

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 69

**Sustav za pretipkavanje signala
temeljen na kratkoj kauzalnoj kubičnoj
splajn interpolaciji**

Luka Karlović

Zagreb, lipanj 2010.

Sadržaj

1.	Uvod	6
	Pretipkavanje signala	6
	Implementacija sustava	7
2.	Kratka kauzalna kubična splajn interpolacija	8
	Metode interpolacije ^[6]	9
	Kauzalna kubična splajn interpolacija	10
3.	Sustav za pretipkavanje signala	20
	Evaluator kubičnog polinoma	21
	Generator kubičnog polinoma	26
	Filtar frakcionalnog kašnjenja	38
	Međuspremnik za pretipkavanje	40
	Tehničke značajke sustava za pretipkavanje	53
4.	Ispitivanje rada sustava za pretipkavanje	56
	Simulacija na PC računalu	56
	Ispitivanje na sklopovlju	72
5.	Zaključak	75
6.	Popis oznaka i kratica	77
7.	Literatura	78
	Dodatak A: hrvatski naslov, ključne riječi, sažetak	79
	Dodatak B: engleski naslov, ključne riječi, sažetak	80

1. Uvod

Razvoj digitalne tehnologije omogućio je nove metode pohrane, prijenosa i obradbe signala. U većini područja primjene, digitalna obradba signala je u potpunosti zamijenila analognu.

Digitalni sustavi za pohranu, obradbu i prijenos, koriste isključivo reprezentacije stvarnih signala kvantizirane po amplitudi, i diskretizirane po vremenu.

Performanse (kapacitet) digitalnog sustava ograničavaju broj kvantizacijskih razina amplitude, i vremenski razmak uzimanja uzoraka izvornog signala. Posljedica kvantizacije i uzorkovanja (otipkavanja) je gubitak (beskonačne količine) informacija pri *digitalizaciji* realnog signala.

Različiti digitalni sustavi mogu imati različite performanse, a pritom i različite reprezentacije digitalnog signala (koje se razlikuju po broju kvantizacijskih razina amplitude i periodu otipkavanja). Problemi se javljaju pri povezivanju takvih sustava. Realan primjer je prijenos ili pohrana digitalnog signala sa nešto nižom frekvencijom otipkavanja, a korištenje (npr. reprodukcija) sa višom frekvencijom otipkavanja. Kako bi opisani sustavi mogli biti integrirani, potrebno je povećati broj uzoraka signala dobivenog iz pohrane ili prijenosnog sustava.

Moguća je i obrnuta situacija, gdje je potrebno smanjiti broj uzoraka signala.

Oba slučaja pretvorbe signala se nazivaju pretipkavanje signala.

Pretipkavanje signala

Pretipkavanje signala podrazumijeva promjenu broja uzoraka otipkanog signala (ili u specijalnom slučaju nepromijenjen broj uzoraka, ali sa unošenjem frakcionalnog kašnjenja).

Pretipkavanje signala sa povećanjem broja uzoraka nužno uvjetuje pojavu novih, prethodno nepostojećih uzoraka u pretipkanom signalu. Kod pretipkavanja signala sa smanjenjem broja uzoraka se također mogu pojaviti prethodno nepostojeći uzorci, ako se radi o necjelobrojnom omjeru smanjenja broja uzoraka izvornog i pretipkanog signala.

U oba slučaja, postupak generiranja nepostojećih uzoraka se vrši interpolacijom (konstrukcija kontinuiranog modela signala, a zatim otipkavanje modela u proizvoljnim trenutcima).

Interpolacija se tipično provodi koristeći znanje o reprezentiranim podacima. Kod dvodimenzionalnih signala (npr. slika) može se koristiti prostorna bliskost za interpolaciju nepostojećih uzoraka u slici. Kod jednodimenzionalnih signala koji imaju vremensko trajanje, za interpolaciju nepostojećih uzoraka se tipično koristi vremenska bliskost postojećih uzoraka.

Točan postupak interpolacije ovisi o konkretnoj metodi koja se koristi. Metode interpolacije mogu biti raznolike.

Implementacija sustava

U ovom radu je opisana implementacija cijelog sustava za pretipkavanje sa proizvoljnim omjerom frekvencija otipkavanja ulaznog i izlaznog signala.

Korištena tehnika interpolacije je kratka kauzalna kubična splajn interpolacija.

Ova interpolacija koristi beskonačnu količinu uzoraka koji vremenski prethode, a podesivu količinu uzoraka koji vremenski slijede nakon interpoliranog uzorka.

Sustav za pretipkavanje je implementiran na FPGA platformama Spartan-3 te Virtex, proizvođača Xilinx. Korišten je jezik za opis sklopovlja VHDL.

Sustav omogućava potpuno asinkroni rad sklopa proizvođača i sklopa potrošača signala.

Za ispitivanje su implementirani i referentni modeli sustava, u programskom jeziku Matlab. Referentni modeli uključuju idealni referentni model (radi u aritmetici pomičnog zareza dvostruke preciznosti), te referentni model koji ponašajno odgovara FPGA implementaciji u svakom bitu rezultata.

Rad sustava je ispitan i na sklopovlju, korištenjem analogno-digitalnih i digitalno-analognih pretvarača, vezanih na digitalnu (FPGA) platformu.

2. Kratka kauzalna kubična splajn interpolacija

Interpolacija je postupak konstrukcije novih točaka unutar niza postojećih točaka u nekom prostoru. Cijeli skup točaka reprezentira kontinuiranu pojavu, koja je prethodno diskretizirana (svedena na skup točaka). Interpolacija je poseban slučaj problema prijanjanja krivulje, kod kojeg konstruirana krivulja prolazi točno kroz postojeće točke.

Konačni cilj je izgradnja sustava za pretipkavanje signala u realnom vremenu. Težnja radu u realnom vremenu se suprotstavlja sa odabirom metode interpolacije za konstrukciju novih točaka signala: interpolacija po definiciji zahtijeva poznate točke iz budućnosti, odnosno sustav ne može biti kauzalan. Dubina akauzalnog zahvata ovisi o konkretnoj metodi interpolacije, no on je uvijek prisutan. Alternativna metoda konstrukcije novih točaka je ekstrapolacija. Kod ekstrapolacije se konstruirana točka nalazi izvan skupa postojećih točaka, odnosno može se ekstrapolirati buduća točka signala u realnom vremenu (stvarno kauzalan sustav). Nedostatak ekstrapolacije je bitno manja točnost od interpolacije (izrazito velika nesigurnost rezultata). Uzimajući u obzir činjenicu da realne implementacije ekstrapolacije na stvarnom sklopovlju ne mogu konstruirati budući uzorak u zadovoljavajuće kratkom vremenu (potrebno uvođenje dodatnog kašnjenja), odabir interpolacije je primjeren (uz uvijek prisutno kašnjenje sklopovlja pri konstrukciji novog uzorka, uvodi se dodatno kašnjenje kako bi nekauzalan sustav postao kauzalan).

Akauzalni zahvati interpolacije mogu biti različitih dužina, ovisno o korištenoj metodi interpolacije. Za rad sustava u realnom vremenu nije moguće primijeniti beskonačan akauzalni zahvat, jer nisu poznati svi budući uzorci signala. Potrebno je odabrati interpolacijsku metodu sa relativno kratkim akauzalnim zahvatom, kako nakon uvedenog kašnjenja kauzalni sustav ne bi imao nezadovoljavajuće veliko ukupno kašnjenje. Dužina akauzalnog zahvata tipično utječe i na točnost interpolacije, pa se metoda interpolacije odabire u skladu sa zahtjevima na rad sustava.

Metode interpolacije^[6]

Utrošak resursa i točnost interpoliranog uzorka ovise o odabranoj metodi interpolacije.

Po dijelovima konstantna interpolacija

Najjednostavnija metoda interpolacije je *po dijelovima konstantna* interpolacija. Interpolirani uzorak poprima vrijednost najbližeg poznatog uzorka. Metoda je vrlo neprecizna, no ekstremno jednostavna za implementaciju. Metoda može biti potpuno kauzalna, ako se najbliži poznati uzorak uzima iz prošlosti, ili nekauzalna sa akauzalnim zahvatom od jednog uzorka, ako se za najbliži uzorak dozvoljava odabir i prvog budućeg uzorka, u slučaju interpolacije točke u drugoj polovini frakcionalnog razmaka (između poznatih uzoraka).

Linearna interpolacija

Linearna interpolacija pretpostavlja linearnu promjenu funkcije između dva susjedna poznata uzorka. Metoda je relativno neprecizna, no računski jednostavna za implementaciju. Interpolacijska funkcija je *po dijelovima linearna* i nije diferencijabilna u točkama poznatih uzoraka.

Polinomijalna interpolacija

Polinomijalna interpolacija modelira kontinuiranu funkciju polinomom. Ova interpolacija je generalizacija linearne interpolacije. Konstruirani polinom prolazi kroz sve poznate točke funkcije (Lagrangeov polinom). Ako je poznato n diskretnih točaka, može se konstruirati polinom stupnja $n - 1$ koji prolazi kroz svaku od njih. Polinom je tipično visokog reda, te je $n - 1$ puta kontinuirano diferencijabilan. Ova interpolacijska metoda je računski vrlo zahtjevna (numerička konstrukcija, a zatim evaluacija polinoma visokog reda). Rezultati interpolacije su vrlo točni, međutim javlja se problem oscilacija (posebice na rubovima područja definicije interpolacijskog polinoma, pojava poznata kao Rungeov fenomen).

Splajn interpolacija

Splajn (*eng. spline*) interpolacija rješava probleme polinomijalne interpolacije. Splajn je funkcija *po dijelovima polinomijalna* (*eng. piecewise polynomial*). Za polinom reda n , potrebno je definirati dodatnih $n - 1$ stupnjeva slobode (tipično preko iznosa viših derivacija). Definira se prirodni kubični splajn – funkcija koja se

sastoji od $n - 1$ kubičnih polinoma za n poznatih uzoraka izvorne funkcije. Prirodni kubični splajn je *po dijelovima kubičan*, kontinuirano dvostruko diferencijabilan, sa drugom derivacijom jednakom 0 na krajevima područja definicije splajna. Kubični polinomi splajn funkcije su konstruirani na način da pristaju glatko jedan do drugoga (do razine druge derivacije). Svaki pojedini kubični polinom se koristi za interpolaciju na samo jednom od intervala između poznatih uzoraka funkcije. Splajn funkcija ima pogrešku interpolacije manju od linearne interpolacije, te nije podložna Rungeovom fenomenu (uporaba polinoma niskog stupnja). Zbog uporabe polinoma niskog stupnja, evaluacija interpolacijske funkcije nije računski zahtjevnija kao kod polinomijalne interpolacije. Tipičan stupanj polinoma je 3, na čemu se bazira familija *kubičnih splajn interpolacija*. Konstrukcija splajn funkcije sadrži akauzalni zahvat. Zbog nemogućnosti izgradnje sustava za rad u stvarnom vremenu sa akauzalnom komponentom, sustav je potrebno pretvoriti u kauzalni dodavanjem potrebnog kašnjenja u dužini (skraćene) akauzalne strane impulsnog odziva. Kako je interpolacija nekauzalna po definiciji, za izgrađeni sustav se naglašava svojstvo kauzalnosti (*kauzalna kubična splajn interpolacija*).

Kauzalna kubična splajn interpolacija

B-splajn interpolacija

B-splajn^[7] (*eng. basis spline*) je splajn funkcija koja za definirani stupanj polinoma, glatkoću i podjelu ulazne domene ima minimalnu podršku (područje vrijednosti različito od nule). B-splajn interpolacija je slična idealnoj pojasno ograničenoj *sinc* interpolaciji, međutim *sinc* ima beskonačno područje podrške. Svaka splajn funkcija određenog stupnja polinoma, glatkoće i podjele ulazne domene se može reprezentirati linearnom kombinacijom B-splajnova jednakog stupnja polinoma, glatkoće i podjele ulazne domene. Evaluacija splajna u obliku linearne kombinacije B-splajnova se može provesti tridiagonalnim matičnim de Boor-ovim algoritmom, ili filtracijom (pogodno za sustav za pretipkavanje). Filtracijska B-splajn interpolacija je postala popularna u obradi signala u realnom vremenu zbog jednostavne konstrukcije B-splajn koeficijenata filtriranjem ([3]), uz uvjet ekvidistantnosti poznatih točaka (tridiagonalna evaluacija omogućava korištenje različitih udaljenosti između poznatih točaka). Tridiagonalna formulacija također zahtijeva konačan niz uzoraka za implementaciju (i dodatne rubne operacije,

poput zrcaljenja signala, jer se vrši nad nizom konačne duljine koji je potrebno proširiti), dok je filtracijom moguća uporaba beskonačnog niza uzoraka. Za pretipkavanje signala u stvarnom vremenu su potrebne eksplicitne vrijednosti svih kubičnih polinoma, no moguće ih je konstruirati jednostavnim FIR filtriranjem B-splajn koeficijenata. Zbog navedenih razloga B-splajn interpolacija je osnova kauzalne kubične splajn interpolacije korištene u sustavu za pretipkavanje.

Članak [3] detaljno opisuje B-splajn interpolaciju i konstrukciju pripadnih koeficijenata.

Konstrukcija B-splajn koeficijenata se vrši preko izraza 2.1 ([3]) izraženog u z-domeni.

$$W(z) = \frac{Y(z)}{A(z)} \quad (2.1)$$

B-splajn koeficijenti $W(z)$ dobivaju se filtriranjem ulaznog signala $Y(z)$ *direktnim* B-splajn filtrom čija je prijenosna funkcija za kubični splajn definirana izrazom 2.2 ([3]). Potpuna prijenosna funkcija B-splajna uključuje faktor 6 ispred $H_B(z)$, koji je izbačen zbog kraćenja pri rekonstrukciji u izrazu 2.4 ([3]).

$$H_B(z) = \frac{1}{A(z)} = \frac{1}{z + 4 + z^{-1}} \quad (2.2)$$

Kubični B-splajn trećeg reda ima podršku od četiri ulazna uzorka, te se sastoji od četiri kubična polinomijalna segmenta. Da bi se iz koeficijenata B-splajna ($w[n]$) dobio kubični polinom za interpolaciju u intervalu od $y[n]$ do $y[n+1]$, potrebno je težinski zbrojiti (izraz 2.4) četiri posmaknuta B-splajn polinomijalna segmenta trećeg reda (izraz 2.3) ([3]).

$$\beta^3(x) = \begin{cases} 4 - 3x^2(2 - |x|), & |x| \leq 1 \\ (2 - |x|)^3, & 1 < |x| \leq 2 \\ 0, & \text{inače} \end{cases} \quad (2.3)$$

$$\bar{Y}_n(x) = \sum_{i=-1}^2 w[n+i] \beta^3(x - (n+i)) \quad (2.4)$$

Umjesto izraza 2.4, tipično se koristi izraz 2.5 za evaluaciju gdje se područje definicije evaluacijske funkcije ograničava na frakcionalni interval Δx između uzoraka $y[n]$ i $y[n+1]$ ([3]).

Izraz 2.5 je praktičan za primjenu u sustavu za pretipkavanje u realnom vremenu, zbog eksplicitne kubične forme pogodne za višestruku evaluaciju.

$$Y_n(\Delta x) = \bar{Y}_n(n + \Delta x) = d_B[n]\Delta x^3 + c_B[n]\Delta x^2 + b_B[n]\Delta x + a_B[n], \quad (2.5)$$

$$\Delta x = x - n, \quad 0 \leq \Delta x \leq 1$$

U izrazu 2.5 su uporabljeni eksplicitni kubični koeficijenti d_B , c_B , b_B , a_B , dobiveni iz B-splajn koeficijenata pomoću matrične formulacije 2.6 ([3], [7]), a predstavljaju združeno djelovanje četiri susjedna B-splajn koeficijenta (transformacija je sustav trećeg reda). Formu je jednostavno izračunati unaprijed i jednaka je za sve kubične segmente splajn funkcije.

$$cub_B[n] = \begin{bmatrix} d_B[n] \\ c_B[n] \\ b_B[n] \\ a_B[n] \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} w[n-1] \\ w[n] \\ w[n+1] \\ w[n+2] \end{bmatrix} \quad (2.6)$$

Izraz 2.6 prebačen u z-domenu je prikazan izrazima 2.7 i 2.8 ([3]).

$$Cub_B(z) = \begin{bmatrix} D_B(z) \\ C_B(z) \\ B_B(z) \\ A_B(z) \end{bmatrix} = T_B(z)W(z) \quad (2.7)$$

$$T_B(z) = \begin{bmatrix} D_B(z) \\ C_B(z) \\ B_B(z) \\ A_B(z) \end{bmatrix} = \begin{bmatrix} -z^{-1} + 3 - 3z + z^2 \\ 3z^{-1} - 6 + 3z \\ -3z^{-1} + 3z \\ z^{-1} + 4 + z \end{bmatrix} \quad (2.8)$$

Prijenosna funkcija filtra $A_B(z)$ (izraz 2.8) je jednaka nazivniku prijenosne funkcije filtra za izračunavanje koeficijenta B-splajna $H_B(z)$ (izraz 2.2). Filtri se nalaze u kaskadi, pa se izrazi poništavaju. Izračunata vrijednost koeficijenta $a_B[n]$ je jednaka ulaznom uzorku $y[n]$, što je konzistentno sa uvjetom interpolacije (kubični polinom za $\Delta x = 0$ kao rezultat izračunava točan iznos koeficijenta $a_B[n]$, odnosno interpolacijski polinom na lijevom rubu područja definicije prolazi točno kroz interpolacijsku točku).

Matrica $T_B(z)$ (izraz 2.8) je prijenosna matrica sustava sa jednim ulazom i više izlaza (SIMO). Sustav se sastoji od četiri FIR filtra, od kojih je jedan trećeg reda. Indeks B u svim prethodnim izrazima označava B-splajn formulaciju (u izvornom obliku).

Kauzalna interpolacija

Prethodno opisana B-splajn interpolacija nije kauzalna, pa je izgradnja sustava za pretipkavanje u realnom vremenu nemoguća (nisu poznati svi uzorci ulaznog signala u svakom trenutku).

B-splajn interpolaciju je potrebno pretvoriti u kauzalnu, uz minimalne gubitke kvalitete interpolacije.

Sustav za pretvorbu B-splajn koeficijenata u koeficijente kubičnog polinoma opisan matricom $T_B(z)$ (izraz 2.8) nije kauzalan. Filtar $D_B(z)$ zahtijeva dva uzorka signala $w[n]$ iz budućnosti. Uvodi se kašnjenje z^{-2} kako bi sustav postao kauzalan. Kašnjenje se uvodi u sva četiri FIR filtra, kako bi izlazi ostali poravnati.

Filtar za generiranje B-splajn koeficijenata $H_B(z)$ (izraz 2.2) je IIR (rekurzivni) filter savršeno simetričnog impulsnog odziva. Zahvat na akauzalnu stranu je beskonačan (svi budući uzorci signala moraju biti poznati).

z^{-1} transformacijom prijenosne funkcije $H_B(z)$ dobiva se beskonačni impulсни odziv $h_B[n]$ (izraz 2.9) ([3]).

$$h_B[n] = g \sum_{k=-\infty}^{\infty} \alpha^{|k|} \delta[n - k] \quad (2.9)$$

$$\alpha = \sqrt{3} - 2, \quad g = \frac{1}{2\sqrt{3}}, \quad \delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (2.10)$$

Impulсни odziv za $n < 0$ mora biti skraćen za proizvoljan broj uzoraka (M). Utjecaj dalekih uzoraka na iznos interpolacije opada eksponencijalno (intuitivno poznato), pa je takvo skraćivanje moguće obaviti. Uvodi se dodatno kašnjenje od z^{-M} kako bi sustav postao kauzalan.

Skraćivanjem $H_B(z)$ nastaje prijenosna funkcija $\tilde{H}_B(z)$, odnosno uvođenjem kašnjenja prijenosna funkcija postaje $z^{-M} \tilde{H}_B(z)$. U generirane B-splajn koeficijente je unesena pogreška ovisna o M .

Izraz 2.11 ([3]) definira B-splajn koeficijente (u z -domeni i diskretnoj vremenskoj) sa pogreškom $ew_M[n]$ unesenom od skraćene prijenosne funkcije $z^{-M} \tilde{H}_B(z)$.

$$\tilde{W}(z) = z^{-M} \tilde{H}_B(z) Y(z) \xrightarrow{z^{-1}} \tilde{w}[n+M] = w[n] - ew_M[n] \quad (2.11)$$

Izraz 2.12 ([3]) definira pogrešku $ew_M[n]$ u z-domeni. $HEW_M(z)$ je prijenosna funkcija pogreške skraćivanja, odnosno razlika idealne i skraćene prijenosne funkcije $H_B(z)$ (izraz 2.13) ([3]).

$$EW_M(z) = W(z) - z^M \tilde{W}(z) = HEW_M(z) Y(z) \quad (2.12)$$

$$HEW_M(z) = H_B(z) - \tilde{H}_B(z) \quad (2.13)$$

Pogreška unesena u slijed B-splajn koeficijenata utječe i na koeficijente kubičnog polinoma, koji se generiraju korištenjem neizmijenjene matrice sustava $T_B(z)$ (izraz 2.14) ([3]).

$$\tilde{cub}_B(z) = \begin{bmatrix} \tilde{D}_B(z) \\ \tilde{C}_B(z) \\ \tilde{B}_B(z) \\ \tilde{A}_B(z) \end{bmatrix} = T_B(z) \tilde{W}(z) \xrightarrow{z^{-1}} \tilde{cub}_B[n] = \begin{bmatrix} \tilde{d}_B[n] \\ \tilde{c}_B[n] \\ \tilde{b}_B[n] \\ \tilde{a}_B[n] \end{bmatrix} \quad (2.14)$$

Pogreška $ew_M[n]$ je unesena u koeficijente kubičnog polinoma, a uvjet interpolacije je narušen. Kod interpolacije kontinuirana funkcija nužno prolazi kroz poznate uzorke. Kako je kauzalni koeficijent $\tilde{a}_B[n+M] \neq a_B[n]$ (opaska: kauzalni koeficijent $\tilde{a}_B[n]$ kasni M uzoraka), uvrštavanjem $\Delta x = 0$ u kubični polinom, dobiva se rezultat $\tilde{a}_B[n+M] \neq y[n]$.

Izraz 2.15 ([3]) definira pogrešku interpolacije na lijevom rubu svakog polinomijalnog segmenta splajna, koja je jednaka pogreški koeficijenata B-splajna, filtriranoj kroz izlazni $A(z)$ filter.

$$ei_M[n] = y[n] - \tilde{a}_B[n+M] \xrightarrow{z} EI_M(z) = Y(z) - z^M \tilde{A}_B(z) = EW_M(z) A(z) \quad (2.15)$$

Iako rezultirajući kauzalni splajn ne zadovoljava uvjet interpolacije, i dalje je dvostruko kontinuirano diferencijabilan (funkcija u klasi C^2), što pokazuje da izlazni sustav $T_B(z)$ osigurava očuvano svojstvo ([3]).

Implementacije direktnog kauzalnog B-splajn filtra

Prethodni odlomak opisuje pogrešku koju unosi skraćeni B-splajn filter $\tilde{H}_B(z)$ u niz B-splajn koeficijenata i kubičnih koeficijenata. Moguća je implementacija kauzalnog filtra na nekoliko načina: kaskadna, paralelna i FIR.

Kaskadna implementacija

Kaskadna implementacija rastavlja $H_B(z)$ u kaskadu filtara: $H_f(z)$ i $H_b(z)$ ([3]).

$$H_B(z) = \frac{1}{z + 4 + z^{-1}} = H_f(z) H_b(z) \quad (2.16)$$

$$H_f(z) = \frac{1}{1 - \alpha z^{-1}} \quad (2.17)$$

$$H_b(z) = \frac{z^{-1}}{1 - \alpha^{-1} z^{-1}} \quad (2.18)$$

$H_f(z)$ je kauzalni (*eng. forward*), a $H_b(z)$ akauzalni (*eng. backward*) filter. Problem je implementacija filtra $H_b(z)$ sa beskonačnim akauzalnim impulsnim odzivom (izraz 2.19) ([3]).

$$h_b[n] = -\sum_{k=1}^{\infty} \alpha^k \delta[n-1+k] \xrightarrow{z} H_b(z) = -z^{-1} \sum_{k=1}^{\infty} (\alpha z)^k \quad (2.19)$$

Skraćivanjem impulsnog odziva $h_b[n]$ nastaje $\tilde{h}_b[n]$ (izraz 2.20) i prijenosna funkcija pripadnog filtra $\tilde{H}_b(z)$ (izraz 2.21), koja se sastoji od izvornog $H_b(z)$ i dodatnog člana. ([3]).

$$\tilde{h}_b[n] = -\sum_{k=1}^{M+1} \alpha^k \delta[n-1+k] \xrightarrow{z} \tilde{H}_b(z) = -z^{-1} \sum_{k=1}^{M+1} (\alpha z)^k \quad (2.20)$$

$$\tilde{H}_b(z) = \frac{z^{-1}}{1 - \alpha^{-1} z^{-1}} (1 - (\alpha z)^{M+1}) = H_b(z) (1 - (\alpha z)^{M+1}) \quad (2.21)$$

Izraz 2.22 prikazuje ukupnu prijenosnu funkciju kaskadne kauzalne realizacije direktnog B-splajn filtra ([3]).

$$\tilde{H}_{cas}(z) = H_f(z) \tilde{H}_b(z) = H_B(z) - (\alpha z)^{M+1} H_B(z) \quad (2.22)$$

Posljednji izraz pokazuje razliku između idealnog i realnog kaskadnog B-splajn filtra. Kako je $|\alpha| < 1$, povećanjem broja uzoraka iz budućnosti (M) se smanjuje razlika idealnog i realnog kaskadnog B-splajn filtra. Za svaki dodatan uzorak iz budućnosti, točnost aproksimacije se povećava 3.7 puta ([3]).

Kako bi sustav bio kauzalan, mora biti uvedeno kašnjenje, pa B-splajn koeficijenti kasne za z^{-M} (izraz 2.23), a pogreška u z-domeni je prikazana izrazom 2.24 ([3]).

$$\tilde{W}_{cas}(z) = z^{-M} \tilde{H}_{cas}(z) Y(z) \xrightarrow{z^{-1}} \tilde{w}_{cas}[n+M] = w[n] - e w_{cas,M}[n] \quad (2.23)$$

$$E W_{cas,M}(z) = (\alpha z)^{M+1} H_B(z) Y(z) = (\alpha z)^{M+1} W(z) \quad (2.24)$$

Efikasna kauzalna realizacija $\tilde{H}_{bM}(z)$ akauzalnog FIR predfiltra $\tilde{H}_b(z)$ je prikazana izrazom 2.25. Tok signala implementacije je prikazan u narednim poglavljima (slika 3.6), no bez izlaznog množenja sa -1 , koje je prebačeno u IIR dio kaskade.

$$\tilde{H}_{bM}(z) = z^{-M}\tilde{H}_b(z) = -\sum_{k=0}^M \alpha^{M+1-k}z^{-k} = -(\alpha^{M+1} + \alpha^M z^{-1} + \dots + \alpha z^{-M}) \quad (2.25)$$

Paralelna implementacija

Paralelna implementacija rastavlja $H_B(z)$ u paralelu IIR filtara prvog reda: $H_c(z)$ i $H_a(z)$ ([3]).

$$H_B(z) = \frac{1}{z+4+z^{-1}} = H_c(z) + H_a(z) \quad (2.26)$$

$$H_c(z) = \frac{g}{1-\alpha z^{-1}} \quad (2.27)$$

$$H_a(z) = \frac{-g}{1-\alpha^{-1}z^{-1}} \quad (2.28)$$

Podjela na paralelne implementacije je također vođena principom podjele na akauzalni i kauzalni filter. U paralelnoj kombinaciji filter $H_a(z)$ je akauzalan i njegov odziv mora biti skraćen i zakašnjen. Detaljan opis implementacije je izostavljen jer je za sustav za pretipkavanje odabrana kaskadna realizacija, zbog boljih performansi na nižim frekvencijama. Kaskadna implementacija najbolje aproksimira idealni B-splajn filter ispod frekvencije $\omega_1 = \frac{\pi^2}{5}$, do najviše 2.7 dB nižu pogrešku od paralelne implementacije, prema omjeru pogrešaka definiranom u [3]. Na frekvenciji ω_1 aproksimacije su jednake, dok iznad frekvencije ω_1 bolju aproksimaciju idealnog B-splajna daje paralelna implementacija (na $\omega = \pi$ ima najviše 2.1 dB nižu pogrešku od kaskadne realizacije, prema omjeru pogrešaka definiranom u [3]).

Potpun opis paralelne implementacije se može pronaći u [3].

FIR implementacija

FIR implementacija ([3]) aproksimira idealni B-splajn IIR filter FIR filtrom (izraz 2.29). Impulsni odziv IIR filtra se skraćuje na obje strane, akauzalnoj i kauzalnoj.

$$\tilde{H}_{fir}(z) = \sum_{n=-M}^M h[n]z^{-n} \quad (2.29)$$

Prema [3] FIR implementacija se pokazala inferiornija kaskadnoj realizaciji u gotovo svim slučajevima, osim u $M + 1$ točno određenih frekvencija kod kojih $\tilde{H}_{fiiir}(z)$ radi idealno ($\tilde{H}_{fiiir}(z) = H_B(z)$). U većini slučajeva spektar ulaznog signala nije poznat i nije ga moguće poravnati sa tim frekvencijama, pa FIR implementacija nije odabrana za sustav za pretipkavanje. Također, (računska) kompleksnost FIR implementacije je veća od prethodno opisanih ([3]).

Kratka kauzalna kubična splajn interpolacija

Prethodno opisana kauzalna kubična splajn interpolacija sa kaskadnom realizacijom može biti unaprijeđena izmjenama predloženima u [2].

Izvorni SIMO sustav $T_B(z)$ (izraz 2.8) na kojem su bazirane prethodne interpolacije sadrži FIR filtre najviše trećeg reda, te zahtijeva dva uzorka iz budućnosti (nekauzalni sustav). Ukupno kašnjenje sustava za generiranje kubičnih koeficijenata je $M + 2$: M je kašnjenje zbog akauzalnog zahvata B-splajn filtra, a 2 je kašnjenje zbog nekauzalnog sustava $T_B(z)$.

Predložen je sustav sa više ulaza i više izlaza (MIMO), sa prijenosnom matricom $T_C(z)$ ([2]).

$$T_C(z) = \begin{bmatrix} D_C(z) \\ C_C(z) \\ B_C(z) \\ A_C(z) \end{bmatrix} = \begin{bmatrix} 6(2 + z^{-1})(-z - 1) & (z + 5) \\ 6(2 + z^{-1})(z + 2) & -9 \\ 6(2 + z^{-1})(-1) & 3 \\ 0 & 1 \end{bmatrix} \quad (2.30)$$

Uporabu kratkog izlaznog MIMO filtra $T_C(z)$ za generiranje kubičnih koeficijenata iz B-splajn koeficijenata generiranih kaskadnim kauzalnim B-splajn filtrom prikazuje izraz 2.31. Nije nužna uporaba kaskadne implementacije B-splajn filtra, mogu se koristiti i ostale implementacije.

$$\tilde{Cub}_C(z) = \begin{bmatrix} \tilde{D}_C(z) \\ \tilde{C}_C(z) \\ \tilde{B}_C(z) \\ \tilde{A}_C(z) \end{bmatrix} = T_C(z) \begin{bmatrix} \tilde{W}(z) \\ z^{-M}Y(z) \end{bmatrix} = T_C(z) \begin{bmatrix} \tilde{H}_{cas}(z) \\ 1 \end{bmatrix} z^{-M}Y(z) \quad (2.31)$$

Moguće je izlučiti faktor $6(2 + z^{-1})$ iz izraza 2.30, čime se dobiva niz vrlo jednostavnih izlaznih filtara prvog reda. Sustav $T_C(z)$ zahtijeva samo jedan uzorak iz budućnosti (za razliku od izvornog sustava $T_B(z)$, izraz 2.8, koji zahtijeva 2 uzorka).

U [3] je opisan alternativni MIMO sustav, no također sa akauzalnim zahvatom od 2 uzorka. Zbog navedenog se sustav $T_C(z)$ naziva *kratkim* izlaznim filtrom.

MIMO sustav $T_C(z)$ koristi B-splajn koeficijente $\tilde{W}(z)$ dobivene kaskadnom kauzalnom implementacijom. Na drugom ulazu koristi $z^{-M}Y(z)$, izvorne uzorke signala, samo zakašnjele zbog poravnanja (kašnjenje mora odgovarati kašnjenju akauzalnog B-splajn filtra). Za razliku od B-splajn koeficijenata $\tilde{W}(z)$, niz uzoraka $Y(z)$ nije pogođen pogreškom zbog skraćanja akauzalnog odziva idealnog B-splajn filtra $H_B(z)$. Zbog toga se drugi ulaz koristi za korekciju kubičnih koeficijenata, kako bi se umanjila pogreška unesena skraćanjem odziva.

Performanse

MIMO sustav za idealni B-splajn filter $H_B(z)$ daje identičan odziv kao izvorni SIMO sustav (izraz 2.32) ([2]).

$$T_B(z)H_B(z) = T_C(z) \begin{bmatrix} H_B(z) \\ 1 \end{bmatrix} \quad (2.32)$$

Međutim za konačni M i neidealni $\tilde{H}_B(z)$ postoje razlike u preciznosti aproksimacije idealnog B-splajna. Procjena performansi je u [2] izvedena usporedbom kardinalnih splajnova. Uspoređeni su kardinalni splajnovi kauzalnih interpolatora (kaskadna i paralelna realizacija sa izlaznim filterima $T_B(z)$ i $T_C(z)$) sa idealnim kubičnim splajnom (dobivenim idealnim B-splajn filtrom $H_B(z)$), na potpunom pojasu frekvencija i niskim frekvencijama.

Za jednaki iznos uzoraka iz budućnosti M, MIMO filter $T_C(z)$ daje bolje rezultate od SIMO filtra $T_B(z)$ do 6.3dB sa kaskadnom realizacijom, i 7.5dB sa paralelnom realizacijom $\tilde{H}_B(z)$, za široki pojas frekvencija ([2]).

Zbog kraćeg filtra $T_C(z)$, moguće je povećati njegov kauzalni B-splajn predfilter za dodatan uzorak, kako bi kašnjenja uspoređivanih sustava bila jednaka. Zbog produženja predfiltera, dobiva se dodatno poboljšanje performansi od 11.44dB, koje je moguće ostvariti za svaki dodatni uzorak iz budućnosti ([2]). Time je kaskadna interpolacija sa filtrom $T_C(z)$ u širokom rasponu frekvencija za jednaku dužinu kašnjenja bolja od interpolacije sa filtrom $T_B(z)$ za 18dB.

Nedostatak kraćeg $T_C(z)$ filtra je prekinutost druge derivacije generiranog kubičnog splajna (splajn je u klasi C^1).

Splajn generiran izlaznim filtrom $T_B(z)$ je u klasi C^2 , no ne zadovoljava uvjet interpolacije ($\tilde{a}_B[n + M] \neq y[n]$, ali $\tilde{a}_C[n + M] = y[n]$) ([2]).

Dodatna mjera performansi opisana u [2] (ukupna kvadratna pogreška u propusnom i nepropusnom pojasu interpolatora, u usporedbi sa idealnim pojasno ograničenim interpolatorom, varirana po frekvenciji) analizira ponašanje interpolatora na specifičnim frekvencijama. Pokazano je da na niskim frekvencijama sustav sa izlaznim filtrom $T_C(z)$ postiže bolje rezultate od sustava sa $T_B(z)$ filtrom za čak 25dB ([2]). Na visokim frekvencijama splajn generiran sa $T_C(z)$ se približava idealnom kubičnom splajnu.

C^1 interpolator MOMS ukupnog kašnjenja 2 uzorka na niskim frekvencijama puno bolje aproksimira idealni kubični splajn od svih navedenih interpolatora, no na cijelom frekvencijskom pojasu odstupa od idealnog kubičnog splajna za 3.1 dB ([2]), što je lošiji rezultat od ostalih prethodno navedenih interpolacija.

Opisana kratka kauzalna kubična interpolacija je korištena za implementaciju sustava za pretipkavanje, opisanog u nastavku. Blok shema generatora polinoma baziranog na ovoj interpolaciji je prikazana slikom 3.5.

3. Sustav za pretipkavanje signala

Na temelju opisane kratke kauzalne kubične splajn interpolacije izgrađen je sustav za pretipkavanje signala. Sustav je izgrađen za ciljnu platformu Xilinx FPGA: Spartan-3 ili Virtex-4. Korišten je jezik za opis sklopovlja VHDL.

Sustav za pretipkavanje signala omogućava pretipkavanje signala otipkanog sa određenom frekvencijom otipkavanja, na proizvoljnu različitu frekvenciju otipkavanja. Potrebni uzorci za pretipkani signal koji ne postoje u izvornom signalu se dobivaju opisanom tehnikom interpolacije. Proizvoljna frekvencija pretipkanog signala ne može biti u bilo kojem odnosu prema izvornoj frekvenciji otipkanog signala, ali može biti u širokom dozvoljenom rasponu (opisano kasnije).

Sustav omogućuje spajanje sa potpuno asinkronim proizvođačem i potrošačem signala (podataka). Signal vremenskog vođenja otipkavanja ulaznog signala generira proizvođač ulaznog, a signal vremenskog vođenja otipkavanja izlaznog signala generira potrošač izlaznog signala. Opisanim dizajnom je omogućen protočni rad sustava, u realnom vremenu (*eng. „on-line“* primjena sustava).

Potrošač i proizvođač signala generiraju svoje signale vremenskog vođenja iz vlastitih, fizički različitih oscilatora. Zbog toga se na sustavu za pretipkavanje susreću dva međusobno asinkrona signala vremenskog vođenja (u ulogama signala otipkavanja izvornog i pretipkanog signala). Uporaba međusobno asinkronih signala u digitalnom sklopovlju može dovesti do problema metastabilnosti i pogrešnog prijenosa podataka. Sustav za pretipkavanje koristi poseban mehanizam kako bi otklonio ovaj problem (opisano kasnije).

Iako je moguće analitički precizno opisati odnos izvorne i pretipkane frekvencije otipkavanja, u stvarnosti frekvencije otipkavanja oba oscilatora odstupaju od nazivnih, ili čak variraju kroz vrijeme. Zbog toga nije dovoljno podesiti sustav za pretipkavanje na nepromjenjivu postavku omjera pretipkavanja. U sustav mora biti uključen regulacijski sustav koji kompenzira odstupanja svih frekvencija od nazivnih.

U nastavku slijedi opis implementacije cijelog sustava za pretipkavanje. Dizajn je podijeljen u modularne cjeline.

Glavne cjeline sustava za pretipkavanje su: generator kubičnog polinoma, međuspremnik za pretipkavanje, evaluator kubičnog polinoma.

Evaluator kubičnog polinoma

Evaluator kubičnog polinoma je komponenta koja nije specifična za primjenu u području interpolacije, te se u identičnom obliku može iskoristiti za druge primjene.

Kod interpolacije se evaluacija polinoma koristi u završnoj fazi. Nad koeficijentima polinoma dobivenim interpolacijskom metodom se evaluira točka koju se želi interpolirati. Evaluator kubičnog polinoma je posebno pogodan za splajn interpolaciju, koja tipično koristi polinom niskog stupnja, kao što je kubična splajn interpolacija.

Evaluacija kubičnog polinoma u svom izvornom obliku (izraz 3.1) na sklopovlju ima svoje nedostatke. Sklopovski resursi potrebni za uzastopna potenciranja argumenta dx su veliki (intenzivna uporaba množila).

$$y = Ddx^3 + Cdx^2 + Bdx + A \quad (3.1)$$

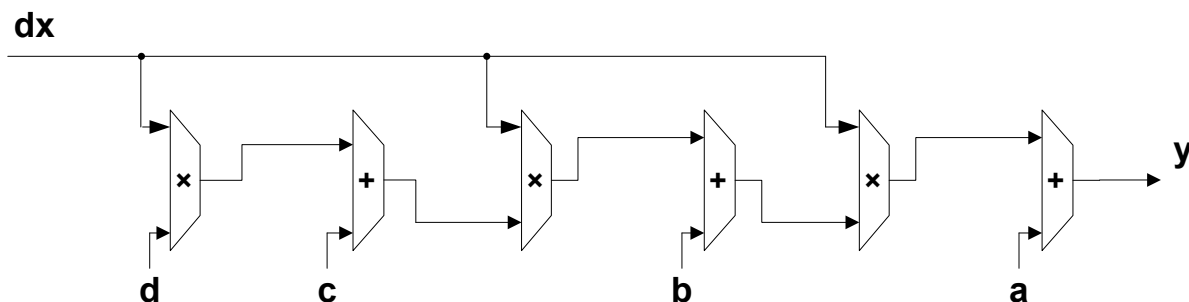
Kubični polinom se zapisuje u Hornerovom obliku (izraz 3.2), kako bi se uštedili resursi za računske operacije.

$$y = ((Ddx + C)dx + B)dx + A \quad (3.2)$$

Evaluator je prilagođen uporabi sa frakcionalnom aritmetikom. Sve širine koeficijenata polinoma, rezultata i međurezultata, kao i argumenta, su podesive.

Asinkroni evaluator

Slika 3.1 prikazuje shemu evaluatora kubičnog polinoma izvedenog u obliku Hornerove sheme.



Slika 3.1: Asinkrona shema evaluatora kubičnog polinoma u Hornerovom obliku

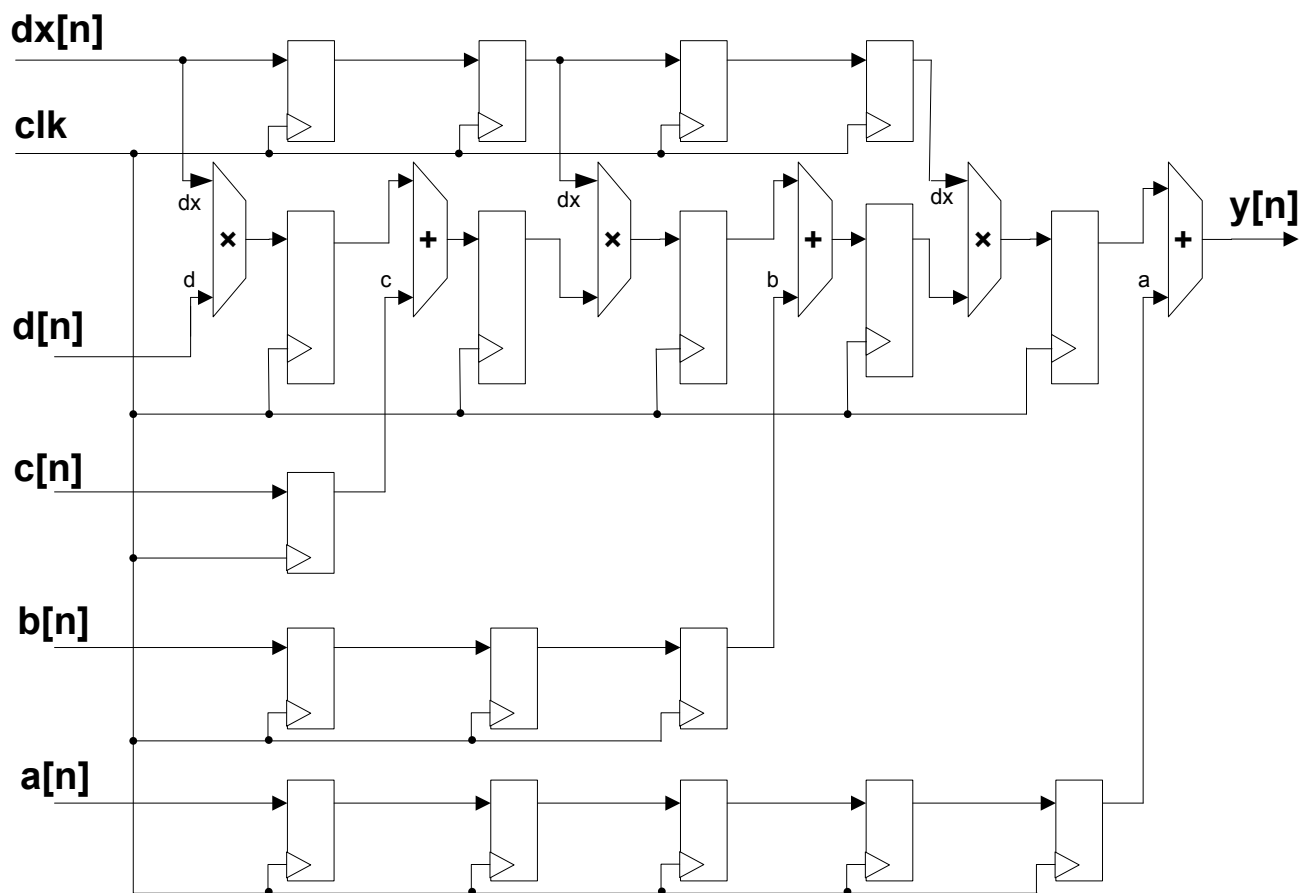
Prikazana izvedba evaluatora je asinkrona, bez pogona signalom vremenskog vođenja. Generalizirano, izvedba je Booleova kombinacijska funkcija sa većim brojem ulaza i izlaza. Nedostatak izvedbe je vrlo velika kompleksnost Booleovih funkcija, sa jako velikim brojem razina logike. Kašnjenje od ulaza do izlaza je vrlo visoko. Posljedica kašnjenja je nemogućnost rada sklopa na velikim frekvencijama signala vremenskog vođenja (rezultat se mora pojaviti na izlazu kombinacijske funkcije prije pojave sljedećeg aktivnog brida signala vremenskog vođenja, koji pokreće novu evaluaciju funkcije i pohranjuje izračunati rezultat).

Sinkroni evaluator

Zbog otklanjanja navedenog problema sa performansama, uvodi se koncept cjevovoda. Sam pojam RTL dolazi od *eng. Register Transfer Level*. Arhitektura digitalnog sklopovlja se svodi na uzastopno izmjenjivanje (sinkronih) registara i asinkrone logike. Asinkrona logika obavlja kombinacijsku logičku funkciju, a registri kao memorijski elementi pohranjuju rezultate, koji su također i argumenti sljedeće logičke funkcije u lancu. Najdulja logička funkcija (sa najviše logičkih razina) u lancu je usko grlo sustava, te ograničava maksimalnu moguću frekvenciju rada sklopa. Jedan period signala vremenskog vođenja odgovara intervalu od pojave argumenata logičke funkcije na izlazu ulaznog registra, do pojave rezultata logičke funkcije na izlazu kombinacijskog bloka. U to razmatranje je potrebno uračunati i *eng. setup time* (vrijeme postavljanja), minimalno vrijeme koje je potrebno proteći od pojave podatka na ulazu u registar (nakon izračunavanja prethodne kombinacijske funkcije) do pojave aktivnog brida signala vremenskog vođenja. Također je bitan (ali u praksi manje problematičan) *eng. hold time* (vrijeme zadržavanja), minimalno vrijeme koje je potrebno proteći od pojave aktivnog brida signala vremenskog vođenja do mogućnosti sigurnog uklanjanja učitano podatka sa ulaza registra. Narušavanje vremena postavljanja ili zadržavanja može dovesti do metastabilnosti sklopa, o čemu će biti riječi kasnije.

Cjevovod je naziv za niz registara i asinkronih kombinacijskih funkcija koje ih povezuju u lanac. Ako performanse sklopa ne zadovoljavaju, potrebno je skratiti najdužu kombinacijsku funkciju između dva susjedna registra.

Slika 3.2 prikazuje implementaciju evaluatora kubičnog polinoma pomoću cjevovoda.



Slika 3.2: RTL shema evaluatora kubičnog polinoma u Hornerovom obliku implementirana cjevovodom

Dodavanjem registara prethodno složena kombinacijska logika se rastavlja na manje cjeline, koje se mogu evaluirati na višem taktu. Kašnjenje rezultata sada iznosi 6 perioda signala vremenskog vođenja, ali nakon inicijalnog punjenja cjevovoda moguće je dobiti novi izračunati rezultat u svakom ciklusu takta.

Implementacija pomoću cjevovoda unosi nove komplikacije. Potrebno je obratiti veliku pozornost na vremenske odnose svih signala, kako bi bili evaluirani pravovremeno (problem poravnanja cjevovoda).

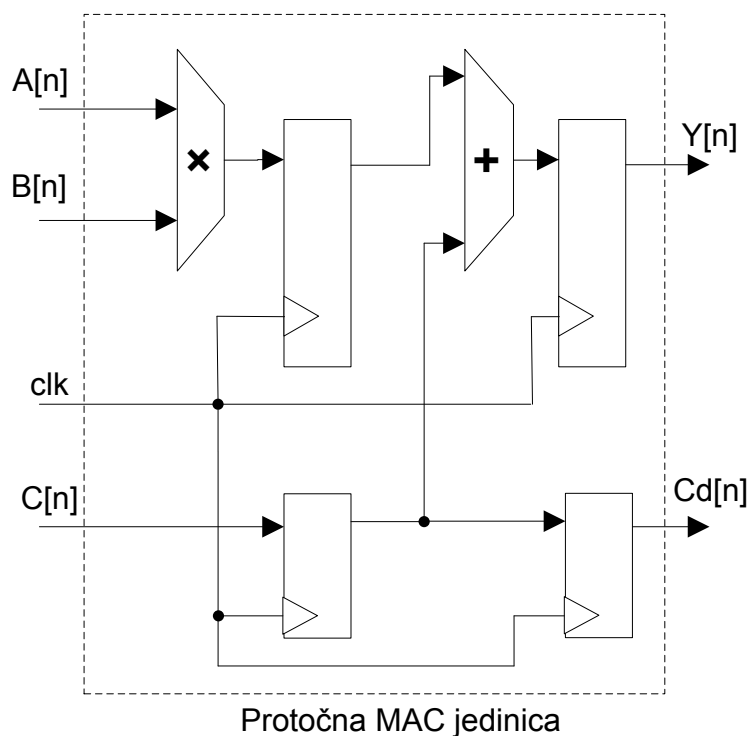
Za većinu koeficijenata polinoma, kao i za argument dx , potrebno je uvoditi dodatne linije za kašnjenje.

Protočna MAC jedinica

U prikazanom sinkronom dizajnu evaluatora se mogu uočiti neke pravilnosti. Blok sa množenjem dva operanda i pribrajanjem trećeg operanda se ponavlja tri puta. Druga pravilnost u ponavljanju je sklopovlje za kašnjenje podataka (za poravnanje cjevovoda). Zbog toga se ovi dijelovi izdvajaju kao zasebni moduli, kako bi se unaprijedila razina apstrakcije, omogućilo ispitivanje, i ponovna uporaba gotovih modula.

Jedan izdvojeni modul je jedinica za kašnjenje, čije je djelovanje trivijalno: zakasniti ulazni podatak za zadani broj perioda signala vremenskog vođenja.

Drugi modul je jedinica za množenje i pribrajanje, koja radi kao protočna MAC jedinica, prikazana na slici 3.3.



Slika 3.3: Protočna MAC jedinica

Množila su izvedena pomoću ugrađenih množila Xilinx FPGA arhitekture. Operandi ugrađenih množila su 18-bitni, a rezultat je 36-bitni. Svi ulazi i izlazi množila su u formatu dvojnog komplementa.

Kod množila se javlja problem sa ograničenom širinom operanda. Ulaz množila je 18-bitni, a time podržava frakcionalnu aritmetiku Q1.17. Izlaz množila je širine 36 bita (frakcionalna aritmetika Q2.34). Prije uporabe u sljedećim fazama, potrebno je skratiti broj bita množila, uz zaokruženje.

Rezultat je potrebno skratiti minimalno na širinu množila sljedeće faze. Rezultat, frakciju u obliku Q2.34 je prvo potrebno posmaknuti jedno mjesto ulijevo, pa rezultat postaje Q1.34. Ovo je potrebno obaviti jer se množenjem umnaža bit predznaka na MSB. Zatim se uzima najviših 18 bitova, i pribraja im se pola LSB. Ovo pribrajanje pola LSB se tipično izvodi postavljanjem najnižeg ulaza FPGA lanca za prijenos preljeva (*eng. carry chain*) u logičku jedinicu.

Rezultat je zaokruživanje rezultata, bez uobičajenog zanemarivanja decimalnog dijela. Dobiven je zaokruženi rezultat, frakcija Q1.17. Bitno je napomenuti da se zbog generičke konfigurabilne širine rezultata i međurezultati zaokružuju na specificiranu širinu izlaza (ne nužno na maksimalnih 18 bita).

Svi ulazi protočne MAC jedinice su poravnati: operandi A, B, C se na ulaz stavljaju istovremeno. Nakon drugog aktivnog brida takta rezultat po izrazu 3.3 se pojavljuje na izlazu.

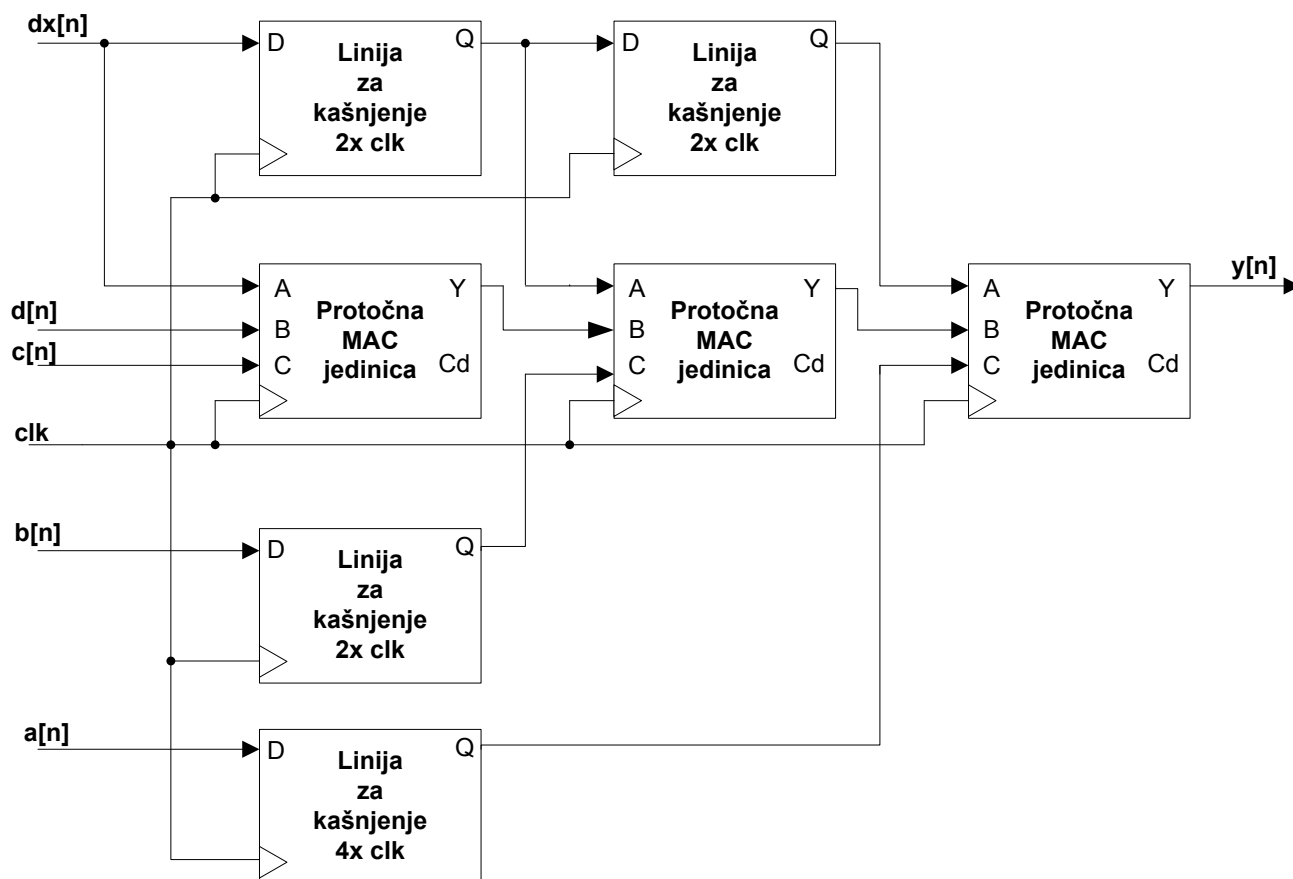
$$Y = A * B + C \tag{3.3}$$

Dodatan pomoćni izlaz je Cd. Na tom izlazu se pojavljuje neizmijenjen operand C, zakašnjen za dva ciklusa takta (poravnat sa izlazom Y). Namjena ovog izlaza je olakšavanje implementacije određenih FIR filtarskih struktura (opisano kasnije). Izlaz se ne primjenjuje za evaluaciju kubičnih polinoma.

Evaluator sa protočnom MAC jedinicom

Evaluator je izgrađen modularno. Upotrebljavaju se protočna MAC jedinica, i jedinica za kašnjenje. Sustav je prikazan na slici 3.4.

Prikazana izvedba evaluatora je konačna izvedba uporabljena u sustavu za pretipkavanje. Sve širine (rezultat i međurezultati, operand, koeficijenti) su generički podesive. Svi argumenti se učitavaju sinkrono. Vrijeme evaluacije je 6 perioda takta.



Slika 3.4: Evaluator kubičnog polinoma sa protočnom MAC jedinicom

Generator kubičnog polinoma

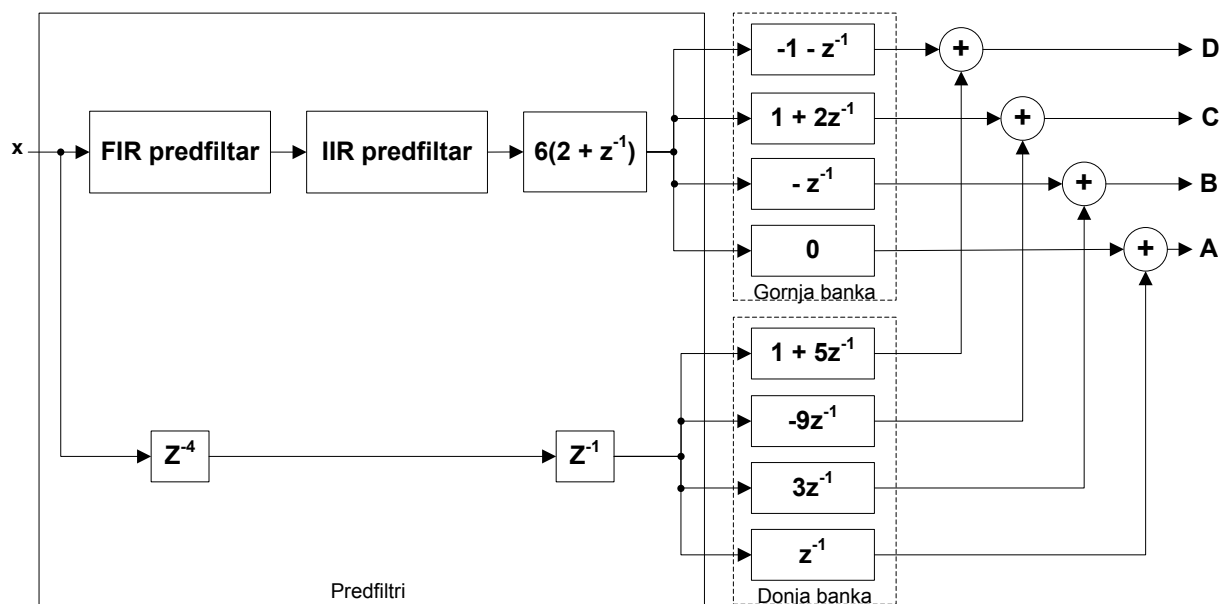
Generator kubičnog polinoma je komponenta sustava koja implementira konkretnu interpolacijsku tehniku. U ovom slučaju radi se o opisanoj kratkoj kauzalnoj kubičnoj splajn interpolaciji. Reimplementacijom ovog modula je moguće zadržati ostatak sustava za pretipkavanje, uz korištenje druge tehnike interpolacije. Jedini uvjet je da nova metoda interpolacije koristi polinom maksimalno trećeg reda, jer je ostatak sustava dizajniran za uporabu četiri koeficijenta polinoma.

Ulaz u generator kubičnog polinoma je ulazni signal sustava, koji je potrebno pretipkati. Izlaz generatora kubičnog polinoma su koeficijenti polinoma A, B, C i D, čijom evaluacijom po izrazu 3.1 se vrši interpolacija u proizvoljno odabranoj nenegativnoj točki dx .

Pozitivna vrijednost točke dx određuje trenutak interpolacije između dvije postojeće uzastopne točke, odnosno između dva postojeća uzorka izvornog signala. Koeficijenti A, B, C i D pripadaju vremenski prethodećoj točki iz para točaka između kojih je definiran interpolacijski polinom. U osnovnoj izvedbi

interpolacijski polinom je definiran samo za pozitivne vrijednosti, i aproksimira funkciju samo između dvije susjedne točke.

Blok-shema generatora kubičnog polinoma je prikazana na slici 3.5. Blok-shema opisuje tok signala.



Slika 3.5: Blok-shema sustava za generiranje polinoma, [1], *Zadatak i opis sustava*

Ovaj tip generatora je implementiran kao tri nezavisne podcjeline. Prefiltri su skupina filtara koji pripremaju signal za donju i gornju filtarsku banku. Gornja filtarska banka sadrži filtre generatore koeficijenata, a donja filtarska banka sadrži filtre za popravljivanje iznosa koeficijenata. IIR i FIR prefiltri su komponente koje će biti detaljnije pojašnjene u nastavku.

Generator polinoma sve koeficijente na izlaz postavlja sinkrono. Ulazni signal je podesive širine, no preporuča se uporaba 12-bitnog signala zbog ograničenja dinamičkog raspona međurezultata.

Kašnjenje generatora polinoma je varijabilno i ovisi o broju FIR stupnjeva FIR prefiltra (pojašnjeno u nastavku). Korišteni broj FIR stupnjeva u trenutnoj implementaciji je 4. Kašnjenje d generatora (u broju perioda takta) je definirano izrazom 3.4.

$$d = \text{FIR_stupnjeva} * 2 + 6 \quad (3.4)$$

Kašnjenje se na najvišoj razini sastoji od kašnjenja prefiltra (izraz 3.6), i kašnjenja gornje i donje filtarske banke (obje banke kasne 1 ciklus takta) te izlaznog sloja zbrajala.

Osim kašnjenja koje filtri unose u sustav, definira se i kašnjenje utitravanja. Utitravanje u generatora (u broju perioda takta) je definirano izrazom 3.5.

$$u = FIR_stupnjeva * 3 + 9 \quad (3.5)$$

Utitravanje je broj ciklusa takta potreban da se napune svi memorijski elementi u sustavu. To je inicijalno kašnjenje sustava, broj perioda takta potreban da se prvi rezultat (nakon inicijaliziranja sustava) pojavi na izlazu. Bitna opaska za simulaciju je da vrijeme utitravanja može biti i veće, zbog korištenja IIR filtra, ako memorijski element IIR filtra nije u referentnom modelu inicijaliziran na jednaku početnu vrijednost kao na stvarnom sklopovlju.

Predfiltri

Predfiltarska banka je skupina filtara koja priprema ulazni signal za gornju i donju filtarsku banku. Donja filtarska banka ne zahtijeva stvarno predprocesiranje ulaznog signala, pa se kao predprocesiranje signala za donju granu koristi samo linija za kašnjenje. Gornja filtarska banka se sastoji od FIR predfiltra, IIR predfiltra, te gornjeg predfiltra. Kašnjenje predfiltarske banke je definirano izrazom 3.6, a utitravanje izrazom 3.7. Broj FIR stupnjeva reprezentira broj stupnjeva FIR predfiltra, te je promjenjiv. U trenutnoj implementaciji iznosi 4.

$$d = FIR_stupnjeva * 2 + 4 \quad (3.6)$$

$$u = FIR_stupnjeva * 3 + 6 \quad (3.7)$$

Gornja grana banke predfiltara se sastoji od FIR predfiltra, IIR predfiltra, te gornjeg predfiltra (koji je FIR tipa). Donja grana banke predfiltara samo prenosi ulazni signal do izlaza za donju banku, uz kašnjenje koje se izračunava po izrazu 3.6.

Kašnjenje predfiltarske banke ovisi posredno o kašnjenju FIR predfiltra, IIR predfiltra i gornjeg predfiltra.

FIR predfilter

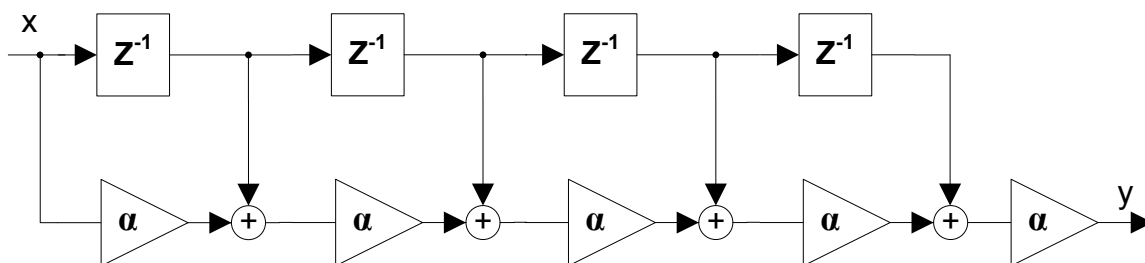
FIR predfilter je FIR komponenta prva u lancu predprocesiranja ulaznog signala. Filter može imati varijabilan broj stupnjeva. FIR koeficijenti korišteni u filteru su vrlo specifičnog oblika: potencije istog koeficijenta. Kod uporabe većeg broja FIR stupnjeva, koriste se koeficijenti koji su veće potencije osnovnog koeficijenta.

Prijenosna funkcija FIR predfiltra ovisi o broju faza, kao i shema toka signala. U trenutnoj izvedbi koriste se 4 stupnja FIR predfiltra.

Prijenosna funkcija filtra sa 4 stupnja je opisana izrazom 3.8. Koeficijent α je osnovni koeficijent koji se potencira u stupnjevima.

$$H(z) = \alpha^5 + \alpha^4 z^{-1} + \alpha^3 z^{-2} + \alpha^2 z^{-3} + \alpha z^{-4} \quad (3.8)$$

Izraz sa potenciranim koeficijentom donosi sklopovski problem, koji se može vrlo lako riješiti. Problem je u utrošku sklopovskih resursa za potenciranje koeficijenata, ili čak za pohranu potenciranih koeficijenata. Rješenje je uporaba FIR filtarske strukture vrlo slične direktnoj formi 1, ali sa alternativnim položajem množila. FIR filtarska struktura je prikazana na slici 3.6. Koriste se 4 FIR stupnja.



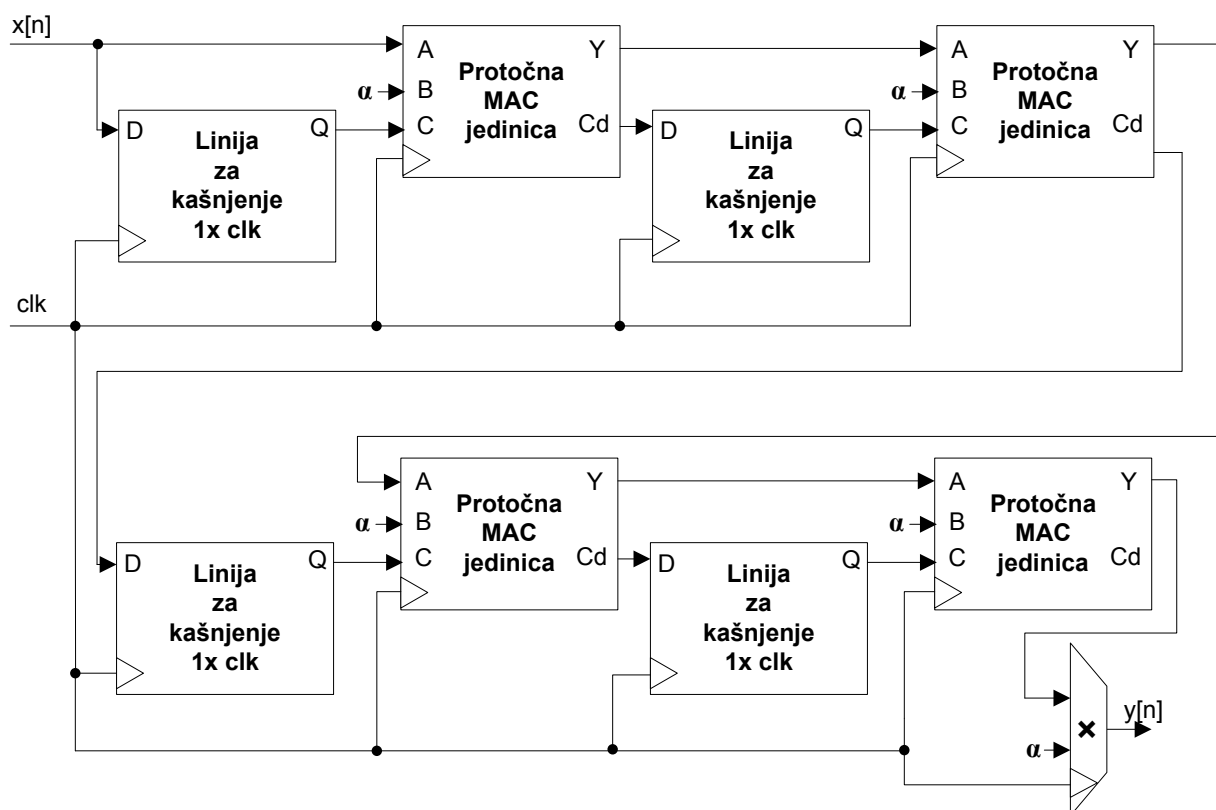
Slika 3.6: Filtarska struktura FIR predfiltra, [1], *Zadatak i opis sustava*

Sa slike je vidljivo da je dovoljno pohraniti filtarski koeficijent α u samo jednoj instanci. Nije potrebno njegovo izravno potenciranje, niti uvođenje numeričke pogreške tim postupkom. Ova forma nije primjenjiva na generalni slučaj FIR filtra. Filtarska struktura omogućuje vrlo jednostavno kaskadiranje FIR stupnjeva, u slučaju evaluacije duže prijenosne funkcije.

Posebnu pažnju kod realizacije filtarskih struktura općenito, pa tako i ove, potrebno je posvetiti RTL modelu. Korištena množila i zbrajala nemaju beskonačno kratko vrijeme evaluacije. Ako bi se shema sa slike 3.6 direktno realizirala u sklopovlju, rezultirajuće sklopovlje bi sadržavalo veliku kombinacijsku funkciju sa zbrajalima i množilima, sa mnogo razina logike, čije vrijeme evaluacije bi bilo nezadovoljavajuće za visoke frekvencije rada. Potrebno je uvesti sinkrona zbrajala i množila, odnosno efektivno iza svakog zbrajala i množila postaviti registar. Rezultirajući RTL model bi imao zadovoljavajuće performanse rada jer je pretvoren u cjevovod, no više ne bi realizirao prijenosnu funkciju sa izraza 3.8 jer

su uvedena nova kašnjenja. Kako bi se RTL model ispravio da realizira ispravnu prijenosnu funkciju, potrebno je u gornju granu dodati dodatna kašnjenja. Efektivno, svaki element za kašnjenje gornje grane iznosa z^{-1} mora postati z^{-3} jer su i u donju granu dodana dva dodatna elementa za kašnjenje.

Već ranije je uočena pravilnost u FIR strukturi koja bi se mogla dodatno enkapsulirati. Funkcionalnost je poznata – umnožak, uz pribrajanje trećeg argumenta. Riječ je o opisanoj protočnoj MAC jedinici, upotrijebljenoj kod evaluacije polinoma. Slika 3.7 prikazuje FIR predfilter izveden pomoću protočne MAC jedinice, uz realizaciju četiri FIR stupnja.



Slika 3.7: RTL shema FIR predfiltera realizirana protočnim MAC jedinicama

Do sada neiskorišteni izlaz Cd MAC jedinice nalazi primjenu u kašnjenju ulaznog signala. Interno kašnjenje MAC jedinice od ulaza C do izlaza Cd je dva perioda takta, kao i izlaz Y. Potrebno kašnjenje signala između dva FIR stupnja je tri perioda takta, stoga se dodaje dodatani vanjski element za kašnjenje. Prikazana RTL shema ispravno realizira prijenosnu funkciju iz izraza 3.8, uz zadovoljavajuće performanse zbog korištenja protočne strukture.

Kašnjenje filtarske strukture je definirano izrazom 3.9, a inicijalno utitravanje izrazom 3.10.

$$d = \text{FIR_stupnjeva} * 2 + 1 \quad (3.9)$$

$$u = \text{FIR_stupnjeva} * 3 + 1 \quad (3.10)$$

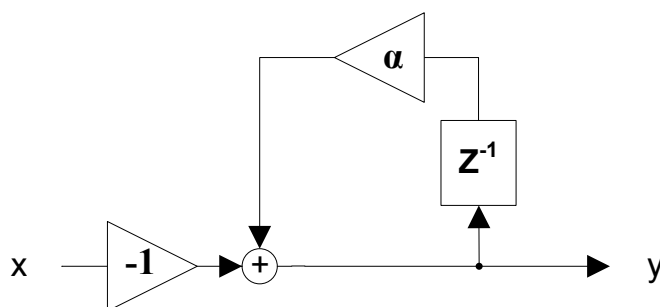
Broj FIR stupnjeva u trenutnoj implementaciji je 4.

IIR predfilter

IIR predfilter je drugi predfilter u slijedu za predprocesiranje ulaznog signala u gornjoj grani predfiltarske banke. Ovaj filter je jedini filter IIR (rekurzivnog) tipa u cijelom generatoru polinoma, ali i sustavu za pretipkavanje općenito. Prijenosna funkcija IIR filtra je prikazana izrazom 3.11.

$$H(z) = \frac{-1}{1 + \alpha z^{-1}} \quad (3.11)$$

Filtarska struktura je prikazana na slici 3.8. Inicijalno množenje sa koeficijentom -1 logički pripada u FIR predfilter (izraz 2.25). Međutim, zbog efikasnosti i uštede resursa, umnožak sa -1 je smješten u IIR filter, gdje je izveden kao oduzimanje na samom zbrajalu unutar IIR strukture (inverzija ulaza i pribroj bita za prijenos na zbrajalu).

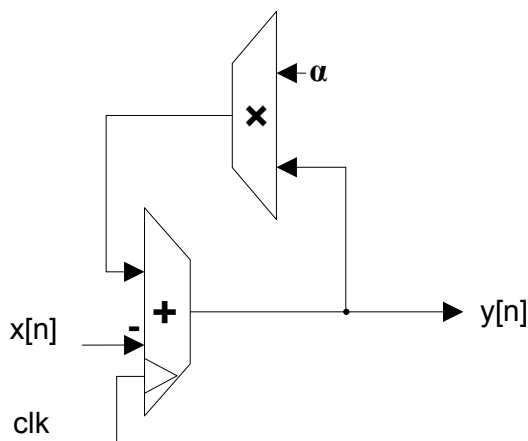


Slika 3.8: Filtarska struktura IIR predfiltra

Po svojoj prirodi, rekurzivni filteri su problematični za implementaciju u sklopovlju. Većinu nerekurzivnih operacija nad signalima (kod FIR filtera) je moguće usporavati ugradnjom memorijskih elemenata do proizvoljne razine (prikazano kod FIR predfiltra), kako bi sustav radio u protočnom načinu rada (na većem taktu). Rekurzivne operacije kod IIR filtera nije moguće usporavati ugradnjom memorijskih elemenata, jer su rezultati obradbe potrebni za obradbu sljedećeg

uzorka. Zbog toga je moguća pojava kombinacijske logike sa mnogo razina, a bez mogućnosti povećanja frekvencije signala vremenskog vođenja.

Opisani problem se javlja i kod ovog IIR predfiltra. RTL shema filtra je prikazana na slici 3.9.



Slika 3.9: RTL shema IIR predfiltra

Množilo u rekurzivnoj grani radi u asinkronom načinu rada. Rezultat je kombinacijska funkcija sa većim brojem logičkih razina, koju čine spomenuto asinkrono množilo rekurzivne grane i kombinacijska logika zbrajala. Zbog toga je ovaj IIR filter najsporija komponenta sustava za pretipkavanje signala. Uz 18-bitni ulaz i izlaz, maksimalni radni takt filtra na Virtex-4 implementacijskoj platformi je približno 120 MHz.

Na izlazu zbrajala se nalazi registar (zbrajalo je sinkrono). Registar na izlazu zbrajala predstavlja memorijski element rekurzivne grane.

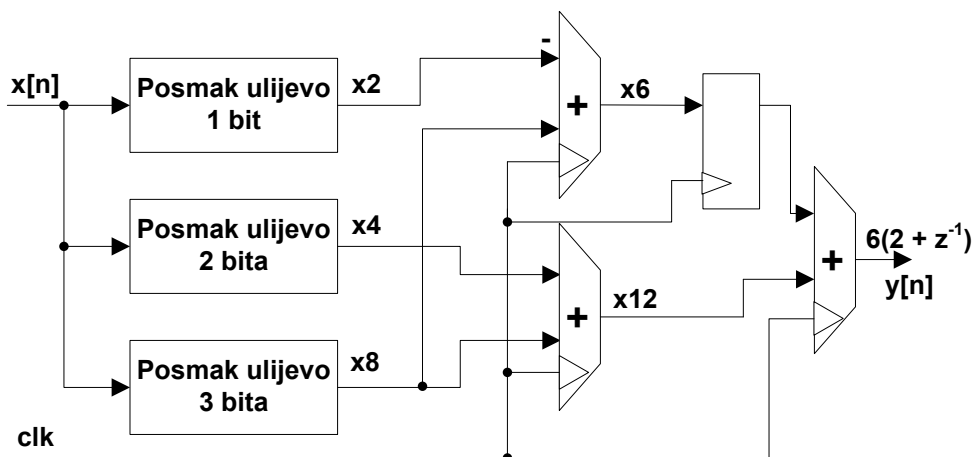
Kašnjenje filtarske strukture iznosi 1 period takta, a inicijalno utitravanje 2 perioda takta.

Gornji predfilter

Ovaj filter je posljednji predfilter za predprocesiranje signala u gornjoj grani. Filter je FIR tipa, sa prijenosnom funkcijom zadanom izrazom 3.12.

$$H(z) = 6(2 + z^{-1}) \quad (3.12)$$

RTL izvedba filtra je prikazana na slici 3.10.



Slika 3.10: RTL realizacija gornjeg predfiltra

RTL implementacija je izvedena uporabom zbrajala kako bi se uštedjeli resursi. Svi posmaci su asinkroni (nema utroška logike), a zbrajala sinkrona (sa registrom na izlazu).

Kašnjenje filtarske strukture iznosi 2 perioda takta, a inicijalno utitravanje 3 perioda takta. Iako se radi o jednostavnom filtru, nije moguće smanjiti kašnjenje filtra bez povećanja broja razina kombinatorne logike.

Gornja filtarska banka

Gornja filtarska banka se sastoji od 3 FIR filtra prvog reda. Filtri su ključni za generiranje koeficijenata polinoma B, C i D. Koeficijent A nije potrebno generirati, jer mu je iznos jednak iznosu ulaznog uzorka signala.

Prijenosne funkcije koeficijenata B, C, D su dane izrazima 3.13, 3.14, 3.15 respektivno.

$$H_B(z) = -z^{-1} \quad (3.13)$$

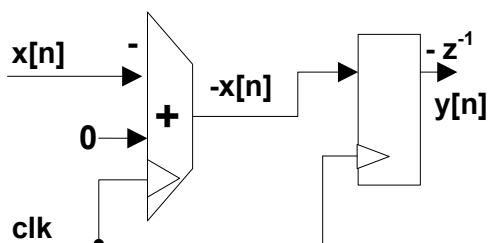
$$H_C(z) = 1 + 2z^{-1} \quad (3.14)$$

$$H_D(z) = -1 - z^{-1} \quad (3.15)$$

Kašnjenje svih izlaza gornje filtarske banke je 1 period takta (npr. za ostvarenje prijenosne funkcije $H(z) = 1$), uz inicijalno utitravanje od 2 perioda takta.

Svi filtri filtarske banke su implementirani bez utroška množiła, korištenjem zbrajala i sklopova za posmak, zbog uštede sklopovskih resursa.

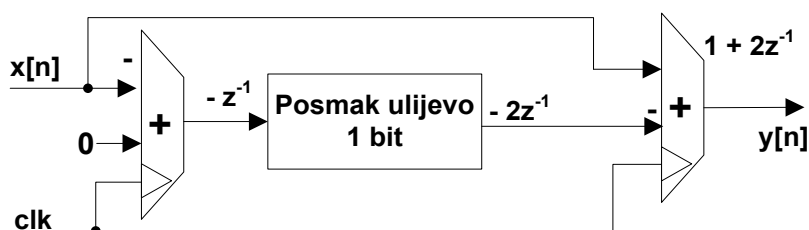
Implementacija prijenosne funkcije definirane izrazom 3.13 je prikazana na slici 3.11.



Slika 3.11: RTL shema filtra definiranog prijenosnom funkcijom 3.13

Zbrajalo je sinkrono, te nakon invertiranja signal već kasni jedan ciklus takta. Dodatno kašnjenje je uvedeno zbog poravnanja za izlazom banke (z^{-1}).

Implementacija prijenosne funkcije definirane izrazom 3.14 je prikazana na slici 3.12.



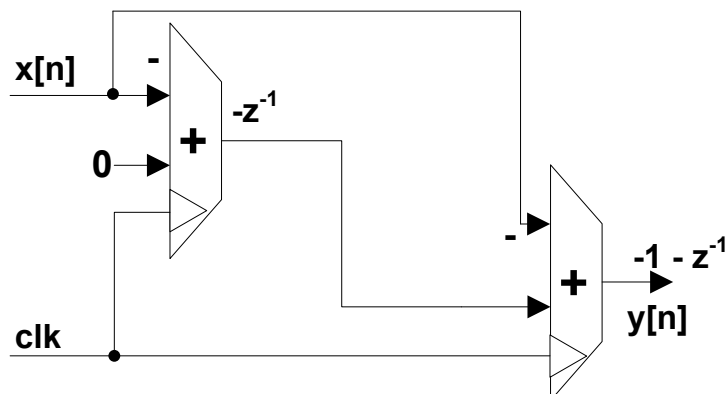
Slika 3.12: RTL shema filtra definiranog prijenosnom funkcijom 3.14

Sklop za posmak radi asinkrono, bez utroška logike. Zbrajala su sinkrona. Inicijalno invertiranje signala također unosi i kašnjenje, te se na sljedećem zbrajalu evaluira prijenosna funkcija u potpunosti.

Filtar dijeli resurse sa ostalim filtrima iz filtarske banke. Zbog toga se ulazni signal u prvoj fazi invertira, a u sljedećoj fazi oduzima na zbrajalu, jer je u filtarskoj banci signal $-z^{-1}$ već dostupan (iz drugog filtra). Stvaranje signala z^{-1} bi bio dodatan utrošak resursa.

Ovo je tipičan primjer povrede enkapsulacije kod modularnog dizajna – implementacija filtera u filterskoj banci se isprepliće (filtri nisu potpuno nezavisni). Kada bi svaki filter bio implementiran kao zaseban modul, dizajn bi bio jasniji, ali uz veći broj utrošenih resursa. Problem je naveden u poglavlju Zaključak.

Implementacija prijenosne funkcije definirane izrazom 3.15 je prikazana na slici 3.13.



Slika 3.13: RTL shema filtra definiranog prijenosnom funkcijom 3.15

Zbrajala su sinkrona, te nakon invertiranja signal već kasni jedan ciklus takta. Invertirani i zakašnjeli signal se dovodi na izlazno zbrajalo, koje evaluira funkciju filtra zbrajanjem invertiranog zakašnjelog signala, i negativnog trenutnog .

Donja filteraska banka

Donja filteraska banka se sastoji od 4 FIR filtera prvog reda. Filtri potpomažu generiranje koeficijenata polinoma A, B, C i D. Rezultati ove banke se pribrajaju rezultatima gornje filteraske banke. Zbrojeni rezultati gornje i donje filteraske banke čine izlazne koeficijente generatora kubičnog polinoma.

Koeficijent A nije potrebno generirati, jer mu je iznos jednak iznosu ulaznog uzorka signala (kod interpolacije aproksimirana funkcija mora prolaziti točno kroz poznate točke), stoga se FIR filter za njegovo generiranje sastoji samo od elementa za kašnjenje.

Prijenosne funkcije koeficijenata A, B, C, D su dane izrazima 3.16, 3.17, 3.18, 3.19 respektivno.

$$H_A(z) = z^{-1} \quad (3.16)$$

$$H_B(z) = 3z^{-1} \quad (3.17)$$

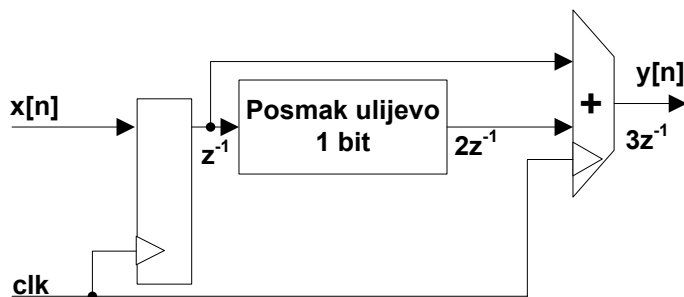
$$H_C(z) = -9z^{-1} \quad (3.18)$$

$$H_D(z) = 1 + 5z^{-1} \quad (3.19)$$

Kašnjenje svih izlaza donje filtarske banke je 1 period takta, uz inicijalno utitravanje od 2 perioda takta. Izlazi banke moraju biti poravnati za izlazima gornje banke.

Svi filtri filtarske banke su implementirani bez utroška množila, korištenjem zbrajala i sklopova za posmak, zbog uštede sklopovskih resursa.

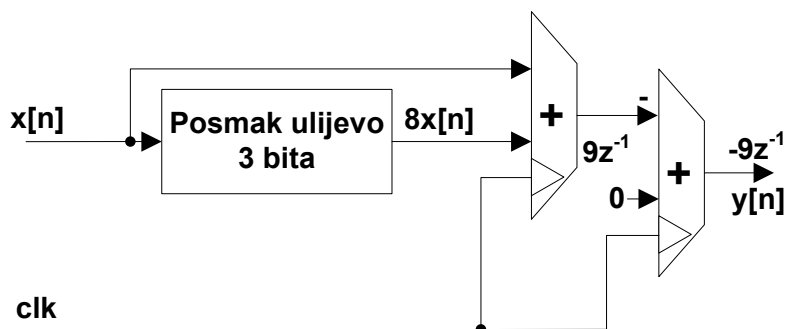
Implementacija prijenosne funkcije definirane izrazom 3.17 je prikazana na slici 3.14.



Slika 3.14: RTL shema filtra definiranog prijenosnom funkcijom 3.17

Zbrajalo je sinkrono, posmak je asinkron (bez utroška logike). Trostruko veći iznos signala je dobiven zbrajanjem posmaknutog i izvornog signala. Inicijalni registar realizira kašnjenje prijenosne funkcije.

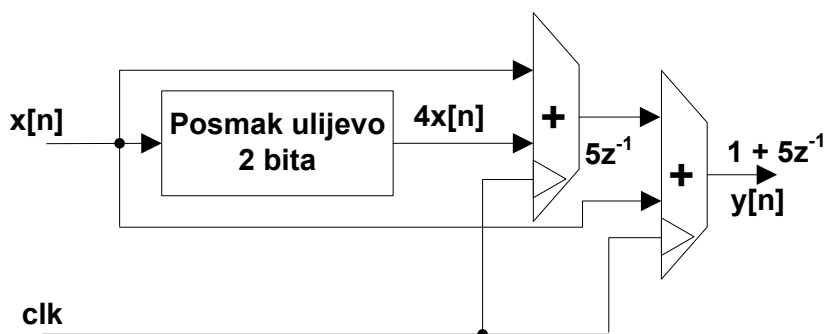
Implementacija prijenosne funkcije definirane izrazom 3.18 je prikazana na slici 3.15.



Slika 3.15: RTL shema filtra definiranog prijenosnom funkcijom 3.18

Zbrajala su sinkrona, posmak je asinkron (bez utroška logike). Deveterostruko veći iznos signala je dobiven zbrajanjem posmaknutog i izvornog signala. Izlazno zbrajalo obavlja invertiranje signala, i unosi dodatno kašnjenje (izlaz je poravnat sa izlazom banke).

Implementacija prijenosne funkcije definirane izrazom 3.19 je prikazana na slici 3.16.



Slika 3.16: RTL shema filtra definiranog prijenosnom funkcijom 3.19

Zbrajala su sinkrona, posmak je asinkron (bez utroška logike). Peterostruko veći iznos signala je dobiven zbrajanjem posmaknutog i izvornog signala. Sinkrono zbrajalo izvornog i posmaknutog signala unosi kašnjenje z^{-1} . Izlazno zbrajalo realizira prijenosnu funkciju zbrajanjem ulaznog signala i zakašnjelog skaliranog.

Opisane su sve komponente generatora kubičnog polinoma. Izlazi iz gornje i donje banke se sumiraju u sloju zbrajala, te je generiran kubični polinom koji predstavlja dio splajn funkcije, metodom kratke kauzalne kubične splajn interpolacije.

Pri implementaciji cijelog sustava, a posebice generatora kubičnog polinoma, potrebno je obratiti pozornost na dinamiku (raspon) svih signala. Bitno je osigurati najveći mogući raspon kretanja svih vrijednosti signala, kako bi se ostvarila najveća preciznost rada sustava. Proračun dinamike je moguće analizirati Matlab m-skriptom `ProracunDinamike.m`.

Filtar frakcionalnog kašnjenja

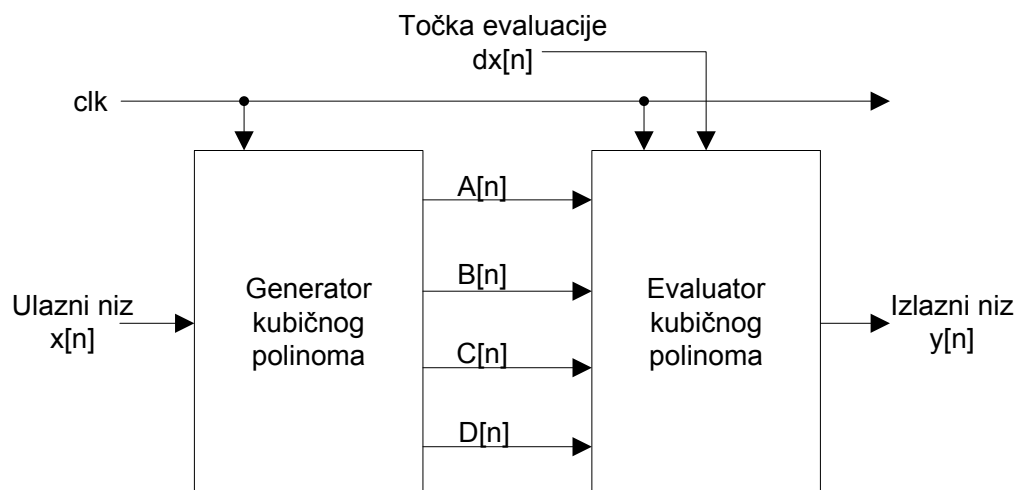
Filtar frakcionalnog kašnjenja je linija za kašnjenje diskretnog signala za necjelobrojni iznos perioda signala vremenskog vođenja. Izlazni signal je također diskretni signal, ali sa uzorcima koji ne postoje u izvornom signalu, već su interpolirani u određenoj točki između postojećih uzoraka.

Unatoč tome što filtar frakcionalnog kašnjenja nije potpun sustav za pretipkavanje, moguće je ispitati velik dio sustava korištenjem ovakvog filtra, uključujući i metodu generacije i evaluacije polinoma, odnosno sam proces interpolacije.

Filtar frakcionalnog kašnjenja se ostvaruje izravnim spajanjem generatora i evaluatora kubičnog polinoma, te specificiranjem argumenta kubičnog polinoma, odnosno točke evaluacije dx , kao trenutka u vremenu u kojem se obavlja interpolacija (frakcionalni iznos kašnjenja).

Opisani sustav je poznat i kao *Farrow* struktura (generiranje polinoma i evaluacija Hornerovom shemom).

Slika 3.17 prikazuje blok shemu filtra frakcionalnog kašnjenja.



Slika 3.17: Filtar frakcionalnog kašnjenja

Filtar frakcionalnog kašnjenja ne mijenja broj uzoraka signala. Izlazni signal ima jednak broj uzoraka, koji se na izlazu pojavljuju vremenski sinkrono sa učitavanjem novih uzoraka u sustav.

Kašnjenje filtra frakcionalnog kašnjenja (broj perioda takta za propagaciju signala od ulaza do izlaza) je jednako zbroju kašnjenja generatora i evaluatora kubičnog polinoma, a definirano je izrazom 3.20. Inicijalno utitravanje sustava je definirano izrazom 3.21.

$$d = FIR_stupnjeva * 2 + 12 \quad (3.20)$$

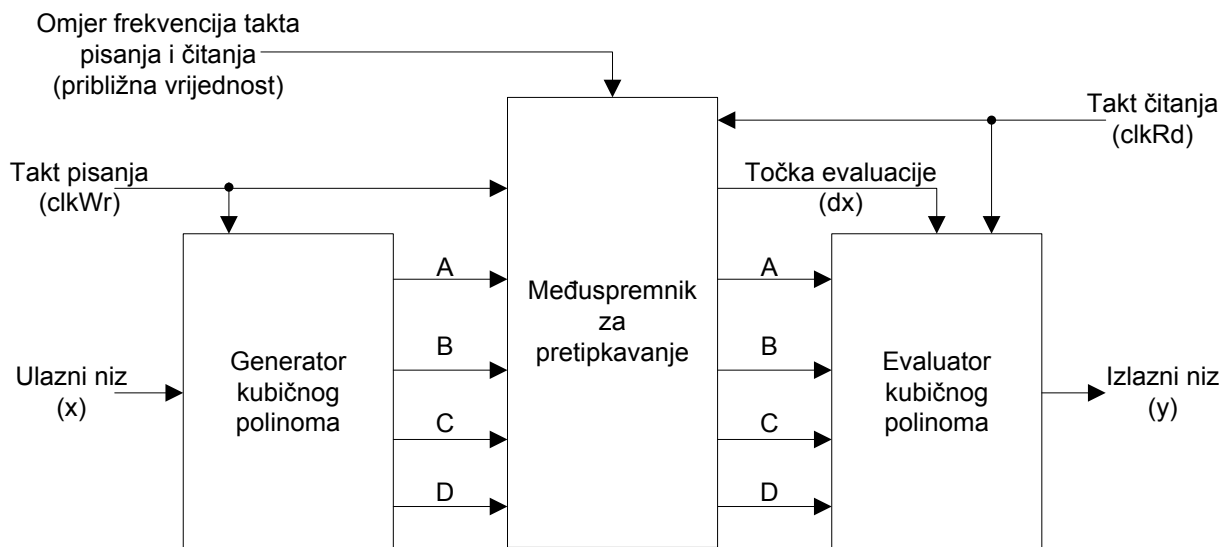
$$u = FIR_stupnjeva * 3 + 15 \quad (3.21)$$

Ispitivanje rada filtra frakcionalnog kašnjenja je izvedeno u narednim poglavljima.

Međuspremnik za pretipkavanje

Prethodno opisani primjer filtra frakcionalnog kašnjenja već omogućava iskorištavanje interpolacijskih mogućnosti sustava. Međutim cilj je izgraditi sustav koji bi omogućavao pretipkavanje signala na bilo koji period otipkavanja, potpuno nezavisan od izvornog. Sa druge strane, filter frakcionalnog kašnjenja ne iskorištava generirane polinome u potpunosti. Svaka četvorka generiranih polinoma se kod ovog filtra iskorištava samo jednom za evaluaciju, u samo jednoj određenoj točki koju definira iznos potrebnog frakcionalnog kašnjenja. Jasno je vidljivo, da se pretipkavanje na vrlo velik broj uzoraka može izvesti višestrukim evaluacijama iste četvorke koeficijenata polinoma – ne samo jednom. Pretipkavanje na manji broj uzoraka uključuje evaluaciju svake četvorke koeficijenata manje od jednom – broj izlaznih uzoraka je manji, a četvorki koeficijenata polinoma ima onoliko koliko ima ulaznih uzoraka.

Vidljivo je da se problem može riješiti ugradnjom međuspremnika za koeficijente polinoma. Sustav filtra frakcionalnog kašnjenja se nadograđuje ugradnjom međuspremnika između generatora i evaluatora kubičnog polinoma. Međuspremnik prihvaća nove koeficijente od generatora polinoma, te isporučuje koeficijente evaluatoru polinoma potrebnom brzinom. Ovakav sustav se naziva sustav za pretipkavanje, i prikazan je shemom na slici 3.18.



Slika 3.18: Sustav za pretipkavanje signala

Osim što pohranjuje koeficijente generirane od generatora polinoma, te ih isporučuje evaluatoru polinoma, međuspremnik za pretipkavanje također određuje i točku evaluacije polinoma. Na taj način je moguće postići precizno pretipkavanje na širok raspon izlaznih frekvencija otipkavanja.

Ako je potrebno povećati broj uzoraka signala 100 puta (pretipkavanje 1:100), za svaki ulazni uzorak generirati će se jedna četvorka koeficijenata, koju će međuspremnik isporučiti evaluatoru 100 puta (istu četvorku), ali svaki puta sa drugom točkom evaluacije (svaki puta kasnijom u vremenu, većom po iznosu).

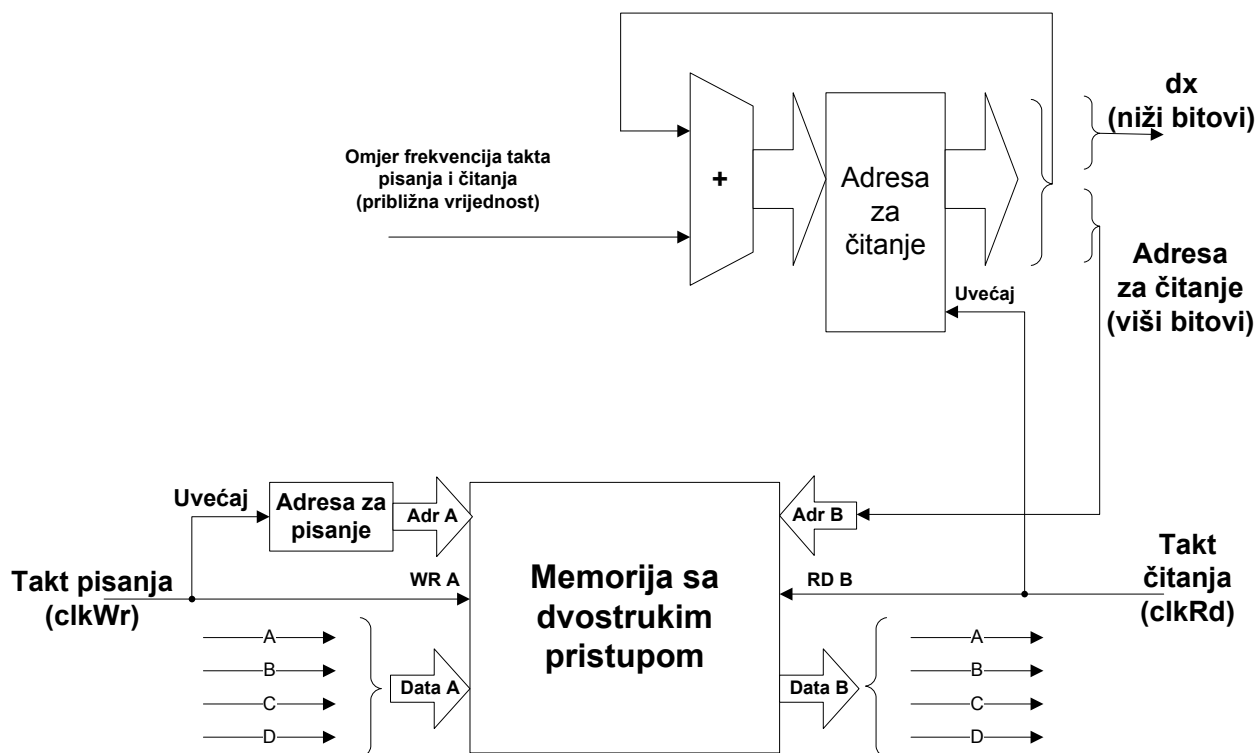
Ako je potrebno smanjiti broj uzoraka signala za cjelobrojni broj, primjerice $N:1$, interpolacija nije potrebna jer je dovoljno uporabiti svaki N -ti uzorak izvornog signala. Kod ovog sustava za pretipkavanje, evaluira se polinom nad svakom N -tom četvorkom koeficijenata (ostalih $N-1$ četvorki se odbacuju). Također je moguće smanjiti broj uzoraka za necjelobrojni faktor N . Tada se također odbacuje određeni broj četvorki koeficijenata, a argument dx je pri svakoj evaluaciji tipično različite vrijednosti.

Međuspremnik za pretipkavanje zahtijeva i dodatan konfiguracijski podatak – specifikaciju omjera frekvencija otipkavanja ulaznog i izlaznog signala. Kasnije će biti opisani razlozi zašto zadavanje ovog podatka nije od ključne važnosti, ali njegovo specificiranje doprinosi performansama međuspremnika.

Sa slike 3.18 su također vidljive i izmjene nad signalom vremenskog vođenja. Kod filtra sa frakcionalnim kašnjenjem je bio jednak broj ulaznih i izlaznih uzoraka, te je bio dovoljan jedinstven signal vremenskog vođenja. Sustav za pretipkavanje mora omogućiti potpuno neovisne frekvencije otipkavanja ulaznog i izlaznog signala. Generiranje kubičnog polinoma je usko vezano uz ulazni signal, pa se pogon ovog modula obavlja signalom vremenskog vođenja otipkavanja ulaznog signala. Evaluacija kubičnog polinoma je usko vezana uz izlazni signal, pa se pogon ovog modula obavlja signalom vremenskog vođenja otipkavanja izlaznog signala.

Međuspremnik za pretipkavanje je ovisan o oba signala vremenskog vođenja. Na ovom modulu se susreću oba signala, koja su međusobno asinkrona. Problem asinkronih signala vremenskog vođenja stvara probleme sa prijenosom podatka iz domene jednog signala vremenskog vođenja u domenu drugog signala.

Pojednostavljena RTL shema međuspremnika za pretipkavanje prikazana je na slici 3.19.



Slika 3.19: Pojednostavljena RTL shema međuspremnika za pretipkavanje

Koeficijenti polinoma se pohranjuju u spremnik tipa FIFO. Ulazni kraj FIFO spremnika se povezuje na generator kubičnih polinoma, a izlazni kraj na evaluator kubičnih polinoma.

FIFO spremnik je na FPGA sklopovlju izveden pomoću memorije sa dvostrukim pristupom i dvije adresne kazaljke.

Adresna kazaljka za pisanje je neovisna o ostatku sustava i uvećava se na takt pisanja. Osim uvećanja adresne kazaljke pisanja, na takt pisanja memorija sa dvostrukim pristupom preko pristupa A pohranjuje koeficijente polinoma na lokaciju na koju pokazuje adresna kazaljka za pisanje.

Adresna kazaljka za čitanje je implementirana na složeniji način, jer određuje brzinu pražnjenja međuspremnik. Kazaljka je implementirana kao registar veće širine (ima svoj cjelobrojni i decimalni dio). Cjelobrojni dio adresne kazaljke čitanja su njeni viši, a decimalni dio niži bitovi. Svakim čitanjem međuspremnik preko pristupa B (takt čitanja), kazaljka se uvećava za iznos određen omjerom frekvencija čitanja i pisanja.

Konfiguracijski iznos za uvećanje adresnog pokazivača čitanja je također broj sa decimalnim i cjelobrojnim dijelom (kao i sam adresni pokazivač). Ako je iznos za uvećanje pokazivača čitanja postavljen na vrijednost 1.0, radi se pretipkavanje na jednaku (nepromijenjenu) frekvenciju otipkavanja. Ako je iznos uvećanja manji od 1.0, vrši se pretipkavanje signala na veći broj uzoraka (u više uzastopnih uvećavanja adresni pokazivač čitanja ima jednak cjelobrojni dio, te pokazuje na istu četvorku koeficijenata). Ako je iznos uvećanja veći od 1.0, vrši se pretipkavanje signala na manji broj uzoraka (povremeno se događa da se cjelobrojni dio adresnog pokazivača čitanja uveća za više od 1, te se preskaču određene četvorke koeficijenata).

Decimalni dio adresnog pokazivača čitanja (niži bitovi) se iskorištava kao točka evaluacije kubičnog polinoma. Primjena se može demonstrirati na primjeru pretipkavanja 1:2, odnosno dvostrukom povećanju broja uzoraka signala. Tada se iznos za uvećanje adresnog pokazivača čitanja postavlja na 0.5. Uzastopnim uvećanjima adresnog pokazivača čitanja, njegov cjelobrojni dio će se uvećavati tek svaki drugi put, što znači da će svaka četvorka koeficijenata biti postavljena na izlaz memorije dva puta. Decimalni dio adresnog pokazivača će alternirati između

0.0 i 0.5. Kako je decimalni dio iskorišten za točku evaluacije polinoma, vidljivo je da će se izmjenjivati točka evaluacije između vremenskog trenutka 0, i polovine vremenskog intervala interpolacijskog polinoma. Evaluacijom polinoma u trenutku 0, rezultat je izvorni koeficijent A polinoma. Ovaj koeficijent odgovara uzorku izvornog signala uz koji su asocirani generirani koeficijenti, odnosno svaki drugi pretipkani uzorak će biti jednak izvornom signalu. Evaluacijom polinoma u trenutku 0.5 interpolira se vrijednost vremenski točno između dva postojeća signala. Dakle u pretipkanom signalu će se izmjenjivati postojeće vrijednosti, te interpolirane vrijednosti vremenski na sredini intervala između postojećih vrijednosti. Sustav uspješno pretipkava signal na dvostruko veću frekvenciju otipkavanja od izvorne.

FIFO međuspremnik je memorijski element određenog kapaciteta. Zbog otpornosti na varijacije u frekvencijama otipkavanja, cilj je održati međuspremnik na približno 50% popunjenosti.

Iako se u opisanom sustavu sa slike 3.19 pojavljuju asinkroni signali vremenskog vođenja, problem sa prijenosom podataka (još) ne postoji, jer je jedina komponenta na kojoj se signali dotiču FIFO međuspremnik, međutim kako je cilj održavati međuspremnik popunjenim približno 50% u svakom trenutku, upisivanje i čitanje iz međuspremnika (operacije koje su međusobno asinkrone) se odvijaju na njegovim suprotnim stranama.

Realizacija međuspremnika treba pohraniti četiri koeficijenta polinoma na jednu memorijsku lokaciju. Ukupna količina bitova potrebna za koeficijente polinoma je 72 bita (maksimalna širina koeficijenta), odnosno toliko mora pohraniti svaka lokacija međuspremnika. Broj potrebnih lokacija ovisi o željenim karakteristikama sustava.

Za ostvarenje međuspremnika se koristi ugrađeno sklopovlje Spartan-3 FPGA sklopova. Ovi FPGA sklopovi sadrže dva namjenska tipa memorije: blokovski i distribuirani RAM. Oba tipa memorije podržavaju dvostruki pristup, što je nužan uvjet za izradu FIFO međuspremnika. Moguće konfiguracije blokovskog RAM-a su prikazane u tablici 3.1, a distribuiranog u tablici 3.2.

Tablica 3.1: Podržane konfiguracije blokovskog RAM-a sa dva pristupa, [12],
Memory Organization/Aspect Ratio

		Pristup A					
		16K x1	8K x2	4K x4	2K x9	1K x18	512 x36
Pristup B	16K x1	_S1_S1					
	8K x2	_S1_S2	_S2_S2				
	4K x4	_S1_S4	_S2_S4	_S4_S4			
	2K x9	_S1_S9	_S2_S9	_S4_S9	_S9_S9		
	1K x18	_S1_S18	_S2_S18	_S4_S18	_S9_S18	_S18_S18	
	512 x36	_S1_S36	_S2_S36	_S4_S36	_S9_S36	_S18_S36	_S36_S36

Da bi blokovski RAM (tablica 3.1) bilo moguće koristiti kao 72-bitni FIFO međuspremnik, potrebno je instancirati dvije instance, sa dvostrukim pristupom, u konfiguraciji sa dva 36-bitna pristupa (_S36_S36). U svakoj instanci su 4 bita namijenjena za pohranu pariteta, ali će biti korišteni za pohranu podatkovnih bitova. Svaka instanca može pohranjivati 36 bita koeficijentata polinoma. Dubina FIFO međuspremnika u ovom slučaju je 512 lokacija.

Tablica 3.2: Podržani načini rada distribuirane RAM memorije, [11], *Library Primitives*

Ime instance	Broj riječi x dužina riječi	Broj pristupa	Adresne linije
RAM16X1S	16x1	Jedan	A3,A2,A1,A0
RAM32X1S	32x1	Jedan	A4,A3,A2,A1,A0
RAM64X1S	64x1	Jedan	A5,A4,A3,A2,A1,A0
RAM16X1D	16x1	Dva	A3,A2,A1,A0

Alternativna memorija za realizaciju FIFO međuspremnika je distribuirana RAM memorija. Kako je nužan uvjet mogućnost dvostrukog pristupa memoriji, jedina moguća konfiguracija za korištenje distribuiranog RAM-a je 16x1 (tablica 3.2). Kako je u toj konfiguraciji širina podatka na jednoj lokaciji jedan bit, potrebno je instancirati 72 instance distribuirane RAM memorije konfiguracije 16x1. Dubina takvog FIFO međuspremnika je 16 lokacija, a za njegovu realizaciju su potrebna 36 CLB blokova Spartan-3 arhitekture.

Iz opisanog je vidljivo da je za dublje međuspremnike jednostavnija uporaba blokovske RAM memorije. Ukoliko je na FPGA sklopu dostupan nedovoljan broj komponenti blokovskog RAM-a, moguće je za dublje međuspremnike koristiti

ulančani distribuirani RAM, no uz podizanje kompleksnosti samog memorijskog podsustava i uz dodatan utrošak CLB blokova.

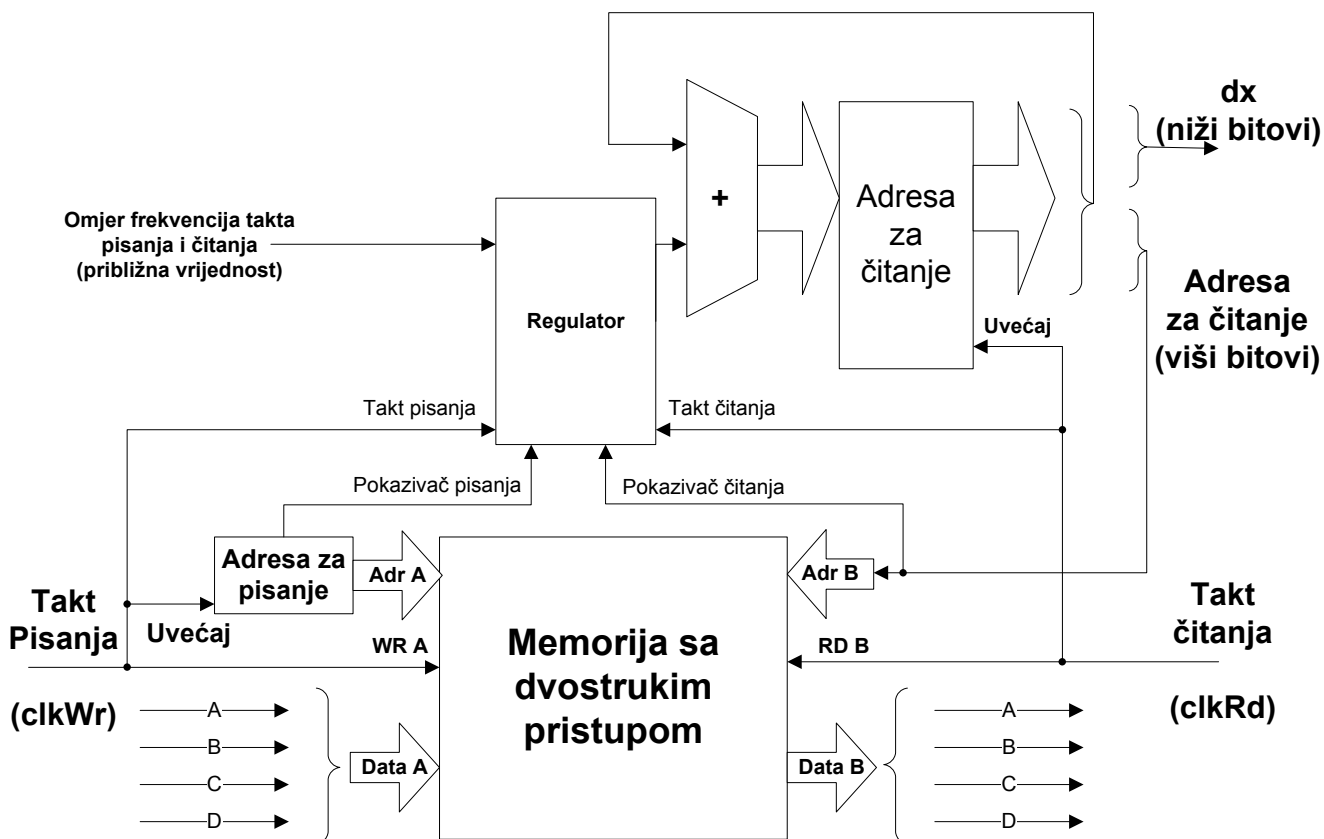
Zbog boljih svojstava kompenzacijskih mogućnosti, za realizaciju je odabran blokovski RAM. Dubina međuspremnik je 512 lokacija. Optimalna popunjenost međuspremnik je 256 lokacija (zbog mogućnosti regulacije, opisano u nastavku). Ovakva konfiguracija unosi dodatno kašnjenje u rad sustava od 256 uzoraka.

Regulacija popunjenosti međuspremnik

Opisano rješenje (slika 3.19) je idealizirano. Pretpostavljen je točan iznos frekvencija obje frekvencije otipkavanja, te zadana ispravna vrijednost uvećanja izlaznog adresnog pokazivača spremnik. U navedenim uvjetima je očekivano zadržavanje popunjenosti međuspremnik na 50% kapaciteta.

Na stvarnom sklopovlju nije moguće ostvariti točan iznos niti jedne od frekvencija otipkavanja, a tako ni njihovog omjera. Osim neidealnosti oscilatora po iznosu, prisutne su i varijacije frekvencija otipkavanja (a time i njihovog omjera). U navedenim, realnim uvjetima, nije moguće definirati konstantnu vrijednost za uvećanje izlaznog adresnog pokazivača. Zbog pogrešno definirane vrijednosti uvećanja, FIFO međuspremnik se počinje podpunjavati ili prepunjavati, sve do konačnog preljeva ili podljeva, što dovodi do grube pogreške u radu sustava.

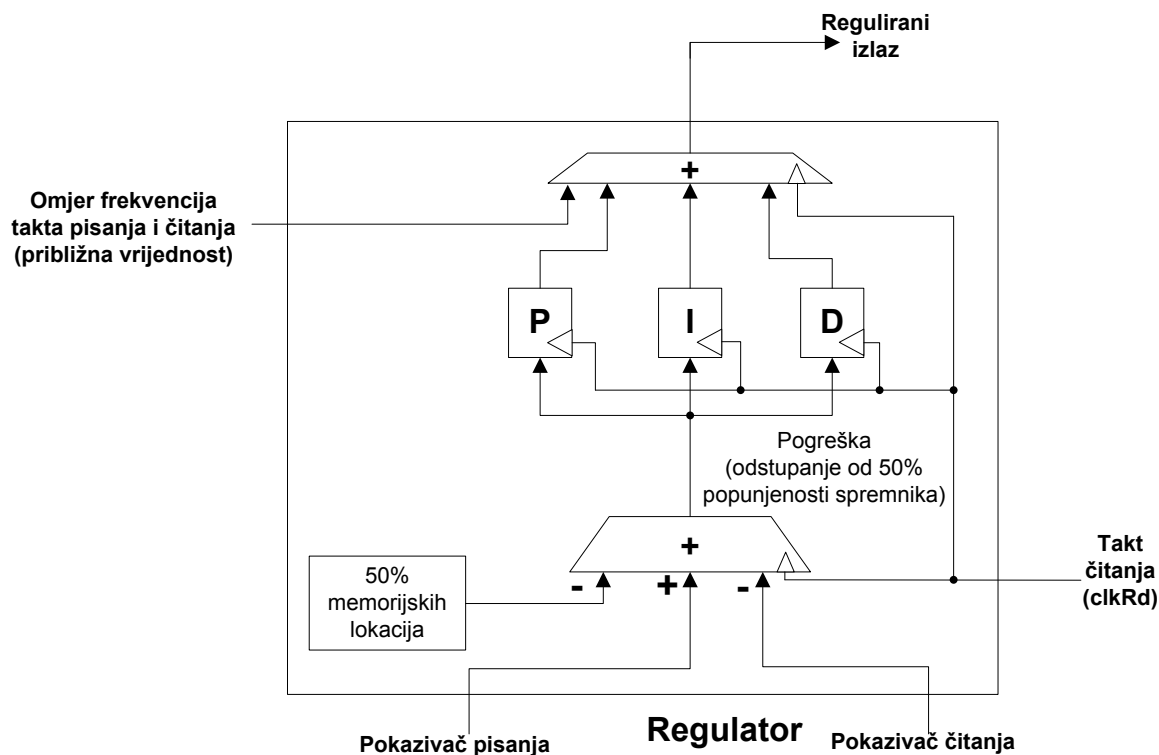
Kako bi se problem otklonio, potrebno je dodati regulacijski mehanizam za korekciju konstantne vrijednosti za uvećanje izlaznog adresnog pokazivača. Korišten je regulator tipa PID (proporcionalno integracijsko derivacijski regulator). PID regulator je među najpopularnijim tipovima regulatora, a posebno je pogodan za regulaciju nepoznatih procesa čije mijenjanje nije poznato *a priori*. Iako je raskorak frekvencija otipkavanja relativno poznat proces, ugradnjom PID regulatora omogućene su optimalne reakcije i na skokovite promjene frekvencija. Regulacijska pogreška je definirana kao odstupanje popunjenosti međuspremnik od 50%. Međuspremnik za pretipkavanje sa ugrađenim regulacijskim mehanizmom je prikazan na slici 3.20.



Slika 3.20: Međuspremnik za pretipkavanje sa regulacijskim krugom

Ako je međuspremnik popunjen iznad 50%, regulator ubrzava pražnjenje povećavajući pribroj za uvećanje adresnog pokazivača čitanja. Suprotno, ako je međuspremnik popunjen ispod 50%, regulator usporava pražnjenje umanjujući pribroj za uvećanje adresnog pokazivača čitanja.

Slika 3.21 prikazuje pojednostavljenu shemu implementiranog PID regulatora.



Slika 3.21: Pojednostavljena shema PID regulatora

Pogreška je definirana kao odstupanje popunjenosti međuspremnika od 50% popunjenosti. Izračunata pogreška se dovodi na proporcionalni, integracijski i derivacijski član regulatora. Odzivi svih članova se zbrajaju, uključujući i zadani omjer frekvencija otipkavanja, te se na izlaz postavlja ukupan, regulirani pribroj adresnom pokazivaču čitanja.

Shema je pojednostavljena, jer sva višestruka zbrajanja i slične kombinacijske funkcije moraju biti izvedene u fazama, odnosno preko cjevovoda. Također je bitno spriječiti preljev reguliranog izlaza ispod 0 ili iznad maksimalne vrijednosti, pa je potrebno koristiti aritmetiku sa zasićenjem u većini regulacijskih komponenata.

Implementacija regulacijskih komponenata

Odziv PID regulatora se sastoji od proporcionalne, integracijske i derivacijske komponente (izraz 3.25). Proporcionalna komponenta (izraz 3.22) svojom brzinom djeluje na brzo kompenziranje pogreške. Integracijska komponenta (izrazi 3.23) vrši integraciju svih prethodnih pogrešaka (memorijski element). Memorijsko djelovanje ublažava oscilacije oko traženog odziva. Derivacijska komponenta (izrazi 3.24) ubrzava odziv (pri naglim promjenama pogreške), koji je postao trom dodavanjem integracijskog člana.

Algebarski zapis rada PID regulatora je prikazan izrazima u nastavku.

$$P_odziv = pogreška * P_konstanta \quad (3.22)$$

$$I_suma = I_suma + (pogreška * I_konstanta) \quad (3.23)$$

$$I_odziv = I_suma$$

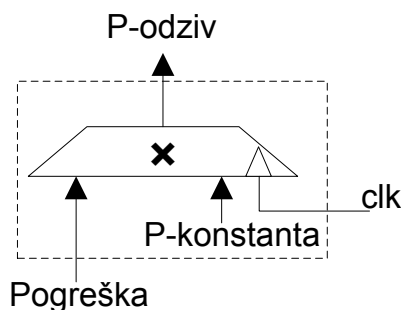
$$D_odziv = (pogreška - D_stanje) * D_konstanta \quad (3.24)$$

$$D_stanje = pogreška$$

$$PID_odziv = P_odziv + I_odziv + D_odziv \quad (3.25)$$

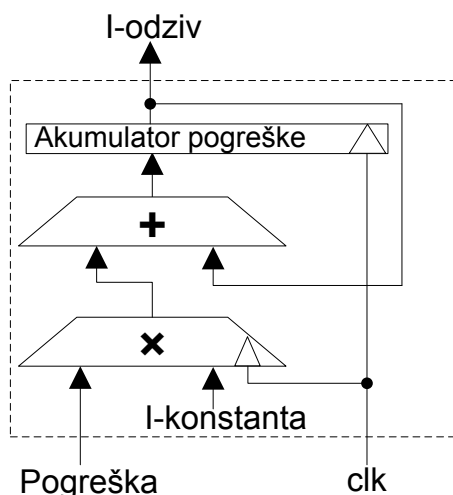
PID regulator je moguće podesiti uporabom različitih P, I, D konstanti, koje pojačavaju djelovanje pojedine regulacijske komponente, kako bi se dobio idealan odziv.

RTL izvedba P-komponente je prikazana na slici 3.22. Komponenta obavlja jednostavno skaliranje pogreške P-konstantom. Izlaz komponente je potrebno dodatno zakasniti kako bi bio poravnat sa izlazima ostalih regulacijskih komponenata.



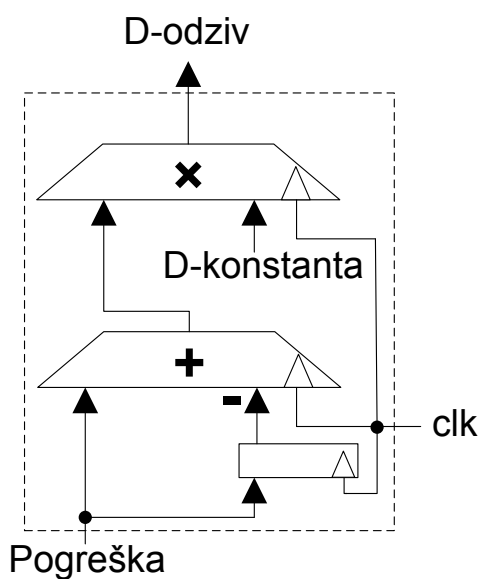
Slika 3.22: RTL implementacija P regulacijskog člana

RTL izvedba I-komponente je prikazana na slici 3.23. Komponenta obavlja integraciju pogreške. Množenje sa I-konstantom je moguće obaviti i na izlazu člana (distributivnost množenja prema zbrajanju). I komponenta je posebno kritična zbog mogućnosti preljeva akumulatora pogreške, te je potrebno implementirati operacije korištenjem aritmetike sa zasićenjem.



Slika 3.23: RTL izvedba I regulacijskog člana

RTL izvedba D-komponente je prikazana na slici 3.24. Komponenta izračunava promjenu (derivaciju) pogreške i skaliranje promjene D-konstantom.



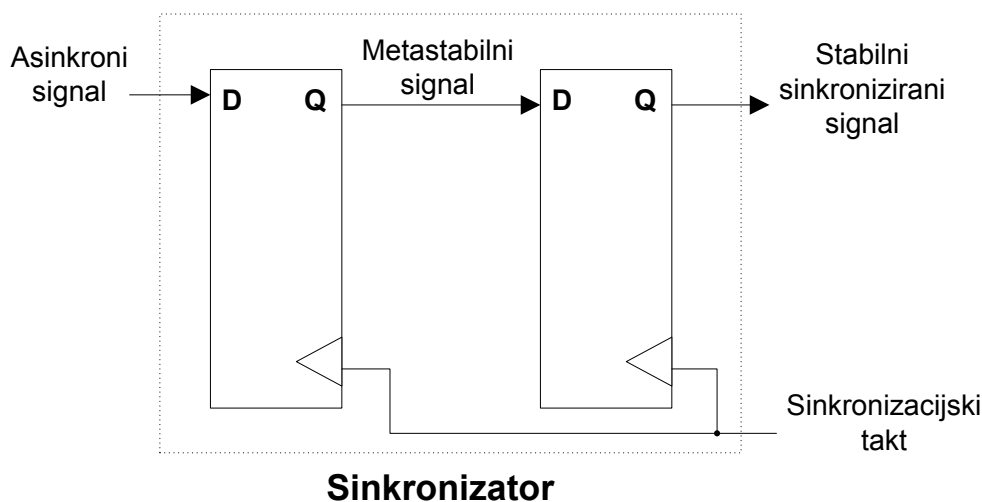
Slika 3.24: RTL izvedba D regulacijskog člana

Višestruki asinkroni signali vremenskog vođenja

Sve dosad opisane regulacijske komponente zahtijevaju pogonski signal vremenskog vođenja. Rezultat regulacije se upotrebljava za uvećavanje adresnog pokazivača čitanja, pa se i pogon regulacijskog kruga obavlja izlaznim taktom čitanja (slika 3.21). Problem se javlja sa adresnim pokazivačem pisanja, koji se

uvećava asinkronim taktom pisanja. Evaluacija zbrajala za izračun regulacijske pogreške zahtijeva stabilne signale pokazivača čitanja i pisanja. Dok se pokazivač čitanja ne mijenja za vrijeme evaluacije (uvećava se sinkrono na izlazni takt čitanja), ulazni pokazivač pisanja bi se mogao mijenjati u tom trenutku. Zbog toga bi evaluacija razlike pokazivača mogla rezultirati metastabilnim stanjem nekih bistabila u registru za pohranu razlike pokazivača (narušena *eng. set-up/hold* vremena bistabila). Navedeni problem nije postojao u prethodno opisanom sustavu bez regulacijskog kruga (slika 3.19), jer su asinkroni signali bili isključivo na suprotnim krajevima FIFO međuspremnika.

Problem se može ublažiti korištenjem sinkronizacijskog sklopovlja (slika 3.25), koje pomoću N ulančanih bistabila ($N \geq 2$) sinkronizira asinkroni signal (ulazni pokazivač) na takt sinkronog signala (izlaznog).



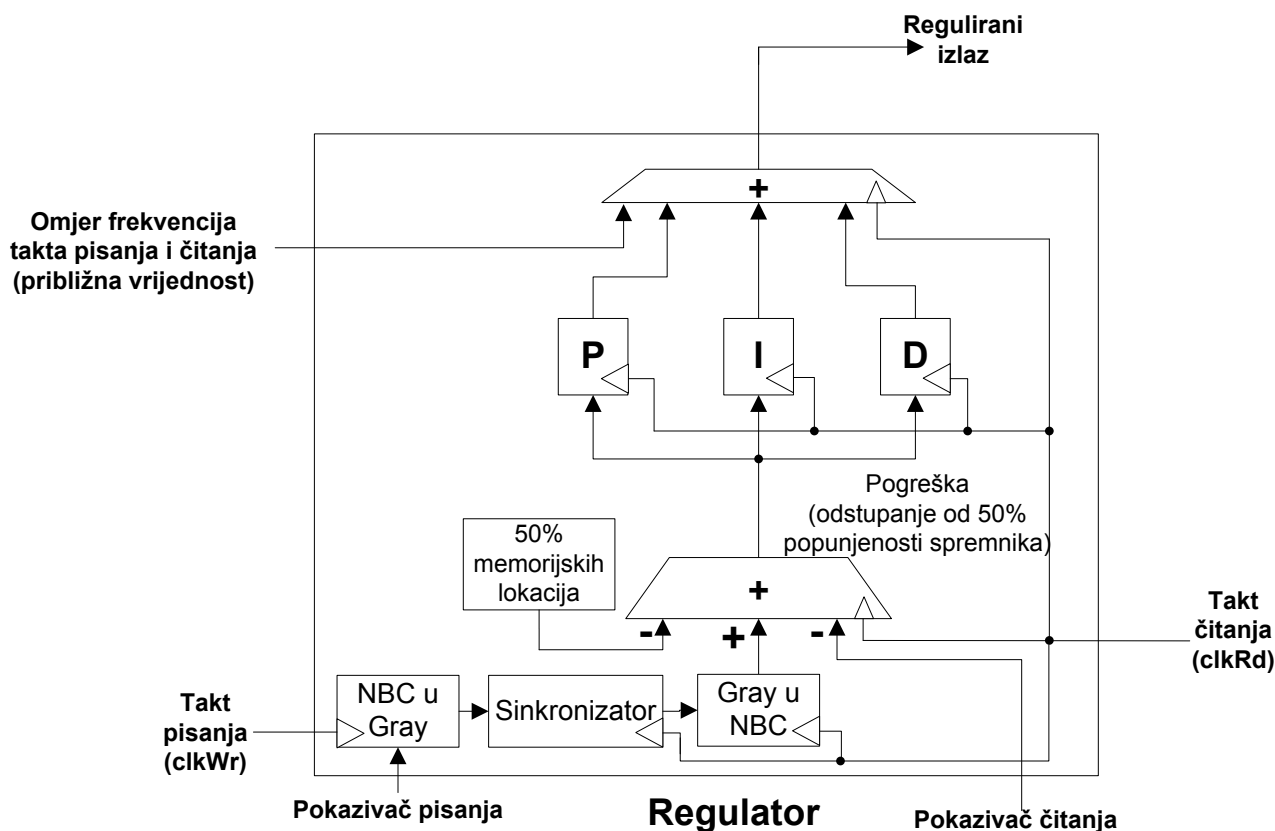
Slika 3.25: Sinkronizator sa dva stupnja

Korištenjem sklopovlja za sinkronizaciju se mogu otkloniti pojave metastabilnog stanja u regulacijskom sustavu, no zbog mogućih velikih razlika uzastopnih brojeva u binarnom brojevnom sustavu (*eng. Hamming cliff*) se ulazni pokazivač može pogrešno sinkronizirati na takt izlaznog.

Stoga je potrebno iznos ulaznog (asinkronog, promatrajući sa izlaza) pokazivača reprezentirati u Grayevom binarnom zapisu, obaviti sinkronizaciju na izlazni takt, te dekodirati iznos ulaznog pokazivača iz Grayevog zapisa u prirodni binarni kod (NBC), te zatim obaviti oduzimanje razlike pokazivača sinkrono na izlazni takt.

Grayev kod je pogodan jer je Hammingova udaljenost uzastopnih brojeva uvijek jednaka 1 (nema *Hammingovih cliffova*) pa je maksimalna moguća pogreška pri sinkronizaciji asinkronog (metastabilnog) signala jednaka 1, što ne utječe u velikoj mjeri na rad PID regulatora.

Regulator sa riješenim problemom metastabilnosti je prikazan na slici 3.26.



Slika 3.26: PID regulator sa sinkronizacijom asinkronog ulaznog pokazivača

Pretvorba NBC (prirodnog binarnog) koda u Grayev kod je izvedena jednostavnom kombinacijskom funkcijom (isključivo-ILI susjednih bitova NBC koda).

Pretvorba Grayevog koda u NBC reprezentaciju zahtijeva kombinacijsku funkciju sa velikim brojem logičkih razina. Zbog toga je ova pretvorba izvedena pohranjivanjem svih mogućih parova vrijednosti NBC i Grayevog koda u čitljivo memorijsko polje (*eng. look-up table*), izvedeno distribuiranom ROM memorijom. Za generiranje svih mogućih 2^9 iznosa adresa kao inicijalizacije za memorijsko polje, napisana je dodatna programska podrška.

Opisana je implementacija PID regulatora popunjenosti međuspremnika. Zbog implementacije regulatora korištenjem cjevovoda, uvedeno je regulacijsko

kašnjenje od 10 ciklusa takta čitanja. Ovo kašnjenje pridonosi pojavi regulacijskih oscilacija.

Ako podatak o omjeru frekvencija pretipkavanja nije zadan, ili je podešen na pogrešnu vrijednost, PID regulator može kompenzirati pogrešku i podesiti sustav da radi u ispravnom načinu rada (u skladu sa stvarnim frekvencijama otipkavanja). Međutim ovakvo ponašanje uvodi inicijalni oscilatorni period kod početka rada sklopa, dok regulator ne kompenzira pogrešku. Ovakav način rada se ne preporuča zbog iskorištenja raspona dinamike regulacijskog sustava, pa su regulacijske performanse umanjene.

Tehničke značajke sustava za pretipkavanje

Ciljna implementacijska platforma je Xilinx Spartan-3 FPGA ili Virtex-4 FPGA.

Širine signala

Preporučena širina ulaznog signala: 12 bita.

Preporučena širina izlaznog signala : 18 bita.

Preporučena širina frakcionalnog pomaka (dx): 18 bita.

Maksimalni radni takt

Implementacijska platforma Spartan-3:

Takt pisanja (clkWr): 117 MHz

Takt čitanja (clkRd): 160 MHz

Implementacijska platforma Virtex-4:

Takt pisanja (clkWr): 124 MHz

Takt čitanja (clkRd): 172 MHz

Kašnjenje i utitravanje

Približno kašnjenje cijelog sustava se može opisati sa stanovišta ulaznog (izraz 3.27) i izlaznog (izraz 3.26) signala vremenskog vođenja. Inicijalno utitravanje cijelog sustava se može opisati sa stanovišta ulaznog (izraz 3.29) i izlaznog (izraz 3.28) signala vremenskog vođenja.

Broj korištenih FIR stupnjeva je 4.

$$N_{kašnjenje} = (FIR_stupnjeva * 2 + 6 + 256) \frac{f_{čitanje}}{f_{pisanje}} + 6 \quad (3.26)$$

$$N_{kašnjenje} = (FIR_stupnjeva * 2 + 6 + 256) + \frac{f_{pisanje}}{f_{čitanje}} 6 \quad (3.27)$$

$$N_{utitravanje} = (FIR_stupnjeva * 3 + 9 + 256) \frac{f_{čitanje}}{f_{pisanje}} + 6 \quad (3.28)$$

$$N_{utitravanje} = (FIR_stupnjeva * 3 + 9 + 256) + \frac{f_{pisanje}}{f_{čitanje}} 6 \quad (3.29)$$

Kašnjenje regulacijskog sustava je 10 perioda izlaznog signala vremenskog vođenja (regulacijski sustav djeluje na rastući brid izlaznog signala vremenskog vođenja).

Podešavanje omjera pretipkavanja

Izračun faktora uvećanja izlaznog registra ovisno o traženom omjeru frekvencija pretipkavanja je prikazan izrazom 3.30.

$$R = \frac{f_{pisanje}}{f_{čitanje}} 2^{\text{širina_frakcionalnog_pomaka} - 1} \quad (3.30)$$

Ekstremni slučajevi pretipkavanja

Djelovanje regulacijskog sustava na izlazni signal vremenskog vođenja, te dubina međuspremnik od 512 uzoraka, imaju sljedeće posljedice pri odnosima frekvencija otipkavanja ulaznog i izlaznog signala:

- povećanje broja uzoraka – *eng. upsampling*: izlaznu frekvenciju otipkavanja je moguće povećavati od iznosa ulazne frekvencije sve do maksimalne podržane na implementacijskoj platformi (Spartan-3 / Virtex-4), no ako je izlazna frekvencija $2^{\text{širina_frakcionalnog_pomaka} - 1}$ puta veća od ulazne, više nema dodatnog povećanja kvalitete interpoliranog signala ($2^{\text{širina_frakcionalnog_pomaka} - 1}$ je broj mogućih nenegativnih frakcionalnih pomaka dx). Daljnjim povećanjem izlazne frekvencije moguće su minimalne distorzije izlaznog signala zbog rada regulacijskog sustava, te regulacijskog kašnjenja.

- smanjenje broja uzoraka – *eng. downsampling*: ulaznu frekvenciju otipkavanja je moguće povećavati od iznosa izlazne frekvencije otipkavanja sve do maksimalnih 256 puta više od izlazne frekvencije (50% FIFO međuspremnik). Daljnjim povećanjem ulazne frekvencije otipkavanja dolazi do prepunjenja međuspremnik zbog tromog rada regulatora (koji radi na izlaznom taktu) Dostizanjem 256 puta veće frekvencije regulator radi nepravilno, te se zbog prepunjenja učestalo aktivira signal za indikaciju prepunjenja međuspremnik (a time i zaustavljanje punjenja). Sustav radi oscilatorno (periodičko uključenje signala indikacije prepunjenja, jake oscilacije u popunjenosti međuspremnik). Sve to uzrokuje velike distorzije na izlaznom signalu pa se ne preporuča smanjenje broja uzoraka (*downsampling*) za tako velik faktor.

Sustav podržava indikaciju podpunjenosti i prepunjenosti FIFO međuspremnik.

Utrošak resursa sustava za pretipkavanje

Spartan-3:

BUFGMUX:	2
IOB:	67
MULT18X18:	9
RAMB16:	2
Slices:	782

Virtex-4:

BUFG	2
DSP48	11
ILOGIC	18
IOB	67
OLOGIC	20
RAMB16	2
Slices	757

U sljedećim poglavljima će biti opisano ispitivanje implementiranog sustava za pretipkavanje signala.

4. Ispitivanje rada sustava za pretipkavanje

Potrebno je ispitati funkcionalnost izgrađenog sustava za ispitivanje. Ispitivanje se vrši simulacijom i verifikacijom pomoću referentnih modela, te validacijom na sklopovlju.

Simulacija na PC računalu

Simulacija se vrši korištenjem simulatora ModelSim i usporedbom rezultata sa referentnim modelima. Obavlja se funkcijska (*eng. behavioral*, RTL) i vremenska (*eng. timesim*, PPR) simulacija. Referentni modeli su izrađeni u programskom jeziku Matlab. Referentni modeli uključuju idealni model (aritmetika pomičnog zareza dvostruke preciznosti) i bitovno jednaki model (*eng. bit-exact*) koji u potpunosti simulira sklopovlje (simulacija svakog bita).

Ispitivanje je važan dio procesa razvoja sklopovlja, pa je dizajn sustava za pretipkavanje prilagođen modularizacijom. Izgrađeno je mnoštvo funkcijski nezavisnih cjelina – modula, koje je moguće nezavisno simulirati (ispitati). Povezivanjem modula moguće je obaviti integracijsko testiranje, sve do najviše razine (simulacija cijelog sustava).

Za ispitivanje su razvijene ispitne datoteke (ModelSim „do“ datoteke), Matlab m-skripte za pripremu i analizu podataka, te VHDL ispitna okruženja. Većina modularnih komponenata ima svoje VHDL ispitno okruženje, ModelSim „do“ datoteke, a po potrebi i Matlab skripte za pripremu i vizualizaciju referentnih podataka i podataka dobivenih simulacijom.

Datoteke i pokretanje simulacije

VHDL ispitna okruženja imaju tipičan naziv:
NazivKomponente_tb.vhd

ModelSim „do“ datoteke za pokretanje RTL (funkcijske) simulacije imaju tipičan naziv:
NazivKomponente_tb_rtl.do

ModelSim „do“ datoteke za pokretanje PPR (vremenske) simulacije imaju tipičan naziv:
NazivKomponente_tb_ppr.do

Matlab m-skripte za pripremu referentnih rezultata imaju tipičan naziv:
NazivKomponente_tb.m

Matlab m-skripte za iscrtavanje rezultata imaju tipičan naziv:
NazivKomponente_tb_plot.m

Proces simulacije uključuje korake u navedenom redoslijedu:

- pokretanje Matlab skripte za pripremu referentnih rezultata, ako postoji:
NazivKomponente_tb.m
- pokretanje ModelSim „do“ datoteke:
 - RTL simulacija: pokrenuti:
NazivKomponente_tb_rtl.do
 - PPR simulacija: pokrenuti izgradnju vremenskog modela NazivKomponente_timesim.vhd u Xilinx ISE, a zatim pokrenuti:
NazivKomponente_tb_ppr.do
- pokretanje Matlab skripte za vizualizaciju rezultata, ako postoji:
NazivKomponente_tb_plot.m

Potrebno je obratiti pozornost na ModelSim „do“ datoteku, koja generira pogrešku u slučaju neispravne vrijednosti dobivene simulacijom sklopovlja. Iznimka su neka ispitivanja kod kojih se pogreška uočava vizualizacijom rezultata Matlab skriptom, ili nije definirana numerička mjera ispravnosti rezultata (primjerice ocjena ispravnosti odziva PID regulatora promatranjem grafa).

Uz opisane ModelSim „do“ datoteke za pokretanje RTL ili PPR simulacije, prisutne su i datoteke:

- za prevođenje VHDL modela:
NazivKomponente_compile_rtl.do,
NazivKomponente_compile_ppr.do
- za iscrtavanje valnih oblika:
NazivKomponente_wave_rtl.do,
NazivKomponente_wave_ppr.do

Kako bi bilo moguće brzo testiranje svih pojedinačnih komponenata sustava, kao i sustava u cjelini, napisan je **slijed** jediničnih testova (*eng. unit tests*). Slijed testova se sastoji od svih pojedinačnih testova komponenata, organiziran u redoslijed od najjednostavnijih atomarnih komponenata (koje se ispituju prve), prema sve složenijim, do vršnog dizajna.

Matlab m-skripta za pripremu slijeda jediničnih testova:
UnitTest_System.m

ModelSim „do“ datoteka za pokretanje slijeda jediničnih testova:

- RTL test: UnitTest_System_rtl.do
- PPR test: UnitTest_System_ppr.do

Nakon pokretanja ModelSim „do“ datoteke sa slijedom jediničnih testova, moguća je pojava pogreške. Prikazana pogreška je na najnižoj komponenti u hijerarhiji, zbog organizacije redoslijeda jediničnih testova.

Prije pokretanja PPR slijeda jediničnih testova, za svaku od komponenata u slijedu testova (od najnižih u hijerarhiji do vršne) je potrebno izgraditi vremenski simulacijski model (NazivKomponente_timesim.vhd) korištenjem Xilinx ISE alata.

Vršne komponente dizajna

Vršna komponenta sustava za pretipkavanje je naziva `Interpolator.vhd`. Alternativna vršna komponenta sustava za pretipkavanje je `InterpolatorNR.vhd`, a u svemu je identična izvornoj komponenti interpolatora, osim u sustavu za regulaciju, koji je izvorno onemogućen u ovoj inačici komponente. Komponenta se upotrebljava za bitovno jednaku simulaciju pretipkavanja sa idealno zadanim omjerom frekvencija pretipkavanja, bez uporabe regulatora.

Vršna komponenta `all_design.vhd` se koristi kod ispitivanja na sklopovlju. Sadržava sustav za interpolaciju i dodatnu logiku za rad na razvojnoj ploči.

Vršna komponenta `FractionalDelayFilter.vhd` sadrži filtar frakcionalnog kašnjenja, ispitan u nastavku.

Ispitivanje filtra frakcionalnog kašnjenja

Prethodno opisani filtar frakcionalnog kašnjenja je ispitan dovođenjem signala definiranog u datoteci `ulazniSignal.m` (sinusni signal, uz prigušenje drugim, sporim sinusnim signalom). Korištena je frekvencija $\omega = 0.2\pi$ za izvorni sinusni signal. Točke interpolacije (dx) su odabrane slučajnim odabirom.

Redoslijed izvođenja:

1. `FractionalDelayFilter_tb.m`
2. `FractionalDelayFilter_tb_rtl.do`
3. `FractionalDelayFilter_tb_plot.m`

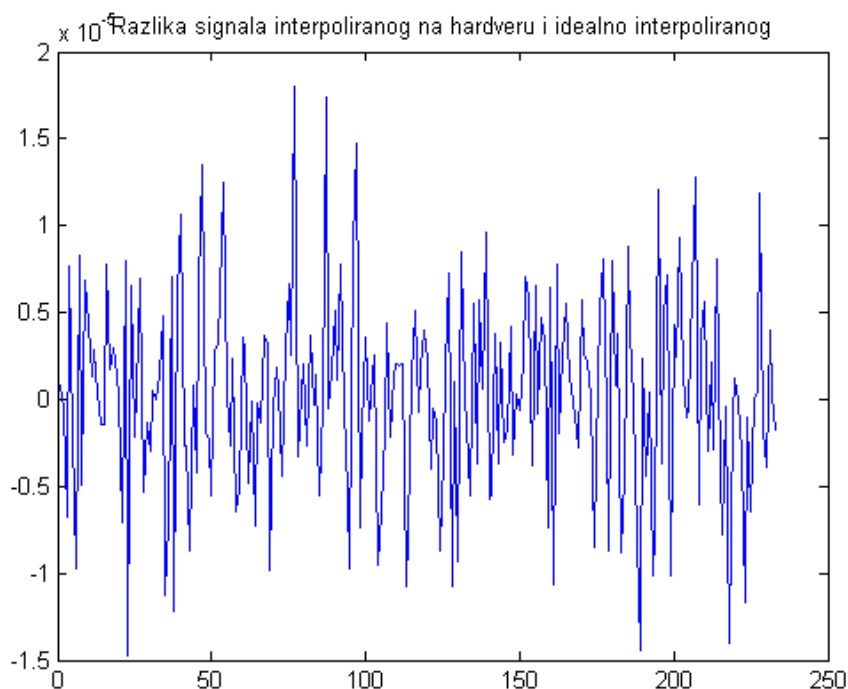
Simulacijom su dobiveni sljedeći podaci:

Maksimalna razlika signala interpoliranog na sklopovlju i interpoliranog bitovno jednakim referentnim modelom:

0

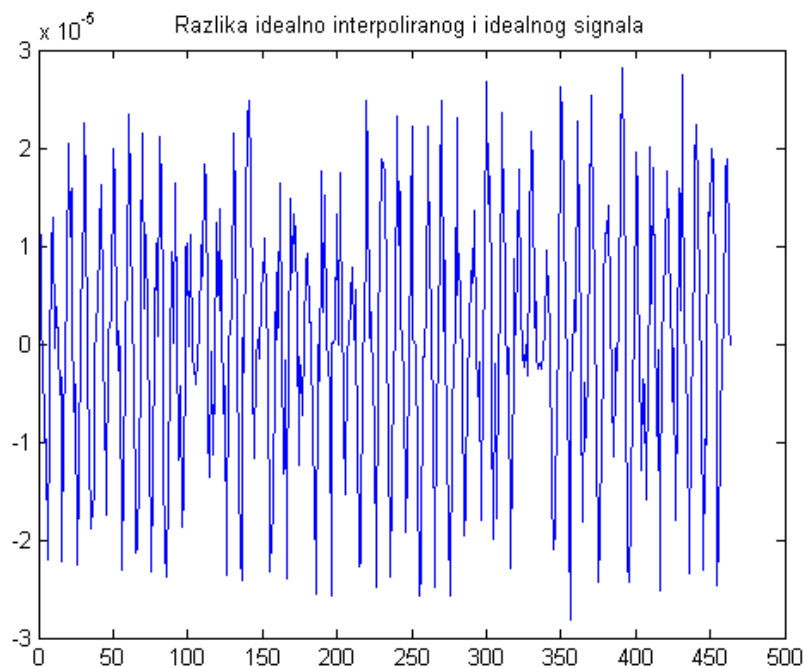
Maksimalna razlika signala interpoliranog na sklopovlju i interpoliranog idealnim referentnim modelom (slika 4.1):

1.8022e-005



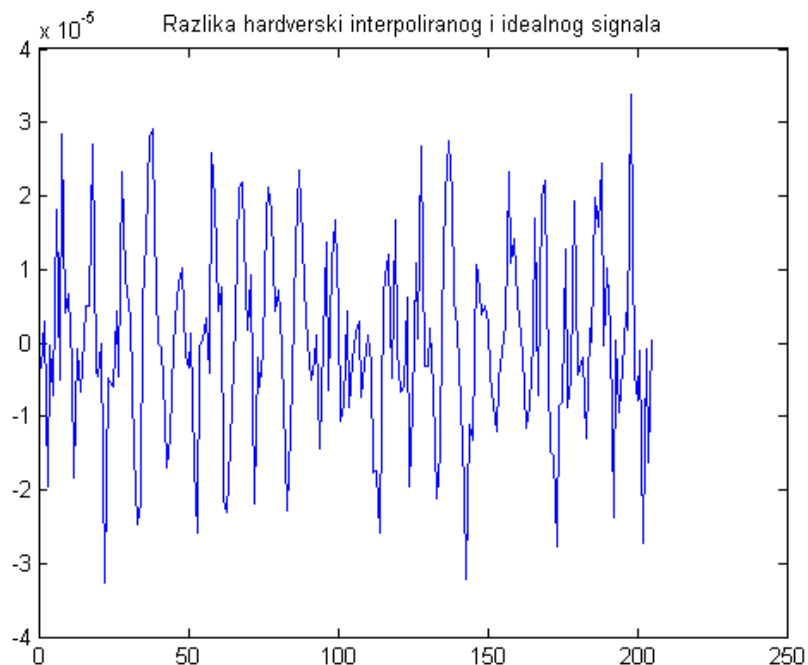
Slika 4.1: Razlika signala interpoliranog na sklopovlju i interpoliranog idealnim referentnim modelom

Maksimalna razlika signala interpoliranog idealnim referentnim modelom i stvarnog signala (matematički model) (slika 4.2):
2.8151e-005



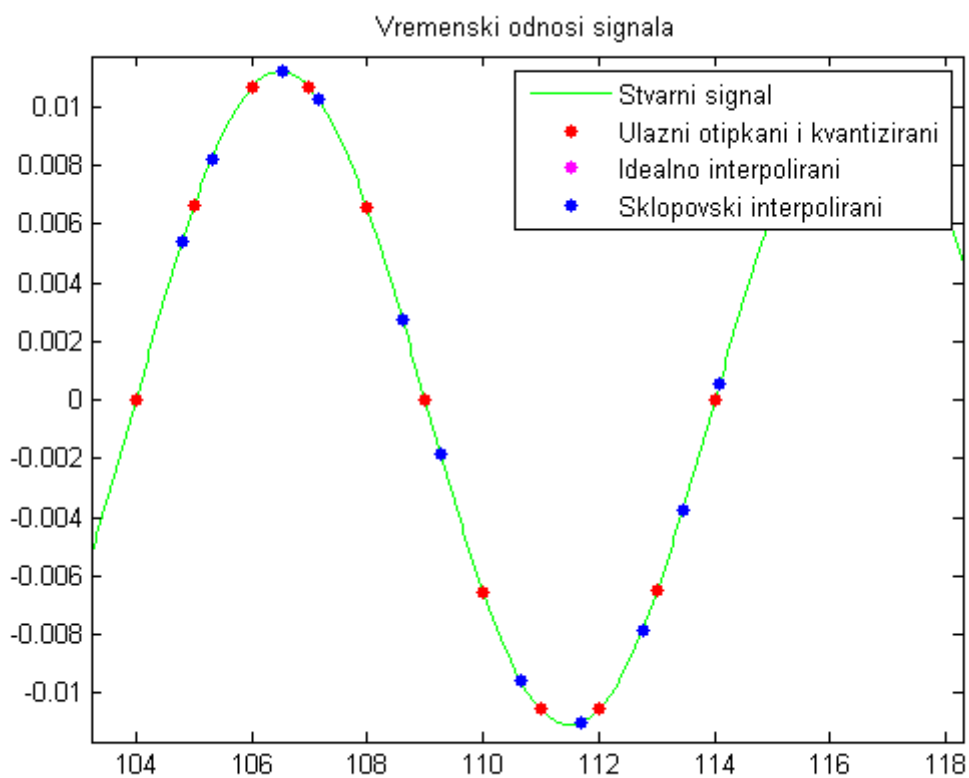
Slika 4.2: Razlika matematičkog modela signala i signala interpoliranog idealnim referentnim modelom

Maksimalna razlika signala interpoliranog na sklopovlju i stvarnog signala (matematički model) (slika 4.3):
3.3776e-005



Slika 4.3: Razlika matematičkog modela signala i signala interpoliranog sklopovljem

Slika 4.4 prikazuje vremenski odnos svih signala. Zelenom linijom je označen matematički model signala. Crvene točke predstavljaju ekvidistantni, otipkani i kvantizirani ulazni signal. Plave točke reprezentiraju sklopovljem interpolirane uzorke, u slučajno odabranim trenutcima. Ružičaste točke se nalaze iza plavih točaka i nisu vidljive bez povećanja prikaza, a reprezentiraju idealno interpolirani signal.



Slika 4.4: Vremenski odnosi svih signala dobivenih filtrom frakcionalnog kašnjenja

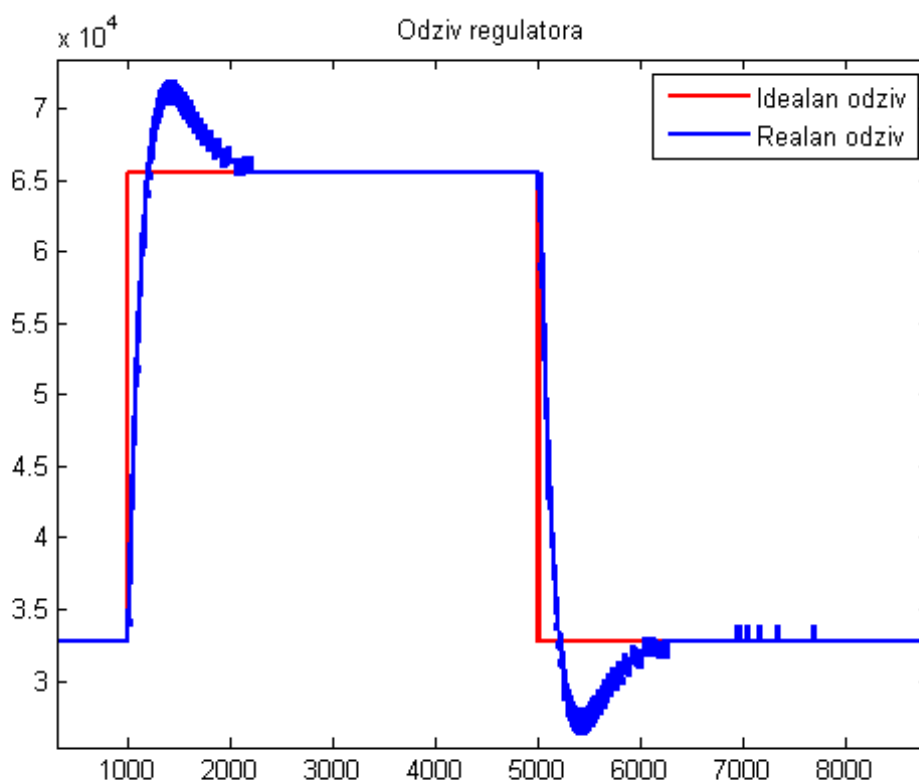
Ispitivanje PID regulacijskog kruga

U nastavku je ispitan PID regulator sustava. Regulator je pobuđen skokovitom pobudom. Regulator održava popunjenost FIFO međuspremnik na 50% popunjenosti (256 uzoraka). Korišten je inicijalni omjer pretipkavanja 1:4 (povećanje broja uzoraka), uz skokovitu promjenu na omjer 1:2. Izlaz reprezentira regulirani pribroj izlaznom adresnom pokazivaču. PID konstante su određene iterativnim ručnim postupkom ([8], *Manual tuning*).

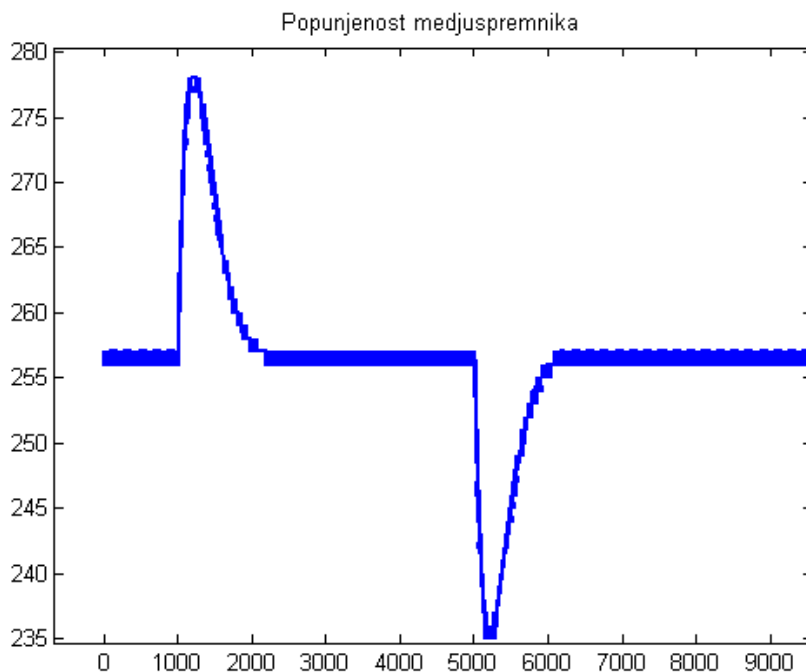
Redoslijed izvođenja:

1. Regulator_tb_rt1.do
2. Regulator_tb_plot.m

Slika 4.5 prikazuje odziv PID regulatora na skokovitu pobudu uz konstante $P = 1100$, $I = 3$, $D = 0$, a slika 4.6 popunjenost međuspremnik za jednak iznos konstanti.



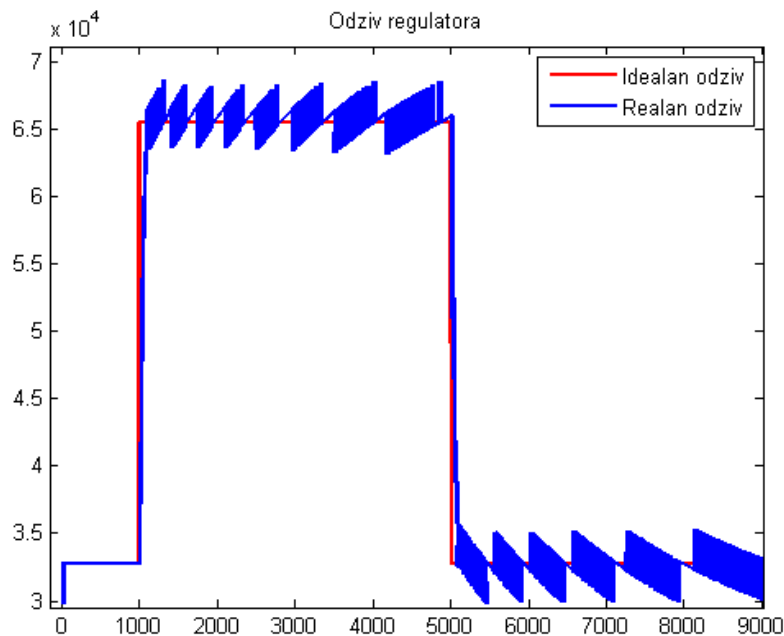
Slika 4.5: Odziv PID regulatora uz konstante $P = 1100$, $I = 3$, $D = 0$.



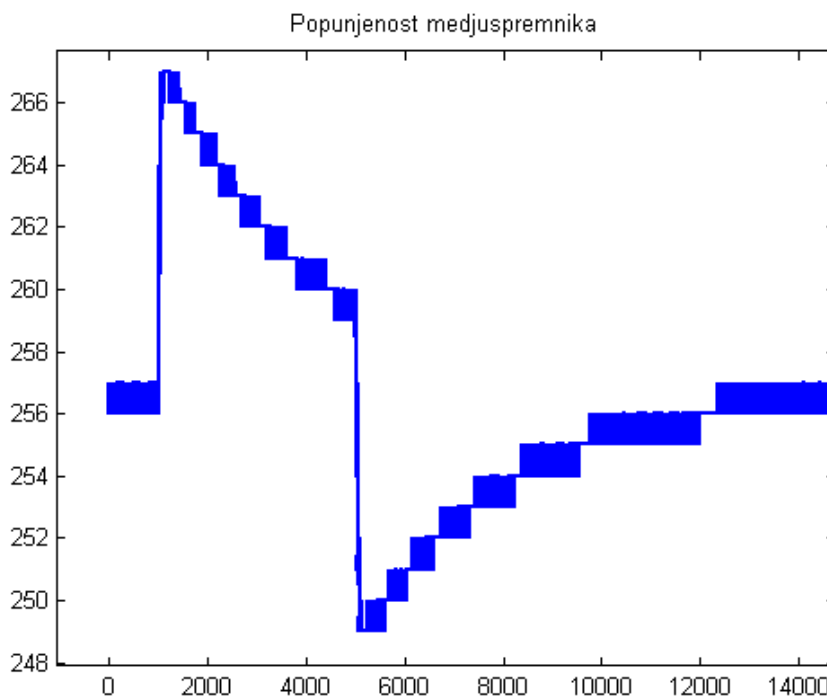
Slika 4.6: Popunjenost međuspremnika uz konstante $P = 1100$, $I = 3$, $D = 0$.

Vidljivo je da vrlo brza regulacija sprječava odstupanje popunjenosti međuspremnika za više od 25 uzoraka. Osim nadvišenja (*eng. overshot*) nema dodatnih oscilacija u regulaciji. Problem nadvišenja je teško u potpunosti otkloniti zbog regulacijskog kašnjenja od 10 uzoraka. Regulacijsko nadvišenje iznosi približno 500. Regulator je vidljivo unaprjeđenje regulatora opisanog u [4], koji ima regulacijsko nadvišenje približno 1000, uz prisutne oscilacije. Regulator u ovom sustavu je unaprjeđen ugradnjom množila za sve konstante regulatora (regulator opisan u [4] može obavljati množenje samo sa PID konstantama koje su potencije broja 2). Ugradnja množila za regulacijske konstante se pokazala izrazito bitna, te je precizna regulacija P-pojačanja vrlo povoljno djelovala na regulaciju. Vidljivo je da iznos D-pojačanja nije toliko bitan za ovu primjenu (D-pojačanje se koristi za povećanje strmine odziva, ali uzrokuje nadvišenja, koja su problematična i bez uporabe ovog regulacijskog člana). Ugradnja množila je uzrokovala produljenje regulacijskog cjevovoda na 10 perioda takta čitanja (regulator opisan u [4] ima regulacijski cjevovod od 9 perioda takta), no vidljivo je da novi regulator postiže bolje performanse preciznim odabirom konstante P, unatoč dužem cjevovodu. Moguć je i alternativni odabir konstanti.

Slika 4.7 prikazuje odziv PID regulatora na skokovitu pobudu uz konstante $P = 3000$, $I = 1$, $D = 0$, a slika 4.8 popunjenost međuspremnik za jednak iznos konstanti.



Slika 4.7: Odziv PID regulatora uz konstante $P = 3000$, $I = 1$, $D = 0$.



Slika 4.8: Popunjenost međuspremnik uz konstante $P = 3000$, $I = 1$, $D = 0$.

Vidljivo je da još agresivniji odabir P-konstante umanjuje regulacijsko nadvišenje. Cijena koju je potrebno platiti je nestabilniji odziv regulatora, čak i u stacionarnom stanju. Međutim, prikazana skokovita pobuda je izazvala odstupanje popunjenosti međuspremnika od samo 10 uzoraka od 50% popunjenosti. To pokazuje da iznimno agresivna regulacija može kompenzirati i veće skokovite promjene frekvencija otipkavanja, bez prepunjenja ili podpunjenja međuspremnika dubine 512 uzoraka.

Daljnje povećanje P konstante nije preporučeno zbog uvođenja dodatne nestabilnosti i oscilacija.

Tipična uporaba PID regulatora može uključivati sustav za adaptivno određivanje regulacijskih koeficijenata, ovisno o stanju procesa, no uz uslozňjavanje implementacije.

Ispitivanje sustava za pretipkavanje

Sustav za pretipkavanje je ispitan dovođenjem signala definiranog u datoteci `ulazniSignal.m` (sinusni signal, uz prigušenje drugim, sporim sinusnim signalom). Korištena je frekvencija $\omega = 0.2\pi$ za izvorni sinusni signal. Sustav je ispitan pretipkavanjem ulaznog signala na četiri puta veći broj uzoraka (*eng. upsampling* 1:4). Vršni dizajn korišten za ispitivanje je `InterpolatorNR`, koji je identičan `Interpolator` dizajnu, ali ne koristi ugrađenu regulaciju popunjenosti (u simulacijskom okruženju su svi signali vremenskog vođenja potpuno deterministički, a regulacija onemogućava bitovno jednaku verifikaciju dizajna).

Redosljed izvođenja:

1. `InterpolatorNR_tb.m`
2. `Interpolator_tb_rtl.do`
3. `Interpolator_tb_plot.m`

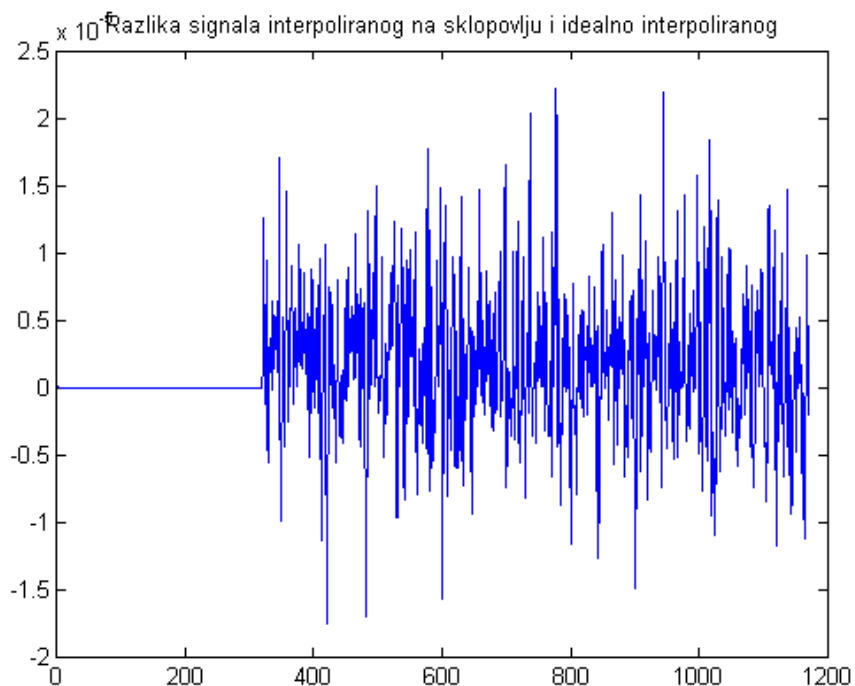
Simulacijom su dobiveni sljedeći podaci:

Maksimalna razlika signala interpoliranog na sklopovlju i interpoliranog bitovno jednakim referentnim modelom:

0

Maksimalna razlika signala interpoliranog na sklopovlju i interpoliranog idealnim referentnim modelom (slika 4.9):

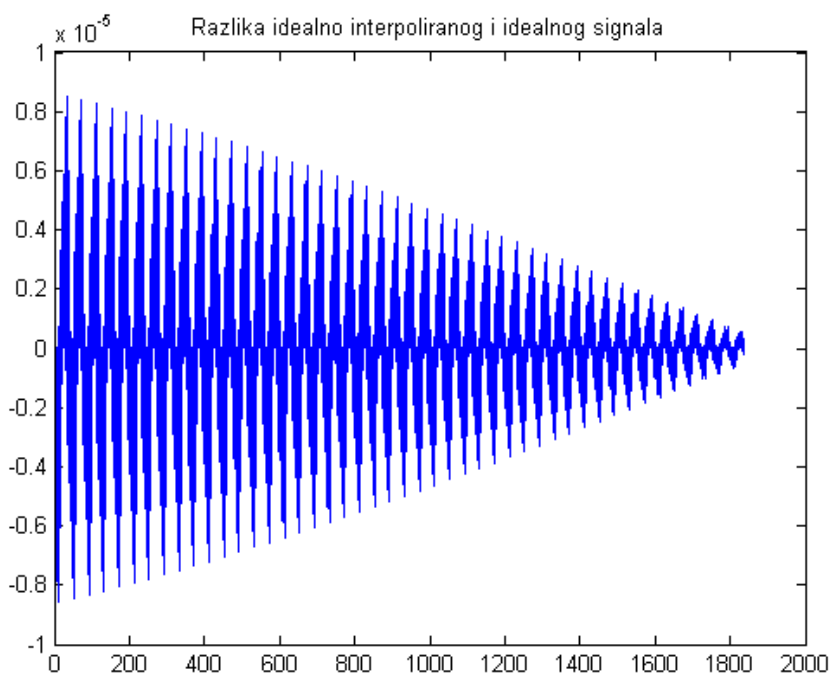
2.2155e-005



Slika 4.9: Razlika signala interpoliranog na sklopovlju i interpoliranog idealnim referentnim modelom

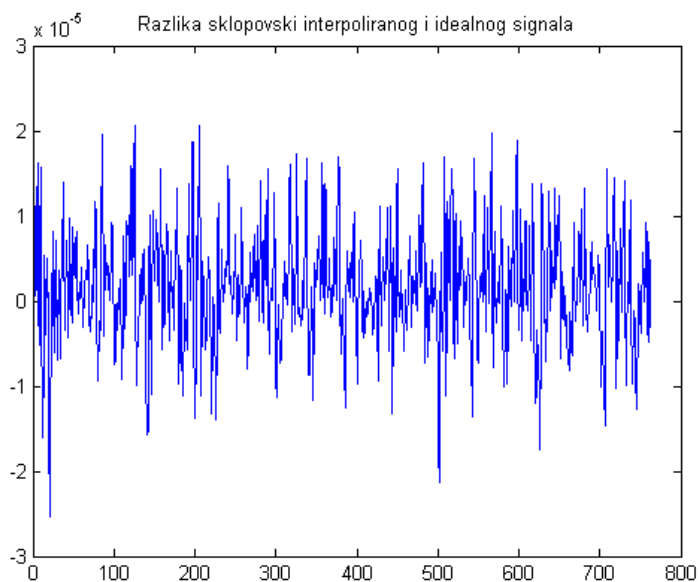
Maksimalna razlika signala interpoliranog idealnim referentnim modelom i stvarnog signala (matematički model) (slika 4.10):

8.5746e-006



Slika 4.10: Razlika matematičkog modela signala i signala interpoliranog idealnim referentnim modelom

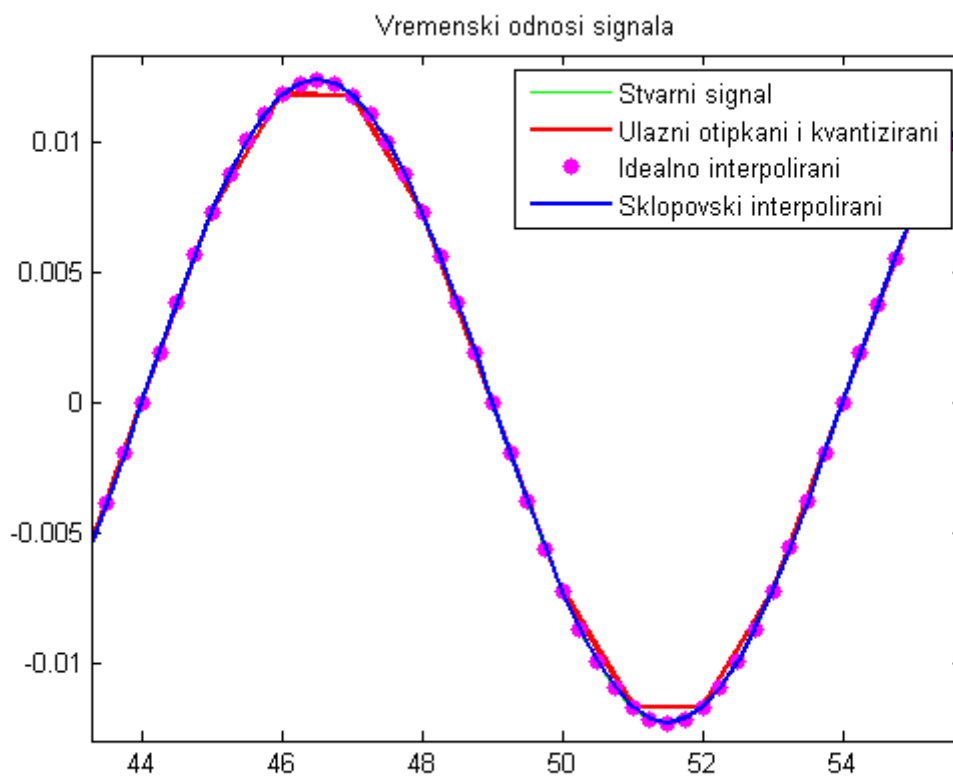
Maksimalna razlika signala interpoliranog na sklopovlju i stvarnog signala (matematički model) (slika 4.11):
 $2.5309e-005$



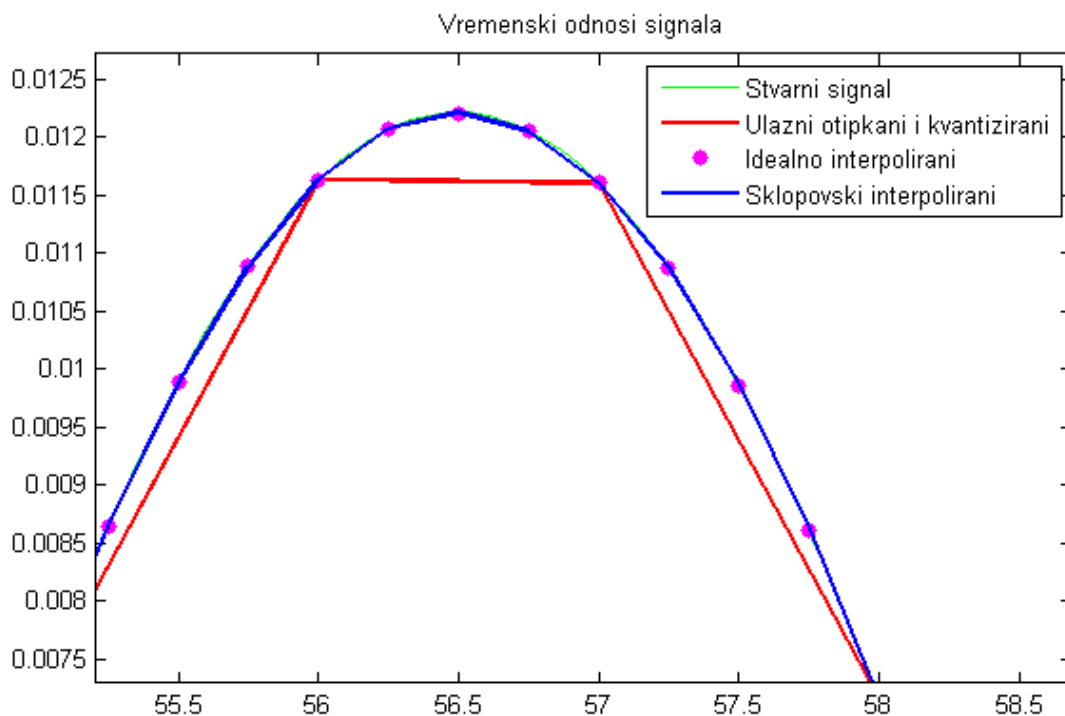
Slika 4.11: Razlika matematičkog modela signala i signala interpoliranog sklopovljem

Slika 4.12 prikazuje vremenski odnos svih signala, a slika 4.13 uvećani prikaz svih signala. Zelenom linijom je označen matematički model signala. Crvena linija reprezentira otipkani i kvantizirani ulazni signal. Plava linija reprezentira

sklopovljem interpolirane uzorke, u omjeru pretipkavanja 1:4. Ružičaste točke se nalaze iza plave linije, a reprezentiraju idealno interpolirani signal.



Slika 4.12: Vremenski odnosi svih signala dobivenih sustavom za pretipkavanje



Slika 4.13: Vremenski odnosi svih signala dobivenih sustavom za pretipkavanje – povećani prikaz

Uočen je tipičan red pogreške 10^{-5} , dok je tipična pogreška sustava opisanog u [4] iznosila 10^{-4} (no potreban je oprez pri usporedbi zbog nešto veće dinamike signala kod [4]).

Uočena je i pravilnost pogreške kod usporedbe idealno interpoliranog signala sa matematičkim modelom signala. Pogreška nije značajna (10^{-6}) pa je umanjen značaj pravilnosti, a one se pojavljuju zbog veće preciznosti idealnog sustava i ekvidistantnog otipkavanja (interpolacije) signala. Pogreška je minimalna na svakom 4. uzorku (nema interpolacije, na izlaz se prenosi ulazni uzorak otipkanog i kvantiziranog signala), a maksimalna između izvornih uzoraka, pa je vidljiva pravilna oscilatorna pogreška.

Ispitivanje ovisnosti točnosti interpolacije o broju FIR stupnjeva

Proveden je eksperiment za koji su prethodno postavljene teoretske osnove: kako ovisi točnost interpolacije o broju stupnjeva FIR pred filtra (broju akauzalnih uzoraka).

Uspoređen je rezultat sklopovlja sa referentnim modelima (idealnim i bitovno jednakim interpolatorima), te u odnosu na matematički model signala, kao i odnos idealnog modela interpolatora i matematičkog modela signala.

Rezultati su prikazani u tablici 4.1.

Za ispitivanje je korišten filter frakcionalnog kašnjenja, no sa zadanom konstantnom točkom evaluacije $\Delta x = 0.5$ kako bi rezultati bili više deterministički. Korišten je ulazni sinusni prigušeni signal (opisan u datoteci `ulazniSignal.m`), osnovne frekvencije $\omega = 0.0002\pi$.

Tablica 4.1: Odnosi najvećih pogrešaka ovisno o broju akauzalnih uzoraka

Pogreška:	Broj akauzalnih uzoraka			
	2	4	6	8
Sklopovlje u odnosu na bitovno jednaki model	0	0	0	0
Sklopovlje u odnosu na idealni model	2.0162e-005	1.7735e-005	1.7737e-005	1.7737e-005
Idealni model u odnosu na matematički model signala	4.8622e-006	4.7608e-006	4.7190e-006	4.6811e-006
Sklopovlje u odnosu na matematički model signala	1.8902e-005	1.6817e-005	1.6817e-005	1.6817e-005

Vidljivo je da generalno postoji ovisnost pogreške o broju akauzalnih uzoraka: povećanjem akauzalnog zahvata povećava se točnost. Međutim vidljivo je da poboljšanje idealnog modela interpolatora u odnosu na matematički model signala može slijediti povećanje broja akauzalnih uzoraka, no sklopovska implementacija ne može. Sklopovlje nakon četvrtog akauzalnog uzorka daljnjim povećanjem akauzalnog zahvata ne dobiva na točnosti interpolacije. Rezultat je očekivan, jer su korišteni ulazni i izlazni signali 12-bitovne preciznosti. Prikazana pogreška je maksimalna na nizu konačnog trajanja. Bez kvantizacije ulaznih i izlaznih signala, pad **srednje** pogreške bi trebao biti 11.44dB za svaki dodatan akauzalni uzorak ([2]).

Za bolje performanse sklopovske implementacije potrebno je povećati dinamiku aritmetike sklopovlja.

Ispitivanje na sklopovlju

Sustav za pretipkavanje je ispitan na Xilinx Virtex-4 FPGA sklopovlju (razvojni sustav DS-KIT-4VLX25LC^[14], slika 4.14), uz korištenjem analognog modula (Memec P160^[15], slika 4.15) za prijenos signala iz analogne domene u digitalnu i obrnuto. Izlazi i ulazi su promatrani na osciloskopu. Ulazni sinusni signal je pretipkan sa nizom proizvoljnih omjera pretipkavanja. Za ispitivanje sustava za pretipkavanje korišten je vršni dizajn `all_design.vhd`.

Dizajn `all_design.vhd` vrši sljedeće prilagodbe signala za ciljno sklopovlje:

- inverzija i monostabil 1s za sve LED diode (aktivna niska razina)
- inverzija i sinkronizacija reset tipkala (aktivna niska razina)
- pomak ulaznog signala (sa AD – nije u NBC formatu već ima odmak)
- pomak izlaznih signala (na DA – nije u NBC formatu već ima odmak)

Dizajn `all_design` iskorištava interni oscilator 100MHz za sintezu takta za pisanje u sustav za pretipkavanje (`clkWr`). Pri vrhu datoteke `all_design.vhd` je definirana konstanta:

```
constant clk100divide_limit : std_logic_vector(27 downto 0) := x"0000096";
```

Konstanta određuje iznos takta `clkWr`, koji je jednak:

$$100\text{MHz} / (2 * 96\text{hex}) = 333.333\text{kHz} \quad (4.1)$$

To je takt otipkavanja ulaznog signala.

Otipkavanje se vrši AD#2 AD pretvornik na P160 modulu frekvencijom 333.333 kHz. Ulazni analogni signal se dovodi na IN2 ulaz P160 modula. Pri ispitivanju obratiti pozornost na Shannon-Nyquistov teorem kao i na dinamiku ulaznog signala (maksimalno 1Vpp, uz 50Ω terminaciju na ulazu AD pretvornika).

Izlaz iz DAC#1 DA pretvornika daje izvorni signal, samo propušten kroz FIFO međuspremnik za kašnjenje (da kašnjenje približno odgovara kašnjenju sustava za interpolaciju). Na izlazu OUT#1 P160 modula je moguće promatrati „digitalizirani“ signal ekvivalentan ulaznom u sustav za interpolaciju.

Izlaz iz DAC#2 DA pretvornika daje pretipkani signal (izlazni signal iz sustava za pretipkavanje). Dinamika izlaznog signala je smanjena sa izvornih 18 bita na razlučivost DAC (12 bita).

Takt clkWr je sintetiziran iz kristalnog oscilatora na ploči i određen je konstantom `clk100divite_limit`, a takt clkRd (takt otipkavanja izlaznog - interpoliranog signala) se dovodi eksterno, izravno na priključak CLKIN#1 P160 modula.

Pri ispitivanju obratiti pozornost na oblik (pravokutni) i dinamiku ulaznog signala takta (minimalno 0V, maksimalno 2.5V, uz 50Ω terminaciju na ulazu CLKIN#1).

Postavke kratkospojnika P160 analognog modula:

- JP5: 1-2 zatvoren
- JP4: oba otvorena
- JP1: zatvoren
- JP2: zatvoren
- JP6: nebitno

Konstanta u datoteci `all_design.vhd`

```
constant delay_size : std_logic_vector(9 downto 0);
```

određuje kašnjenje prijenosa signala sa ulaza IN#2 na OUT#1 (FIFO međuspremnik za kašnjenje).

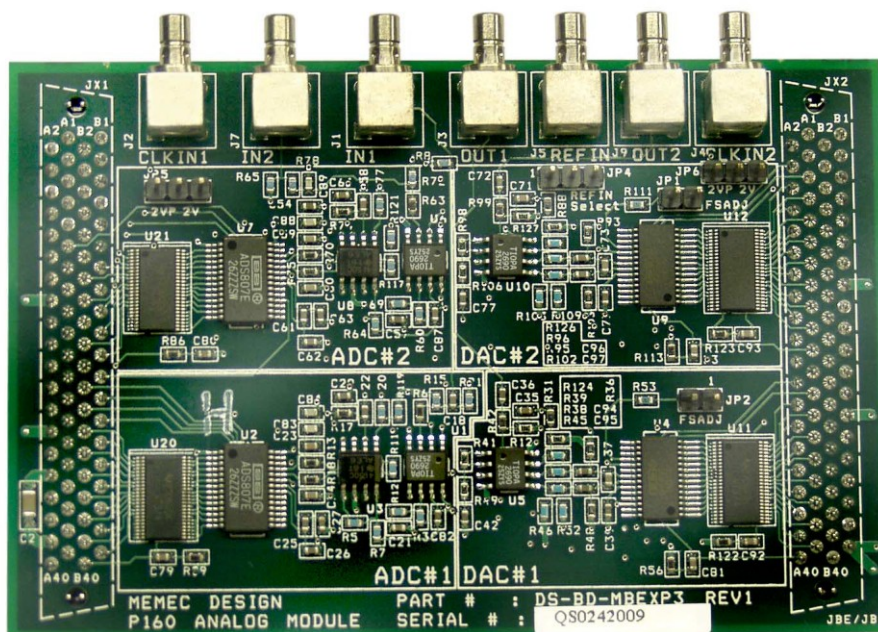
Na Virtex-4 razvojnoj ploči se koristi sljedeće korisničko sučelje:

- Tipkalo SW5 – reset sustava (pritisnuti)
- Dioda LED1 – podpunjenje međuspremnika sustava za interpolaciju (ako svijetli)
- Dioda LED2 – prepunjenje međuspremnika sustava za interpolaciju (ako svijetli)
- Dioda LED4 – prevelika dinamika ulaznog analognog signala na AD#2 IN#2 (ako svijetli)

Sve LED diode su upravljane monostabilom sa trajanjem astabilnog stanja od približno 1s.



Slika 4.14: Memec razvojna ploča (Virtex 4), [14]



Slika 4.15: Memec P160 analogni modul, [15]

5. Zaključak

Kratka kauzalna kubična splajn interpolacija je podesiva interpolacijska tehnika koja omogućava jednostavnu prilagodbu ciljanoj primjeni. Odabir točnije interpolacije (izrazito na niskim frekvencijama) podrazumijeva utrošak veće količine sklopovskih resursa, te veće kašnjenje rada sustava.

Izgrađen je modularan, podesiv sustav za pretipkavanje temeljen na ovom tipu interpolacije, za ciljnu arhitekturu Xilinx FPGA.

Popularna FPGA arhitektura omogućava implementaciju sustava po prihvatljivim cijenama za niski proizvodni obujam (i akademske primjene), a ostavlja mogućnost relativno jednostavnih prilagodbi u slučaju izrade čvrsto ožičenih sklopova za veći obujam proizvodnje (primjerice MPGA kopije FPGA sklopovlja).

Modularnost izgrađenog sustava je izrazito bitna sa stanovišta apstrakcije, analize rada sustava, ponovne uporabljivosti i zamjene pojedinih komponenti, te ispitljivosti.

Lagana zamjenjivost podsustava za generiranje koeficijenata kubičnog polinoma, kao i podesivost postojećeg generatora (baziranog na kratkoj kauzalnoj kubičnoj splajn interpolaciji) omogućuju laganu primjenu sustava za pretipkavanje na širok spektar problema.

Ispitivanje je bitan i neizostavan faktor svakog razvojnog procesa. Modularni dizajn sustava sa jasno definiranim sučeljima omogućuje ispitivanje sustava komponentu po komponentu (jedinično testiranje), kao i integracijsko testiranje, što je bitan korak pri efikasnoj dijagnostici i lociranju problema.

Negativan efekt razlaganja (modularizacije) sustava može biti gubitak performansi, ili utrošak veće količine resursa nego što je minimalno potrebno (kako bi se definirala jasna i konzistentna sučelja, ili zbog dupliciranja funkcijskih blokova u različitim jedinicama). Problem se javlja rijetko, posebice zbog dupliciranja funkcionalnosti (oslanjanje na inteligentne optimizacijske mogućnosti alata za sintezu sklopovlja). Problem je općenito puno više prisutniji u softverskim

razvojnim procesima, gdje objektni i strukturirani dizajn često donosi gubitak performansi.

Posebnu pozornost je potrebno posvetiti dizajnu sustava kod kojih se javlja više signala vremenskog vođenja, koji su međusobno asinkroni. Dodirne točke dvije domene signala vremenskog vođenja neizbježno stvaraju probleme pri prijenosu podataka (uz pojavu metastabilnosti).

Generalna preporuka je izbjegavanje uporabe višestrukih asinkronih signala vremenskog vođenja (sinkronizacija svih asinkronih ulazno-izlaznih signala na zajednički signal vremenskog vođenja, po potrebi visoke frekvencije).

Opisani sustav za pretipkavanje je specifičan slučaj kod kojeg bi uvođenje trećeg (globalnog) signala vremenskog vođenja degradiralo performanse, pošto bi ulazni i izlazni signal vremenskog vođenja morali biti niže frekvencije (jer bi se otipkavali pomoću globalnog signala vremenskog vođenja).

Zbog očuvanja maksimalne frekvencije signala vremenskog vođenja, nužno je složene operacije (poput regulatora popunjenosti međuspremnik) rastaviti na faze i izvoditi pomoću cjevovoda. To usporava ukupnu brzinu reakcije komponenata. Kod regulatora usporavanje brzine reakcije pridonosi pojavi oscilacija u regulatorskom krugu, koje je teško otkloniti u potpunosti (za opći slučaj), čak i uz uporabu potpunog PID regulatora.

Luka Karlović

6. Popis oznaka i kratica

AD	Analog to Digital
CLB	Configurable Logic Block
D	Derivacijski
DA	Digital to Analog
eng.	engleski
FIFO	First In First Out
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
I	Integracijski
IIR	Infinite Impulse Response
LSB	Least Significant Bit
MAC	Multiply And Accumulate
MIMO	Multiple Input Multiple Output
MOMS	Maximum Order Minimum Support
MPGA	Mask/Metal Programmable Gate Array
MSB	Most Significant Bit
NBC	Natural Binary Code
npr.	na primjer
P	Proporcionalni
PID	Proporcionalno-Integracijsko-Derivacijski
PPR	Post Place and Route
RAM	Random Access Memory
ROM	Read Only Memory
RTL	Register Transfer Level
SIMO	Single Input Multiple Output
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit

7. Literatura

- [1] Petrović Sebastian: *Sustav za interpolaciju*, seminarski rad – NARDS, FER-ZESOI, siječanj 2010.
- [2] Petrinović Davor: *Short output filter for causal cubic spline interpolation*, FER-ZESOI, listopad 2009. (nije još objavljeno)
- [3] Petrinović Davor: *Causal Cubic Splines: Formulations, Interpolation Properties and Implementations*, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 56, NO. 11, NOVEMBER 2008
- [4] Karlović Luka: *Nadogradnja i ispitivanje sustava za pretipkavanje signala*, tehnička dokumentacija – Diplomski projekt, FER-ZESOI, 2009.
- [5] Haus Mario: *MOMS interpolator*, seminarski rad – NARDS, FER-ZESOI, 2009.
- [6] *Interpolation*, Wikipedia
<http://en.wikipedia.org/wiki/Interpolation>
(10.06.2010.)
- [7] *B-spline*, Wikipedia
<http://en.wikipedia.org/wiki/B-spline>
(10.06.2010.)
- [8] *PID controller*, Wikipedia
http://en.wikipedia.org/wiki/PID_controller
(11.06.2010.)
- [9] *Spartan-3 FPGA Family: Complete Data Sheet*, Xilinx Inc., DS099, 25.06.2008.
http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf
(30.03.2009.)
- [10] *Using Embedded Multipliers in Spartan-3 FPGAs*, Xilinx Inc., XAPP467 (v1.1), svibanj 2003.
http://www.xilinx.com/support/documentation/application_notes/xapp467.pdf
(20.04.2008.)
- [11] *Using Look-Up Tables as Distributed RAM in Spartan-3 Generation FPGAs*, Xilinx Inc., XAPP464 (v2.0), ožujak 2005.
http://www.xilinx.com/support/documentation/application_notes/xapp464.pdf
(30.03.2009.)
- [12] *Using Block RAM in Spartan-3 Generation FPGAs*, Xilinx Inc., XAPP463 (v2.0) ožujak 2005.
http://www.xilinx.com/support/documentation/application_notes/xapp463.pdf
(30.03.2009.)
- [13] *Virtex-4 FPGA User Guide*, UG070 v2.6, Xilinx Inc., 01.12.2008.
http://www.xilinx.com/support/documentation/user_guides/ug070.pdf
(01.12.2009.)
- [14] *Virtex-4™ LC Development Board User's Guide*, Revision 1 (DS-KIT-4VLX25LC), Memec Inc., 26.01.2005.
- [15] *Memec P160 Analog Module User Guide*, PN# DS-MANUAL-ANALOG1 v1.2, Memec Inc., srpanj 2003.

Dodatak A: hrvatski naslov, ključne riječi, sažetak

Naslov

Sustav za pretipkavanje signala temeljen na kratkoj kauzalnoj kubičnoj splajn interpolaciji

Ključne riječi

pretipkavanje, signal, interpolacija, kratka, kauzalna, kubična, splajn, FPGA, Xilinx, VHDL, simulacija, Matlab, model

Sažetak

Tema rada je implementacija sustava za pretipkavanje signala temeljena na kratkoj kauzalnoj kubičnoj splajn interpolaciji. Sustav omogućava pretipkavanje ulaznog otipkanog signala na proizvoljnu izlaznu frekvenciju otipkavanja. Ugrađeni regulator omogućava varijabilan odnos ulazne i izlazne frekvencije otipkavanja. Ciljna platforma za implementaciju sustava je Xilinx Spartan-3/Virtex-4 FPGA platforma. Sustav je implementiran u jeziku za opis sklopovlja VHDL. Ispravnost rada sustava je ispitana usporedbom sa referentnim Matlab modelima (idealnim modelom sa aritmetikom pomičnog zareza dvostruke preciznosti, i realnim bitovno-jednakim modelom), te na stvarnom FPGA sklopovlju uz korištenje AD i DA pretvornika. Dokument također opisuje i kratku teoretsku osnovu korištene metode interpolacije.

Dodatak B: engleski naslov, ključne riječi, sažetak

Title

Signal resampling system based on short causal cubic spline interpolation

Keywords

resampling, signal, interpolation, short, causal, cubic, spline, FPGA, Xilinx, VHDL, simulation, Matlab, model

Summary

Main topic of this document is implementation of signal resampling system based on short causal cubic spline interpolation. System can resample sampled input signal to any other sample rate. Embedded regulator supports variable input-to-output sample rate ratio. Target platform for system implementation is Xilinx Spartan-3/Virtex-4 FPGA. System is implemented in hardware description language VHDL. System is verified by comparison with Matlab reference models (ideal double precision floating point arithmetic model, and real bit-exact model), and on real FPGA hardware with AD and DA converters. Document also describes short theoretical basis of used interpolation method.