

Sistem za direktnu digitalnu sintezu

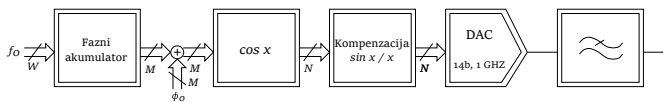
Kristijan Mitrović
Odsek za elektroniku
Elektrotehnički fakultet
Beograd, Srbija
mk150214d@student.etf.bg.ac.rs

Dragan Božinović
Odsek za elektroniku
Elektrotehnički fakultet
Beograd, Srbija
bd150211d@student.etf.bg.ac.rs

Sažetak—Ovaj rad predstavlja rešenje projekta iz predmeta *Hardversko-softverska obrada signala*. U odeljku I opisan je sistem za direktnu digitalnu sintezu koji je potrebno realizovati. U odeljku II izvršena je detaljna teorijska analiza navedenog sistema i dat predlog za implementaciju. Razvojno okruženje u okviru kojeg je izvršena implementacija, kao i sama implementacija, dati su u odeljcima III i IV, respektivno.

I. POSTAVKA PROBLEMA

Potrebno je realizovati sistem za direktnu digitalnu sintezu (DDS), prikazan na slici 1, koji se koristi za generisanje sinusnog signala u opsegu učestanosti $[\Delta f, f_{max}]$.



Slika 1. Blok dijagram DDS sistema

Učestanost generisanog signala se zadaje W -bitnom kontrolnom rečju f_0 , dok se početna faza može zadati kontrolnom rečju ϕ_0 . Pretpostavka je da se ceo sistem taktuje signalom učestanosti $f_{clk} = 100 \text{ MHz}$ i da je $f_{max} = 40 \text{ MHz}$.

Potrebno je:

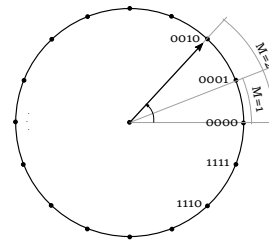
- Odrediti širinu W kontrolne reči f_0 tako da frekventijska rezolucija DDS sistema bude $\Delta f = 100 \mu\text{Hz}$.
- Predložiti arhitekturu generatora odbiraka $\cos x$ i moguće optimizacije. Proceniti složenost implementacije.
- Odrediti red i koeficijente FIR filtra za korekciju $\sin x/x$ frekventijske karakteristike kola zadržke nultog reda, tako da varijacija amplitude izlaznog signala bude $\pm 0.05 \text{ dB}$ u opsegu učestanosti od interesa.
- Odrediti specifikacije analognog rekonstrukcionog filtra tako da spektralne replike budu potisnute bar 60 dB . Na osnovu specifikacija predložiti tip i red analognog filtra.
- Predložiti arhitekturu faznog modulatora i/ili modifikaciju arhitekture DDS-a za potiskivanje spurova usled kvantizacije faze i amplitude.
- Izračunati maksimalni džiter signala takta $t_{j,clk}$ za koji ne dolazi do degradacije signala maksimalne učestanosti u prvoj i višim Nikvistovim zonama. Predložiti adekvatan izvor signala takta i izračunati džiter na osnovu profila faznog šuma.

- Razmotriti potrebne modifikacije i ograničenja za generisanje signala u trećoj Nikvistovoj zoni. Proračunati parametre modifikovanih blokova.

II. TEORIJSKO REŠENJE

A. Fazni akumulator

Fazni akumulator je deo sistema koji se taktuje sa f_{clk} i koji na svaki takt uvećava svoj sadržaj za bezdimenzionu vrednost $\Delta\theta$. Neka je, primera radi, $W = 4$ i $\Delta\theta = 0001_b$, a početna vrednost u faznom akumulatoru 0000_b . Na svaki takt vrednost u faznom akumulatoru se inkrementira za 0001 , odakle je posle jednog takta u njemu zapisano 0001_b , posle drugog 0010_b itd., sve dok nakon $2^4 = 16$ taktova ne dođe do prekoračenja $1111_b \rightarrow 0000_b$, nakon čega se ciklus ponavlja. Ovakvo ponašanje, nastalo kao posledica ograničene aritmetike, omogućava mapiranje vrednosti faznog akumulatora u ugao jediničnog vektora koji rotira u faznoj ravni, kao što je prikazano na slici 2.



Slika 2. Rotacija vektora u faznoj ravni

Krug je podeljen na 2^W delova, gde razmak između dve sukcesivne tačke na kružnici predstavlja najmanji mogući inkrement faze koji je jednak 40 -obitnom broju kod koga su svi biti sem najnižeg nule. Dakle, najmanji inkrement faze je $\Delta\theta_{min} = 00 \dots 01_b$. Pri obilasku kružnice za inkrement faze koji nije najmanji, događa se da se neke tačke preskaču, čime se efektivno postiže da se krug obiđe više puta za isto vreme, tj. veća frekvencija sinusa čiji je argument ova faza. Na osnovu toga može se napisati

$$f_0 = M_{inc} \frac{f_{clk}}{2^W}, \quad (1)$$

gde je M_{inc} faktor koji diktira "preskakanje" tačaka na kružnici, a f_0 učestanost izlaznog sinusa. Oдавde se može izraziti najmanji inkrement frekvencije kao

$$\Delta f = \frac{f_{clk}}{2^W}, \quad (2)$$

i faktor inkrementa kao

$$M_{inc} = \frac{f_0}{f_{clk}} 2^W. \quad (3)$$

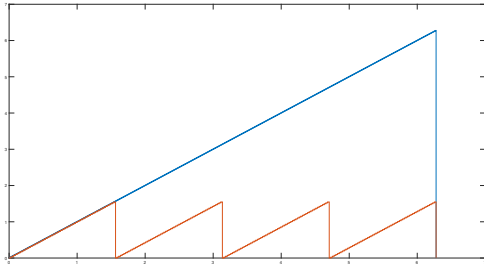
Iz (2) se preuređivanjem dobija da je

$$W = \left\lceil \log_2 \frac{f_{clk}}{\Delta f} \right\rceil, \quad (4)$$

što daje $W = 40$ za zadate brojne vrednosti. Jednačina (3) omogućava da se za unetu željenu učestanost izlaznog sinusa dobije inkrement $\Delta\theta = M_{inc} \Delta\theta_{min}$ kojim se sadržaj faznog akumulatora uvećava na svaki takt.

S obzirom da je DAC 14-obitni, N je fiksirano na 14. Kao maksimalna smisljena vrednost za M takođe se nameće 14, budući da je time obezbeđeno jednoznačno mapiranje ulaza u generator odbiraka u vrednosti njegovih izlaza.

Najviših 14 bita faznog akumulatora iskorišćeno je za mapiranje faze u $[0, 2\pi]$. Od tih 14 bita, 12 nižih bita iskorišćeno je za mapiranje ugla $[0, \pi/2]$ u $[0, 2^{12}]$, dok su preostala dva iskorišćena kao indikator kvadranta u kom se ugao nalazi. Na slici 3 plavom bojom prikazana je vrednost unutar faznog akumulatora, a crvenom njegov 12-obitni izlaz za jednu punu rotaciju u faznoj ravni pri $M_{inc} = 1$. Crveni signal se direktno vodi na generator odbiraka.



Slika 3. Izlaz faznog akumulatora

B. Generator odbiraka $\cos x$

Jedna mogućnost za generisanje amplitude na osnovu argumenta kosinusne funkcije je *look-up* tabela. Prednost ovog pristupa je što se vrednost amplitude ne proračunava u toku izvršavanja, već se prostim indeksiranjem tabele (niza) iz nje dovhatu prethodno izračunata vrednost, što se može obaviti vrlo brzo. Ukoliko nikakve metode kompresije nisu primenjene (recimo, čuvanje amplituda samo za četvrtinu uglova, imajući u vidu simetriju sinusnog signala, ili neke optimizacije sa bitima), tabela će imati $2^M = 2^{14}$ ulaza, što je $14 \times 2^{14} = 28 \text{ kB}$ podataka smeštenih u ROM memoriju. Imajući u vidu da je, recimo, veličina EEPROM memorije kod *Arudino Uno* mikrokontrolera 1 kB, postaje jasno zašto ovakav pristup nije praktičan. Povećanjem broja bita sa n na

$n+1$, broj ulaza u tabelu se duplira, što znači eksponencijalnu složenost u zahtevima za memorijom.

Sa druge strane, *CORDIC* algoritam se vrlo često primenjuje kod generisanja elementarnih, hiperboličkih i trigonometrijskih funkcija, kao i kod numerički kontrolisanih oscilatora, što je situacija sa *DDS* sistemom koji ovde razmatramo. Čisto softverska implementacija *CORDIC*-a zahteva proste iteracije kroz petlju u kojoj se vrše operacije sabiranja, oduzimanja i pomeranja, a maksimalna greška aproksimacije ugla, kao i povećanje rezolucije za jedan bit, mogu se obezbediti povećanjem broja iteracija.

Za izračunavanje sinusa i kosinusa ugla, *CORDIC* algoritam se primenjuje u tzv. *rotacionom* modu. Algoritam se sastoji iz M iteracija kroz petlju u kojoj se vrše sledeća izračunavanja:

$$x_{i+1} = x_i - \sigma_i y_i \gg i, \quad (5)$$

$$y_{i+1} = y_i + \sigma_i x_i \gg i, \quad (6)$$

$$z_{i+1} = z_i - \sigma_i \arctan 2^{-i}, \quad (7)$$

$$\sigma_i = \text{sgn}(z_i). \quad (8)$$

Rotacija vektora za ugao θ može se izvršiti u više koraka, odnosno mikrorotacija:

$$\text{ROT}(\theta) = \prod_i \text{ROT}(\sigma_i \theta_i) \quad (9)$$

Imajući u vidu da je

$$\text{ROT}(\sigma_i \theta_i) = \begin{bmatrix} \cos \sigma_i \theta_i & -\sin \sigma_i \theta_i \\ \sin \sigma_i \theta_i & \cos \sigma_i \theta_i \end{bmatrix}, \quad (10)$$

i pogodnim odabirom $\theta_i = \arctan 2^{-i}$, dobija se da je

$$\text{ROT}(\sigma_i \theta_i) = \frac{1}{\sqrt{1+2^{-2i}}} \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix}. \quad (11)$$

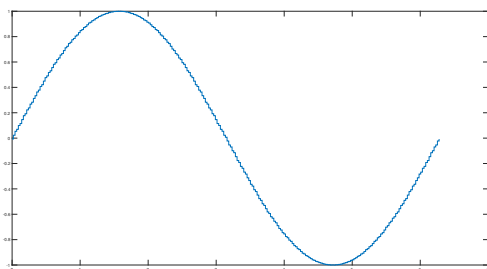
Izraz koji množi matricu je multiplikativni faktor $K(i)$, koji se za zadati broj mikrorotacija n može unapred izračunati kao

$$K(n) = \prod_{i=0}^{n-1} K_i = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1+2^{-2i}}}, \quad (12)$$

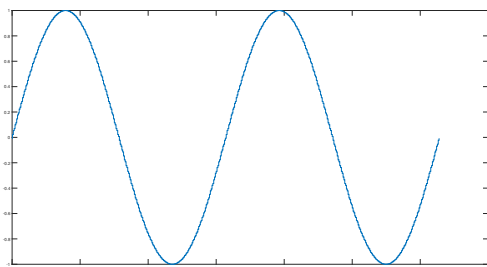
čija granična vrednost za $n \rightarrow \infty$ iznosi $K_\infty \approx 0.60726$.

Kako se generatoru signala, na osnovu blok dijagrama sa slike 1, dostavlja vrednost argumenta na osnovu koje treba da generiše amplitudu, mora se u obzir uzeti i vreme potrebno za izračunavanje izraza (5) – (8). Od W bita, generatoru odbiraka dovodi se M najviših bita faznog akumulatora, čime je obezbeđeno 2^{W-M} taktova vremena da algoritam sračuna amplitudu za vrednost argumenta koja je trenutno na njegovom ulazu, pre nego fazni akumulator promeni taj argument.

Na slici 4 prikazan je izlaz generatora odbiraka kada se na njegov ulaz dovede crveni signal sa slike 3. Prostim povećanjem M_{inc} sa 1 na 2 dobija se sinus duplo veće učestanosti sa slike 5.



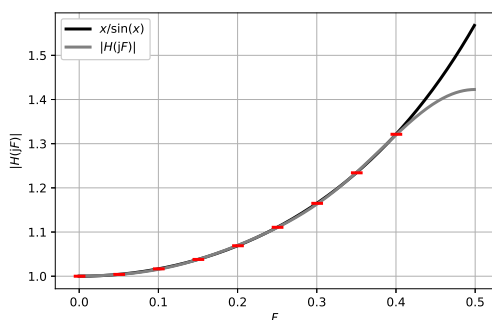
Slika 4. Sinus osnovne učestanosti



Slika 5. Sinus duplo veće učestanosti od osnovne

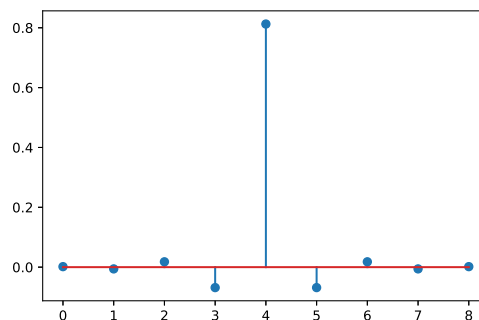
C. FIR filter

U cilju korekcije prenosne funkcije kola zadržke nultog reda projektovan je FIR filter. Kako bi varijacija amplitude izlaznog signala u opsegu $[\Delta f, f_{max}]$ bila ± 0.05 dB, potrebno je iznaći najmanji red filtra koji u željenom opsegu dovoljno dobro aproksimira inverznu funkciju funkcije $\sin x/x$. To je obezbeđeno tako što je funkciji `firls` u *Python*-u prosleđen niz diskretnih frekvencija i vrednosti željene amplitude u tim frekvencijama, kao što je na slici 6 prikazano crvenom bojom.



Slika 6. Koeficijenti projektovanog FIR filtra

Kao rezultat dobijen je FIR filter osmog reda sa slike 7, čiji su koeficijenti prikazani u tabeli I.

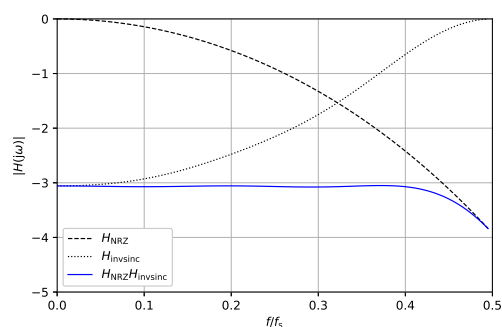


Slika 7. Koeficijenti projektovanog FIR filtra

Tabela I
KOEFIČIJENTI FIR FILTRA

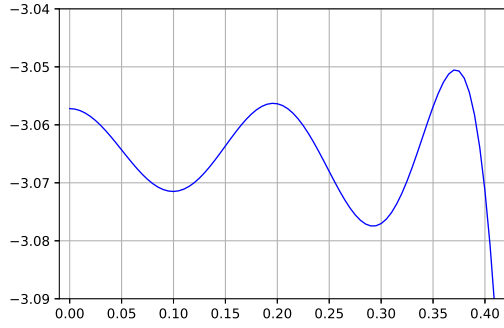
Koeficijent	Vrednost
$h_{inv\text{sinc}}[0]$	0.00170522
$h_{inv\text{sinc}}[1]$	-0.00583712
$h_{inv\text{sinc}}[2]$	0.01786389
$h_{inv\text{sinc}}[3]$	-0.06833815
$h_{inv\text{sinc}}[4]$	0.81251125
$h_{inv\text{sinc}}[5]$	-0.06833815
$h_{inv\text{sinc}}[6]$	0.01786389
$h_{inv\text{sinc}}[7]$	-0.00583712
$h_{inv\text{sinc}}[8]$	0.00170522

Frekvencijski odzivi kola zadržke nultog reda, projektovanog kompenzacnog FIR filtra i njihov proizvod, prikazani su na slici 8.



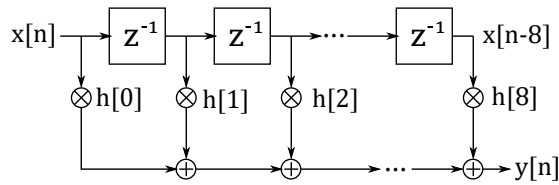
Slika 8. Kompenzacija NRZ kola

Uvećan prikaz kompenzovanog kola zadržke nultog reda dat je na slici 9, sa koje se jasno vidi da je amplituda u opsegu $[-3.05, -3.08]$ [dB], čime su zadovoljeni zahtevani gabariti.



Slika 9. Varijacija amplitude u opsegu od interesa

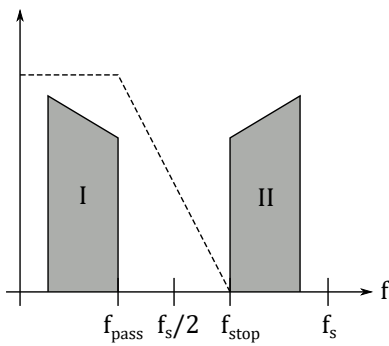
Filtriranje signala je ostvareno pomoću direktne realizacije sa slike 10.



Slika 10. Direktna realizacija FIR filtra

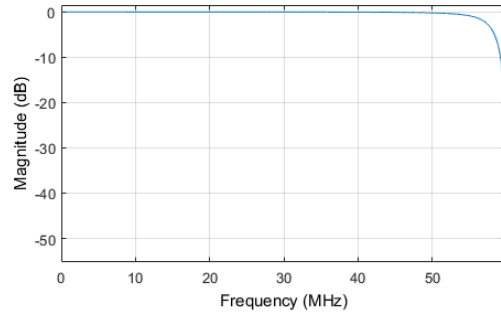
D. Analogni rekonstrukcioni filter

U cilju suzbijanja spektralnih replika na višim učestanostima, na izlazu DAC-a postavljen je analogni NF filter. Zbog specifičnog načina preslikavanja spektralnih replika, kao na slici 11, moguće je zaći u drugu Nikvistovu zonu za $f_s/2 - f_{max}$, tj. do $f_{stop} = 60$ MHz, čime se dobija prelazna zona NF filtra od 20 MHz.



Slika 11. Analogni NF filter

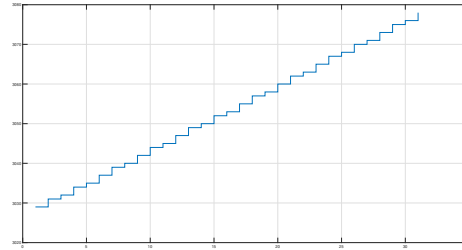
Pomoću *Filter Design & Analysis Tool*-a u *Matlab*-u projektovan je Batervortov filter sa parametrima $F_{pass} = 40$ MHz, $F_{stop} = 60$ MHz, $A_{pass} = 0.05$ dB i $A_{stop} = 60$ dB. Time je dobijen filter prvog reda, gde je na slici 12 prikazan njegov frekvencijski odziv.



Slika 12. Analogni rekonstrukcioni filter

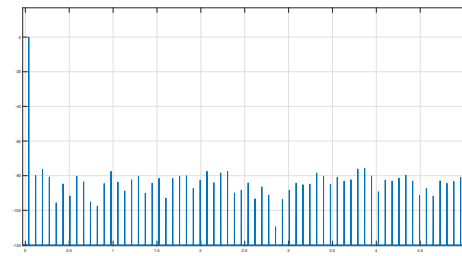
E. Potiskivanje spurova

Usled konačne dužine reči za predstavu argumenta generatora odbiraka, dolazi do neuniformne kvantizacije faze. Na slici 13 dat je uveličan segment izlaza faznog akumulatora već prikazanog na slici 3.



Slika 13. Neuniformna kvantizacija faze

Kao što se dá uočiti, razlika između dva uzastopna nivoa nije uniformna, što implicitno uzrokuje neuniformnu kvantizaciju amplitude, usled čega se u spektru izlaznog signala javljaju periodične smetnje, tzv. spurovi, prikazani na slici 14.



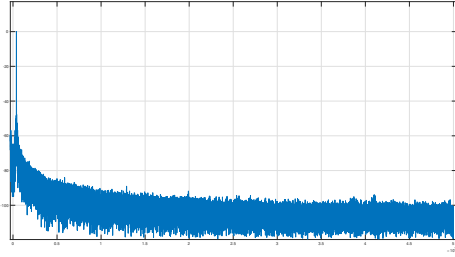
Slika 14. Spektr izlaznog signala sa uočljivim spurovima

Do pojave spurova u spektru dolazi ukoliko je zadovoljena relacija

$$\frac{f_{out}}{f_{clk}} = \frac{P}{M}, \quad (13)$$

gde je f_{out} željena učestanost signala, f_{clk} učestanost odabiranja, P paran broj, a $M = 2^m$ broj odbiraka. Jedna od tehnika za razbijanje spurova je tzv. *ditering*. Ova tehnika sastoji se

od dodavanja nekorelisanog ili pseudoslučajnog signala na "spurovit" signal, čime se razbijaju periodične komponente. Na slici 15 prikazan je spektar izlaznog signala popravljen nesupstraktivnim diterom, na kom se više ne uočavaju spurovi.



Slika 15. Spektar izlaznog signala bez spurova

F. Džiter

Podrtavanje takta unosi nesigurnost u trenutke odabiranja signala, čime se efektivno unosi šum u signal. Odnos signal-šum usled džitera takta je

$$SNR = 10 \log_{10} \frac{P_{signal}}{P_{noise}} = 20 \log_{10} \frac{1}{2\pi f t_{j,clk}}. \quad (14)$$

Imajući u vidu da je

$$SNR = SQNR = 6.02N + 1.76 \text{ dB}, \quad (15)$$

dobija se da je

$$t_{j,clk} = \frac{1}{2\pi f} 10^{-\frac{6.02N+1.76}{20}}. \quad (16)$$

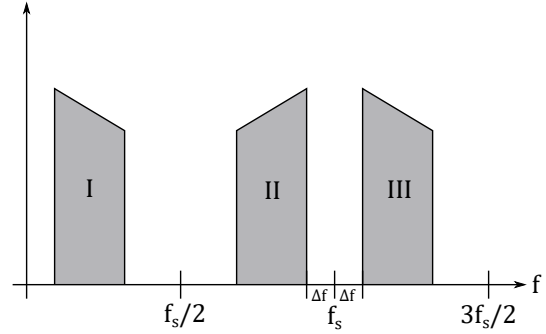
Zamenom $f = f_{max}$ i $N = 14$, dobija se $t_{j,clk} \approx 0.2 \text{ ps}$. Ovako mali džiter može se realizovati pomoću *LTC6951* i *LMX2594* PLL-ova ili *ADS5401* i *ADC32RF42* AD konvertora.

G. Generisanje signala u trećoj Nikvistovoj zoni

Kolo zadržke nultog reda pogodno je za rekonstrukciju signala u prvoj Nikvistovoj zoni, pošto potiskuje spektralne replike u višim zonama. Rekonstrukcija signala u višim Nikvistovim zonama ostvariva je pogodnim odabirom kola zadržke. Jedan od mogućih izbora je bipolarno kolo zadržke nultog reda kod kojeg slabljenje u trećoj Nikvistovoj zoni ne premašuje $\approx 13 \text{ dB}$. Ipak, ovo kolo unosi neravnomerno slabljenje u opsegu $f/f_s = [1, 1.5]$, što nije poželjno. Ovaj problem može se prenebrežniti bipolarnim kolom zadržke nultog reda sa povratkom na nulu, sa slabljenjem nešto većim od oko 10 dB , ali koje je približno konstantno na pomenutom opsegu.

Takođe, potrebno je modifikovati analogni rekonstrukcioni filter sa NF na propusnik opsega $[f_s, \frac{3}{2}f_s]$. Kao što se može videti sa slike 16, ovakav filter zahtevao bi izuzetno usku prelaznu zonu između $f_s - \Delta f$ koje pripada drugoj i $f_s + \Delta f$ koje pripada trećoj Nikvistovoj zoni. Prelazna zona širine $2\Delta f = 200 \mu\text{Hz}$ iziskuje veoma veliki red filtra. Jedino vidno

rešenje tog problema je povećanje minimalne učestanosti signala koje bi obezbedilo širu prelaznu zonu i ujedno manji red filtra.



Slika 16. Replike spektra u višim Nikvistovim zonama

Najviša Nikvistova zona u kojoj je moguće izvršiti odabiranje je

$$k_{max} = \left\lfloor \frac{f_H}{B} \right\rfloor. \quad (17)$$

Kako bismo obezbedili $k = 3$, pri $f_{max} = 40 \text{ MHz}$, potrebno je da bude $B = 12 \text{ MHz}$, usled čega se dobija da je

$$\frac{2f_H}{k} \leq f_s \leq \frac{2f_L}{k-1}, \quad (18)$$

$$26.7 \text{ MHz} \leq f_s \leq 28 \text{ MHz}. \quad (19)$$

Dakle, da bi se obezbedilo odabiranje u trećoj Nikvistovoj zoni sa manjim f_{clk} potrebno je smanjiti širinu prelazne zone na samo 12 MHz , čime se zaista obezbeđuje manja učestanost odabiranja, ali opseg učestanosti koji je moguće realizovati postaje značajno manji. Stoga je najbolje zadržati prvobitnu učestanost odabiranja od $f_{clk} = 100 \text{ MHz}$, a Δf povećati dovoljno da se obezbedi šira prelazna zona koja bi omogućila dogledan red prethodno opisanog modifikovanog analognog filtra na izlazu.

III. RAZVOJNO OKRUŽENJE

Programski kod koji obavlja zahtevanu funkcionalnost sistema napisan je u programskom jeziku C++, dok je razvojno okruženje korišćeno za izradu ovog projekta *Visual Studio Code*.

Za potrebe fiksne aritmetike korišćene su biblioteke *ac_fixed.h* i *ac_int.h*, pri čemu prva omogućava rad sa brojevima sa fiksnim zarezom dok druga biblioteka služi za korišćenje celobrojnih promenljivih ograničenih na određen broj bita.

Funkcionalnost je ostvarena u okviru fajla *"main.cpp"*. Vrednosti koje se po uslovu zadatka vode na DAC konvertor i određeni međurezultati bivaju smešteni u fajl *"DDSlog.csv"*, dok fajl *"logProcess.m"* predstavlja skriptu koja se pokreće iz *MATLAB*-a i služi za prikaz rezultata smeštenih u okviru *log* fajla.

IV. FUNKCIONALNI BLOKOVI REALIZOVANOG SISTEMA

A. Fazni akumulator

Fazni akumulator realizovan je u vidu promenljive `phaseAcc`. Ova promenljiva, budući da je ograničena širinom od $W = 40$ bita, predstavlja broj u opsegu $[0, 2^{40} - 1]$. Frekvencija generisanog signala određena je inkrementom faznog akumulatora, označenog sa $d\theta$, pri čemu minimalna vrednost inkrementa predstavlja frekvencijsku rezoluciju. Ipak, budući da su obe promenljive bezdimenzioni brojevi, omogućena je intuitivna manipulacija ovim vrednostima. Imajući u vidu da je najmanji inkrement faznog akumulatora jednak $\Delta\theta_{min} = 2^0 = 1$ i jednačinu (3) $d\theta$ se računa po formuli

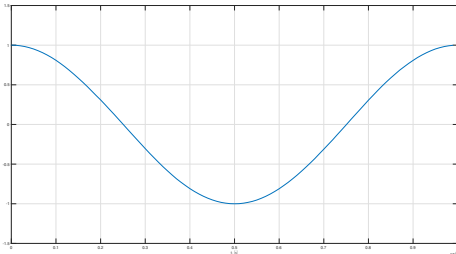
$$\Delta\theta = M_{inc} \Delta\theta_{min} = \frac{f_0}{\Delta f} = \frac{f_0}{f_{clk}} 2^W, \quad (20)$$

pri čemu je sa f_0 označena učestanost željenog signala a sa f_{clk} učestanost takta. Ovakvim pristupom, korisniku je omogućeno da unese željenu frekvenciju u vidu promenljive f_0 koja nadalje biva skalirana u odgovarajući inkrement faznog akumulatora, čime u potpunosti biva lišen unošenja neintuitivnih brojeva. Ujedno se za najmanju ostvarivu frekvenciju Δf dobija da je $M_{inc} = 1$, odnosno, inkrement faznog akumulatora jednak je bitu najmanje težine. Maksimalna ostvariva učestanost ovakvog sistema, $f_s/2$, biva preslikana u inkrementalni pomeraj od 2^{W-1} , odnosno, u pomeraj koji je jednak π .

Početna faza, označena sa ϕ_0 , predstavlja takođe ceo broj dužine $W = 40$ bita. Korisnik unosi željenu početnu fazu preko promenljive `phi0`, pri čemu njena vrednost pripada intervalu $[0, 2\pi]$. Tako unesena vrednost se dalje skalira: kako maksimalni pomeraj od 2^W odgovara inkrementu faze za 2π , to znači da za $\phi_0 = 2\pi$ inkrement akumulatora mora biti jednak 2^W . Stoga, početna faza koja se pridodaje faznom akumulatoru računa se po formuli

$$\phi = \frac{\phi_0}{2\pi} 2^W \quad (21)$$

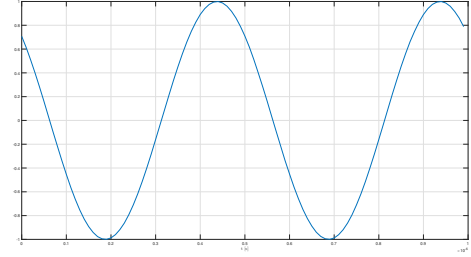
Na slici 17 prikazan je kosinusoidalni signal čija je učestanost $f_0 = 1$ MHz.



Slika 17. Kosinus učestanosti 1 MHz

Promenom korisnički definisanih parametara na $f_0 = 2$ MHz i $\phi_0 = \pi/4$ dobija se kosinusoida prikazana na slici 18.

Kao što se da videti, učestanost kosinusoida je duplo veća, pri čemu je izvršen fazni pomeraj koji je unet od strane korisnika.



Slika 18. Kosinus učestanosti 2 MHz i početne faze $\pi/4$

B. Generator odbiraka $\cos x$

Izlaz iz faznog akumulatora, promenljiva `phaseAccOut`, dovodi se na ulaz u *CORDIC* algoritam, gde je nadalje potrebno izračunati vrednost amplitude na osnovu dovedene vrednosti faze.

Kako je maksimalna vrednost ugla koja se može proslediti *CORDIC* algoritmu jednaka 99.88° , potrebno je na neki način ograničiti vrednost ugla koja se prosleđuje za izračunavanje amplitude. To je učinjeno tako što se najviših dva bita dovodi na ulaz multipleksera u vidu promenljive `muxSel`, dok se ostalih 12 bita koristi za mapiranje vrednosti u opsegu od $[0, \frac{\pi}{2}]$ u vidu `phaseAccOut`.

Sama `for` petlja koja realizuje $M = 14$ iteracija *CORDIC* algoritma implementira jednačine (5) – (8). Radi optimizacije, pre samog početka rada napravljena je svojevrsna tabela vrednosti za $\arctan 2^{-i}$. Budući da se taj niz vrednosti pojavljuje u svakoj iteraciji algoritma, ovim je postignuta manja redudansa kao i brže izvršavanje samog programa. Takođe, imajući u vidu zaključak izveden iz jednačine (12), $K(n) = K_\infty$ je sračunato pre samih iteracija *CORDIC*-a.

Kod izračunavanja trigonometrijskih funkcija pomoću *CORDIC*-a postavljaju se sledeće početne vrednosti promenljivih: $x_0 = K_\infty$, $y_0 = 0$. Ipak, kako kontrolna promenljiva na ulazu u *CORDIC* algoritam, `phaseAccOut`, a koja bi trebalo da nosi informaciju o željenoj vrednosti ugla z_0 , nema nikakvu realnu dimenziju, potrebno je izvršiti skaliranje. Budući da je maksimalna vrednost `phaseAccOut` jednaka $2^{M-2} - 1$ i da ona odgovara fazi od $\pi/2$, promenljiva z se računa po formuli:

$$z_0 = \frac{\text{phaseAccOut} \pi}{2^{M-2} - 1} \frac{\pi}{2}. \quad (22)$$

Izlazne vrednosti, x i y , smeštaju se zatim u promenljive `sampleSin` i `sampleCos`, na osnovu vrednosti promenljive `muxSel`.

C. Diter

Spurovi se javljaju kao neželjene periodične smetnje u spektru izlaznog signala i posledica su neuniformne kvantizacije faze. Pri koherentnom odabiranju uslovljenom jednačinom (13), spurovi su jasno vidljivi u spektru signala, kao što se dá videti na slici 14.

Taktovanje celog sistema u C++-u implementirano je `for` petljom koja se, na osnovu jednačine (13), izvršava 2^n puta, čime je obezbeđeno logovanje celog broja perioda sinusnog signala. U implementaciji je uzeto da je $n = 16$, koje omogućava izuzetno brzo izvršavanje programa u C++-u, dok, s druge strane, *MATLAB*-u pruža dovoljan broj odbiraka za detaljno iscrtavanje signala i njegovog spektra. Naravno, ovo uzrokuje da je frekvencijski bin u *MATLAB*-u širine $\Delta f = f_{clk}/2^{16}$, a ne $\Delta f = f_{clk}/2^{40}$, ali je dovoljno dobar kompromis između oprečnih zahteva za preciznošću i kratkim vremenom izvršavanja.

Tehnika koja je iskorišćena za razbijanje spurova je tzv. *ditering*. Ova tehnika sastoji se od dodavanja nekorrelisanog ili pseudoslučajnog signala na "spurovit" signal, čime se razbijaju periodične komponente. Pseudoslučajan signal u C++-u implementiran je instanciranjem objekata `default_random_engine` i `normal_distribution` iz `<random>` biblioteke, koji omogućavaju generisanje slučajnih brojeva sa normalnom raspodelom $\mathcal{N}(\mu, \sigma^2)$ sa proizvoljnim μ i σ . Generisani pseudonasumični brojevi sa $\mu = 0$ i $\sigma = 1$ pomnoženi su sa 2^{26} kako bi se šiftovali do bita koji se prosleđuju *CORDIC* algoritmu, kao i sa `ditherAmp` čime se podešava nivo ditera koji se dodaje na fazni akumulator u svakoj iteraciji pomenute `for` petlje.

Na slici 15 prikazan je spektar izlaznog signala popravljen nesupstraktivnim diterom, na kom se više ne uočavaju spurovi.

D. FIR filter

FIR filter za korekciju $\sin x/x$ frekvencijske karakteristike kola zadržke nultog reda implementiran je direktnom realizacijom sa slike 10, koja se može matematički zapisati kao

$$y[n] = \sum_{i=0}^8 h[i] x[n-i], \quad (23)$$

što predstavlja konvoluciju koeficijenata filtra $h[0] \dots h[8]$ sa trenutnim $x[n]$ i prethodnim $x[n-1] \dots x[n-8]$ izlazima *CORDIC*-a.

Na identičan način je u nizove `cordicSin` i `cordicCos` smeštano poslednjih devet izlaza generatora odbiraka, i ti nizovi su po jednačini (23) konvoluirani sa koeficijentima filtra prethodno izračunatim u *Python*-u tako da filter zadovoljava zahtevane uslove, i smeštenim u `filter` niz. Rezultati konvolucije, tj. filtriranja `DACinputCos` i `DACinputSin` potom su upisani u izlazni fajl za logovanje.

V. ZAKLJUČAK

U ovom radu dati su teorijski aspekti nuži za razumevanje rada sistema za direktnu digitalnu sintezu, kao i opis čisto softverske implementacije navedenog sistema u programskom jeziku C++. Svi dobijeni rezultati simulacija u potpunosti se slažu sa onim što teorija predviđa.

LITERATURA

- [1] Dr. Dušan Grujić, Predavanja iz predmeta *Hardversko-softverska obrada signala*, Elektrotehnički fakultet, Beograd
- [2] Bruce Land, *Direct Digital Synthesis*, Cornell University, New York, <https://www.youtube.com/watch?v=YDC5zaEZGhM>
- [3] Analog Devices, *Fundamentals of Direct Digital Synthesis*, <https://www.analog.com/media/en/training-seminars/tutorials/MT-085.pdf>
- [4] HLS LIBS, *AC DATATYPES*, <https://hlslibs.org/#collapseACDatatypes>
- [5] C++, *std::normal_distribution*, http://www.cplusplus.com/reference/random/normal_distribution/