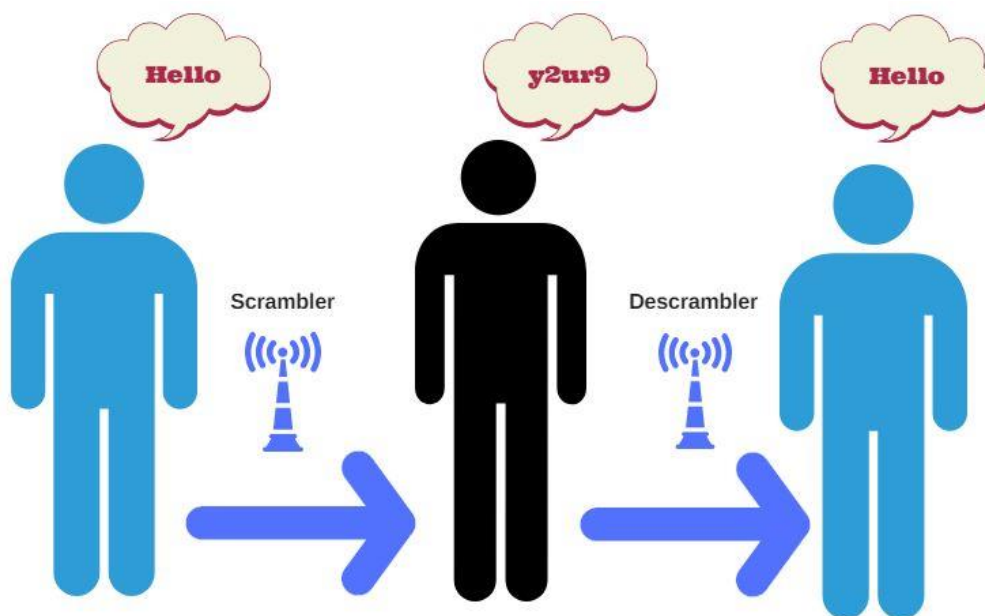


Slika 2.1.1 Idejna blok šema enkripcije

Skremblovanje se u velikoj meri poklapa sa enkripcijom. Suštinski imaju isti cilj: "da onemoguće interpretaciju poruka koje su presretene". Dok se enkripcija najčešće vezuje za digitalni domen signala, skremblovanje se u najvećem broju slučajeva odnosi na analogni domen. U osnovi, to je proces dodavanja ili modifikacije komponenata signala tako da je teško utvrditi koji deo signala je koristan i nosi informacije. U današnje vreme, kao posledica velike rasprostranjenosti digitalne elektronike koja u sve većem broju slučajeva zamenjuje odgovarajuća analogna kola, dosta je teško uočiti jasnu razliku između skremblovanja i enkripcije. Imajući u vidu da je tema ovog diplomskog rada obrada audio signal, koristiće se termin skremblovanje. Bez gubitka preciznosti i opštosti, mogao se koristiti i termin enkripcija.

2.2 Problem

Problematika ovog rada se svodi na implementaciju IP jezgra za skremblovanje audio signala. IP jezgro je predviđeno da bude posrednik između predajnika i prijemnika, kao što je prikazano na 2.2.1.



Slika 2.2.1. Blok šema Sistema koju čine privilegovani članovi : prijemnik i predajnik i tzv "third party"

Ovo podrazumeva da IP jezgro na predajnoj strani radi adekvatnu manipulaciju audio signala, pri čemu je nemoguće interpretirati audio signal na izlazu skremlera. Na prijemnoj strani se vrši inverzna operacija, tj vrši se deskremblovanje audio signala, tako da je audio signal na izlazu prijemnika identičan audio signalu na ulazu predajnika. To znači da prijemna i predajna strana imaju uvid u koristan signal, dok signal između prijemnika i predajnika nema izražene harmonijske komponente iz kojih se može izvući korisna informacija o sadržaju početnog audio signala.

Signal na ulazu u jezgro se dovodi kao periodična povorka odbiraka na frekvenciji određenoj Nikvistovim kriterijumom³. Frekvencija rada IP jezgra je nezavisna od učestanosti odbiraka i skremler treba da bude imun na zastoje u prijemu paketa, kao i na fluktuacije u brzini pristizanja odbiraka sve dok je učestanost pristizanja odbiraka manja od frekvencije rada samog IP jezgra.

Signal na izlazu jezgra je periodična povorka odbiraka skremblovanog signala. Harmonijske komponente izlaznog signala treba da se razlikuju u što većoj meri u odnosu na harmonijske komponente ulaznog signala, kako bi proces prepoznavanja korisnih informacija u okviru deskremblovanog bio u što većoj meri otežan.

³ Nikvistov kriterijum definiše vezu između frekvencije odabiranja i frekvencije najvišeg harmonika u signalu. Za vernu reprezentaciju komponenata, neophodno je da frekvencija odabiranja bude najmanje duplo veća u odnosu na najviši harmonik. Za čujni opseg, najviša komponenta ima učestanost oko 20 KHz, pa je učestanost odabiranja 48 KHz što je industrijski standard.

2.3 Predlog rešenja

Predloženo rešenje za implementaciju IP jezgra za skremblovanje audio signala, koje ispunjava zahteve definisane u poglavlju 2.2 je implementirano digitalnim sistemom koji vrši permutaciju audio odbiraka. Ovakav tip audio obrade se svodi na promenu redosleda odbiraka ulaznog signala na osnovu adekvatnog algoritma. Kako bi se poboljšao efekat skremblovanja, predloženo rešenje vrši gorepomenute permutacije u frekvencijskom domenu signala. Pre nego što se permutacije izvrše, neophodno je preći iz vremenskog u kompleksni domen. Permutacije, tj. sam proces skremblovanja se vrši nad odbircima signala nastalog nakon DFT⁴. Signal se zatim, procesom inverzne DFT signala tj. permutovanog niza vraća u vremenski domen. Ovim je proces skremblovanja završen.

Na prijemnoj strani se vrši inverzni postupak, tj. signal se procesom DFT prebacuje u kompleksni domen, gde se ponovo vrši adekvatna permutacija odbiraka. Time se dobija početni niz kompleksnih odbiraka, pre permutacija na predajnoj strani. Inverznom DFT se signal vraća u početni oblik, čime je process deskremblovanja uspešno okončan. Greška koja se ovim sistemom unosi je posledica rada sa brojevima konačne preciznosti. Od sistema se očekuje da unesena greška ne bude registrovana od strane korisnika kako se kvalitet signala ne bi degradirao.

$$a(t) \rightarrow A(n) \rightarrow A'(n) \rightarrow a'(t) \dots \rightarrow \dots a'(t) \rightarrow A'(n) \rightarrow A(n) \rightarrow a(t)$$

$$a(t) \rightarrow A(n) - DFT$$

$$A(n) \rightarrow a(t) - Inverzna DFT$$

$$A(n) \rightarrow A'(n) - Permutacija elemenata$$

2.4 Permutacije nizova

Prilikom obrade audio signala, vrši se diskretizacija po vremenu i amplitudi, nakon čega se dobija digitalni signal. Suštinski, to je sekvenca brojeva koja se predstavlja u obliku brojevni nizova. U ovom odeljku će biti opisane osnovne zakonitosti vezane za permutacije nizova koje predstavljaju dobru osnovu za razumevanje procesa koji se odvijaju u IP jezgru za skremblovanje audio signala. Pretpostavimo da imamo skup od N elemenata $A = \{a_0, a_1, a_2, a_3, a_4, \dots, a_{N-1}\}$. Permutacija elemenata skupa je svaka funkcija koja preslikava skup A u skup A, pri čemu se početni i krajnji skup razlikuju u rasporedu elemenata. Permutacija se formalno definiše kao:

$$\sigma : A \rightarrow A$$

Bitno je zapaziti da je σ bijektivna funkcija. Ovo podrazumeva da je σ :

- Sirjekcija – različiti elementi domena se preslikavaju u različite elemente kodomena
- Injekcija – čitav kodomen funkcije je pokriven funkcijom

Ilustrativni primer permutacije niza od osam elemenata:

$$\sigma = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\ a_4 & a_8 & a_6 & a_3 & a_2 & a_1 & a_5 & a_7 \end{pmatrix}$$

Bitno je napomenuti da ista permutacija, primenjena prvo na niz, a zatim na njegov permutovani ekvivalent rezultuje početnim nizom.

⁴ DFT – Diskretna Furijeova transformacija

Formalno:

$$\sigma(A) = A'; \sigma(A') = A$$

Ova osobina permutacija je osnova za razumevanje rada IP jezgra za skremblovanje audio signala.

Više o ovome, čitalac može da pronađe u [2].

2.5 Furijeova transformacija signala

Pre samog ulaženja u tematiku Furijeove transformacije, definišimo signal kao funkciju jedne ili više nezavisnih promenljivih koje preslikavaju elemente vremenske ose u elemente nekog drugog skupa, koji definiše prirodu signala.⁵ Na osnovu prethodnog, realni signal se matematički definiše kao:

$$x : \mathbb{R} \rightarrow \mathbb{R}.$$

Ukoliko između bilo koja dva elementa domena postoji najmanje jedan element, tada se kaže da je signal kontinualan po vremenu. U suprotnom je diskretan. Analogni zaključak se izvodi za kodomen, pa se tada kaže da je signal diskretan odnosno kontinualan po amplitudi. Generalno, ovaj zaključak važi za bilo koju dimenziju signala u koliko je u pitanju višedimenzionalni signal. Na osnovu prethodnog, signali mogu biti:

- Kontinualni po vremenu i amplitudi
- Diskretni po vremenu i amplitudi
- Diskretni po vremenu, kontinualni po amplitudi
- Kontinualni po vremenu, diskretni po amplitudi

Neke signale je pogodnije posmatrati u vremenskom, a neke u frekvencijskom domenu, i Z transformacija omogućava prelazak iz vremenskog u kompleksni domen i obrnuto. Prema granicama odgovarajućih suma, razlikujemo bilateralnu i unilateralnu Z transformaciju.

$$X_b = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$
$$X_u = \sum_{n=0}^{\infty} x(n)z^{-n}$$

Ove dve sume su jednake u slučaju da je signal jednak nuli za sve vrednosti levo od nekog inicijalnog trenutka. Ova osobina realnih signala se naziva kauzalnost. Takođe, jasno je da je signal jednoznačno definisan svojom oblašću konvergencije⁶ i svojom Z transformacijom. Furijeova transformacija diskretnih signala je evaluacija Z transformacije na jediničnom krugu.

$$z^{-n} \rightarrow e^{-j\omega k}$$
$$X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x(k)e^{-j\omega k}$$

⁵ Ako je kodomen signala skup kompleksnih brojeva, kažemo da je u pitanju kompleksni signal. Analogno, kodomen realnih signala je skup realnih brojeva.

⁶ Oblast konvergencije (ROC – Region of convergence) je skup svih vrednosti z za koje red konvergira

Za razliku od Furijeove transformacije kontinualnog signala, Furijeova transformacija diskretnog signala je periodična funkcija.

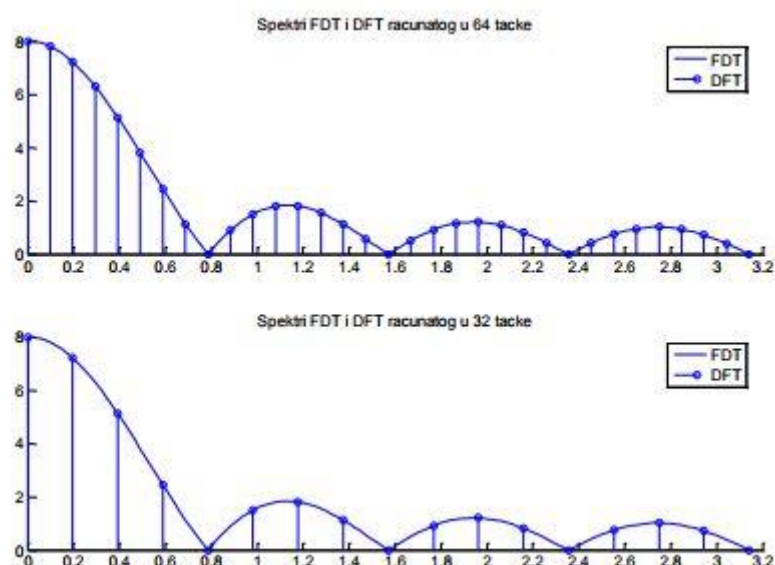
Osnovni nedostatak Furijeove transformacije diskretnih signala je korišćenje beskonačnih suma prilikom njene evaluacije. Kako bi se izbegla ova neželjena osobina, koristi se diskretna Furijeova transformacija, definisana kao:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn} \quad k = 0, 1, 2, \dots, N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi}{N}kn} \quad n = 0, 1, 2, \dots, N-1$$

Spektralni signal koji se dobija kao rezultat Diskretne furijeove transformacije je diskretan što olakšava njegovu obradu digitalnim elektronskim kolima.

Parametar N označava broj tačaka u kojima se računa DFT. Direktno je povezan sa pojmom frekvencijska rezolucija, koji označava priraštaj sukcesivnih elemenata u spektralnom signalu. Rast parametra N je direktno proporcionalan vernosti reprezentacije spektralnog signala. Takođe, može se zapaziti veza između DFT i FDT. Naime, diskretna Furijeova transformacija uzima vrednosti Furijeove transformacije diskretnog signala u tačno definisanim tačkama, čiji je razmak uslovljen frekvencijskom rezolucijom, kao što je prikazano na slici 2.5.1.



Slika 2.5.1. Spektri DFT i FDT

Imajući u vidu računsku kompleksnost evaluacije raznih Furijeovih transformacija, jasno je da je jednostavnije, a samim tim i efikasnije implementirati DFT u odnosu na FDT. Međutim, redukcija računске kompleksnosti se ne završava na DFT. Brza furijeova transformacija - FFT⁷ predstavlja efikasnije algoritme za evaluaciju DFT, uzimajući u obzir broj elementarnih operacija.

⁷ FFT – Fast Fourier Transfer – brza Furijeova transformacija

Usvajimo da je $W_N^{kn} = e^{-j\frac{2\pi}{N}kn}$ i nazovimo ga rotacioni faktor. Tada, DFT postaje :

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad k = 0, 1, 2, \dots, N-1$$

Efikasnost FFT algoritama se krije u interesantnim osobinama rotacionog faktora, a to su kompleksno konjugovana simetričnost i periodičnost, definisanih u nastavku.

$$W_N^{k(N-n)} = W_N^{-kn} = (W_N^{kn})^*$$

$$W_N^{kn} = W_N^{k(N+n)} = W_N^{(k+N)n}$$

Obe ove osobine su direktna posledica simetrije i periodičnosti trigonometrijskih sinus i kosinus funkcija, na koje se dekomponuje eksponencijalna funkcija.

$$e^{-j\frac{2\pi}{N}kn} = \cos j\frac{2\pi}{N}kn - j \sin j\frac{2\pi}{N}kn$$

Usvajimo da je polazni niz sekvenca odbiraka dužine stepena dva, što nije suviše limitirajući faktor ako znamo da se signal može dopuniti nulama do adekvatne dužine (tzv. padding). Tada se DFT može dekomponovati na sumu parnih i neparnih odbiraka.

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{2kn} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{(2n+1)k} = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{2kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{2kn}$$

Za parne vrednosti N važi :

$$W_N^{2kn} = e^{-j\frac{2\pi n}{N} \cdot 2k} = e^{-j\frac{2\pi n}{\frac{N}{2}}k} = W_{\frac{N}{2}}^{kn}$$

Prethodni izraz tada postaje :

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_{\frac{N}{2}}^{kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_{\frac{N}{2}}^{kn}$$

Iz prethodnog sledi da se DFT sekvence dužine N može izračunati računanjem DFT parnih i neparnih odbiraka, uz množenje sa rotacionim faktorom. Matematički :

$$X(k) = X_{1p}(k) + W_N^k * X_{1n}(k) \quad k = 0, 1, 2, 3, \dots, \frac{N}{2} - 1$$

X_{1p} i X_{1n} su DFT parnih, odnosno neparnih podsekvenci početnog niza $x(n)$. Za vrednosti k u opsegu $(\frac{N}{2}, N-1)$ dobijamo :

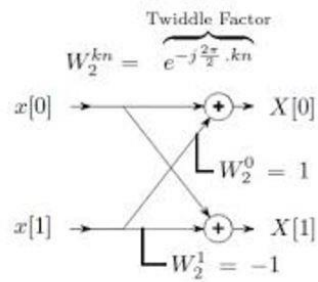
$$X\left(k + \frac{N}{2}\right) = X_{1p}(k) - W_N^{k+\frac{N}{2}} * X_{1p}(k) \quad k = 0, 1, 2, 3, \dots, \frac{N}{2} - 1$$

Na osnovu osobina rotacionog faktora važi sledeće :

$$W_N^{k+\frac{N}{2}} = e^{-j\frac{2\pi}{N}(k+\frac{N}{2})} = e^{-j\frac{2\pi}{N}k} * e^{-j\frac{2\pi}{N}\frac{N}{2}} = -1 * e^{-j\frac{2\pi}{N}k} = -W_N^k$$

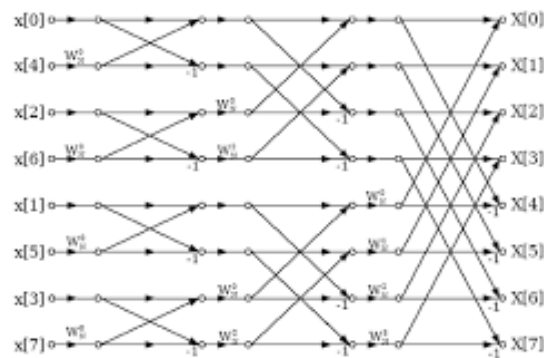
Na osnovu prethodnog zaključujemo da izrazi za računanje DFT vrednosti čiji su indeksi k i $k + \frac{N}{2}$ razlikuju samo u znaku drugog sabirka.

Ovaj bitan zaključak dovodi do suštinskog povećanja efikasnosti koji se ogleda u smanjenju broja operacija. Operacija računanja DFT vrednosti $X(k)$ i $X(k+\frac{N}{2})$ na osnovu DFT vrednosti $X_{1p}(k)$ i $X_{1n}(k)$ naziva se leptir (butterfly) operacija i najčešće se grafički prikazuje kao na slici 2.5.2.



Slika 2.5.2. Radix – butterfly

Leptir operacija sastoji se iz jednog kompleksnog množenja, sa rotacionim faktorom, dva kompleksna sabiranja, odnosno četiri realna množenja i šest realnih sabiranja. Na slici 2.5.3 je prikazano računanje FFT u 8 tačaka, korišćenjem radix-2 dekompozicije.

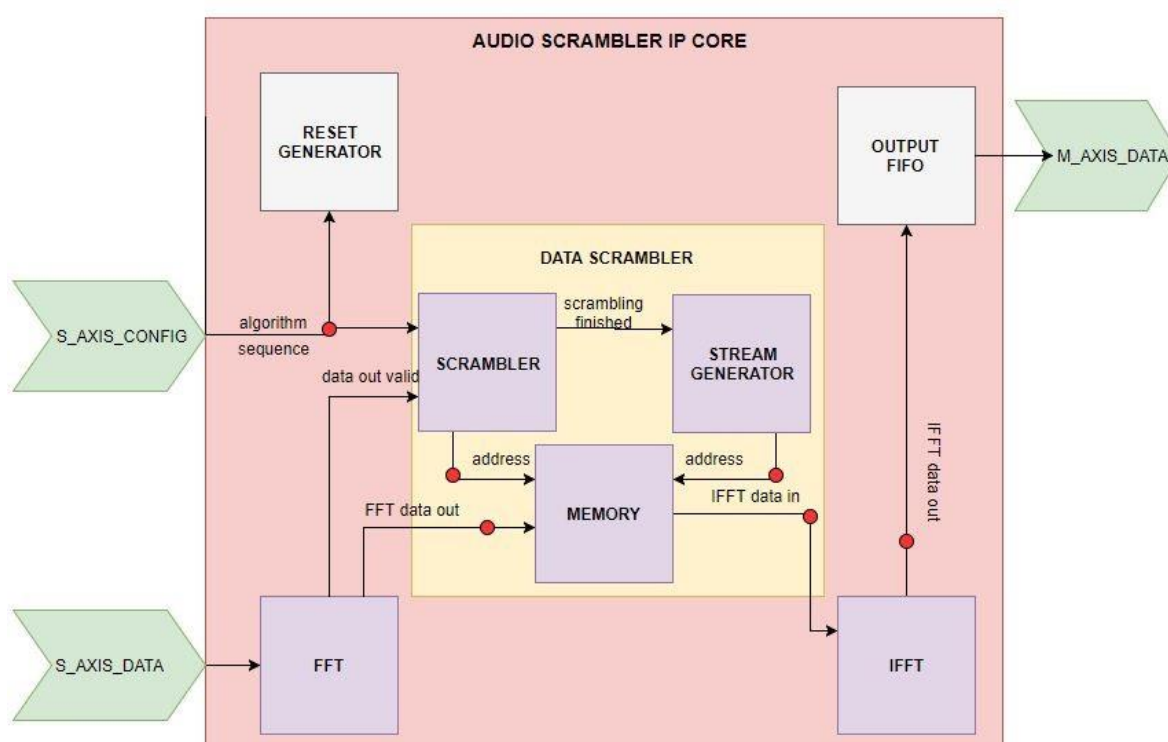


Slika 2.5.3 Radix – butterfly

Više detalja u okviru ove tematike čitalac može da pronaći u [1].

Dizajn IP jezgra za skremblovanje audio signala

U ovom poglavlju biće detaljno izložen proces dizajna IP jezra koje obavlja funkciju skremblovanja u frekvencijskom domenu. Prilikom dizajna, korišćena je RTL metodologija i sistematičan pristup projektovanju, kako bi se postigao što efikasniji dizajn, koji se u što većoj meri poklapa sa očekivanim hardverom nakon sinteze. Detalji o RTL metodologiji se mogu naći u [3]. Takođe, biće detaljno opisani pojedinačni funkcionalni blokovi od kojih se sastoji dato IP jezgro, kao i interfejs za povezivanje sa ostatkom sistema, neophodan za obezbeđivanje koncizne i sistematične integracije IP jezgra u sistemsko okruženje i neophodne modularnosti, koja je neizostavni zahtev modernih digitalnih sistema.



Slika 3.1. Blok šema IP jezgra za obradu audio signala

Sa blok šeme, izložene na slici 3.1. se mogu uočiti sledeći funkcionalni blokovi:

- IP jezgro za izvršavanje direktne Furijeove transformacije
- Skrembler
- Memorija
- Stream generator
- IP jezgro za izvršavanje inverzne Furijeove transformacije
- Reset Generator
- Output FIFO bafer

S_AXIS_CONFIG interfejs služi za prosleđivanje konfiguracije IP jezgra. Interfejs obuhvata kontrolne signale i 16-bitni port za prosleđivanje konfiguracije IP skremlera (slike 3.1.2 i 3.1.3). s_axis_config_tdata je 16-bitni port koji definiše konfiguraciju IP jezgra za skremblovanje audio signala. Niži bajt se koristi za prosleđivanje pomoćnog niza za skremblovanje (scrambling sequence,

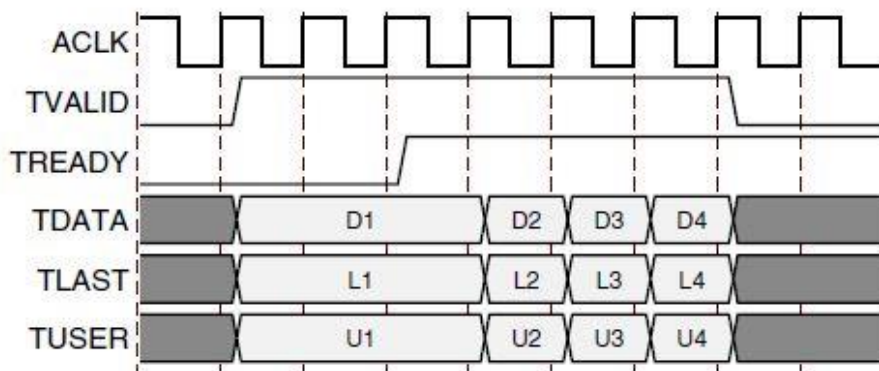
scrambling key). Viši bajt se koristi za prosleđivanje algoritma po kojem se vrši skremblovanje. Više o algoritmima i režimima rada se može naći u poglavlju 3.2. Imajući u vidu AXI protokol, detaljnije opisan u narednom poglavlju, svaki put kada se desi promena algoritma ili pomoćnog niza, vrši se reset čitavog jezgra, za šta je zadužen reset generator. Osim u reset generator, konfiguracioni port se vodi u scrambler, koji na osnovu prosleđenih informacija vrši adekvatne permutacije.

S_AXIS_DATA interfejs je interfejs za prijem audio signala. 16-bitni audio signal se vodi na ulaz FFT IP jezgra, koje je zaduženo za prelazak iz vremenskog u kompleksni domen. Nakon što se procesiranje podataka izvrši i nakon što su podaci na m_axis_data_tdata porta validni, u skladu sa AXI protokolom, scrambler počinje permutaciju memorijskih adresa. Signal m_axis_data_tvalid je ujedno enable signal scramblera. Kada je podatak na izlazu FFT IP jezgra validan, scrambler, na osnovu algoritma i pomoćnog niza generiše na svom izlazu memorijsku adresu. Opisan postupak podrazumeva jedinstvenu dodelu memorijskih lokacija svakom podatku na izlazu FFT IP jezgra. Kada je obuhvaćena celina od 256 permutovanih elemenata, scrambler signalizira da išitavanje elemenata može da počne. Ovu funkciju obavlja stream generator. Stream generator je zadužen za generisanje memorijskih adresa u prirodnom poretku. Imajući ovo u vidu, memorija, konfigurisana kao četiri celine od 256 64-bitnih lokacija, se koristi za privremeno smeštanje permutovanih elemenata pre nego što se kompletira celina od 256 elemenata, nakon čega se u naredni opseg memorije smeštaju novi skremblovani podaci, a iz prethodno kompletirane celine iščitavaju i dovode na ulaz IFFT IP jezgra, koje je zaduženo za inverznu furijeovu transformaciju. Izlazni podaci se privremeno smeštaju u FIFO bafer pre nego što se proslede izvan sistema, preko M_AXIS interfejsa.

3.1 Interfejs IP jezgra za skremblovanje audio signala

IP jezgro za je projektovano tako da spoljašnji interfejsi budu AXI kompatibilni, što omogućava lakšu integraciju u sistemsko okruženje koje je definisano standardnim industrijskim protokolima, kakav je AXI. Ovo jezgro je namenjeno za korišćenje u aplikacijama za obradu zvuka. Kako je kontinualni tok podataka neophodan za ovakav tip aplikacija, tendencija prilikom dizajna samog jezgra je bila lak i brz protok podataka kroz jezgro, kao i jednostavno povezivanje sa drugim komponentama putem AXI protokola. Standardizacijom ovog tipa se povećava modularnost digitalnog dizajna i otvara mogućnost za njegovo ponovno korišćenje u nekim drugim aplikacijama („design reuse“).

Protok ulaznih i izlaznih podataka je jednostavna realizacija „Handshake“ protokola, koji je prikazan na slici 3.1.1. Zastavica TVALID, koja se podiže na predajnoj strani, je indikator da se na magistrali nalaze validni podaci i da je predajnik spreman za transfer. TREADY zastavica se podiže na prijemnoj strani i indikator je spremnosti prijemnika da prima podatke na TDATA magistrali. Detekcija stanja „1-1“ predstavlja početak transfera.

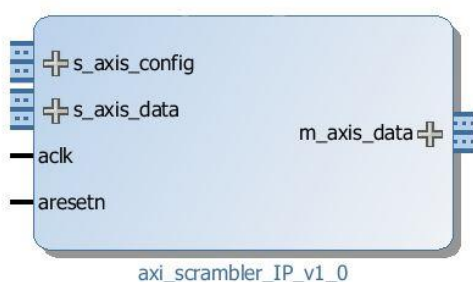


Slika 3.1.1 "Handshake" transfer podataka

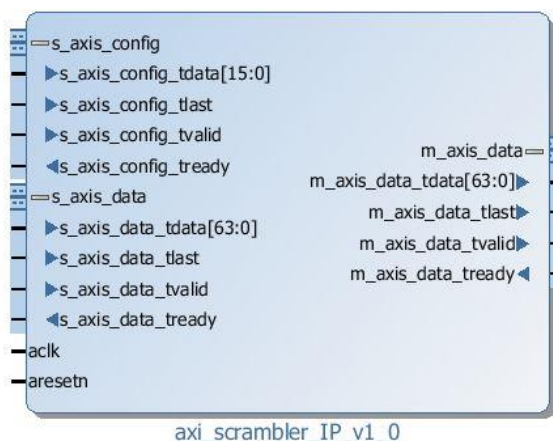
Razne varijacije AXI protokola četvrte generacije (AXI-Lite, AXI-Full, AXI-Stream) se međusobno razlikuju po broju kontrolnih signala i indikatora unutrašnjih stanja mastera i slejva. Složenost transfera podataka diktira složenost (broj portova i kontrolnih signala) i prirodu komunikacionog protokola. Na osnovu inženjerske intuicije, lako je zaključiti da je dobra praksa koristiti što jednostavniji protokol koji omogućava unapred specificirani tok podataka. Primena IP jezgra za skremblovanje audio signala u oblastima u kojima se očekuje kontinualni tok podataka sugerise korišćenje AXI-Stream protokola na svakom od sledećih interfejsa :

- s_axis_config – služi za konfigurisanje jezgra od strane korisnika
- s_axis_data – služi za prijem podataka (slave interface)
- m_axis_data – služi za slanje podataka (master interface)

Krajnji izgled IP jezgra i njegovih interfejsa je prikazana na slikama 3.1.2 i 3.1.3., dok se detaljnije informacije u vezi AXI interfejsa mogu naći u [6].



Slika 3.1.2. Izgled IP jezgra na nivou interfejsa



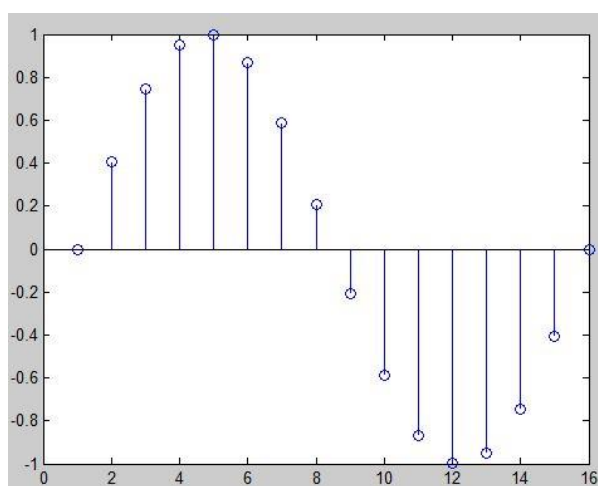
Slika 3.1.3. Detaljan prikaz portova IP jezgra

3.2 Režimi rada data scramblera

IP jezgro za skremblovanje audio signala je predviđeno da radi u jednom od četiri različita režima. Ovi režimi se odnose na algoritam po kojem se vrši permutacija signala. Pod permutacijama signala se podrazumeva promena raspodela odbiraka signala. Permutacije se vrše na uzorku od 256 odbiraka raspoređenih u 16 frejmova po 16 elemenata. U daljem tekstu je prikazana ilustracija rada sa permutacijama 8 elemenata. To je urađeno bez gubitka opštosti, kako bi čitalac lakše i preglednije razumeo ideju skremblovanja. Dva najniža bita u okviru gornjeg bajta s_axis_config porta (9 - 8) služe za definisanje algoritma skremblovanja i na osnovu stanja ovog porta se definišu režimi rada IP jezgra. Režim rada je određen skremblerom, koji kao osnovu ima brojač. Stanje brojača predstavlja primarnu adresu, koja se kasnije adekvatnim bitskim manipulacijama, u zavisnosti od algoritma, menja kako bi se dobile permutacije.

Stanju "00" odgovara režim rada u kojem je isključeno skremblovanje. Elementi se prosleđuju kroz sistem u istom redosledu u kojem su i primljeni, tj adrese generisane brojačem u skrembleru se kao takve prosleđuju na adresni ulaz memorije. Drugim režimima, memorija se adresira u prirodnom poretku.

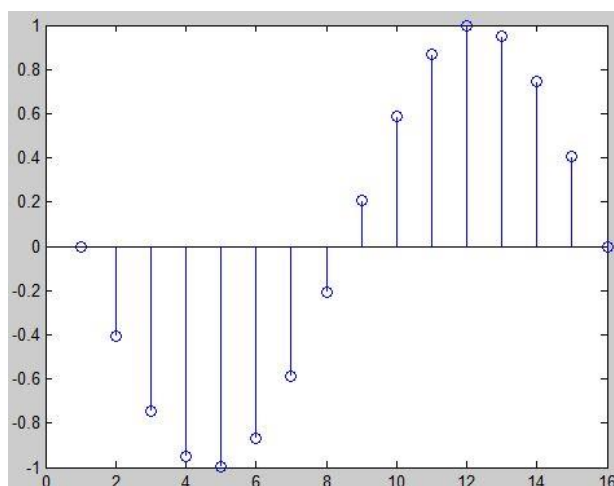
$$\sigma = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \end{pmatrix}$$



Slika 3.2.1. Sinusoida nakon "00" algoritma

Stanju "01" odgovara "frame inversion" algoritam gde se redosled elemenata invertuje i prosleđuju se od poslednjeg ka prvom. Gledajući ovo sa aspekta adrese generisane u okviru skremblera, dolazi do inverzije bita (0101 → 1010) i adresiranja memorijskih lokacija od najviše (255) do najniže (0).

$$\sigma = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{pmatrix}$$

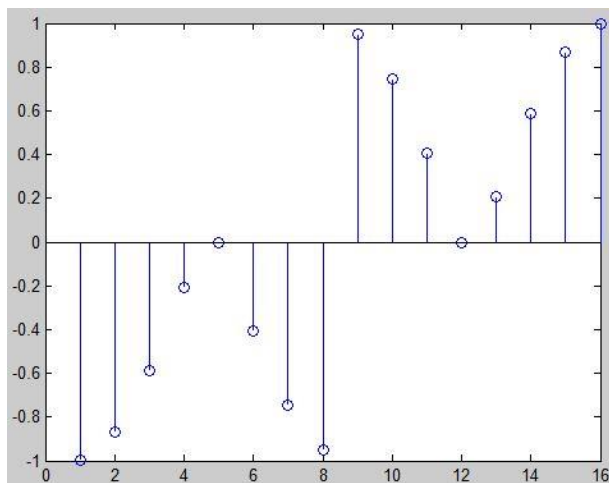


Slika 3.2.2. Sinusoida nakon "01" algoritma

Stanju "10" odgovara algoritam uslovljen pomoćnim nizom (scrambling key), koji prosleđuje korisnik. Imajući u vidu da je memorija sa slike 3.1, čiji je rad (u sprezi sa skremblerom i stream generatorom) detaljno opisana u odeljku 3.4., predstavljena kao kolekcija blokova od 256 elemenata, jasno je da za jedinstveno adresiranje elementa unutar bloka neophodna osmobitna adresa. Generisanje adrese iz prethodno pomenute primarne adrese, generisane u brojaču skremblera, svakog elementa se radi uz pomoć skremblerskog ključa i trenutnog stanja internog brojača u skrembleru.

$$\text{"00100110" xor "00101111" = "00001001"}$$

$$\sigma = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_0 \oplus x_0 & a_1 \oplus x_1 & a_2 \oplus x_2 & a_3 \oplus x_3 & a_4 \oplus x_4 & a_5 \oplus x_5 & a_6 \oplus x_6 & a_7 \oplus x_7 \end{pmatrix}$$



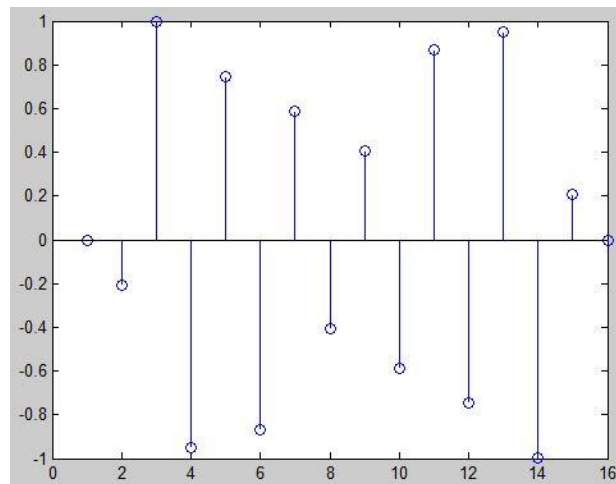
Slika 3.2.3. Sinusoida nakon "10" algoritma i sekvence "1011"

Ovo omogućava da prijemnik i predajnik poseduju osmobitnu sekvencu (scrambling key) koji služi za skremblovanje, odnosno deskremblovanje signala.

Stanju “11” odgovara “bit inversion” algoritam za permutacije. Ovaj algoritam nalaže da se osmobična adresa podatka dobija zamenom težinskih pozicija bita koji je sačinjavaju. Suštinski, ovo znači da će se MSB⁸ menjati u svakom taktu, MSB - 1 u svakom drugom, MSB - 2 u svakom četvrtom, itd. Takođe, ovo omogućava da ni jedna dva podatka neskremblovanog frejma budu u istom skremblovanom frejmu. Ovo otežava tumačenje neskremblovanog signala.

$\text{pos}(0) \Leftrightarrow \text{pos}(7)$, $\text{pos}(1) \Leftrightarrow \text{pos}(6)$, $\text{pos}(2) \Leftrightarrow \text{pos}(5)$, $\text{pos}(3) \Leftrightarrow \text{pos}(4)$

$$\sigma = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_0 & a_4 & a_2 & a_6 & a_1 & a_5 & a_3 & a_7 \end{pmatrix}$$



Slika 3.2.4. Sinusoida nakon “11” algoritma

⁸ MSB (Most significant Bit) bit najveće težine u višebitnoj reči

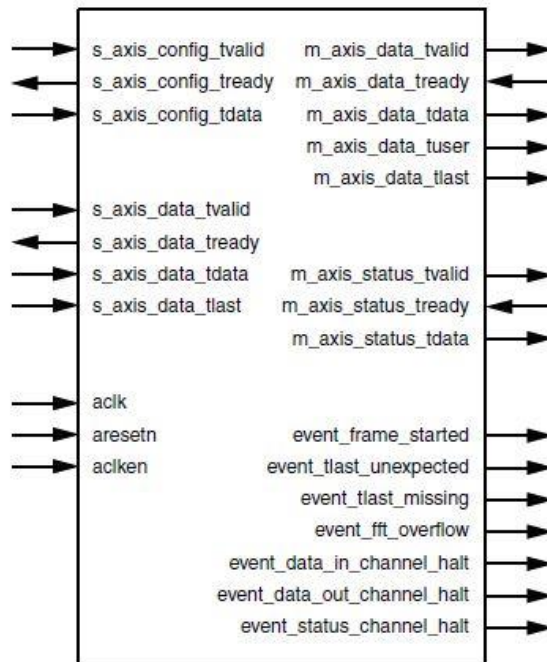
3.3 IP jezgro za implementaciju Furijeove transformacije

Ip jezgro za implementaciju Furijeove transformacije, koje distribuira kompanija Xilinx, je blok koji omogućuje transformaciju iz vremenskog u frekvencijski domen signala i obrnuto. U ovom procesu se koriste efikasni algoritmi za računanje brze Furijeove transformacije. Podložan je rekonfiguraciji, kako prilikom dizajna, tako i prilikom samog rada sistema (run time konfigurabile) , što ga čini izuzetno modularnim i omogućava laku i brzu integraciju u dosta različitih aplikacija. U konkretnom slučaju, konfigurisano je da, u „Realtime“ modu evaluiira FFT i IFFT nad blokovima od 16 odbiraka. Tokom rada se zaokruživanjem održava konstantna veličina signala. Kako ne bi došlo do zastoja u obradi podataka, tj zagušenja protoka, koristi se arhitekturna implementacija sa protočnom obradom. Ovo unosi dodatne hardverske resurse, ali ima pozitivan uticaj na performanse. Posedovanje AXI-Stream interfejsa je dodatna standardizovanost koja pospešuje laku i brzu integraciju u različite sisteme, što povećava produktivnost inženjera koji rade dizajn na sistemskom nivou. Blok šema FFT jezgra je prilazana na slici 3.1.1, dok se lista portova i njihovih objašnjenja se nalazi u tabeli 3.3.1. Dato jezgro izvršava operaciju FFT i IFFT⁹ na osnovu sledeće jednačine:

$$\text{FFT :} \quad X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \quad k = 0, 1, 2, 3 \dots N - 1$$

$$\text{IFFT:} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi kn}{N}} \quad k = 0, 1, 2, 3 \dots N - 1$$

Detalji vezani za računanje Furijeove transformacije su dati u 2.4. , dok se detaljan opis jezgra nalazi u [5].



Slika 3.3.1. Xilinx FFT IP jezgro

⁹ FFT brza Furijeova transformacija , IFFT inverzna brza Furijeova transformacija

Naziv	Orijentacija	Opis
aclk	Input	Takt signal sa aktivnom rastućom ivicom
aclken	Input	Signal dozvole takt signala
aresetn	Input	Sinhroni reset aktivanna niskom logičkom nivou. Ima prednost u odnosu na aclken
s_axis_config_tvalid	Input	TVALID za konfiguracioni kanal. Spoljašnji master podiže signal na visok logički nivo ako su ulazni podaci validni
s_axis_config_tready	Output	TREADY za konfiguracioni kanal. Jezgro podiže ovaj signal na visok logički nivo u koliko je spremno da prihvati podatke.
s_axis_config_tdata	Input	TDATA za konfiguracioni kanal. Sadrži informacije kao što su: CP_LEN, FWD/INV, NFFT and SCALE_SCH.
s_axis_data_tvalid	Input	TVALID za kanal namenjen prijemu podataka. Spoljašnji master podiže signal na visok logički nivo ako su ulazni podaci validni.
s_axis_data_tready	Output	TREADY za kanal namenjen prijemu podataka. Jezgro podiže ovaj signal na visok logički nivo u koliko je spremno da prihvati podatke.
s_axis_data_tdata	Input	TDATA za kanal namenjen prijemu podataka. Prihvata odbirak koji ima realnu i imaginarnu komponentu: XN_RE and XN_IM.
s_axis_data_tlast	Input	TLAST za kanal namenjen prijemu podataka. Spoljašnji master podiže signal na logičku jedinicu kada prosleđuje poslednju podatak. Ovaj signal se ne koristi izuzev za generisanje statusnih signal : event_tlast_unexpected and event_tlast_missing Events
m_axis_data_tvalid	Output	TVALID za kanal namenjen predaji podataka. Jezgro podiže ovaj signal kada su podaci na m_axis_data_tdata validni
m_axis_data_tready	Input	TREADY za kanal namenjen predaji podataka. Spojašnji slejv podiže ovaj signal na visok nivo kada je spreman da prima podatke. Prisutan samo u "Non-realtime" konfiguraciji jezgra.
m_axis_data_tdata	Output	TDATA za kanal namenjen predaji podataka. Vrš predaju realne i imaginarne komponente odbirka: XK_RE and XK_IM.
m_axis_data_tuser	Output	TUSER za kanal namenjen predaji podataka. Nosi informacije, vezane za pojedinačne odbirke, kao što su : XK_INDEX, OVFO and BLK_EXP.
m_axis_data_tlast	Output	TLAST za kanal namenjen predaji podataka. Jezgro podiže ovaj signal na visok nivo u trenutku slanja poslednjeg odbirka u frejmu.
m_axis_status_tvalid	Output	TVALID za statusni kanal. Jezgro podiže ovaj signal na visok nivo kada je spremno da prosledi statusne podatke
m_axis_status_tready	Input	TREADY za statusni kanal. Spoljanji slejv podiže signal na visok nivo kada je spreman da prima podatke. Ovaj signal je prisutan samo u "Non-Realtime" režiu.
m_axis_status_tdata	Output	TDATA za statusni kanal. Nosi informacije kao što su: BLK_EXP or OVFO.
event_frame_started	Output	Visok nivo je indikacija da jezgro počinje obradu novog frejma.
event_tlast_unexpected	Output	Na visokom nivou kad jezgro registruje visok nivo na axis_data_tlastn na odbirku koji nije poslednji.
event_tlast_missing	Output	Na visokom nivou kada je poslednji odbirak stigao praćen axis_data_tlast na niskom nivou.
event_fft_overflow	Output	Indikacija prekoračenja dinamičkog opsega na izlaznom

		kanalu, u koliko se ne koristi opcija skaliranja.
event_data_in_channel_halt	Output	Na visokom nivou kada podaci na ulaznom kanalu nisu dostupni u trenutku kada je jezgro spremno za prihvatanje podataka.
event_data_out_channel_halt	Output	Na visokom nivou kada podaci na izlaznom kanalu nisu prihvaćeni od strane spoljašnjeg slejva. Prisutan je samo u "Non-realtime" aplikacijama
event_status_channel_halt	Output	Na visokom nivou kada podaci na statusnom kanalu nisu prihvaćeni od strane spoljašnjeg slejva. Prisutan je samo u "Non-realtime" aplikacijama

Tabela 3.3.1. Tabela portova i njihovih opisa

Sa slike 3.3.1 i iz tabele 3.3.1 može se primetiti da su interfejsi kompatibilni sa AXI standardom, što povećava dijapazon primene, laku integraciju i povezivanje sa velikim brojem komponenti koji takođe imaju AXI interfejse. Sa slike se mogu uočiti sledeći interfejsi:

AXI slave data interfejs – služi za prijem podataka,

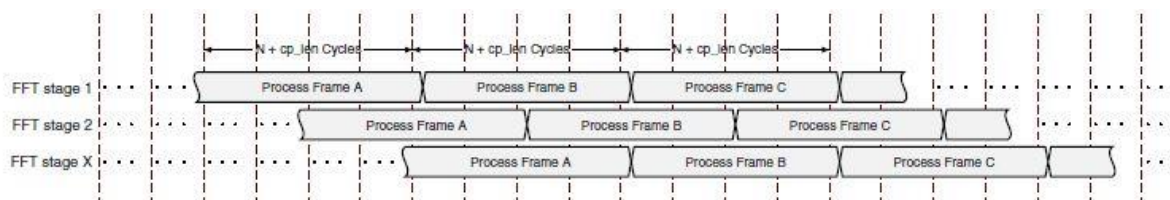
AXI master data interfejs – služi za prosleđivanje podataka,

AXI slave config interfejs – služi za konfiguraciju jezgra tokom samog rada,

kao i kontrolni signali za indicaciju trenutnog stanja u kojem se jezgro nalazi.

U osnovi arhitekture Xilinx FFT IP jezgra je radix-2 dekompozicija. Male promene interne strukture samog jezgra predstavljaju kompenzaciju između performansi i hardverske složenosti. U konkretnoj aplikaciji se koristi arhitektura sa protočnom obradom kako bi se postogao kontinualni tok podataka, neophodan za tzv. streaming aplikacije.

Protočna obrada inherentno podrazumeva podelu toka podataka na faze, pri čemu nakon određenog broja ciklusa, neophodnog da se pipeline napuni, svaka faza vrši obradu podataka, čime se postiže ubrzanje obrade podataka, onoliko puta, koliko ima različitih faza. Time se povećava paralelizam i iskorišćenost hardverskih resursa na raspolaganju. Ovo podrazumeva da se paket podataka (frame) za računanje FFT preuzima pre nego što je prethodni paket podataka obrađen. Ova korisna osobina pospešuje jednostavnost implementacije protokola i brzinu rada na račun dodatnih hardverskih resursa koji se koriste.



Slika 3.3.2. Model protočne obrade

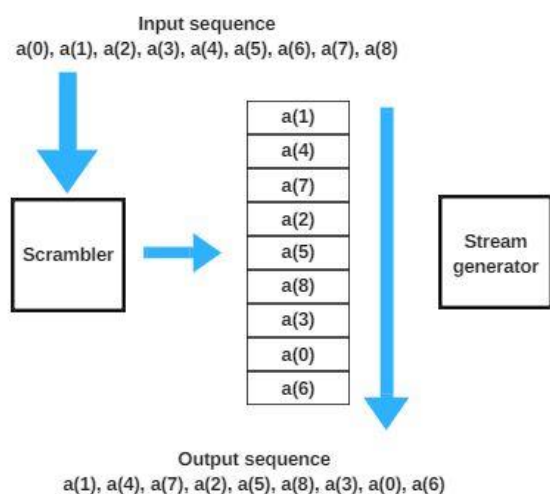
3.4 Data scrambler

Data scrambler je podsistem u okviru IP jezgra za skremblovanje signala u frekvencijskom domenu, koji služi za implementaciju samog procesa skremblovanja. Na svom ulazu je povezan na izlaz FFT IP jezgra. Podaci se primaju u kontinualnom nizu i ova komponenta vrši permutacije elemenata ovog niza. Na svom izlazu je povezan sa IFFT IP jezgrom kojem šalje skremblovaee podatke. Sastoji se iz tri funkcionalna bloka :

- Scrambler
- Memorija
- Stream generator

Scrambler je funkcionalni blok koji je zadužen za permutacije elemenata. Ulazni interfejs obuhvata dvobitni port za specifikaciju skrembling algoritma i osmobitni port za prosleđivanje pomoćnog niza koji služi za skremblovanje (scrambling key). Kako su FFT jezgra konfigurisana da računaju Furijeovu transformaciju u 16 tačaka, a dužina permutovanog niza je 256 elemenata, jasno je da permutacija nije limitirana na jedan blok od 16 tačaka, već na $16^2 = 256$ tačaka. To podrazumeva da se sistem može konfigurisati tako da se elementi blokova (frejmova) permutuju tako, da prilikom inverzne FFT u okviru ulaznog bloka od 16 elemenata ne bude više od jednog elementa iz istog bloka pre skremblovanja. Ova karakteristika je izuzetno bitna jer se dobija mnogo kvalitetnija permutacija elemenata gde se na prijemnoj strani, u odsustvu adekvatnog deskremblerskog algoritma ne može rekonstruisati polazni signal.

Na svom izlazu, skrembler generiše adrese koje se koriste za adresiranje memorijskih lokacija u koje se smeštaju elementi iz FFT jezgra (slika 3.1). Permutacija elemenata se ogleda u stanju memorije prilikom kontinualnog iščitavanja memorijskih lokacija. Imajući u vidu da se memorija sastoji od četiri bloka od 256 lokacija, gde se skremblovanje vrši u okviru jednog bloka, jasno je da je osnova skremblera 10-bitni brojač koji se inkrementira pri svakom prosleđivanju podatka od strane FFT jezgra, tj pri svakom smeštanju podatka u memoriju. Polazna adresa je 10-bitna, od čega se na osnovu algoritama vrši permutacija donjih osam bita, a dva bita najveće težine se ne menjaju. Iz ovoga se može zaključiti da se permutacije vrše na blokovima od 256 elemenata, dok se blokovi popunjavaju jedan za drugim nezavisno od algoritma i načina skremblovanja, što je i zadatak ovog modula.



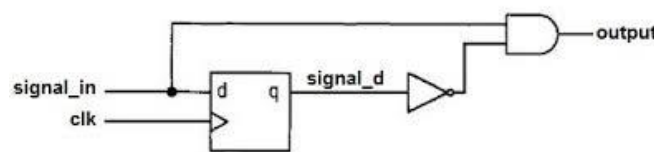
Slika 3.4.1. Idejna blok šema Data scramblera

Memorija na slici 3.4.1 služi za privremeno skladištenje permutovanih elemenata, sve dok se ne kompletira celina od 256 elemenata. Implementirana je kao dvoprístupna memorija sa signalima dozvole upisa i čitanja. Veličina memorije je 1024 64-bitne lokacije, grupisane u 4 bloka po 256 lokacija.

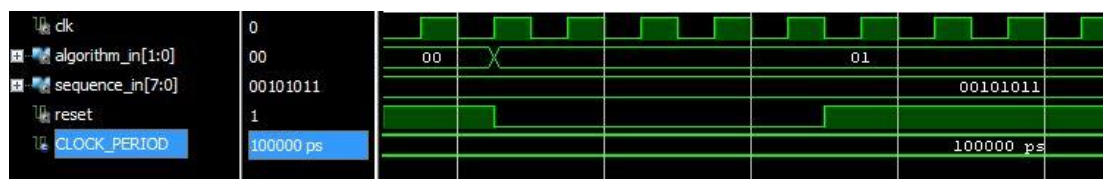
Stream generator predstavlja blok koji se koristi za kontinualno iščitavanje memorijskih lokacija. Nakon što je scrambler permutovao 256 elemenata u memoriji, stream generator se koristi da redom generiše memorijske adrese, što omogućava kontinualan tok podataka, koji se prosleđuju IFFT IP jezgri, zaduženom za inverznu Furijeovu transformaciju.

3.5 Reset generator

Reset je izuzetno bitan deo svakog sistema. U konkretnom slučaju, reset podrazumeva inicijalizaciju skremlera, stream generatora, memorije, FFT, IFFT IP jezgra i izlaznog FIFO bafera. Sa spoljašnjih interfejsa se može primetiti da IP jezgro ima aresetn pin kojim se, zajedno sa ostatkom programabilne logike, ili nekog njenog segmenta, uvodi u reset stanje. Bitno je zapaziti da je reset jezgra neophodan samo u situacijama kada dolazi do promene algoritma za skremljivanje ili skremljivanje sekvence, tj. pomoćnog niza. Zbog toga se spoljašnji aresetn pin ne koristi nego se u okviru ove komponente generiše lokalni reset signal koji resetuje ostatak jezgra kada je to potrebno. Ova komponenta se svodi na detector ivice, čiji je primer prikazan na slici 3.5.1. Detekcija ivice na signalu, koji odgovara algoritmu ili sekvenci generiše reset u trajanju od pet taktova. Situacija detekcije promene algoritma je prikazana na slici 3.5.2



Slika 3.5.1. Detektor opadajuće ivice



Slika 3.5.2. Signali reset generatora

3.6 Output FIFO bafer

U realnim sistemima gde se koriste komunikacioni protokoli nalik na AXI, bitno je voditi računa da master i slejv ne moraju, u opštem slučaju, raditi u sinhronizmu. Situacija da slejv nije spreman da primi podatak u trenutku kada je master signalizirao da je spreman za transfer je vrlo česta. Kako ne bi došlo do zagušenja sistema usled neusklađenosti mastera i slejva ili gubljenja odbiraka, na izlazu jezgra se nalazi FIFO¹⁰ bafer. On omogućava mesto za privremeno skladištenje podataka pre nego što se ti podaci prihvate na slejv strani.

¹⁰ FIFO – First In First Out – Podatak koji prvi uđe u bafer biva prvi poslat iz bafera

3.7 Rezultati sinteze

Tendencija razvoja modernih digitalnih sistema je da se dizajn implementira na što manjoj površini čipa. Prilikom ASIC¹¹ dizajna, površina čipa je direktno proporcionalna ceni, dok kod FPGA dizajna, fizički manji dizajn se može implementirati manjem i jeftinijem čipu. Iako nije u pitanju komercijalni proizvod koji će se distribuirati, dizajn je razvijan sa ciljem da se što bolje mapira u hardver (vodeći se metodologijama u [3]), tj da zauzme što više predefinisanih resursa. Parametri u dizajnu su takođe birani kako bi se jezgro implementiralo sa što manjim hardverskim zauzećem. U tabeli ispod su dati rezultati sinteze.

Report Check Netlist:

	Item	Errors	Warnings	Status	Description
1	Multi driven nets	0	0	Passed	Multi driven nets

Detailed RTL Component Info :

Component	Width	Number
Adder 2 input	24	2
Adder 2 input	11	4
Adder 2 input	10	2
Adder 2 input	1	8
XOR 2 input	64	3
Register	11	2
Register	10	2
Register	8	3
Register	2	3
Register	1	11
MUX 2 input	24	4
MUX 2 input	11	24
MUX 2 input	1	1
RAM	128K	2
RAM	64K	1

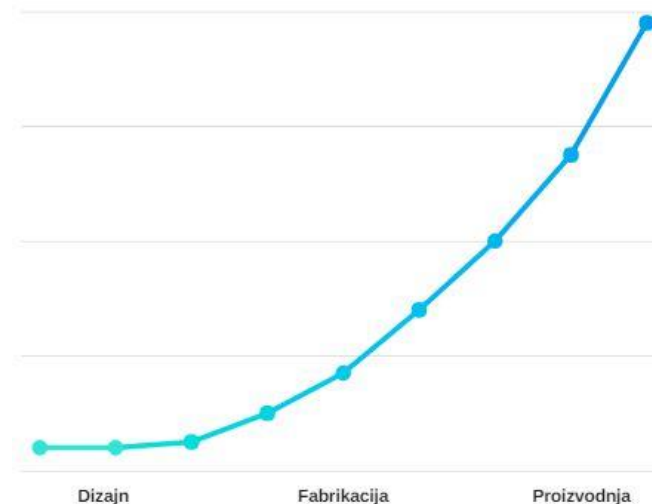
	Module	Cell
1	TOP	762
2	FIFO	110
3	Reset generator	27
4	Scrambler	467
5	FIFO	107
6	Memory	2
7	Scrambler	45
8	Stream generator	21

¹¹ ASIC – Application Specific Integrated Circuit – integrisano kolo dizajnirano i fizički napravljeno za prethodno određenu namenu

Verifikacija IP jezgra za obradu audio signala

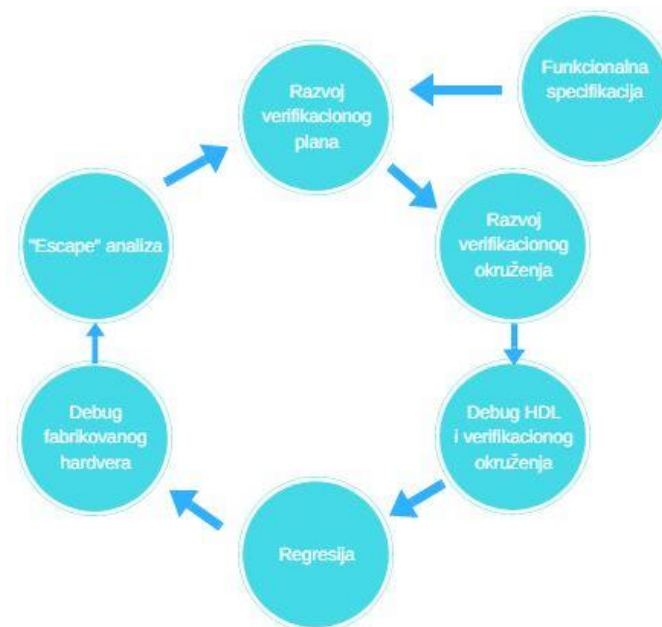
U ovom poglavlju, biće detaljno izložen proces verifikacije IP jezgra za skremblovanje audio signala. Kratak teorijski opis određenih etapa, kao i predstavljanje rezultata relevantnih faza verifikacije je takođe obuhvaćen.

Verifikacija predstavlja neizostavni deo razvoja modernih digitalnih sistema. Svrha verifikacije je detektovanje nepravilnosti u radu dizajna, u odnosu na očekivano ponašanje, koje je specificirano funkcionalnom specifikacijom. U slučaju dizajna ASIC-a, verifikacija je ključni korak, zato što blagovremena detekcija bagova omogućava njihovo pravovremeno otklanjanje, dok detekcija bagova nakon fabrikacije hardvera povećava kašnjenje proizvoda na tržište i zahteva refabrikaciju hardvera, što je dodatno finansijsko opterećenje na proizvođače digitalnih integrisanih kola. Na slici 4.1 je prikazana ilustrativna skica „cene bagova“. Detekcija bagova u procesu verifikacije iziskuje troškove inženjerskog vremena, neophodnog za otklanjanje bagova. Detekcija bagova nakon fabrikacije zahteva ponovnu fabrikaciju, dok otkrivanje bagova od strane korisnika povlači kršenje garancije proizvođača i finansijski ostavlja najnegativniji trag na kompaniju.



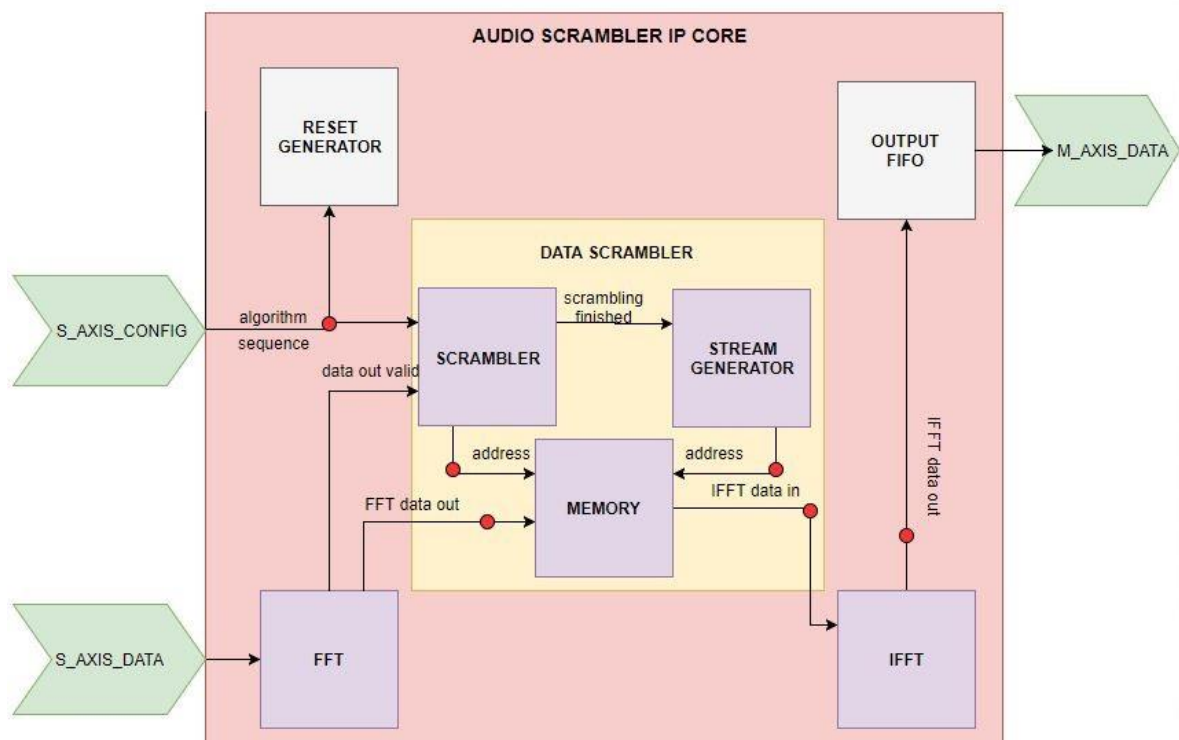
Slika 4.1. "Bug cost"

Kompleksnost modernih digitalnih dizajnova je izuzetno velika, što nameće potrebu za sistematičnim i formalno uređenim verifikacionim ciklusom koji je prikazan na slici 4.2. Metodološki pristup i praćenje verifikacionih faza redukuje broj bagova koji se detektuju nakon fabrikacije. To inherentno smanjuje troškove proizvoda i osigurava njegovu pouzdanost, što je i cilj same verifikacije.

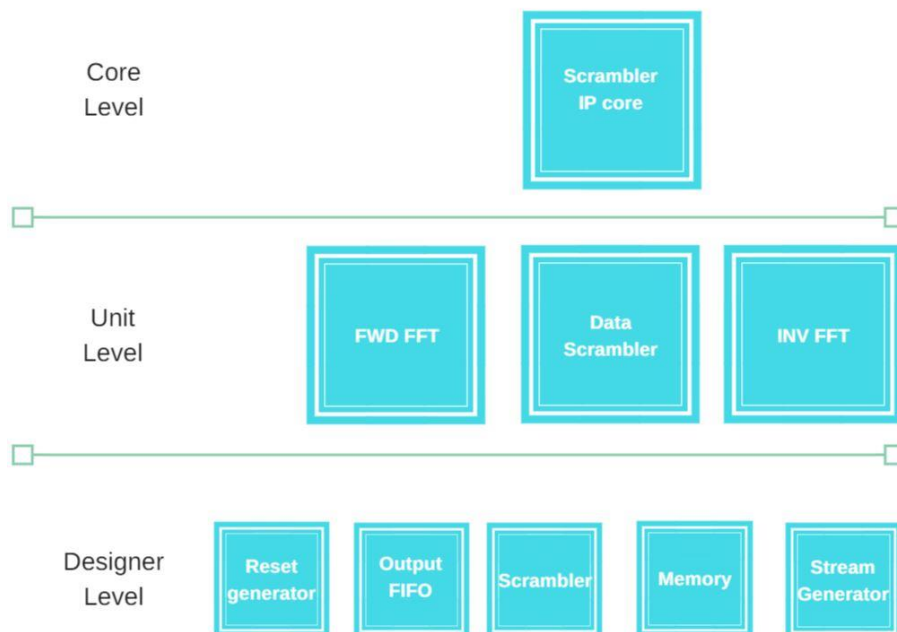


Slika 4.2. Verifikacioni ciklus

Prilikom dizajna samog IP jezgra, korišćen je hijerarhijski pristup, gde se dizajn rekurzivno deli na jednostavnije blokove, koji enkapsuliraju određenu funkcionalnost. Ovaj pristup se koristi i prilikom verifikacije, kada se prvo verifikuju jedinice na nižim nivoima (designer level i unit level), nakon čega se prelazi na verifikaciju na nivou samog jezgra. Ovaj pristup (bottom up approach) doprinosi sistematičnosti otklanjanja bagova. Prvo se verifikuje funkcionalnost gradivnih jedinica, i tek nakon što su one verifikovane, prelazi se na viši hijerarhijski nivo. Na nivou jezgra se verifikuje interakcija gradivnih jedinica, polazeći od toga da je funkcionalnost svake od njih verifikovana. Sa slika 4.3 i 4. 4 se jasno vidi kako verifikacija prati strukturu dizajna.



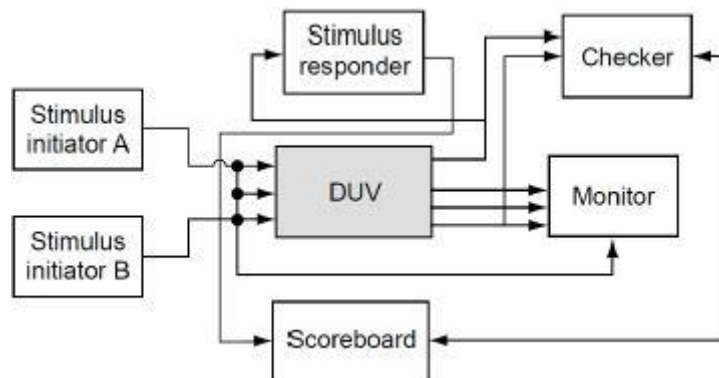
Slika 4.3. Hijerarhijska struktura dizajna



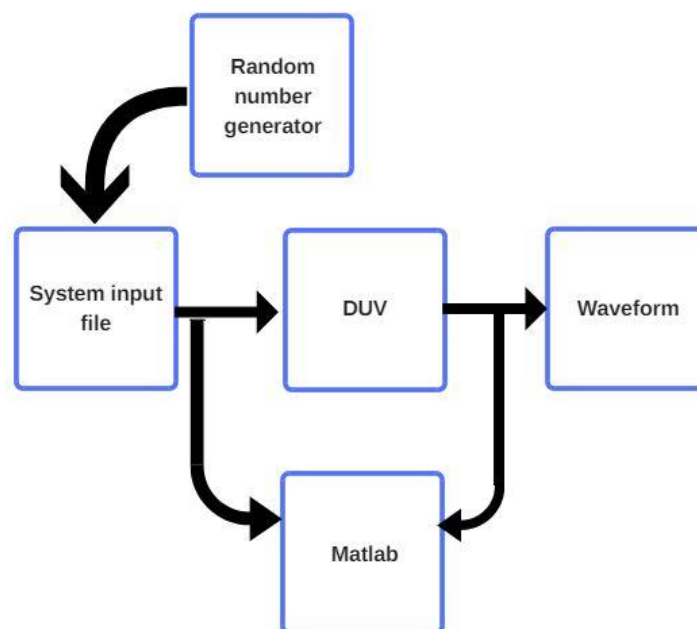
Slika 4.4. Verifikaciona struktura

Na slici 4.3 su mogu uočiti tačke opservacije koje su postavljene unutar dizajna. Prilikom verifikacije na unit i core hijerarhijskim nivoima se koristi „grey box“ pristup. To podrazumeva smeštanje opservacionih tačaka na spoljne interfejse, kao i posmatranje unutrašnjih signala. Ovaj princip doprinosi bržem lokalizovanju i otklanjanju bagova, i predstavlja kompromis između jednostavnosti „black box“ pristupa, gde se posmatraju samo spljašnji interfejsi i preciznosti „white box“ pristupa, gde se posmatraju svi signali.

Tipično verifikaciono okruženje se sastoji od osnovnih verifikacionih komponentata, čija međusobna interakcija je prikazana na slici 4.5. Imajući u vidu relativno mali broj stanja u kojima može da se nađe dizajn, razvijeno je specifično verifikaciono okruženje, prikazano na slici 4.6.



Slika 4.5. Komponente tipičnog verifikacionog okruženja

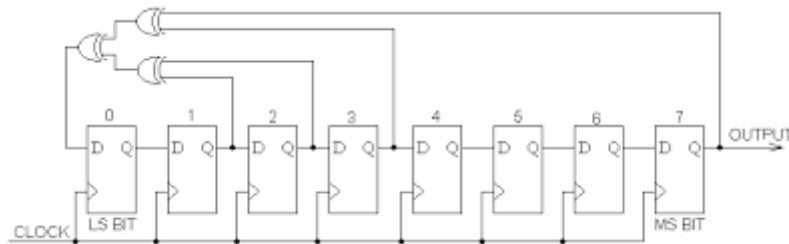


Slika 4.6. Verifikaciono okruženje IP jezgra za skremblovanje audio signala

4.1 Stimulus

Stimulus predstavlja komponentu u verifikacionom okruženju koja je zadužena za adekvatno oponašanje okruženja u kojem će se dizajn nalaziti. Jedna od osnovnih karakteristika verifikacije je kontrolabilnost, i ona je tesno povezana sa stimulus komponentom. Kontrolabilnost podrazumeva mogućnost generisanja adekvatnih sekvenci na ulazima dizajna, koje će omogućiti prolazak kroz stanja u kojima se nalaze potencijalni bagovi (tzv. corner case). Upešan prolazak kroz kolekciju scenarija i detekcija odstupanja od očekivanog ponašanja je suština verifikacije. Stimulus komponenta se može podeliti na stimulus inicijator i stimulus reponder. Stimulus inicijator generiše ulazne sekvence nezavisno u DUV-a dok stimulus responder predstavlja povratnu spregu, koja na osnovu sekvenci na interfejsima DUV-a generiše povratne sekvence signala. Prilikom konkretne implementacije stimulus komponente u okviru verifikacionog okruženja, korišćen je stimulus

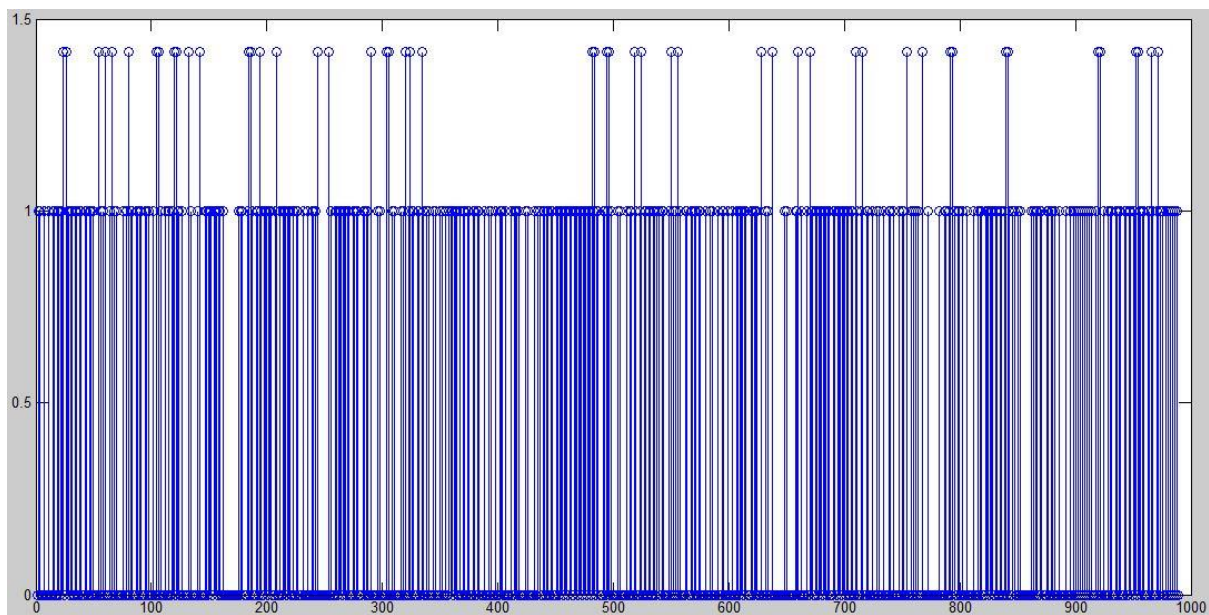
inicijator. Implementiran je kao generator slučajnih brojeva, kako bi se testirao odziv sistema na kvazi-nedeterministički signal, kakav očekujemo prilikom skremblovanja govornog signala čoveka. Implementacija generatora slučajnih brojeva je moguća na mnogo načina. LFSR (linear feedback shift register) je jedan primer kvazi-nedeterminističkog generatora i se zasniva na povratnoj sprezi bita na različitim težinskim pozicijama u okviru broja. Inicijalne vrednosti (eng. seed) predstavljaju početnu vrednost pseudoslučajnog niza. Prilikom konkretne implementacije, stimulus komponenta je zadužena za generisanje uzalnih signala, kao i za njihovo smeštanje u spoljašnju datoteku. Ovo omogućava da komponente implementirane u Matlabu imaju pristup sekvencama na ulazima sistema. Ovo je izuzetno bitno prilikom korišćenja referentnih modela, detaljno opisanih u nastavku. Više informacija vezanih za LFSR, čitalac može naći u [4].



Slika 4.1.1. 8 - bit Linear Feedback Shift Register

4.2 Checker

Checker predstavlja komponentu u verifikacionom okruženju koja je zadužena za prikupljanje podataka sa izlaza i za proveru validnosti rezultata. Služi da detektuje neispravno ponašanje sistema u odnosu na set očekivanih vrednosti, koje se tretiraju kao validne. U konkretnom slučaju, set očekivanih vrednosti se dobija korišćenjem referentnih modela, tj modela razvijanih u višim programkim jezicima, pri čemu se algoritamska implementacija funkcionalnosti modela razlikuje u odnosu na implementaciju u dizajnu koji se verifikuje. Značaj ove komponente je uočen prilikom verifikacije jezgra za računanje brze Furijeove transformacije. Prilikom verifikacije FFT IP jezgra, kompanije Xilinx, korišćen je programski paket Matlab, koji u okviru svojih funkcija ima mogućnost računanja direktne i inverzne Furijeove transformacije. Ovako realizovan referentni model je na svojim ulazima dobijao isti set kao i FFT IP jezgro (slika 4.2.1.). Funkcija checkera se svela na poređenje izlaza FFT IP jezgra sa rezultatima referentnog modela. Na slici 4.2.1. se vidi da je apsolutna greška lokalizovana na LSB ulaznog signala, tj. da uzima vrednosti iz diskretnog skupa $(0, 1, \sqrt{2})$, što znači da je odstupanje realne i imaginarne komponente signala u odnosu na referentni model iz skupa $(0, 1)$. U slučaju ostalih komponentata, ulazni i izlazni podaci su poređeni u Matlabu, kako bi se preglednije i efikasnije pratile greške i kako bi bio dostupan grafički uvid u rezultate.



Slika 4.2.1. Rezultat checkera FFT IP jezgra na uzorku od 1000 odbiraka

4.3 Scoreboard

Scoreboard je komponenta u verifikacionom okruženju koja se koristi za privremeno smeštanje podataka između checker komponente i DUVa. Imajući u vidu da je checker implementiran izvan simulatora kompanije Xilinx, koji se koristi u ovom projektu, svrha scoreboarda je da omogući prenos podataka između softverskih paketa Vivado i Matlab. Ovo se postiže razvojem posebnih procesa u okviru VHDL testbench koda, koji implementiraju upis trenutnih stanja izlaznih signala u tekstualne fajlove i ištavanje istih. Ova komponenta predstavlja spregu između programa koji izvršavaju različite funkcije u verifikacionom okruženju. U slučaju aplikacija gde je glavna funkcionalnost obrada toka podataka, ovaj vid implementacije scoreboarda i checkera se pokazao kao izuzetno pouzdano, pregledno i jednostavno rešenje.

4.4 “Tape-out” kriterijum

„Tape-out“ kriterijum predstavlja kolekciju uslova koje verifikacija mora da zadovolji pre nego što se dizajn proglasi spremnim za fizičku implementaciju na čipu. Među ovim kriterijumima, najbitniju ulogu imaju funkcionalna i kodna pokrivenost. Funkcionalna pokrivenost podrazumeva proveru celokupne funkcionalnosti dizajna. U konkretnom slučaju dizajna koji je verifikovan, ovo obuhvata proveru skremblovanja kvazi-nedeterministički generisanog seta podataka svakim od mogućih algoritama. Regresivni testovi se koriste u ovom slučaju. Ovim se dizajn izlaže kolekciji različitih scenarija sa ciljem da se ispita njegova funkcionalnost u celosti. Regresivni testovi su poslednji i najsloženiji vid testiranja dizajna. U konkretnom slučaju, testirana su dva jezgra, jedno kao prijemnik i jedno kao predajnik. Regresija se ogleda u korišćenju svih algoritama i poređenju ulaznih sekvenci predajnika i izlaznih sekvenci prijemnika. U slučaju da su prijemnik i predajnik isto konfigurisani, očekuje se da se ove sekvence poklapaju, što prilikom različite konfiguracije nije sličaj.

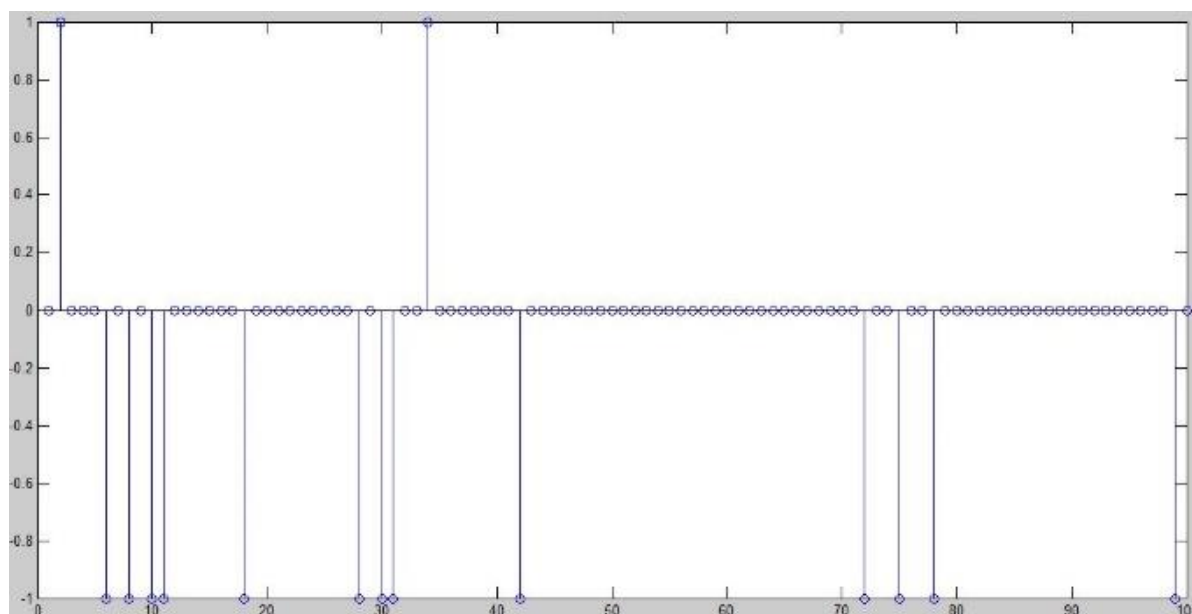
4.5 Rezultati verifikacije

Prilikom verifikacije "Multi level scrambling IP" jezgra, verifikacioni plan nalaže korišćenje DUV-a u ulozi prijemnika i predajnika.

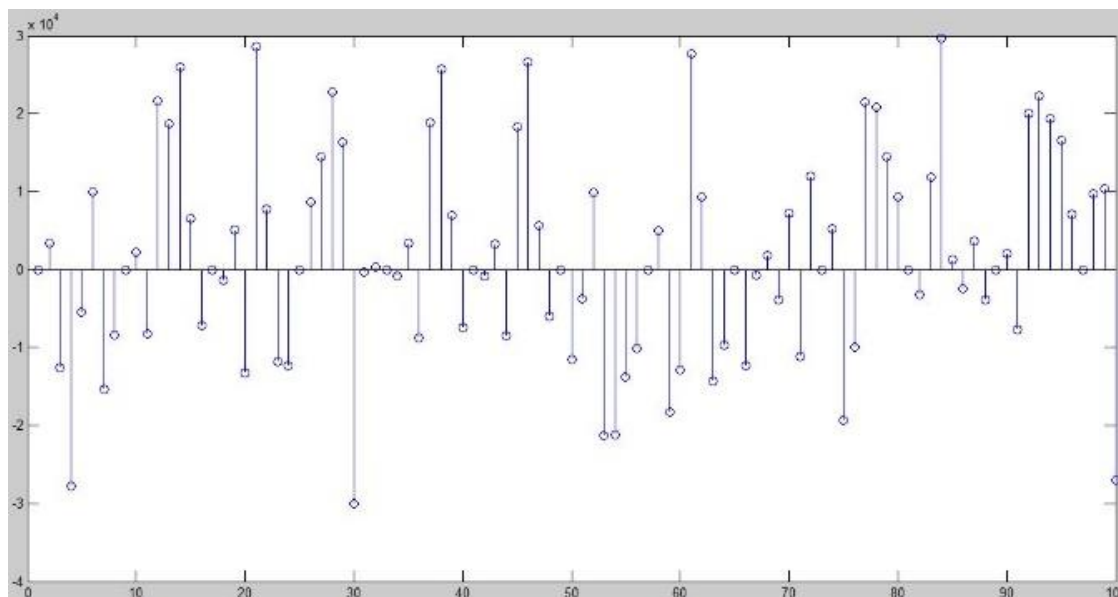


Slika 4.5.1. DUV u ulozi prijemnika i predajnika

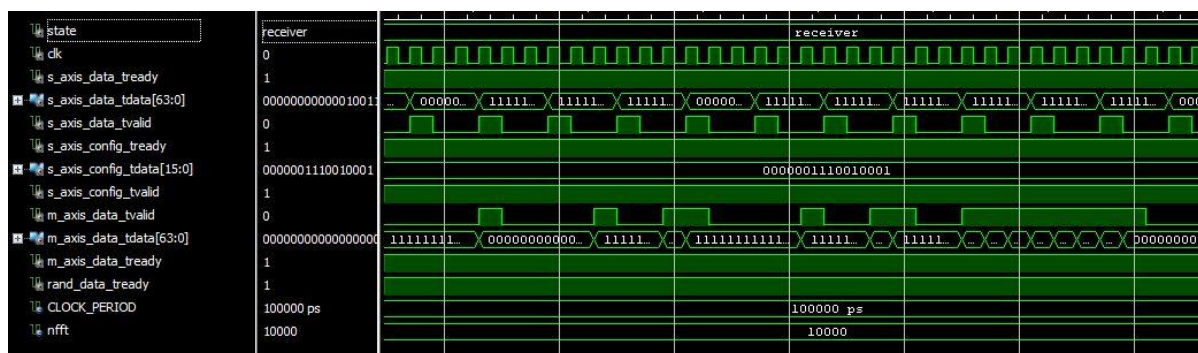
Na ulaz predajnika se prosleđuje pseudo slučajan niz, generisan LFSR generatorom. Predajnik, koristeći adekvatan algoritam i skrembling sekvencu vrši skremblovanje signala. Skremblovani signal se prosleđuje na ulaz prijemnika, koji, koristeći isti algoritam za skremblovanje i skrembling sekvencu, vrši operaciju deskremblovanja, sa ciljem dobijanja niza sličnog polaznom. Razlika ulaznog signala i skremblovanog signala na izlazu predajnika je mera kvaliteta skremblovanja, dok je razlika ulaznog signala i deskremblovanog signala na izlazu prijemnika mera kvaliteta deskremblovanja. Uspešnost dizajna se ogleda u što manjoj razlici ulaznog signala u skrembler i izlaznog signala iz deskremblera. Na slikama 4.5.2 i 4.5.3 su prikazani odnosi signala koji pokazuju uspešnost i kvalitet dizajna. Sa aspekta relativnih odnosa signala, prikazanih na slikama 4.5.2 i 4.5.3, digitalni dizajn, koji implementira funkciju skremblovanja signala se može smatrati uspešnim.



Slika 4.5.2. Odnos ulaznog i deskremblovanog signala



Slika 4.5.3. Odnos ulaznog i skremblovanog signal



Slika 4.5.4. Izgled signala tokom verifikacije (waveform)

4.6 Hardverasko testiranje dizajna

Hardversko testiranje dizajna predstavlja poslednju fazu u okviru verifikacije. Ovom prilikom se dizajnirano jezgro integriše u realno okruženje u kojem je predviđeno da radi. Prednost hardverskog testiranja je u tome što se ovim putem detektuju greške koje nisu uočene u simulaciji. Takođe, simulaciono vreme se evaluira dosta sporo u odnosu na rad sistema u realnom vremenu. Posledica toga je činjenica, da je svega par sekundi hardverskog testiranja ekvivalentno čitavom verifikacionom ciklusu. Proces detektovanja grešaka na gotovom hardveru se naziva „escape analiza“ i ona je izuzetno važan korak prilikom AISIC dizajna, gde je broj revizija čipa limitiran finansijskim ograničenjima. Obzirom da se dizajn evaluira na FPGA platformi, jednostavnim reprogramiranjem se otklanjaju greške, tako da se hardversko testiranje svodi na proces upotpunjavanja verifikacionog ciklusa.

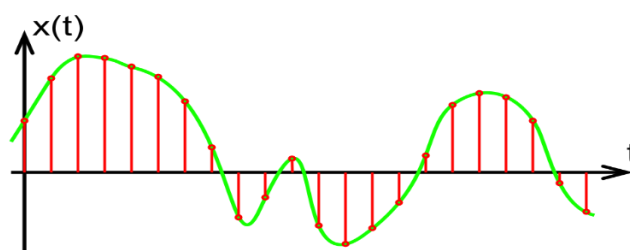
U okviru ovog diplomskog rada, hardversko testiranje predstavlja dokaz ispravnosti konceptualnog dizajna i implementacije. Na Zedboard razvojnoj ploči, kompanije Xilinx je razvijen sistem za obradu zvuka sa IP jezgrom kao centralnom komponentom. Ovaj sistem je detaljno opisan u narednom poglavlju, ali je sa stanovišta verifikacije bitno napomenuti da se sistem na realnom hardveru ponaša u skladu sa specifikacijom.

Projektovanje audio sistema i demonstracija rada IP jezgra za skremblovanje audio signala

Kao što je prethodno naglašeno, ovaj diplomski rad obuhvata demonstraciju rada dizajniranog IP jezgra na razvojnoj ploči. Ovaj deo upotpunjava diplomski rad, demonstrirajući praktičnu primenu dizajniranog jezgra.

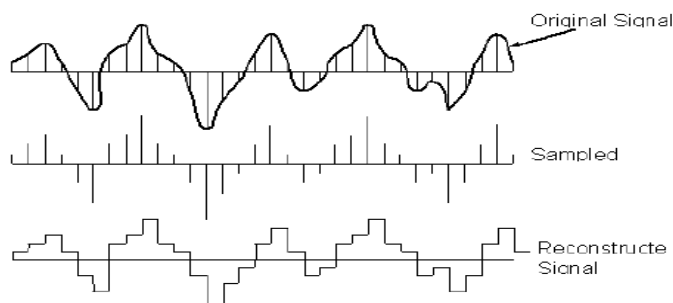
5.1 Blok dizajn i opis rada audio sistema

Projektovani dizajn na sistemskom nivou obavlja funkciju testiranja i demonstracije rada IP jezgra za skremblovanje audio signala. Ulazni signal koji se nalazi na LINE_IN portu Zedboarda je po prirodi kontinualan po vremenskoj i amplitudskoj osi. Imajući u vidu da se koristi digitalno, a ne analogno kolo za obradu signala, neophodno je kvantizovati ulazni signal po vremenu i amplitudi, kao što je prikazano na slici 5.1.1.



Slika 5.1.1. Primer AD konverzije

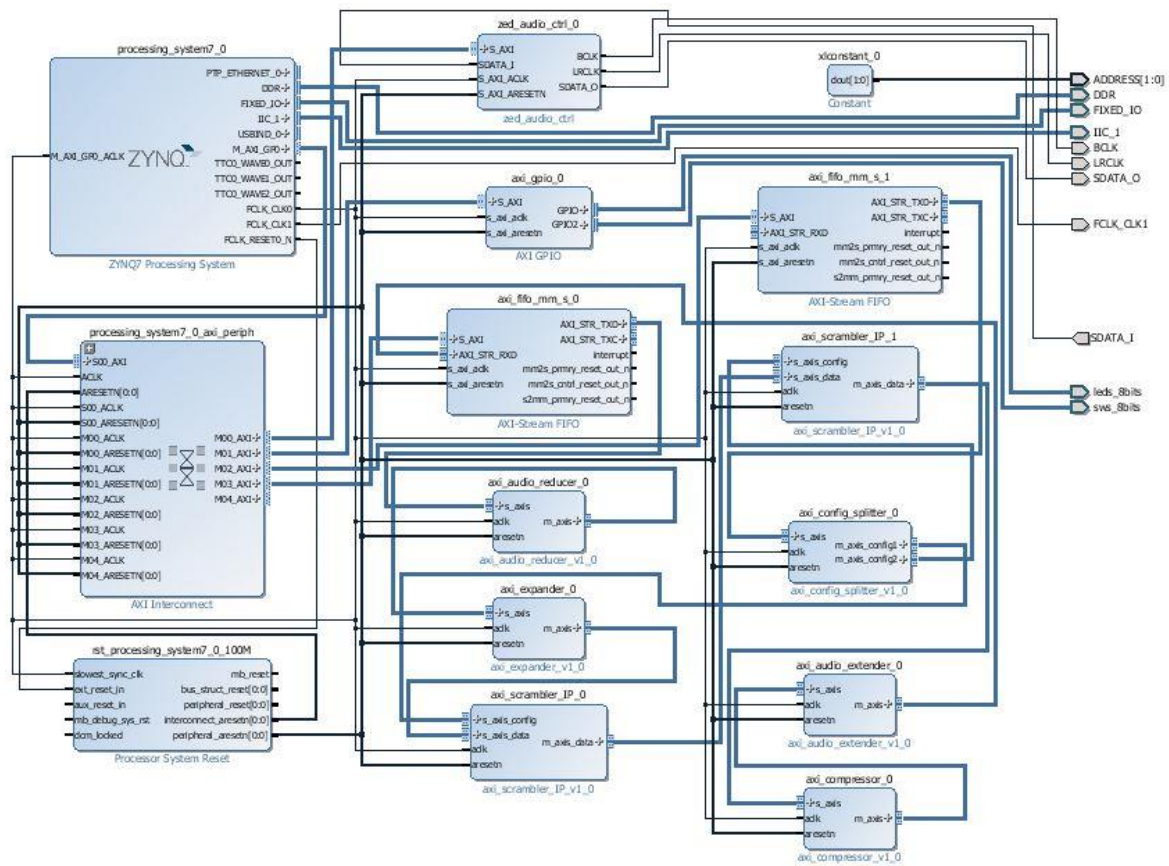
Dobijeni digitalni signal se prosleđuje u FPGA deo Zynq – 7000 čipa, tačnije, IP audio skrembleru, koji se ponaša kao predajnik i vrši proces skremblovanja. Skremblovani signal se zatim prosleđuje IP jezgru, konfigurisanom kao prijemnik, koji vrši operaciju deskremblovanja. IP jezgra se konfigurisu odvojeno, od strane korisnika, korišćenjem UART-a¹². Konfigurabilnost omogućava demonstraciju situacije kada prijemnik i predajnik koriste isti algoritam za skremblovanje (korisni audio signal se razaznaje) i kada prijemnik i predajnik nisu sinhronizovani u smislu korišćenja zajedničkog algoritma (korisni audio signal se ne razaznaje). Na kraju, digitalni signal se inverznim postupkom (DA konverzijom) prebacuje u analogni koji se dovodi na LINE_OUT port Zedboarda.



Slika 5.1.2. Rekonstrukcija signala nakon AD konverzije

¹² UART – Universal Asynchronous Receiver Transmitter – protokol asinhronne serijske komunikacije

Na slici 5.1.3. je prikazan blok dizajn audio sistema, koji obavlja gorenavedenu funkciju. U narednom poglavlju će biti detaljno opisani svi blokovi, njihova funkcija i način povezivanja sa kratkim osvrtom na softver, pomoću kojeg se upravlja datim komponentama.

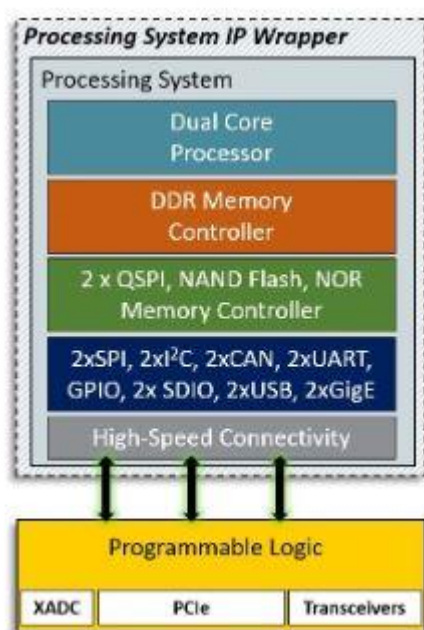


Slika 5.1.3. Blok dizajn sistema

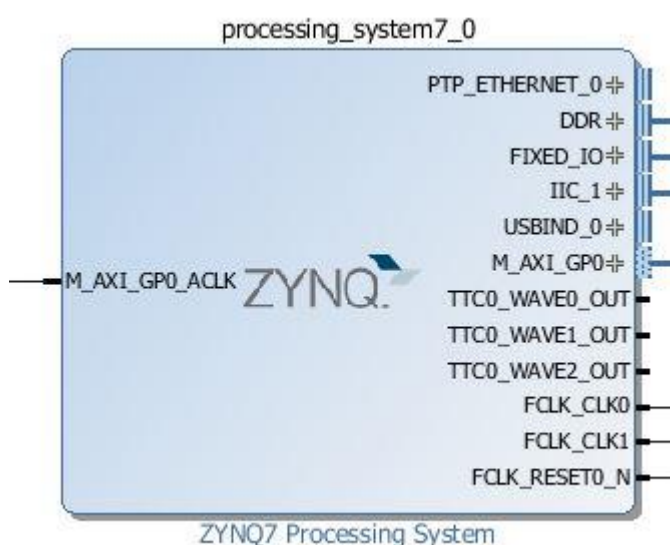
5.2 Opis pojedinačnih komponenata audio sistema

U ovom poglavlju biće detaljno opisana svaka komponenta, imajući u vidu način, na koji se ona uklapa u sistem.

Zynq processing system je IP jezgro koje predstavlja procesor u okviru Zynq – 7000 SoC¹³. Ova komponenta predstavlja vezu između PS (processing system), baziranog na ARM Cortex A9 mikroprocesoru i PL (programabile logic). Takođe obuhvata memoriju na čipu, spoljašnje memorijske interfejsa i veliki broj različitih ulazno – izlaznih interfejsa. U okviru ovog dizajna, koristi se za manipulaciju toka podataka (prihvat signala na izlazu AD konvertora i slanje signala na DA konvertor), prijem komandi poslate UART-om i konfigurisanje jezgra u skladu sa tim.



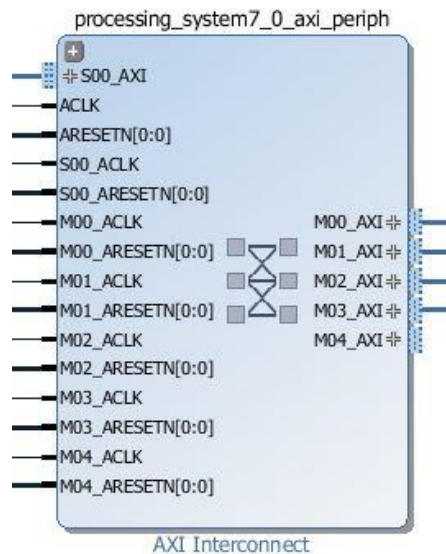
Slika 5.2.1 Blok šema Zynq 7000 SoC



Slika 5.2.2 Zynq 7000 u Vivado radnom okruženju

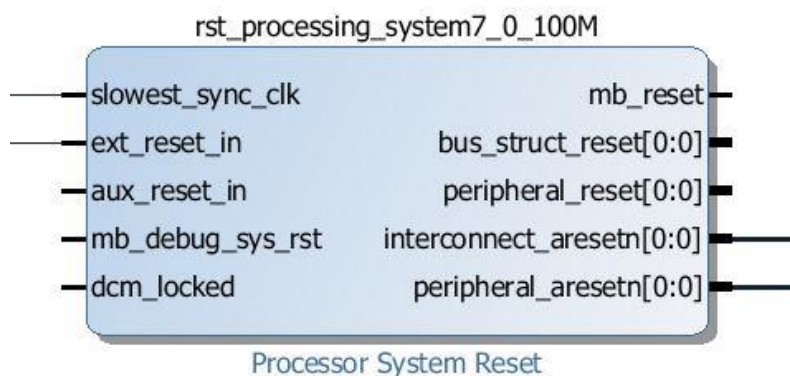
AXI interconnect je IP jezgro zaduženo za memorijsko mapiranje AXI master periferija na AXI slejv periferije. Ovo omogućava da procesor, prilikom rada, ima pristup podacima iz PL dela procesora, koji se nalaze na adekvatnim memorijskim lokacijama, dodeljenih prilikom memorijskog mapiranja komponenata, tokom generisanja bitstreama. Ovo omogućuje slanje podataka između PL i PS dela. AXI interconnect je osnovna komponenta koja predstavlja spregu procesora i GPIO, FIFO bafera i ostalih AXI-kompatibilnih komponenata.

¹³ SoC – System on Chip – Digitalni sistem koji obuhvata procesor sa svim propratnim komponentama na jednom integrisanom kolu



Slika 5.2.3. AXI Interconnect

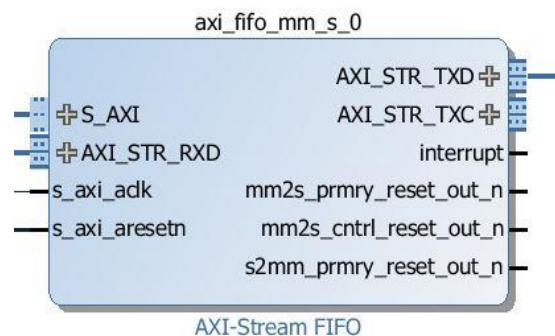
Processor system reset je IP jezgro zaduženo za generisanje reset signala koji mogu biti povezani na procesor AXI Interconnect ili AXI periferije. Ovo IP jezgro je parametrizovano, tako da korisnik može da generiše razne vrste reset sekvenci u zavisnosti od potrebe samog dizajna.



Slika 5.2.4. Processor system reset

AXI Stream FIFO obezbeđuje izuzetno pogodan način prosleđivanja podataka jezgrima koja koriste AXI-Stream komunikacioni protokol. Ovo IP jezgro je implementirano kao prijemni i predajni FIFO bafer, kontrolisani od strane procesora. Mali broj kontrolnih signala, kao i jednostavna hardverska i softverska implementacija su osnovni razlog korišćenja ove komponente. Prilikom rada, predajni FIFO bafer se napuni i u adekvatnom trenutku se isprazni u BURST režimu. To znači da se podaci šalju takt za taktom (pretpostavljajući da je prijemnik spreman za prijem) sve dok se predajni bafer ne isprazni. Imajući u vidu sporost prosleđivanja podataka, radi uštede hardverskih resursa, dubina predajnog FIFO bafra može biti izuzetno mala, čak i jedan 32-bitni registar je dovoljan. Prijem se podataka se vrši u drugačijem maniru. Prijemni FIFO bafer se puni dok ima mesta i kada procesor zatraži prijem podataka, bafer se prazni i svi podaci koji se u tom trenutku nalaze bivaju prosleđeni procesoru. Nakon analize rada Xilinx FFT IP jezgra, ustanovljeno je da se podaci izbacuju u nekonzistentnom maniru u slučaju kada nije obezbeđen kontinualni tok podataka na nivou učestanosti rada programabilne logike. Ako znamo da je učestanost audio signala mnogo manja od učestanosti rada programabilne logike, jasno je ovaj uslov nije ispunjen. Imajući ovo u vidu, dubina

prijemnog FIFO bafera se određuje kao polovina veličine furijeove transformacije koju FFT jezgro izvršava. U konkretnom slučaju, FFT IP radi FFT u 16 tačaka, pa je dubina prijemnog FIFO bafera 8. Ovo je direktna posledica radix algoritma, koji je u osnovi rada ovog jezgra. Detalji vezane za ovu komponentu se mogu naći u [7].

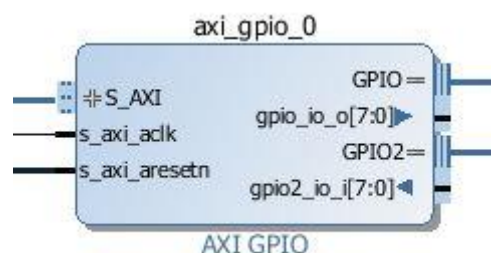


Slika 5.2..5 AXI stream FIFO bafer

AXI GPIO (General Purpose Input Output) je IP jezgro kompatibilno sa AXI protokolom gde se portovi proizvoljne širine koriste kao ulazni, izlazni ili ulazno - izlazni. Moguće su modifikacije:

- Broj kanala (jednokanalni, dvokanalni)
- Širina porta (0 – 32)
- Orijentacija porta (unilateralni – input, output i bilateralni – inout)
- Korišćenje prekidnih rutina

U ovom projektu, GPIO se koristi za generisanje sporopromenljivih signala, čija je vrednost konstantna tokom vremena, izuzev u prelaznim režimima koji se dešavaju relativno retko. Ovo obihvata prikaz skrembling sekvence, tj pomoćnog niza prilikom skremblovanja, na LED. Ova osobina nema direktan uticaj na funkcionalnost dizajna, već služi kao vizuelni pokazatelj stanja predajnika kojim upravlja korisnik. Takodje, prekidači su povezani kao GPIO ulaz i uz malu modifikaciju se mogu koristiti za prosleđivanje sekvence skrembleru. Ova osobina može biti korisna u koliko korisnik nije mogućnosti da koristi PC prilikom rada.

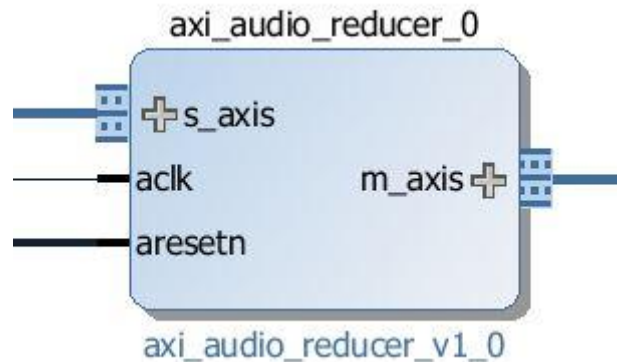


Slika 5.2.6. AXI GPIO

AXI Audio reducer je IP jezgro definisano specifično za potrebe hardverskog testiranja. Imajući u vidu da je audio signal na izlazu AD konvertora 24-bitan i da je empirijski utvrđeno da se u okviru Xilinx FFT IP jezgra događa overflow¹⁴ za 24-bitne signale, AXI Audio reducer ima ulogu redukcije najnižih bita signala, tako da je izlazni signal 16-bitan (1 bit za celobrojni i 15 za razlomljeni deo). Radi sistematičnosti, širina signala ulaznog i izlaznog signala je 32 bita, međutim, pozicije viših bita su

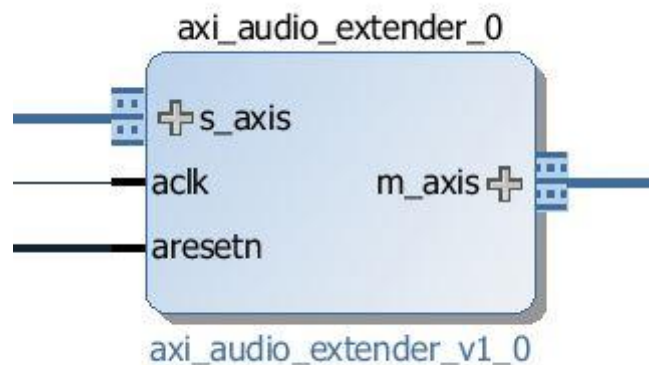
¹⁴ Overflow – prekoračenje dinamičkog opsega nakon čega je rezultat operaciju u većini slučajeva pogrešan.

inicijalno postavljene na nulu, a njihova uloga je da onemoguće prekoračenje dinamičkog opsega usled operacija unutar FFT transformacije.



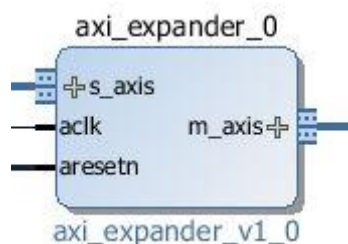
Slika 5.2.7 AXI audio reducer

AXI Audio Extender je IP jezgro, koje ima inverznu funkciju u odnosu na AXI Audio reducer. Služi da vrati signal u prvobitni 24-bitni oblik, pogodan za DA konverziju. Signal na ulazu extendera je identičan (izuzimajući grešku unesenu procesom skremblovanja) signalu na izlazu reducera. Ovo znači da je neophodno da se odgovarajući biti u okviru 32-bitne reči transliraju za onoliko mesta, koliko je bita bilo odsečeno redukcijom.



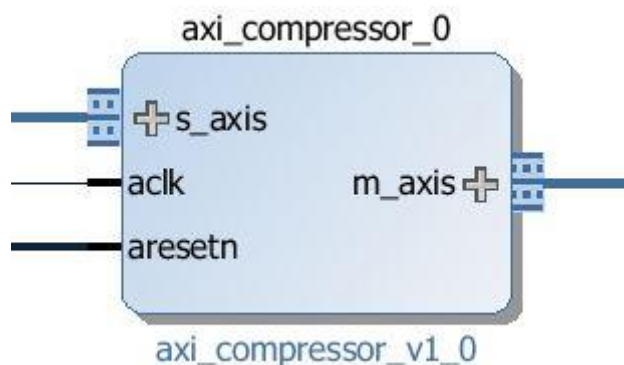
Slika 5.2.8. AXI audio extender

AXI Expander je komponenta koja se koristi da 32-bitne signale konvertuje u 64-bitne kako bi odgovarali interfejsima IP jezgra za skremblovanje audio signala. Komponenta je dizajnirana kao posledica ograničenosti AXI kompatibilnih komponenata na 32-bitnu širinu podataka. Redukcija interfejsa jezgra na 32-bitnu širinu (16 bita za realni, 16 bita za imaginarni deo) bi dovela do degradacije kvaliteta signala uzimajući u obzir da je neophodno uračunati dinamičko povećanje signala usled radix butterfly operacija. Ovaj vid adaptacije signala je samo jedan od mogućih načina. Alternativa bi bila odvojeno prosleđivanje realne i imaginarne komponente, nakon čega bi se formirao 64-bitni signal. Imajući u vidu da se na istoj ploči nalaze i prijemnik i predajnik, radi jednostavnosti i uštede hardverskih resursa, koristi se ova vrsta izjednačavanja širine signala (tzv. padding).



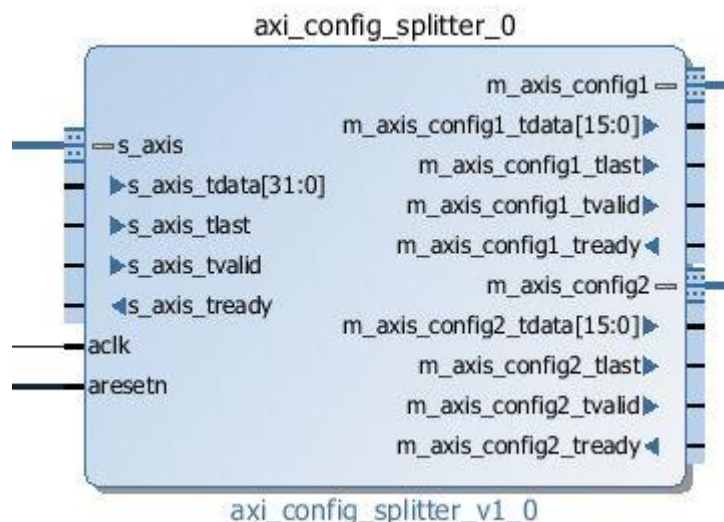
Slika 5.2.9. AXI expander

AXI Compressor je IP jezgro dizajnirano da izvršava inverznu funkciju u odnosu na AXI expander. Služi kako bi 64-bitne signale na izlazu IP jezgra za skremblovanje audio signala prilagodio 32-bitnoj širini FIFO prijemnog bafera. Redukcija podataka na 32-bitne je neophodna radi njihove dalje obrade u Zynq procesoru. Treba primetiti da je izlazni signal nakon procesa deskremblovanja ekvivalentan ulaznom signalu koji je čisto realan pri čemu je dinamička širina 24 bita. To znači da se procesom redukcije signala na 64 na 32 bita bez gubitka na informacionom sadržaju mogu uzeti samo niža 32 bita.



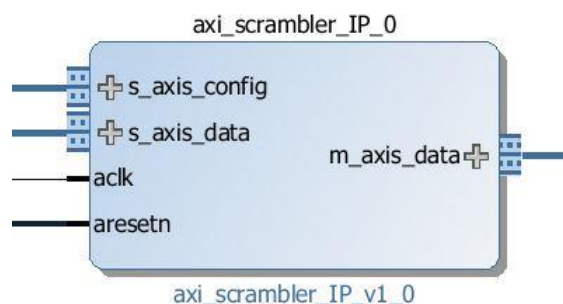
Slika 5.2.10. AXI compressor

AXI Config Splitter je IP jezgro dizajnirano sa ciljem prosleđivanja konfiguracije prijemnika i predajnika. Imajući u vidu da je širina konfiguracionog porta IP skremblera 16 bita i da AXI FIFO ima maksimalnu širinu 32 bita, intuitivno je zapaziti mogućnost da se prijemnik i predajnik konfiguriraju iz istog bafera. Ovim se ostvaruje hardverska ušteda i povećava jednostavnost samog dizajna. U realnoj situaciji, kada se prijemnik i predajnik ne nalaze na istom čipu, ova komponenta je suvišna i prijemnik i redajnik bi se konfigurisali odvojeno, obraćajući pažnju na sinhronizam njihovih konfiguracija.



Slika 5.2.11. AXI config splitter

AXI scrambler IP je centralno IP jezgro u dizajnu. Celokupan dizajn na sistemskom nivou se svodi na proveru i demonstraciju rada ovog jezgra. Detalji ovog jezgra su izloženi u prethodnim poglavljima.



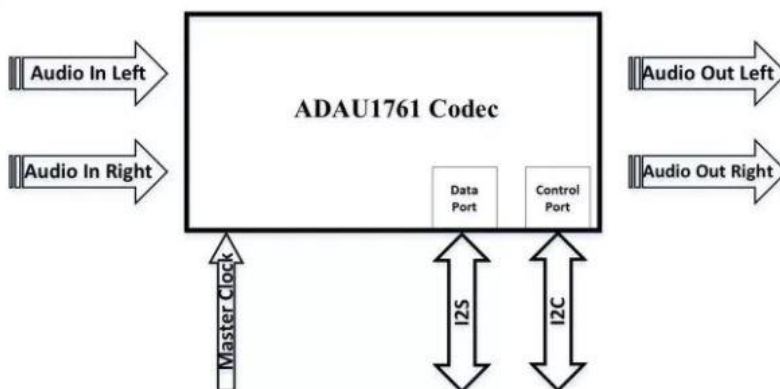
Slika 5.2.12. AXI scrambler IP core

Zed audio control Je IP jezgro koje je omogućava prijem i emitovanje audio signala. Među hardverskim resursima, Zebroard poseduje ADAU1761 audio Codec(Coder Decoder) koji se koristi za obradu audio signala u smislu kvantizacije, odabiranja, digitalno – analogne i analogno – digitalne konverzije. Ulazni signal je analogni i predstavlja uređeni par leve i desne komponente (right channel, left channel). Uzimajući u obzir Nikvistov kriterijum i frekvencijsku limitiranost ljudskog sluha od oko 20 KHz, učestanost odabiranja je 48 KHz. ADAU1761 sadrži dva ADC i dva DAC sa konfigurabilnim učestanostima odabiranja iz opsega (8 KHz – 96 KHz). Komunicira sa ostatkom sistema preko I^2C ¹⁵magistrale, korišćene za konfiguraciju jezgra i prihvatanje ulaznih i predaju izlaznih podataka.

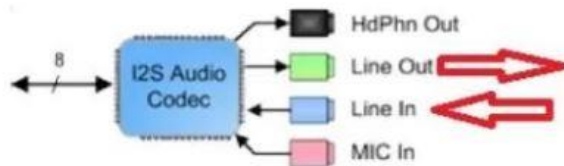
¹⁵ I^2C – Inter Integrated Circuit bidirekciona komunikacija za prenos podataka na kratkim razdaljinama između jednog mastera i više slejvova. Komunikacija se odvija serijski na nivou paketa.



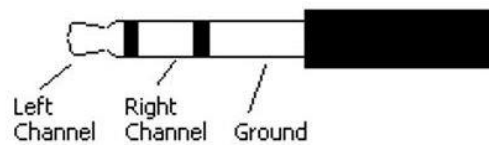
Slika 5.2.13. Zed audio control



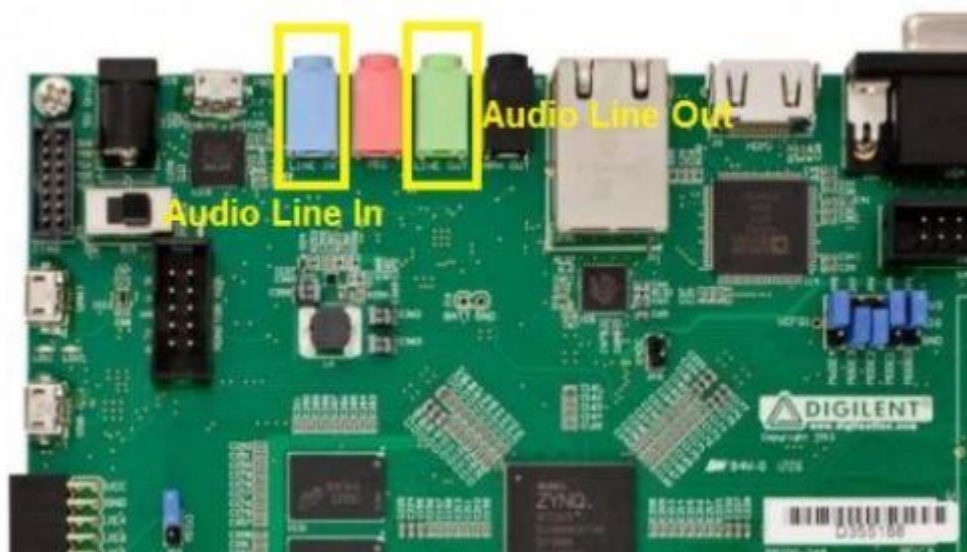
Slika 5.2 14. ADAU1761 Codec blok dijagram



Slika 5.2.15. Audio interfejs



Slika 5.2.16. Standardni audio ulaz



Slika 5. 2. 17. Audio interfejs Zedboard razvojne ploče

5.3 Razvoj softvera, grafičkog interfejsa i korišćenje audio sistema

Do sada smo opisali strukturu audio sistema na sistemskom nivou. Nakon sinteze, implementacije i generisanja bitstreama, razvija se softverska komponenta. Jedna od karakteristika embedded sistema je tesna povezanost hardvera i softvera. Ovo podrazumeva da se softver razvija za prethodno definisan hardver. Ovo usložnjava proces razvoja ali tesna povezanost softvera i hardvera garantuje poboljšanje performansi i ostavlja mogućnost za eventualne optimizacije ili premeštanje funkcionalnosti iz hardvera u softver ili obrnuto. Poboljšanje performansi se ogleda u efikasnom iskorišćenju protočne obrade i kvaziparalelizma. U konkretnom slučaju, zadatak koji treba da ispunjava softver je da omogući adekvatan rad čitavog sistema, implementiranog u PL delu Zedboarda. Takođe, komunikacija između korisnika i jezgra, kojom se vrši konfiguracija je urađena softverski. Definišimo u kratkim crtama softverske zahteve date aplikacije.

Audio signal se dovodi na analogni ulaz Zedboarda, na koji je direktno priključen AD konvertor. Na njegovom izlazu se dobija digitalni signal učestanosti 48KHz. U svakoj iteraciji beskonačne petlje, na kojoj se bazira rad sistema, jedan odbirak se šalje jezgru za obradu audio signala, dok se u istom trenutku vrši prihvatanje jednog, prethodno obrađenog odbirka. Ovaj obrađeni odbirak se zatim šalje na DA konvertor, koji se kasnije vodi na audio izlaz Zedboarda. Iz prethodno opisanog toka rada se može videti moć protočne obrade. U istom trenutku je moguće proslediti novi neobrađeni podatak i prihvatiti jedan ili više obrađenih podataka. Ovaj vid kvaziparalelizma je zasnovan na različitoj brzini rada procesora i programabilne logike. Efikasnost i potpuno iskorišćenje koncepta protočne obrade je suštinska osnova datog rešenja koje garantuje kontinualni tok podataka i rad sistema bez zastoja, što se manifestuje u kvalitetnom audio signalu. Bitno je napomenuti da je procesor takođe zadužen za interpretaciju korisničkih komandi u vidu algoritma za spremblovanje signala, kao i skrembling sekvence, tj pomoćnog niza koji se koristi u svrhu skremblovanja. Imajući u vidu da korisnik retko menja konfiguraciju sistema u odnosu na brzinu rada, pogodno je da procesor ove sekundarne funkcije radi u prekidnoj rutini.

Na osnovu prethodno izloženog, možemo parcelisati dizajn softvera na sledeće faze:

- UART komunikaciju sa korisnikom i interpretaciju komandi
- Prosleđivanje komandi i rekonfiguracija jezgra
- Komunikacija sa IP jezgrom za obradu audio signala
- AD, DA konverziju i konfiguraciju ulazno – izlaznog audio podsistema

Zadavanje komandi i konfiguracija jezgra se izvršava korišćenjem UART-a. Prilikom inicijalizacije UART-a se definiše brzina i veličina poruke, podešavanje bita parnosti i ostale karakteristike protokola. Ključna stvar koja je implementirana se odnosi na prijem i interpretaciju komandi koje zadaje korisnik. Imajući u vidu da se u odnosu na brzinu prijema, obrade i slanja podataka, rekonfiguracija jezgra dešava relativno retko, jasno je da će implementacija UART komunikacije kao prekidne rutine u najmanjoj mogućoj meri opteretiti redovan rad sistema. Kada se desi „timeout“¹⁶ tokom slanja, u okviru prekidne rutine, na osnovu sadržaja primljene poruke se vrši interpretacija, tj. ekstrakcija skremblerskog ključa i algoritma za svako od dva jezgra, kao i uključivanje i isključivanje samog protoka audio signala kroz sistem.

Za rekonfiguraciju jezgra, predaju i prijem podataka se koristi AXI-Stream FIFO bafer, opisan u prethodnom poglavlju. U sprezi sa AXI Interconnect modulom, omogućeno je memorijsko mapiranje AXI-Stream FIFO bafera. Inicijalizacija ovog modula obuhvata memorijsko mapiranje na tačnu adresu

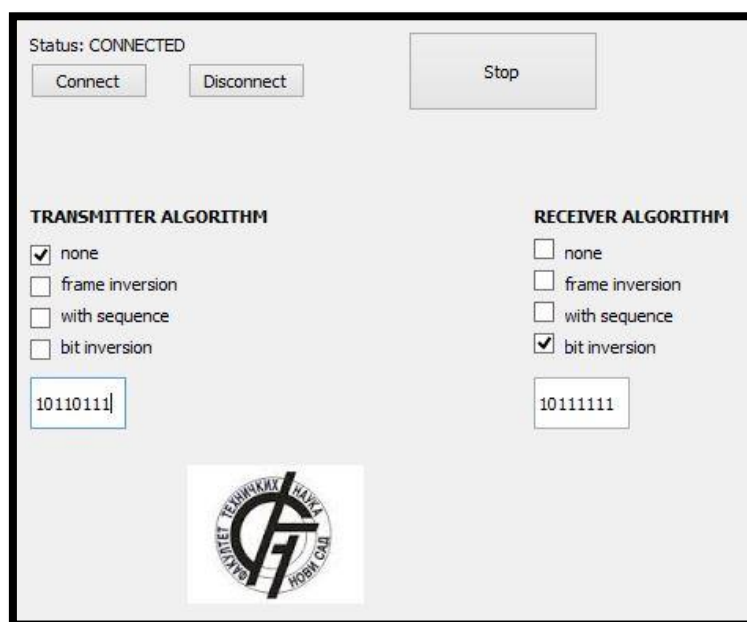
¹⁶ Nakon prijema određenog broja karaktera, nastaje pauza koja je indikator kraja transfera

u implementiranom dizajnu, specifikaciju širine i dubine bafera u skladu sa hardverskom implementacijom, kao i korišćenje predefinisanih Xilinx funkcija, koje se koriste da omoguće prenos podataka iz PL u PS deo Zedboarda i obrnuto.

DA i AD konverzija je implementirana u okviru ZED audio control IP jezgra. Inicijalizacija obuhvata upis adekvatnih vrednosti u registre koji upravljaju DA i AD konvertorima, kao u specifikaciju kvantizacije, frekvencije odabiranja i sl.

Kako bi demonstracija sistema bila preglednija, i kako bi čitava aplikacija bila „User friendly“, razvijen je GUI¹⁷ korišćenjem Qt radnog okvira. Pogodnost korisničkih interfejsa je činjenica da korisnik ne mora biti upoznat sa detaljima aplikacije, već se rad svodi na klikanje dugmića i upisivanje sadržaja u adekvatna polja.

Osnova Qt radnog okvira je koncept signala i slotova. Ovaj koncept podrazumeva veze i reakcije između objekata definisanih u okviru korisničkog interfejsa. To znači da aktiviranjem određenog objekta (npr. pritiskom na dugme ili pomeranjem položaja klizača) dobijamo reakciju tog objekta u smislu nekog algoritma, kao i reakcije svih objekata sa kojima je dati objekat povezan od strane dizajnera. Ovim se postiže „umrežavanje“ različitih objekata i njihova međusobna interakcija.



Slika 5.3.1. GUI

Na slici 5.3.1 se može uočiti konkretna implementacija korisničkog interfejsa za datu aplikaciju. Ovaj GUI u potpunosti demonstrira rad datog sistema. Odgovarajućim dugmićima se uključuje i isključuje sistem. Neophodno je uspostaviti UART vezu sa Zedboardom, što se radi pritiskom na dugme Connect. Isključivanje UART komunikacije se vrši pritiskom na dugme Disconnect. Polje status se koristi kako bi korisnik imao uvid u trenutno stanje sistema. Selekcija prijemnog i predajnog algoritma se vrši u poljima namenjenim u tu svrhu. Takođe, postoji mogućnost da korisnik zadaje pomoćni niz za skremblovanje.

¹⁷ GUI – Graphic user interface

Iz priloženog se može videti demonstracija rada sistema. Moguće je konfigurisati sistem na razne načine, korišćenjem raznih varijacija algoritama na prijemnoj i predajnoj strani, kao zadavanje skremblerskog ključa, što se od demonstrativne aplikacije i očekuje.

Zaključak

Nakon hardverskog testiranja sistema sa IP jezgrom za skremblovanje podataka kao centralnom komponentnom, možemo izvesti zaključke vezane za uspešnost same izrade ovog diplomskog rada, koja se ogleda u :

- Dizajnu IP jezgra za skremblovanje audio signala
- Verifikaciji IP jezgra za skremblovanje audio signal
- Projektovanju audio sistema koji demonstrira funkcionalnost jezgra
- Razvoju softvera za dati embedded sistem

Nakon testiranja utvrđeno je da sistem ispunjava sve uslove da se okarakterise kao uspešan. Skremblovanje signala je efektivno u toj meri da je nemoguće razaznati signal nakon skremblovanja. Deskremblovanje signal se pokazalo takođe kao uspešno realizovano zato što se signal nakon deskremblovanja ne razlikuje u odnosu na početni signal. Greška, unesena računanjem sa brojevima konačne preciznosti se ispostavila kao suviše mala da bi bila registrovana od strane čoveka, što je i očekivano.

Iako je specificirani zadatak ispunjen, dalji rad na ovom projektu je moguć. U nekoj od narednih revizija ovog dizajna bi bilo korisno razviti nove algoritme za skremblovanje koji nakon skremblovanja daju 32-bitni signal. Ovo je moguće uraditi čuvanjem relativnog položaja konjugovano kompleksnih odbiraka u frekvencijskom domenu [1]. Takođe, interesantno bi bilo razviti algoritme koji skrivaju boju glasa, ali bez gubitka na informativnom sadržaju govora.

Literatura

- [1] "Signals and systems", Alan V. Oppenheim
- [2] "Concrete mathematics", Ronald L. Graham, Donald E. Knuth, Oren Patashnik
- [3] "RTL Hardware Design Using VHDL", Pong Chu
- [4] "Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators", Xilinx 1996
- [5] "Fast Fourier Transform v9.0 - LogiCORE IP Product Guide", Xilinx 2015
- [6] "Vivado AXI Reference", Xilinx 2015
- [7] "AXI4-Stream FIFO v4.1 - LogiCORE IP Product Guide", Xilinx 2016