

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1848

**FPGA implementacija vrlo strmih
FIR filtara dobivenih tehnikom
maskiranja frekvencijske
karakteristike**

Luka Lavrnja

Zagreb, srpanj 2019.

Umjesto ove stranice umetnite izvornik Vašeg rada.

Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.

Zahvaljujem se ekipi iz Ericsson Nikola Tesla; doc dr. sc. Goranu Molnaru, mag. ing. Aljoši Dudarinu i mag. ing. Hrvoju Butini na izdvojenom vremenu i pomoći prilikom izrade ovog diplomskog rada.

Također zahvaljujem se i mentoru prof. dr. sc. Davoru Petrinoviću što mi je omogućio izradu diplomskog rada u Ericssonu Nikoli Tesli.

Najviše se zahvaljujem kolegama koji su kao i ja radili diplomski rad u Ericsson Nikola Tesla na svakodnevnim konzultacijama i raspravama.

SADRŽAJ

1. Uvod	1
2. Tehnika maskiranja frekvencijske karakteristike (FRM)	2
2.1. Dizajn uskopojasnih filtara	2
2.2. Dizajn filtara proizvoljne pojasne širine	4
3. MATLAB model FRM filtra	9
3.1. Primjer širokopojasnog FRM filtra	10
3.2. Primjer uskopojasnog FRM filtra	13
4. Fizički ostvariv model FRM filtra	14
4.1. Struktura fizički ostvarivog modela FRM filtra	14
4.2. Opis fizički ostvarivih modela pojedinih filtara	16
4.3. Strukturni model FRM filtra	17
5. Uvodavanje implementiranog filtra na razvojnom sustavu ZedBoard	21
5.1. Komunikacija između PL dijela i Xillinuxa	22
5.2. Priprema podataka za analizu i rezultati	23
6. Zaključak	26
Literatura	27

1. Uvod

FIR (engl. Finite-impulse-response) filtri s linearnom fazom imaju mnoge prednosti. Neke od njih su zajamčena stabilnost i da nema izobličenja faze. Poznato je kako je duljina impulsnog odziva filtra obrnuto proporcionalna širini prijelaznog područja. Zbog toga, složenost postaje previsoka za filtre oštrog prijelaza između područja propuštanja i područja gušenja. Time je i implementacija na sklopovlju uvelike otežana [1]. U literaturi je predloženo nekoliko metoda za smanjenje složenosti oštrog FIR filtra. Jedna od najuspješnijih tehnika za sintezu vrlo strmog filtra je tehnika maskiranja frekvencijske karakteristike, odnosno FRM (engl. Frequency-response-masking) koja je prvo predložena u [2].

Tehnika maskiranja frekvencijske karakteristike je primjenjiva i za uskopojasne i za širokopojasne filtre. Glavna prednost FRM tehnike je da filter ima vrlo oskudni vektor koeficijenata, odnosno puno koeficijenata je jednako nuli. Stoga je njegova aritmetička složenost vrlo niska, iako su njegova duljina i kašnjenje malo duži od onih u konvencionalnim izvedbama. Time fazno linearni FRM filtri uspješno smanjuju složenost realizacije. S druge strane, njihovo grupno kašnjenje je čak i veće od izravnog oblika FIR filtra s istom točnošću.

Detaljan opis potrebne matematičke pozadine FRM tehnike se nalazi u 2. poglavlju ovog rada, VHDL implementacija modela u 3. poglavlju, te testiranje na sklopovlju i konačni rezultati u 4. poglavlju.

2. Tehnika maskiranja frekvencijske karakteristike (FRM)

2.1. Dizajn uskopojasnih filtara

Dizajn uskopojasnog FRM filtra počinje s prototipnim filtrom prijenosne funkcije $H_a(z)$ širine prijelaznog područja Δ_a i njegovom frekvencijskom karakteristikom $H_a(e^{j\omega})$ prikazanom na slici 2.1a. Zamjenom svakog kašnjenja filtra s M kašnjenjem dobivamo periodični filter, čija prijenosna funkcija postaje

$$H_b(z) = H_a(z^M) \quad (2.1)$$

i frekvencijska karakteristika

$$H_b(e^{j\omega}) = H_a(e^{jM\omega}) \quad (2.2)$$

kao što je prikazano na slici 2.1b. Kako bi se uklonile spektralne slike novog periodičnog filtra $H_b(e^{j\omega})$, on se maskira s $H_c(e^{j\omega})$ prikazanim na slici 2.1c koji filtrira periodični filter tako da ostane samo spektralna slika na najnižim frekvencijama. Ukupni frekvencijski odziv se dobije kaskadom periodičnog filtra i njegovog maskirnog

$$H_d(e^{j\omega}) = H_b(e^{j\omega})H_c(e^{j\omega}) \quad (2.3)$$

prikazan na slici 2.1d. Širina prijelaznog područja $H_d(e^{j\omega})$ je Δ_a/M . Ako se $H_b(e^{j\omega})$ maskira s $H_e(e^{j\omega})$ prikazanim na slici 2.1e, dobije se konačni frekvencijski odziv

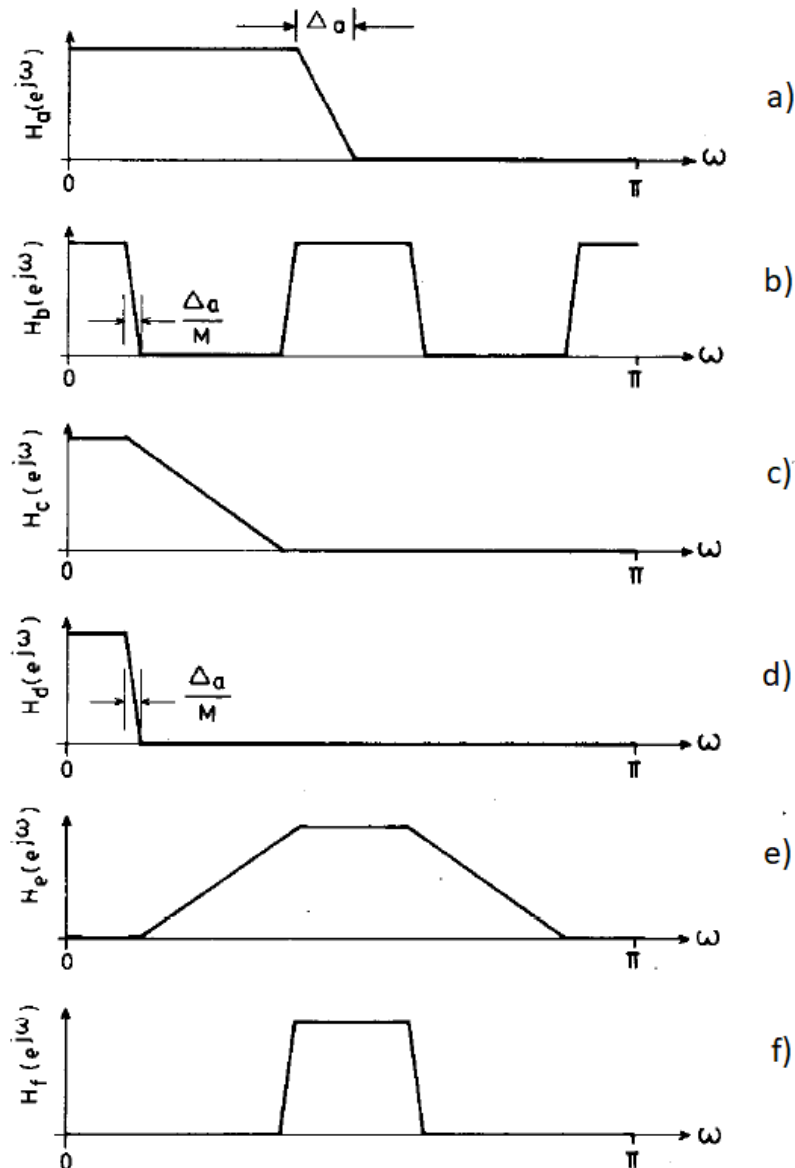
$$H_f(e^{j\omega}) = H_b(e^{j\omega})H_e(e^{j\omega}) \quad (2.4)$$

čija je širina prijelaznog područja opet Δ_a/M . Drugim riječima, prijenosna funkcija FRM uskopojasnog filtra glasi

$$H(z) = H_a(z^M) \cdot H_{ma}(z) \quad (2.5)$$

gdje je H_{ma} frekvencijski odziv maskirnog filtra.

Ovaj pristup omogućava implementaciju vrlo oštrog prijelaznog područja konačnog filtra pomoću filtara s puno širim prijelaznim područjem. FRM tehnika ipak ima nedostataka. Iako se zamjenom svakog kašnjenja $H_a(z)$ s M kašnjenjem smanjuje širina prijelaznog područja, istovremeno se smanjuje i širina područja propuštanja. Upravo iz ovog razloga je ova metoda prikladna samo za dizajn uskopojasnih filtara. U sljedećem potpoglavlju je opisana metoda za dizajn filtara proizvoljne pojasne širine.



Slika 2.1: Jednostavna FRM metoda za dobivanje vrlo strmih uskopojasnih filtara, preuzeto iz [2]

2.2. Dizajn filtara proizvoljne pojasne širine

Problem smanjenja područja propuštanja naveden na kraju prethodnog potpoglavlja se rješava uvođenjem još dodatna 2 filtra: komplementarni filter periodičnog filtra te njegov maskirni filter. Komplementarni filter ima obrnute spektralne slike od periodičnog, a njihovi maskirni filteri drže područje propuštanja do željene širine.

Dva fazno linearna filtra F_a i F_c se smatraju komplementarnim parom ako vrijedi

$$|F_a(e^{j\omega}) + F_c(e^{j\omega})| = 1 \quad (2.6)$$

gdje su $F_a(e^{j\omega})$ i $F_c(e^{j\omega})$ frekvencijski odzivi F_a i F_c . Za FIR filter reda N s linearnom razom može se izraziti frekvencijski odziv kao [2]

$$F_a(e^{j\omega}) = e^{-j((N-1)/2)\omega} R(\omega) \quad (2.7)$$

gdje je $R(\omega)$ trigonometrijska funkcija od ω .

Frekvencijski odziv komplementarnog filtra tada glasi

$$F_c(e^{j\omega}) = e^{-j((N-1)/2)\omega} (1 - R(\omega)) \quad (2.8)$$

Ako z-transformacija funkcije F_a glasi $F_a(z)$, tada je z-transformacija funkcije F_c jednaka

$$F_c(z) = z^{-j((N-1)/2)} - F_a(z) \quad (2.9)$$

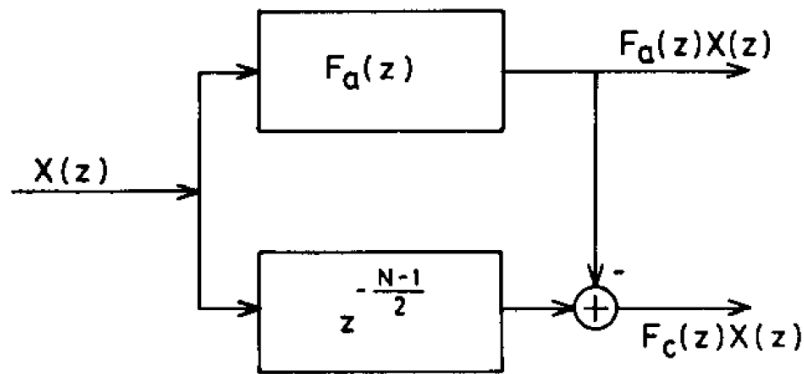
gdje se F_c implementira kao razlika izlaza F_a i zakašnjelog ulaza $X(z)$ kao što je prikazano na slici 2.2.

Pretpostavimo niskopropusni filter F_a s $R(\omega)$ prikazan na slici 2.3a. Njegovo područje propuštanja je ograničeno s frekvencijom Θ , a područje gušenja frekvencijom ϕ . Frekvencijski odziv njegovog komplementarnog filtra je prikazan na slici 2.3b. Stvaraju se dva filtra, F'_a i F'_c zamjenom svakog kašnjenja F_a i F_c s M kašnjenjem, odnosno dobiva se periodični i njegov komplementarni filter. Frekvencijski odzivi F'_a i F'_c za neparni N su $F'_a(e^{j\omega})$ i $F'_c(e^{j\omega})$ gdje je

$$F'_a(e^{j\omega}) = F_a(e^{jM\omega}) \quad (2.10)$$

i

$$F'_c(e^{j\omega}) = F_c(e^{jM\omega}) \quad (2.11)$$



Slika 2.2: Realizacija komplementarnog filtra, preuzeto iz [2]

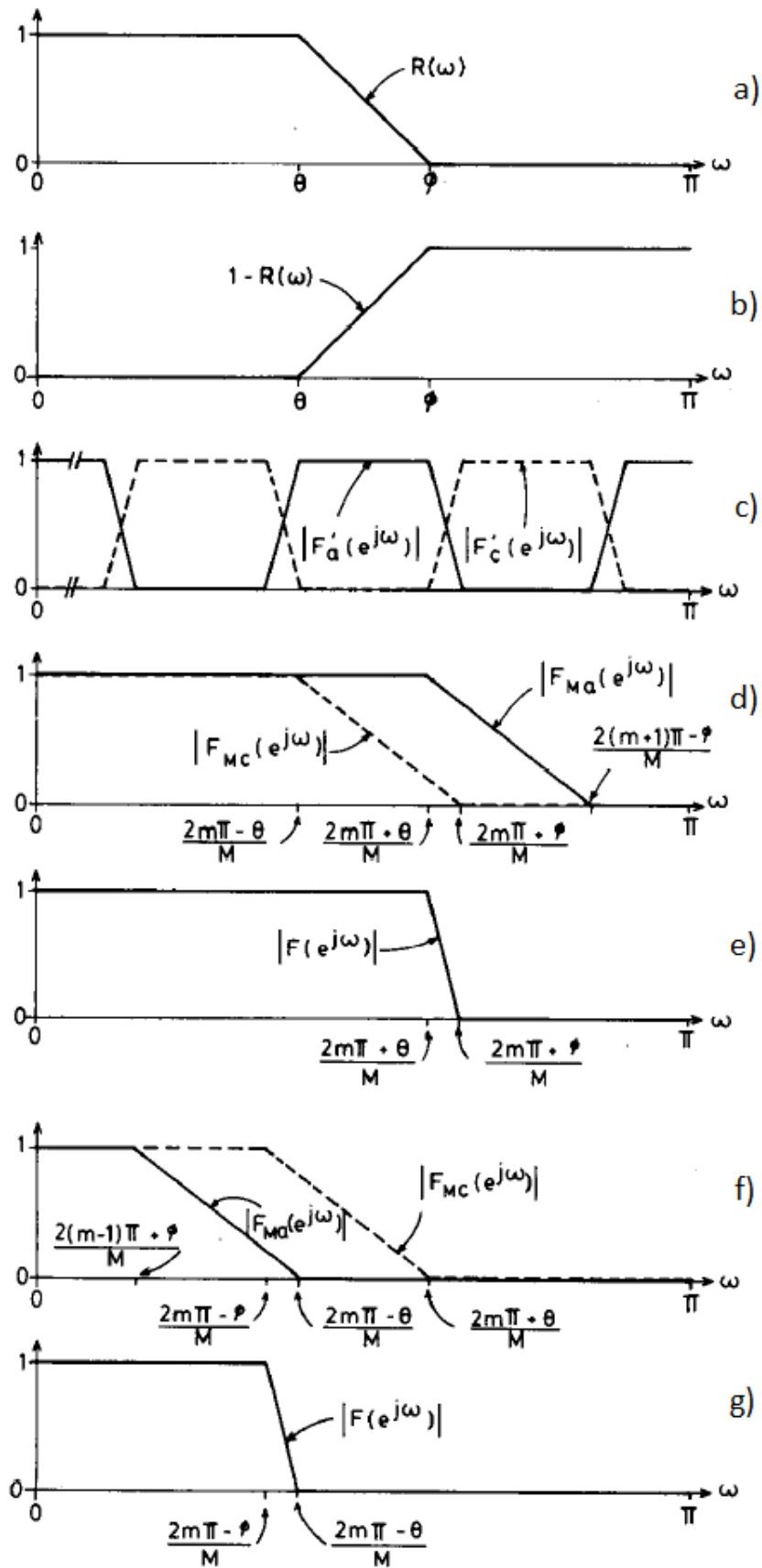
kao što je prikazano na slici 2.3c. Dva maskirna filtra F_{Ma} i F_{Mc} s frekvencijskim odzivima $F_{Ma}(e^{j\omega})$ i $F_{Mc}(e^{j\omega})$ prikazani na slici 2.3d se koriste za maskiranje $F'_a(e^{j\omega})$ i $F'_c(e^{j\omega})$. Ako se zbroji kaskada periodičnog i njegovog maskirnog filtra s kaskadom komplementarnog i njegovog maskirnog filtra, odnosno izlazi F_{Ma} i F_{Mc} kao na slici 2.4, dobije se frekvencijski odziv konačnog filtra F prikazan na slici 2.3e.

Drugim riječima, prijenosna funkcija širokopojasnog FRM filtra glasi

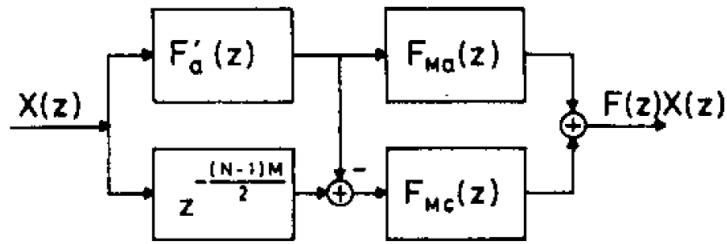
$$H(z) = H_a(z^M) \cdot H_{ma}(z) + H_c(z^M) \cdot H_{mc}(z) \quad (2.12)$$

gdje je

$$H_c(z^M) = z^{\frac{(N-1)M}{2}} \quad (2.13)$$



Slika 2.3: Maskiranje frekvencijske karakteristike komplementarnih filtara za implementaciju širokopojasnih vrlo strmih filtara, preuzeto iz [2]



Slika 2.4: Struktura filtra implementiranog tehnikom maskiranja frekvencijske karakteristike, preuzeto iz [2]

Ako je ω_p granična frekvencija područja propuštanja i ω_s granična frekvencija područja gušenja filtra F , može se zapisati kao [2]

$$\omega_p = \frac{2m\pi + \Theta}{M} \quad (2.14a)$$

$$\omega_s = \frac{2m\pi + \phi}{M} \quad (2.14b)$$

gdje je m cijeli broj manji od M . Treba uzeti u obzir da grupno kašnjenje F_{Ma} i F_{Mc} mora biti jednako što znači da duljine oba filtra moraju biti ili parne ili neparne i da se vodeće kašnjenje mora dodati ili F_{Ma} ili F_{Mc} ako je potrebno kako bi se izjednačilo njihovo grupno kašnjenje. Također, za izbjegavanje kašnjenja od pola uzorka, $(N-1)M$ mora biti paran.

Na slici 2.3e se vidi da je frekvencijska karakteristika F određena uglavnom karakteristikom F'_a , što se još naziva prvim slučajem. Ako su frekvencijske karakteristike F_{Ma} i F_{Mc} kao na slici 2.3f, onda će frekvencijska karakteristika F biti određena uglavnom karakteristikom F'_c što je prikazano na slici 2.3g. Ovo je drugi slučaj. Granične frekvencije kod drugog slučaja se zapisuju kao [2]

$$\omega_p = \frac{2m\pi - \Theta}{M} \quad (2.15a)$$

$$\omega_s = \frac{2m\pi - \phi}{M} \quad (2.15b)$$

Za pronalazak graničnih frekvencija maskirnih filtera moraju se ω_p i ω_s izraziti preko m , M , Θ i ϕ . Faktor frekvencijskog skaliranja M se bira tako da se cjelokupna kompleksnost filtra minimizira. Također, mora se očuvati

$$0 < \Theta < \phi < \pi. \quad (2.16)$$

Da (2.14a) i (2.14b) očuvaju (2.16), zapisuje se [2]

$$m = \left\lfloor \frac{\omega_p M}{2\pi} \right\rfloor \quad (2.17a)$$

$$\Theta = \omega_p M - 2m\pi \quad (2.17b)$$

$$\phi = \omega_s M - 2m\pi \quad (2.17c)$$

gdje je m najveći cijeli broj manji od $\frac{\omega_p M}{2\pi}$. Da (2.15a) i (2.15b) očuvaju (2.16), zapisuje se [2]

$$m = \left\lceil \frac{\omega_s M}{2\pi} \right\rceil \quad (2.18a)$$

$$\Theta = 2m\pi - \omega_s M \quad (2.18b)$$

$$\phi = 2m\pi - \omega_p M \quad (2.18c)$$

gdje je m najmanji cijeli broj veći od $\frac{\omega_s M}{2\pi}$. Za gotovo bilo koji izbor ω_p , ω_s i M samo jedan slučaj od prikazana dva će pronaći Θ i ϕ takve da vrijedi $\phi < \pi$. Granične frekvencije područja propuštanja i gušenja maskirnih filtara su prikazane na 2.3. S obzirom na to da je prijelazno područje filtra $F_a(e^{j\omega})$ jednako $M(\omega_s - \omega_p)$, za dane ω_p i ω_s se povećava i prijelazno područje $F_a(e^{j\omega})$ u skladu s M . Iz tog razloga se smanjuje kompleksnost F_a kako se povećava M . Suma prijelaznih područja $F_{Ma}(e^{j\omega})$ i $F_{Mc}(e^{j\omega})$ je jednaka $1/M$, te se smanjuje povećavanjem M .

3. MATLAB model FRM filtra

U programskom okruženju MATLAB dizajniran je model filtra koji za željene frekvencije područja propuštanja ω_p i gušenja ω_s konačnog filtra, te faktor frekvencijskog skaliranja M određuje frekvencije potrebnog prototipnog filtra i maskirnih filtara. Ovom metodom što se više povećava faktor frekvencijskog skaliranja M to će biti potreban prototipni filtar manjeg reda, no tu se stvara potreba za maskirnim filtrima većeg reda. Primjer ove pojave je prikazan u tablici 3.1 gdje su $\omega_p = 0.3\pi$, $\omega_s = 0.315\pi$.

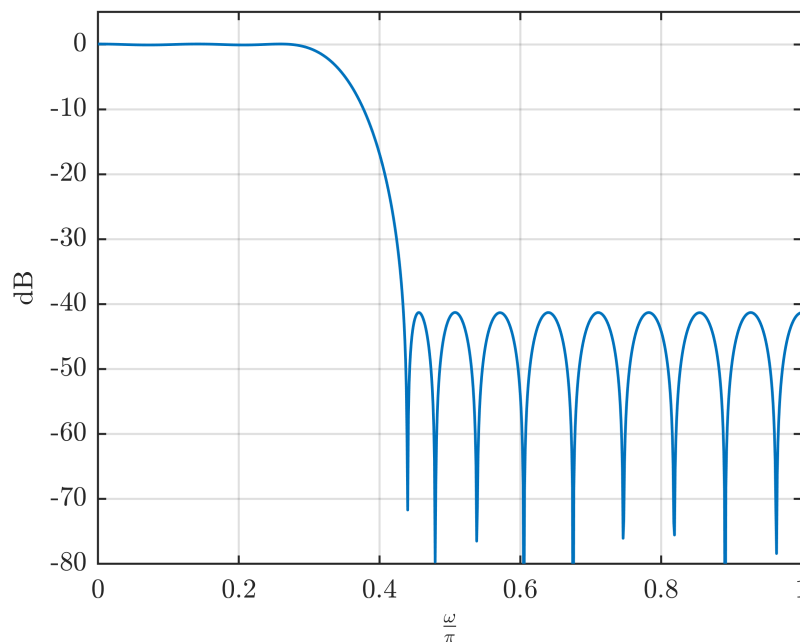
Tablica 3.1: Primjer ovisnosti reda filtra o faktoru frekvencijskog skaliranja

R	N_a	N_{ma}	N_{mc}
4	70	38	10
5	56	20	22
6	48	14	82
7	40	18	98
8	36	32	38
9	32	82	24
10	28	282	22

Također za neke vrijednosti M , ω_p i ω_s postavljaju se "nemogući" zahtjevi. Primjerice, za $\omega_p = 0.174\pi$, $\omega_s = 0.191\pi$ i $M = 11$ prema formulama navedenim u prethodnom potpoglavlju traži se prototipni filtar čije prijelazno područje prelazi raspon $[0, \pi]$. Konkretno, za prvi slučaj se traži filtar s $\Theta = 1.9140\pi$ i $\phi = 2.1010\pi$, a po formulama za drugi slučaj $\Theta = 1.8990\pi$, $\phi = 2.0860\pi$. U ovakvim zahtjevima algoritam ne može pronaći rješenje.

3.1. Primjer širokopojasnog FRM filtra

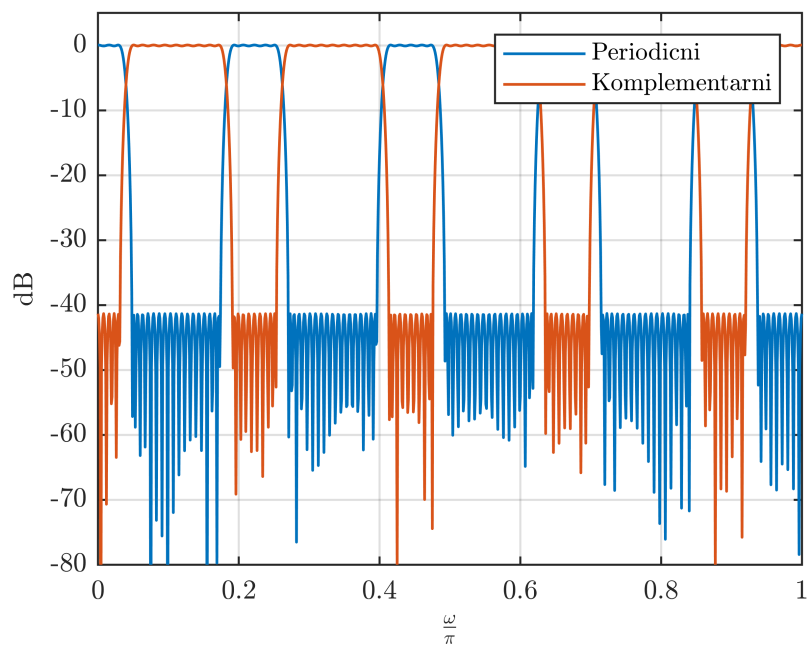
Za prikaz rada algoritma za širokopojasni filtar uzet je slučaj gdje su $\omega_p = 0.174\pi$, $\omega_s = 0.191\pi$ i $M = 9$. Potreban prototipni filtar prema formulama (2.18b) i (2.18c) za drugi slučaj ima frekvenciju propuštanja $\Theta = 0.2810\pi$, te frekvenciju gušenja $\phi = 0.4340\pi$. Dobije se filtar reda $N = 28$, a njegova amplitudno-frekvencijska karakteristika prikazana je na slici 3.1.



Slika 3.1: Amplitudno-frekvencijska karakteristika prototipnog filtra

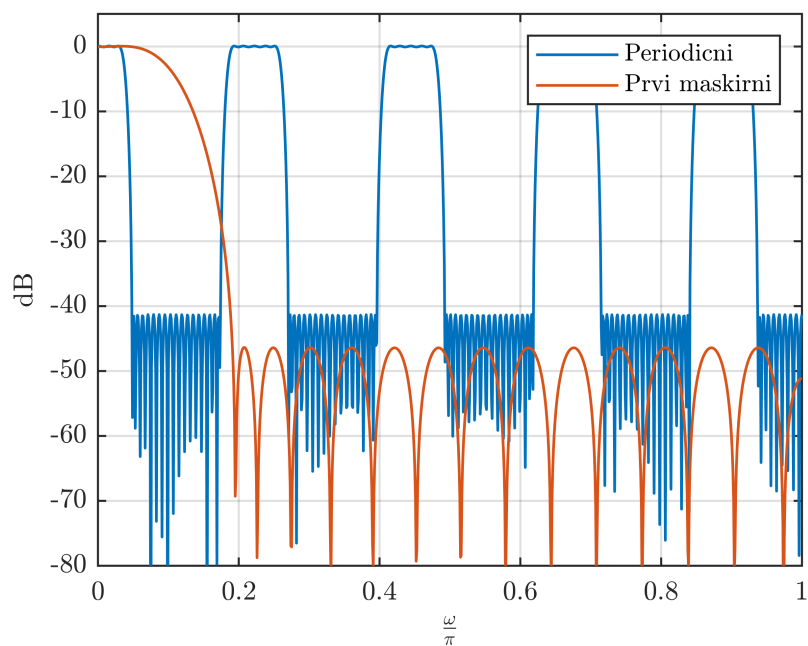
S obzirom da se prototipni filtar proširio s $M - 1$ nula između svakog koeficijenta kako bi se ostvario periodičan filtar, broj koeficijenata komplementarnog filtra postaje M puta veći od prototipnog. Za dobivanje komplementarnog filtra korištena je MATLAB funkcija *grpdelay*[3] te je izračunata razlika između kašnjenja i koeficijenata prototipnog filtra. Ta razlika predstavlja koeficijente komplementarnog filtra. Odnos periodičnog i njegovog komplementarnog filtra prikazan je na slici 3.2.

Potrebni su još njihovi maskirni filtri. Prema formulama opisanim u prethodnom poglavlju na slici 2.3f, granične frekvencije prvog maskirnog filtra za periodični su $\omega_{pMa} = 0.0482\pi$ i $\omega_{sMa} = 0.1910\pi$, a za drugog maskirnog filtra za komplementarni $\omega_{pMc} = 0.1740\pi$, $\omega_{sMc} = 0.2534\pi$. Prvi maskirni filtar je reda $N = 30$, a drugi reda $N = 54$. Da bi se izjednačilo grupno kašnjenje maskirnih filtara, prvi maskirni filtar je proširen nulama kako bi postao također reda $N = 54$. Za jasniju sliku ranije opisanog

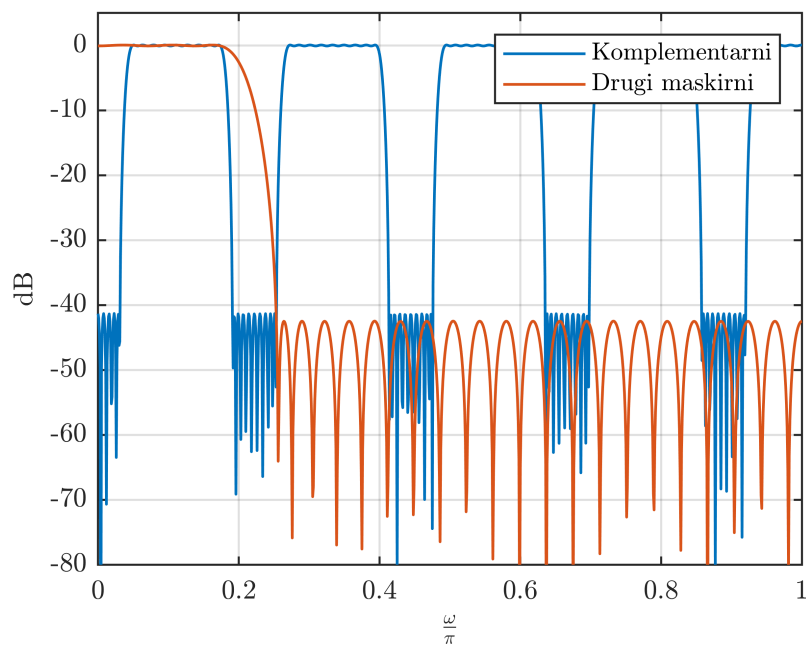


Slika 3.2: Amplitudno-frekvencijske karakteristike komplementarnog i periodičnog filtra

efekta prvog maskirnog filtra na periodični, njihove amplitudno-frekvencijske karakteristike su prikazane na slici 3.3, te komplementarnog i drugog maskirnog filtra na slici 3.4.

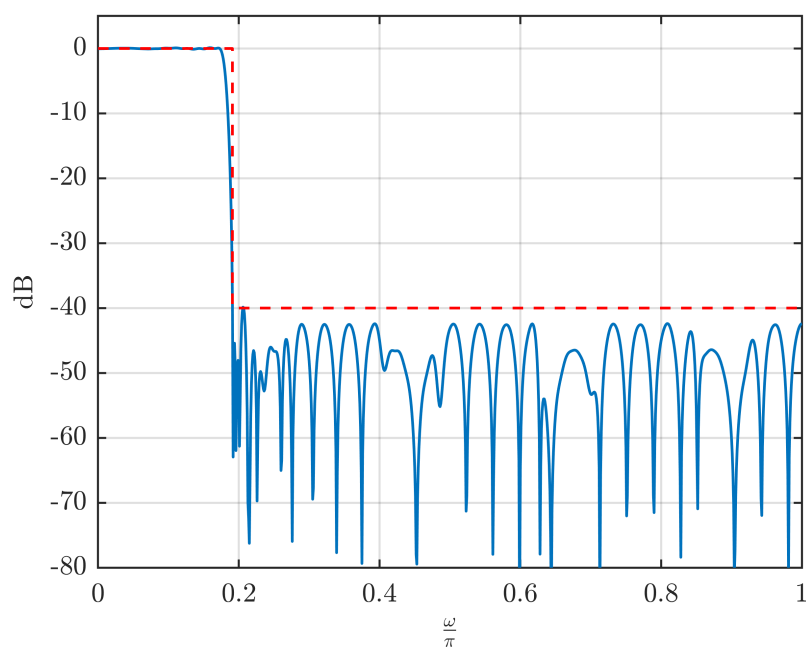


Slika 3.3: Amplitudno-frekvencijske karakteristike periodičnog i prvog maskirnog filtra



Slika 3.4: Amplitudno-frekvencijske karakteristike komplementarnog i drugog maskirnog filtra

Konačni FRM filtar dobije se zbrajanjem kaskade periodičnog i prvog maskirnog filtra s kaskadom komplementarnog i drugog maskirnog filtra. Amplitudno-frekvencijska karakteristika konačnog širokopojasnog filtra prikazana je na slici 3.5.



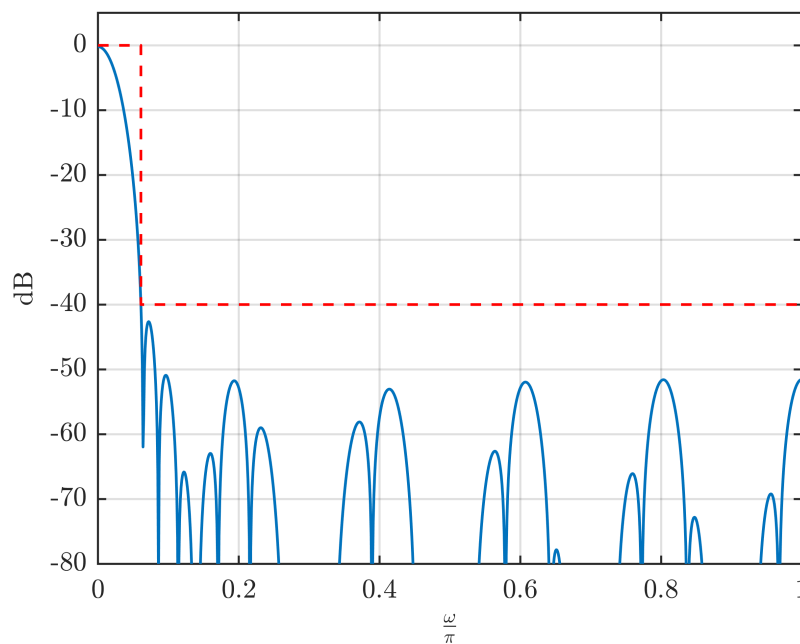
Slika 3.5: Amplitudno-frekvencijska karakteristika širokopojasnog FRM filtra

3.2. Primjer uskopojasnog FRM filtra

Za prikaz rada algoritma za uskopojasni filtar uzet je slučaj gdje su $\omega_p = 0$, $\omega_s = 0.061\pi$ i $M = 10$. Potreban prototipni filtar prema formulama (2.17b) i (2.17c) za prvi slučaj ima frekvenciju propuštanja $\Theta = 0$, te frekvenciju gušenja $\phi = 0.61\pi$. Dobije se filtar reda $N = 4$.

Periodičan filtar se dobiva tako da se koeficijenti prototipnog filtra prošire ubacivanjem $M - 1$ nula između svakog koeficijenta. Potreban je još njegov maskirni filtar čije su granične frekvencije izračunate prema formuli u prethodnom poglavlju na slici 2.3d $\omega_{pMa} = 0$ i $\omega_{sMa} = 0.139\pi$ i red $N_{ma} = 30$.

Konačni uskopojasni FRM filtar čini kaskadu periodičnog i maskirnog filtra. Amplitudno-frekvencijska karakteristika konačnog uskopojasnog filtra je prikazana na slici 3.6.



Slika 3.6: Amplitudno-frekvencijska karakteristika konačnog uskopojasnog FRM filtra

4. Fizički ostvariv model FRM filtra

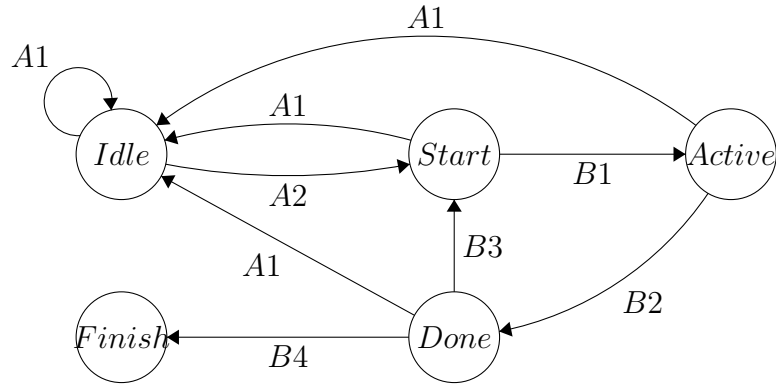
RTL (engl. Register Transfer Level), odnosno fizički ostvariv model FRM filtra, razvijen je korištenjem jezika za opis sklopovlja VHDL (engl. VHSIC Hardware Description Language). Na kraju je model optimiziran za implementaciju na programabilnim logičkim poljima (engl. Field Programmable Arrays, FPGAs). Kao razvojno okruženje korišten je Vivado Design Suite 2018.3. U nastavku je opisana struktura fizički ostvarivog modela FRM filtra.

4.1. Struktura fizički ostvarivog modela FRM filtra

Model je implementiran prema shemi na slici 2.4 iz drugog poglavlja. Svaki FIR filtar ima zadatak provesti konvoluciju ulaznog signala i impulsnog odziva filtra. Cijeli sustav je implementiran kao zbroj kaskada periodičnog filtra i odgovarajućeg maskirnog filtra te komplementarnog filtra i njemu odgovarajućeg maskirnog filtra.

Svi filtri su realizirani kao automati s 5 stanja: *Idle* u koje je sustav inicijalno postavljen te ponovno ulazi kada je resetiran, *Start* kada se svi interni signali i registri pripremaju za početak konvolucije, *Active* u kojem se nalazi sve dok ne završi konvolucija koja traje onoliko ciklusa koliko koeficijentata impulsnog odziva ima sami filtar, *Done* kada konvolucija završi i sustav pušta podatak na izlaz, te *Finished* u koji sustav ulazi kada završi s obradom svih podataka. Dijagram stanja automata je na slici 4.1, a legenda prijelaza stanja je objašnjena u tablici 4.1.

Budući da filtar obavlja uzastopno zbrajanje i množenje, postoji opasnost od preljeva u registru. Da bi se to spriječilo koristi se akumulatorska ekstenzija. Akumulatorska ekstenzija je predznačno proširivanje širine riječi registra za pohranu rezultata kako ne



Slika 4.1: Dijagram stanja automata filtra

Tablica 4.1: Legenda prijelaza automata stanja

Kratica	Objašnjenje
A1	Reset je aktivan
A2	Reset nije aktivan
B1	Početak konvolucije
B2	Kraj konvolucije
B3	Svi ulazni podaci još nisu obrađeni
B4	Svi ulazni podaci su obrađeni

bi došlo do preljeva ni kod najgoreg slučaja. Potreban broj bitova se računa kao

$$AccExt = \left\lceil \log_2 \left(\sum_{h=0}^N |h| \right) \right\rceil \quad (4.1)$$

gdje su h koeficijenti impulsnog odziva, a N red filtra.

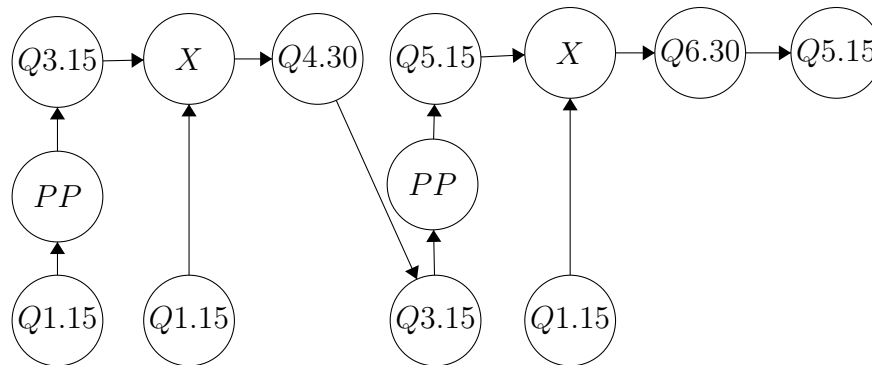
Konačni izlazni podatak je rezultat zbrajanja konvolucija maskirnih filtara. Širina izlazne riječi jednaka je širini ulaznog podatka kojoj je dodan broj bitova kao posljedica dobitka u pojačanju uslijed obrade (engl. Processing Gain, PG) koji se računaju kao

$$PG = \left\lceil \frac{10 \log_{10} \left(\frac{\pi}{\omega_g} \right)}{6dB} \right\rceil \quad (4.2)$$

gdje je ω_g granična frekvencija područja propuštanja filtra. Upravo zbog dobitka u pojačanju uslijed obrade kroz 2 razine filtara je izlazni podatak 20-bitni, a ne 16-bitni kao ulaz.

S obzirom na to da se za generiranje koeficijenata filtra koristila MATLAB funkcija

firpm koja daje realne koeficijente, ulazni 16-bitni podaci i 16-bitni koeficijenti filtra zapisani su u formatu Q1.15, gdje prvi najznačajniji bit označava cjelobrojnu vrijednost, a ostalih 15 bitova označava decimalnu. Konkretno, u odabranom primjeru akumulatorska ekstenzija iznosi 2 bita i dobitak u pojačanju uslijed obrade također 2 bita. Zbog predznačnog proširivanja, podatak postaje formata Q3.15. Nakon množenja s koeficijentom formata Q1.15, podatak postaje formata Q4.30. Zatim, "rezanjem" podatka na dobitak u pojačanju uslijed obrade na izlazu periodičnog i komplementarnog filtra dobiva se podatak u formatu Q3.15. Nakon prolaska podatka kroz maskirne filtre, događa se ista stvar i dobiva se podatak u formatu Q5.15. Jasniji prikaz formata bitova je na slici 4.2 gdje PP označava predznačno proširivanje, a X množenje.



Slika 4.2: Shema efekta predznačnog proširivanja i dobitka u pojačanju uslijed obrade

4.2. Opis fizički ostvarivih modela pojedinih filtara

Modeli periodičnog i komplementarnog filtra su implementirani u jednoj komponenti *per_comp.vhd* zato što se komplementarni filtar izvodi kao razlika zakašnjelog ulaznog podatka i trenutnog izlaza periodičnog filtra. Zato se u samu komponentu ne učitavaju svi koeficijenti periodičnog i komplementarnog filtra, nego prototipnog.

Generičke konstante i priključci entiteta su prikazani u tablici 4.2. Generičke konstante se učitavaju iz novo definirane biblioteke *initialization.vhd*. Unutar samog entiteta definirana su polja tipa *signed* za koeficijente filtra i ulazne podatke. Napravljena je i funkcija koja prilikom sinteze čita koeficijente filtra iz vanjske datoteke te ih odmah zapisuje u polje. Polje ulaznih podataka je duljine broja koeficijenata periodičnog filtra, a budući da je tek svaki *R*-ti koeficijent impulsnog odziva periodičnog filtra različit od nule, polje koeficijenata impulsnog odziva je duljine broja koeficijenata prototipnog

filtra. Nakon što dođe novi podatak, polje ulaznih podataka posmiče podatke za jedno mjesto u desno te se novi podatak sprema u prvo polje. Uz to je definiran i brojač koji broji do broja koeficijenata filtra jer je sustav implementiran s jednim zbrajalom i s jednim množilom zbog čega je na svaki signal takta je moguće obraditi samo jedan podatak. Kada brojač izbroji do kraja, zbrajalo i množilo se postavljaju u nulu te se polje podataka posmiče i prihvaća se novi podatak.

Tablica 4.2: Generičke konstante i priključci periodičnog i komplementarnog filtra

Generičke konstante		Opis
R		faktor frekvencijskog skaliranja
CoeffWidth		širina riječi koeficijenata filtra
SampleWidth		širina riječi ulaznog podatka
CoeffNum		broj koeficijenata filtra
SampleNum		broj ulaznih uzoraka
firCoeffsFileName		ime datoteke koja sadrži koeficijente filtra
Priključci entiteta		Opis
clk	in	signal takta sustava
sample_in	in	ulazni podatak
reset	in	reset sustava
output	out	izlaz periodičnog filtra
output_complementary	out	izlaz komplementarnog filtra
next_sample	in	signal dolaska novog podatka
conv_enable	out	signal nemogućnosti primanja novog podatka

Maskirni filtri su realizirani na gotovo jednak način kao i periodični i komplementarni. Svaki je instanciran komponentom *fir03.vhd* čije su generičke konstante i priključci entiteta prikazani u tablici 4.3. Budući da maskirni filtri nisu periodični, njihovo provođenje konvolucije traje nešto duže od periodičnog i komplementarnog jer imaju više koeficijenata različitih od nule.

4.3. Strukturni model FRM filtra

Radi lakšeg testiranja sustava, napravljen je "omotač" (*engl. wrapper*) za povezivanje ranije navedenih komponenti sustava *fir_frm_wrapper.vhd* čije su generičke konstante i priključci entiteta navedeni u tablici 4.4. Unutar samog omotača su definirani interni

Tablica 4.3: Generičke konstante i priključci entiteta maskirnih filtara

Generičke konstante		Opis
CoeffWidth		širina riječi koeficijenata filtra
SampleWidth		širina riječi ulaznog podatka
CoeffNum		broj koeficijenata filtra
SampleNum		broj ulaznih uzoraka
firCoeffsFileName		ime datoteke koja sadrži koeficijente filtra
Priključci entiteta		Opis
clk	in	signal takta sustava
sample_in	in	ulazni podatak
reset	in	reset sustava
output	out	izlaz maskirnog filtra
next_sample	in	signal dolaska novog podatka
conv_enable	out	signal nemogućnosti primanja novog podatka

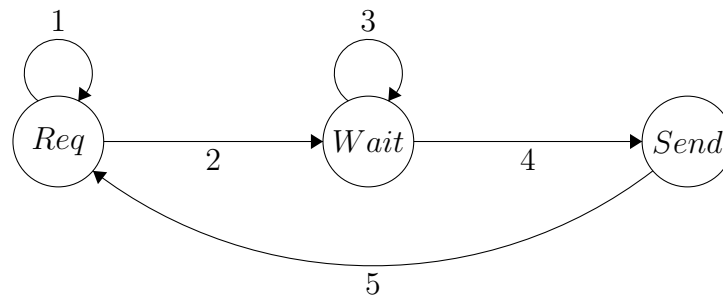
signali potrebni za povezivanje filtara i jedno zbrajalo koje sumira izlaze maskirnih filtara, što predstavlja izlaz cijelog filtra, odnosno FRM filtra.

Tablica 4.4: Generičke konstante i priključci entiteta "omotača" sustava

Generičke konstante		Opis
SampleWidth		širina riječi ulaznog podatka
SampleNum		broj ulaznih uzoraka
Priključci entiteta		Opis
clk	in	signal takta sustava
sample_in	in	ulazni podatak
reset	in	reset sustava
output	out	izlaz maskirnog filtra
next_sample	out	signal dolaska novog podatka
write_enable	out	signal zapisa izlaznog podatka
ready	in	signal spremnosti FIFO memorija

Kao priprema za prijenos sustava na sklopovlje, napravljen je još jedan automat stanja koji upravlja signalima za pokretanje sustava i omogućivanje FIFO (*engl.* First-In-First-out) memorija. Automat ima sljedeća stanja: *Req* kada sustav traži novi podatak

od ulazne FIFO memorije, *Wait* u kojem se sustav nalazi sve dok filter ne završi s obradom podatka i *Send* kada sustav šalje podatak na izlaznu FIFO memoriju. Koriste se pomoćni upravljački signali *next_sample* koji javlja ulaznoj FIFO memoriji kada je sustav spreman prihvatiti novi podatak, *write_enable* koji javlja izlaznoj FIFO memoriji kada je podatak spreman za izlazak iz filtra, *ready* koji javlja sustavu kada su FIFO memorije spremne poslati i primiti podatak, te *next_sample_per* koji javlja sustavu kada može započeti s obradom podatka. Logika automata stanja u "omotaču" je prikazana na slici 4.3, a legenda prijelaza iz jednog u drugo stanje u tablici 4.5.

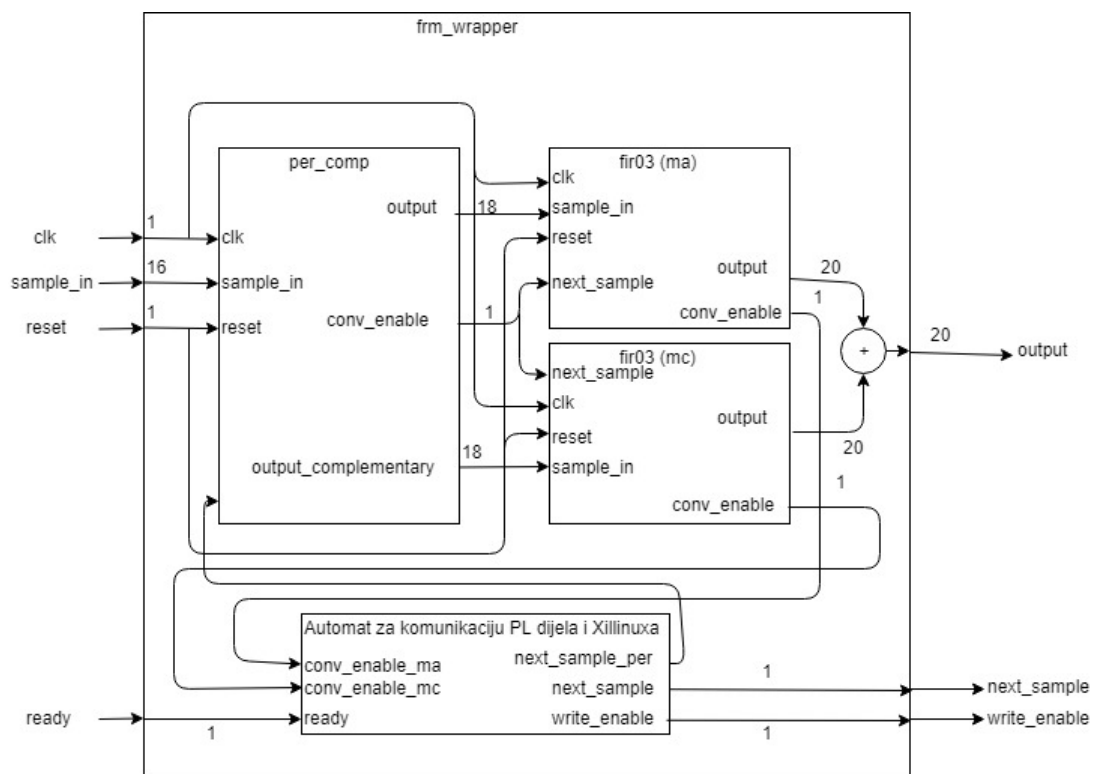


Slika 4.3: Dijagram stanja automata za komunikaciju sustava s FIFO memorijama

Tablica 4.5: Legenda prijelaza stanja automata za komunikaciju sustava s FIFO memorijama

Kratica	Objašnjenje
1	FIFO memorije nisu spremne poslati/prihvatiti podatak
2	FIFO memorija je spremna poslati podatak
3	Konvolucija još traje
4	Konvolucija je završila
5	Slanje obrađenog podatka na izlaznu FIFO memoriju Čekanje sljedećeg podatka iz ulazne FIFO memorije

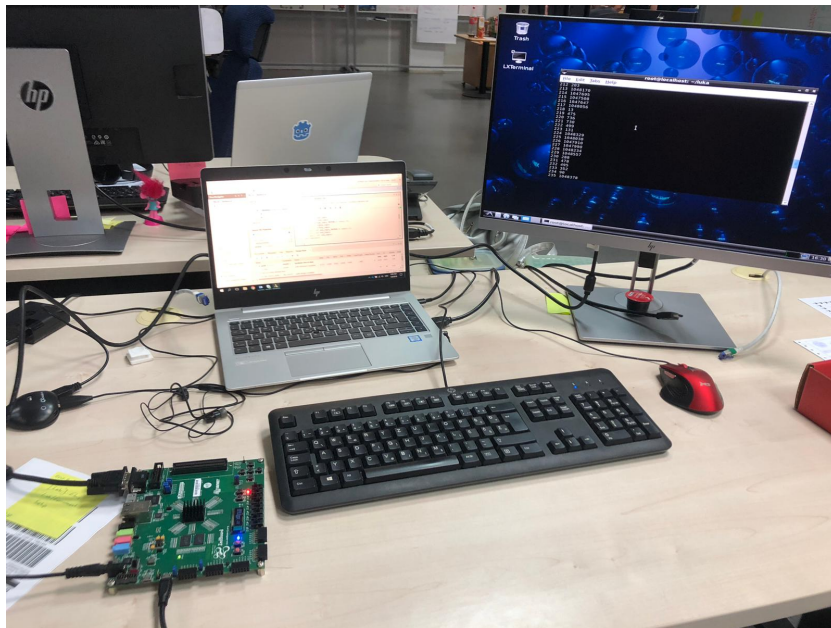
Napravljen je *fir_frm_tb.vhd* u kojem se generira signal takta svakih 2 ns i koji čita datoteku s ulaznim podacima te šalje na FRM filter. Za potrebe simulacije signal spremnosti FIFO memorija *ready* je postavljen u visoku razinu cijelo vrijeme. Blokowska shema RTL sustava je prikazana na slici 4.4. Konačni odziv stvarnog sklopovlja prikazan je u sljedećem poglavlju.



Slika 4.4: Blokvska shema RTL modela širokopolasnog FRM filtra

5. Uvodavanje implementiranog filtra na razvojnom sustavu ZedBoard

ZedBoard je razvojni sustav koji sadrži čip porodice Zynq-7000 koji je baziran na Xilinxovoj All programmable Soc (*engl. System-on-Chip*) arhitekturi. Sastoji se od dvojezrenog ARM Cortex-A9 procesorskog sustava i programabilne logike koji su smješteni u jedno kućište te povezani unutarnjim sabirnicama. Pretežito ciljne aplikacije su obrada videa, rekonfigurabilno računarstvo, kontrola motora, softversko ubrzanje, te razvoj Linux/Android/RTOS aplikacija. Pokrenuta ZedBoard pločica spojena s tipkovnicom i mišem preko USB OTG-a te s monitorom preko VGA kabla je prikazana na slici 5.1.

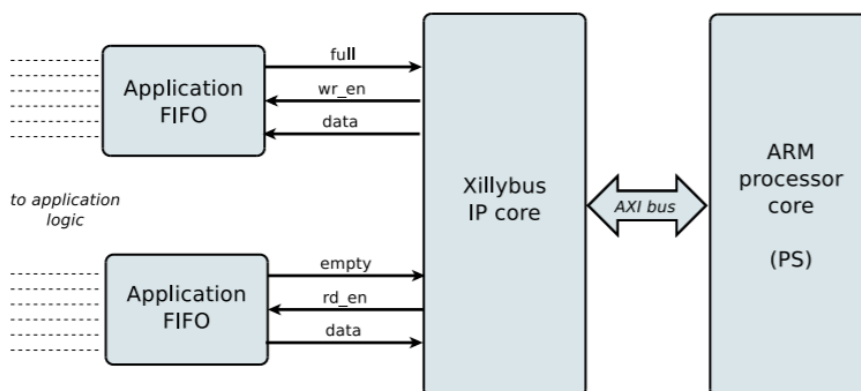


Slika 5.1: ZedBoard spojen s periferijom

Pločica se može programirati na više načina. Način programiranja se definira postavljanjem seta kratkospojnika. U ovom diplomskom radu korišten je bežični način programiranja pomoću SD kartice. Nakon što se SD kartica pripremi s potrebnim datotekama, umetne se s donje strane pločice te se prilikom pokretanja ZedBoard isprogramira ako su ispravno postavljeni kratkospojnici. U ovom radu korišten je operacijski sustav Xillinux koji je baziran na Ubuntu 16.04 Linux distribuciji [4]. Sabirnički sustav upravljan je Xillybus sučeljem koje je detaljnije opisano u sljedećem potpoglavlju.

5.1. Komunikacija između PL dijela i Xillinixa

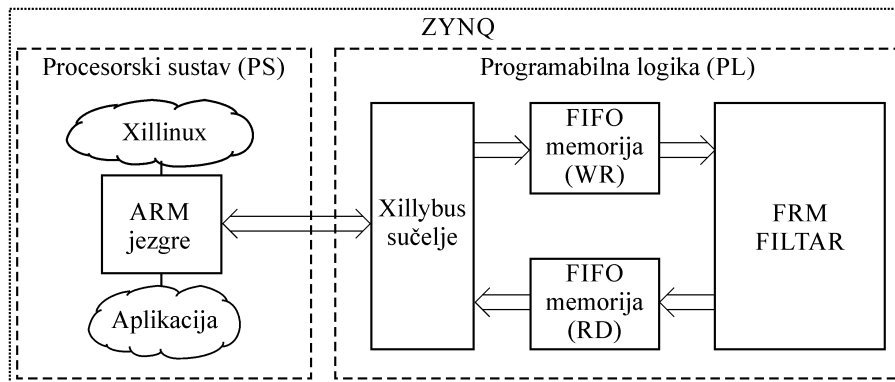
Xillybus sučelje omogućuje komunikaciju, odnosno prijenos podataka, između FPGA i operacijskog sustava, u ovom slučaju Linuxa. Logika u FPGA dijelu povezuje se na IP (*engl. Intellectual Property*) jezgru kroz klasične FIFO memorije. Korisnički program sa strane procesorskog sustava izvodi jednostavne ulazno-izlazne datotečne operacije na cjevovodno oblikovanim datotekama preko funkcija definiranih za samo sučelje. Na slici 5.2 prikazana je blokovska shema Xillybus sučelja. Donja FIFO memorija je ulazna, dok je gornja izlazna. Procesorski sustav preko Xillybus sučelja šalje podatak koji se upisuje u ulaznu FIFO memoriju. Dalje se taj podatak prosljeđuje u implementirani FRM filtar, koji nakon obrade šalje podatak u izlaznu FIFO memoriju. Nakon što se podatak upiše u izlaznu memoriju, Xillybus sučelje očitava podatak te isti šalje procesorskom sustavu.



Slika 5.2: Blokovska shema Xillybus sučelja, preuzeto iz [4]

Sa Xillybusove stranice [5] preuzeto je testno okruženje Xillydemo koje sadrže im-

plementirane funkcionalnosti potrebne za rad platforme. U Xillydemu su na početku instancirane dvije FIFO memorije, jedna širine riječi 8 bita, a druga širine riječi 32 bita. Budući da su u ovom radu ulazni podaci 16-bitni, a izlazni 20-bitni, korištena je 32-bitna FIFO memorija, te je instancirana još jedna 32-bitna FIFO memorija. U datoteci *xillydemo.vhd* je definirano povezivanje FIFO memorija i FRM filtra tako da se iz ulazne FIFO memorije prosljeđuje donjih 16 bitova na FRM filter, te se iz FRM filtra na izlaznu FIFO memoriju šalje na donjih 20 bitova. U procesorskom sustavu se na kraju zapisuje podatak od 32 bita nad kojim treba provesti još potrebne konverzije koje su opisane u sljedećem potpoglavlju. Blok shema implementiranog sustava u ovom radu je prikazana na slici 5.3, a zauzeće potrebnih komponenti na ZedBoardu u tablici 5.1.



Slika 5.3: Blokvska shema implementiranog sustava

Tablica 5.1: Usporedba broja dostupnih i iskorištenih komponenti ZedBoarda za opisani sustav

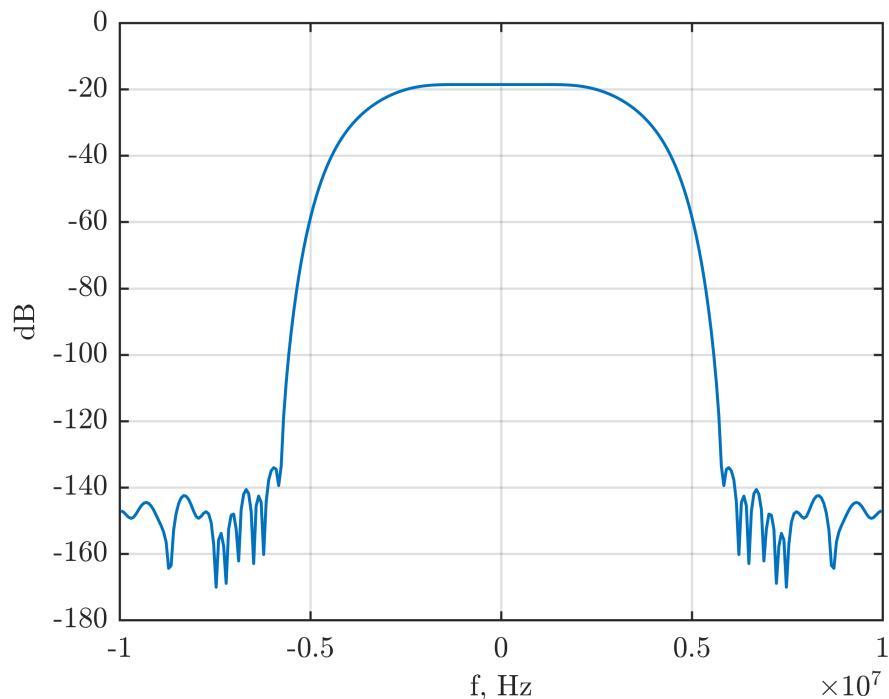
Dijelovi sklopa	Iskorišteni	Dostupni
LUT	5692	53200
Bistabili	6683	106400
F7 multiplekseri	262	26600
F8 multiplekseri	108	13300
RAMB18	3	280
DSP	3	220

5.2. Priprema podataka za analizu i rezultati

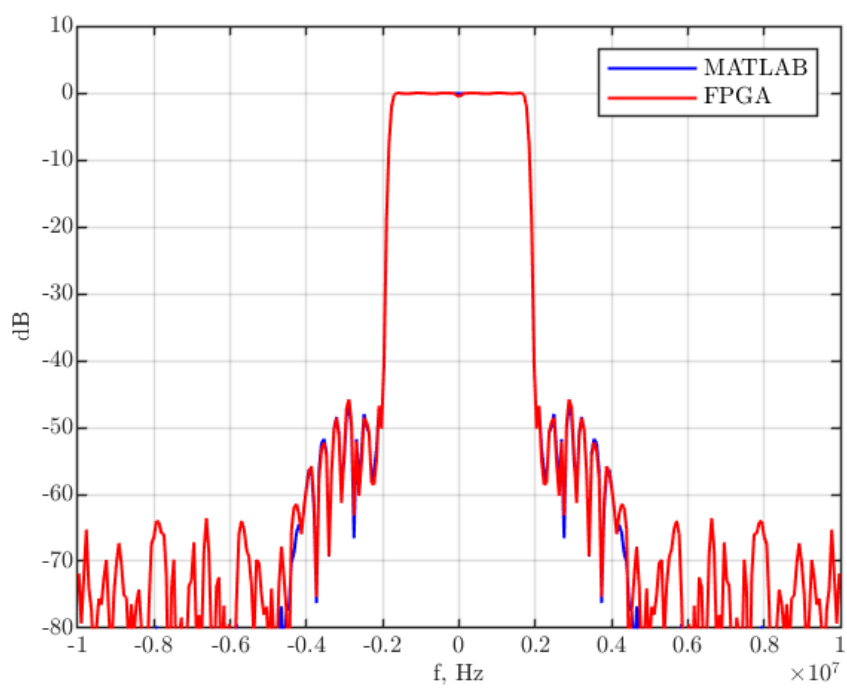
Kao ulazni signal se koristio izdignuti kosinusni puls centralne frekvencije $f_c = 500$ MHz. Signal je generiran u razvojnom okruženju MATLAB gdje se još koristio

fi_objekt[6] kako bi se ulazni signal mogao zapisati kao 16-bitni podatak gdje najznačajniji bit označava cjelobrojnu vrijednost, a 15 nižih bitova decimalnu vrijednost. Osim toga, potrebno je bilo i napraviti program u jeziku C koji na Xilinx-u učitava podatke tipa *unsigned int*, šalje na sklopovski dio sustava preko funkcija za komunikaciju definiranih za Xillybus sučelje, prihvaća podatke iz sklopovskog dijela te ih ponovno zapisuje u tip *unsigned int* u novu datoteku. Dobivene podatke je zatim bilo potrebno ponovno u MATLABu učitati kao *unsigned int*, te pretvoriti u binarni zapis da bi ih se konačno moglo vratiti u Q5.15 zapis. Razlog ove komplikacije s konverzijama je taj što u programskom jeziku C ne postoji bit format s kojim bi se mogao jednostavno zapisati izlazni podatak.

Kako bi rezultati stvarnog FRM filtra bili usporedivi s modelom, sam MATLAB model filtra je također kvantiziran korištenjem *fi_objekta* na Q5.15 zapis. Zbog činjenice da su i koeficijenti pojedinih filtara i sam ulazni signal realnih vrijednosti, očekivano se dobilo nepotpuno poklapanje karakteristika odziva. Na slici 5.4 prikazana je amplitudna karakteristika pobude, dok je na slici 5.5 prikazana usporedba amplitudno-frekvencijskih karakteristika kvantiziranog modela FRM filtra dobivenog u MATLABu i implementiranog filtra na ZedBoardu.



Slika 5.4: Amplitudna karakteristika pobude



Slika 5.5: Usporedba amplitudno-frekvencijskih karakteristika kvantiziranog modela dobivenog u MATLABu i na ZedBoardu

6. Zaključak

U radu je opisan postupak dobivanja vrlo strmih FIR filtara korištenjem tehnike frekvencijskog maskiranja. Koristilo se programsko okruženje MATLAB za izvedbu samog algoritma te pronalaska potrebnih frekvencijskih širina i koeficijenata impulsnih odziva filtara. Filtri potrebni za izvedbu algoritma su: periodički filter, komplementarni filter periodičnog filtra, maskirni filter periodičnog filtra i maskirni filter komplementarnog filtra. Konačni impulsni odziv FRM filtra dobiva se zbrajanjem kaskade periodičnog filtra i njegovog maskirnog te kaskade komplementarnog filtra i njegovog maskirnog.

Za izvedbu fizički ostvarivog modela FRM filtra korišteno je razvojno okruženje Vivado Design Suite 2018.3 i jezik za opis sklopovlja VHDL. Komplementarni filter izveden je tako da se trenutnom izlazu periodičnog filtra oduzeo zakašnjeni ulazni podatak. Trajanje konvolucije periodičnog filtra je manje od maskirnih filtara zato što periodični filter ima puno koeficijenata impulsnog odziva jednakih nuli. Zbog toga se morala provoditi konvolucija nad puno manje podataka. FRM tehnika se pokazala bržom nego kada bi se ista širina prijelaznog područja ostvarila konvencionalnim FIR filtrom. Kada bi se opisani slučaj u ovom radu izveo konvencionalnim načinom, bio bi potreban FIR filter s 249 koeficijenata. S FRM filtrom potrebno je 84 koeficijenata zbog puno koeficijenata u periodičnom filtru koji su jednaki nuli.

Na kraju je implementirani FRM filter uhodan na razvojnoj pločici ZedBoard, na platformi Zynq. Procesorski sustav je upravljan Xilinx operacijskim sustavom, baziranim na Ubuntu 16.04. Komunikacija između procesorskog sustava i programabilne logike je realizirana preko Xillybus sabirničkog sučelja.

LITERATURA

- [1] L. J. G. Joseph Rodrigues, K R Pai, "Synthesis of linear phase sharp transition fir digital filter," 2005.
- [2] Y. S. J. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," 1986.
- [3] "Prosječno kašnjenje filtra." <https://www.mathworks.com/help/signal/ref/grpdelay.html>. [Pristupljeno 8. srpnja 2019.].
- [4] "Getting started with Xilinx for Zynq-7000 v2.0." http://xillybus.com/downloads/doc/xillybus_getting_started_zynq.pdf/. [Pristupljeno 25. lipnja 2019.].
- [5] "Xillybus." <http://xillybus.com/>. [Pristupljeno 25. lipnja 2019.].
- [6] "Fixed-point numerički objekt." <https://www.mathworks.com/help/fixedpoint/ref/fi.html>. [Pristupljeno 8. srpnja 2019.].

FPGA implementacija vrlo strmih FIR filtara dobivenih tehnikom maskiranja frekvencijske karakteristike

Sažetak

Proučena je tehnika maskiranja frekvencijske karakteristike (FRM) za dobivanje vrlo strmih niskopropusnih FIR filtara s linearnom fazom. FRM tehnikom su obuhvaćeni i uskopojasni i širokopojasni filtri. Za realizaciju FRM filtra određeni su prototipni i maskirni filtri te faktor skaliranja frekvencijske osi. Za dane granične frekvencije prijelaznog područja FRM filtra razvijen je postupak određivanja faktora skaliranja te graničnih frekvencija prototipnog filtra i graničnih frekvencija maskirnih filtara. Za dobivene frekvencije te za dane redove ovih filtara određeni su koeficijenti njihovih prijenosnih funkcija koje imaju jednoliku valovitost amplitude u području propuštanja i području gušenja. Za potrebe implementacije FRM filtra razvijen je fizički ostvariv model FIR filtra korištenjem jednog zbrajala i jednog množila, koji podržava generički broj elemenata za kašnjenje između dvije uzastopne sumacije. Korištenjem ovog modela i dobivenih koeficijenata implementiran je prototipni filtar s brojem elemenata za kašnjenje koji odgovara faktoru skaliranja te maskirni filtri s jediničnim kašnjenjem. Na kraju, korištenjem ovih filtara, razvijen je model širokopojasnog FRM filtra. Model filtra razvijen je u jeziku VHDL i uhodan na razvojnom sustavu Zedboard.

Ključne riječi: FIR, frekvencijsko maskiranje, filtar, MATLAB, VHDL, Zedboard, Xillybus, FIFO

FPGA implementation of Very Sharp FIR Filters Obtained by Frequency-Response Masking Technique

Abstract

The frequency response masking (FRM) technique has been studied to obtain very steep non-linear FIR filters with a linear phase. FRM technology included narrowband and wideband filters. For the FRM filter, prototype and masking filters and the frequency axis scaling factor were determined. For given FRM filter cutoff frequencies, the method of determining the scaling factor and the prototype filter and masking filters boundary frequencies has been developed. For the obtained frequencies and for given orders of these filter, the coefficients of the transfer functions were determined, which have a uniform amplitude wavelength in the passband and stopband area. For the implementation of the FRM filter, a physically feasible FIR filter model was developed using one adder and one multiplier, which supports a generic number of delay elements between two successive additions. By using this model and the obtained coefficients, a prototype filter with a number of delay elements corresponding to the scaling factor and the single-delay masking filters were implemented. Finally, using these filters, a wideband FRM filter model was developed. The filter model was developed in VHDL language and run on the Zedboard development system.

Keywords: FIR, frequency masking, FIR, MATLAB, VHDL, Zedboard, Xillybus, FIFO