

Realizacija adaptivnog filtra primenom metode protočne obrade signala

Divna Mičić, Dragana Prokin, Gabrijela Dimić
 Visoka škola elektrotehnike i računarstva strukovnih studija
 Beograd, Srbija
 divnam@viser.edu.rs, dprokin@viser.edu.rs, gdimic@viser.edu.rs

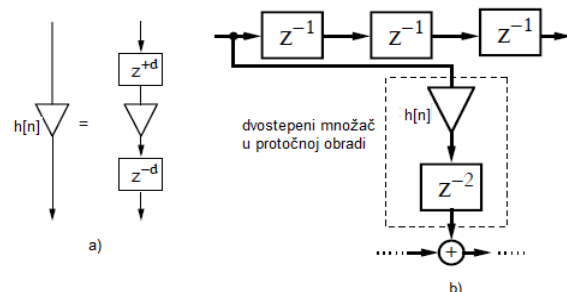
Sadržaj— U ovom radu predstavljen je dizajn protočnog adaptivnog FIR filtra (eng. *Pipelined adaptive finite impulse response filter*) implementiranog DLMS algoritmom (eng. *Delayed Least Mean Square*) koji se koristi za minimizaciju signala greške određivanjem koeficijenata adaptivnog filtra. Predloženi dizajn u odnosu na standardnu implementaciju LMS algoritma (eng. *Least Mean Square*) iterativno poboljšava propusni opseg što utiče na povećanje brzine uz povećanje iskorišćenja logičkih resursa FPGA kola (eng. *Field-programmable gate array*).

Ključne riječi- FPGA logička kola; DLMS algoritam; adaptivni filter; FIR filter; protočna obrada

I. UVOD

Projektovanje optimalnog filtra se zasniva na prethodnom poznavanju statistike signala, a dobijeno rešenje je optimalno samo za slučaj kada signali koji se filtriraju zaista imaju pretpostavljena statistička svojstva. Nažalost, veoma čest slučaj u praksi je da statistička svojstva signala nisu dostupna ili su ta svojstva promenljiva. U tom slučaju se statistički parametri menjaju u vremenu i veoma je teško projektovati odgovarajuće optimalno rešenje. Pogodan pristup u nevedenim situacijama je projektovanje tzv. *samopodešavajućih filtara*, poznatih pod imenom adaptivni filtri. Jedna od najznačajnijih osobina adaptivnih filtara je njihova sposobnost prilagođavanja, koja vodi ka željenom sistemu. Umesto da sistem bude unapred čvrsto definisan, adaptivni filtri koriste podatke iz okruženja, da bi postavili vrednost svojih parametara. To podrazumeva da se vrednost parametara menja u skladu sa izlazom filtra, preko povratne sprege i odgovarajuće kriterijumske funkcije. Svojstva povratne sprege direktno utiču na promenu vrednosti parametara, preko sistemske procedure koja se naziva proces obuke ili trening period. U toku obuke se menja izlaz filtra, u skladu sa referentnim signalom, tj. smanjuje se vrednost greške sa napredovanjem procesa obuke. Osnova ovog pristupa je dizajn adaptivnog algoritma. Namena adaptivnog algoritma je nadgledanje okruženja i prilagođavanje prenosne funkcije filtra u skladu sa uočenim promenama. Algoritam polazi od prethodno definisanih početnih uslova, koji ne moraju biti usklađeni sa okruženjem, i na osnovu trenutnih vrednosti

ulaznog, izlaznog i referentnog signala teži da nađe rešenje koje će biti optimalno u datom trenutku. U stacionarnom okruženju se očekuje da filter konvergira ka optimalnom filtru, dok se u nestacionarnom okruženju očekuje da filter prati vremenske promene, i u skladu sa njima menja svoje parametre. Procesom adaptacije upravlja signal greške, koji predstavlja meru prilagođenja parametara filtra, odnosno pokazatelj je usklađenosti izlaznog i referentnog signala. Često se kao kriterijum optimizacije prilikom estimacije parametara filtra koristi vrednost kvadrata signala greške ili srednja kvadratna greška MSE (eng. *Mean Square Error*). Zavisno od konkretne primene adaptivnog filtra, mera uspešnosti adaptacije može biti zasnovana na vrednosti estimiranih parametara filtra, izlaznom signalu filtra ili signalu greške [1], [2]. U osnovi realizacije adaptivnih filtara pogodnih za implementaciju u FPGA kolima nalaze se najčešće FIR filtri sa programabilnom arhitekturom. Za praktične primene od najvećeg interesa su optimizacije dizajna za postizanje većeg propusnog opsega, odnosno radne učestanosti, dok je sa aspekta implementacije u FPGA kolima bitno i zauzeće hardverskih resursa za predviđanje i ažuriranje vrednosti parametara filtra u toku obrade signala. Ovi resursi obuhvataju kola za sabiranje, množače i registarske komponente. Povećanje brzine programabilne arhitekture FIR filtra obezbeđuje se primenom množača u protočnoj strukturi i primenom stabla sabirača za akumuliranje u izlaznom stepenu. Ponekad određivanje pojedinačnog koeficijenta dovodi do većeg kašnjenja u protočnoj obradi od svih ostalih koeficijenata. Ako se to kašnjenje modelira sa $h[n]z^{-d}$ i ako se ovom kašnjenju doda kašnjenje z^d , dobija se model $z^d h[n]z^{-d}$ kojim mogu da se eliminišu dva kašnjenja (Sl. 1a). Na (Sl. 1b) prikazana je mogućnost primena ove ideje u protočnoj obradi. Slična ideja može da se primeni u hardverskim implementacijama složenijih struktura FIR filtara kao što su adaptivne realizacije na bazi LMS algoritma.



Slika 1. Kompenzacija kašnjenja FIR filtra. (a) princip, b) primena u protočnoj obradi

¹Sredstva za ovaj rad su delimično obezbeđena od strane Ministarstva prosvete, nauke i tehnološkog razvoja Vlade Republike Srbije u okviru projekata TR32039 and TR32043.

D. Prokin, Visoka škola elektrotehnike i računarstva strukovnih studija, Vojvode Stepe 283, 11000 Beograd, Srbija (e-mail: dprokin@viser.edu.rs).

D. Mičić, Visoka škola elektrotehnike i računarstva strukovnih studija, Vojvode Stepe 283, 11000 Beograd, Srbija (e-mail: divnam@viser.edu.rs).

G. Dimić, Visoka škola elektrotehnike i računarstva strukovnih studija, Vojvode Stepe 283, 11000 Beograd, Srbija (e-mail: gdimic@viser.edu.rs)

II. KRITERIJUMSKA FUNKCIJA SREDNJE KVADRATNE GREŠKE

U procesu identifikacije sistema moguća je primena različitih modela. Jedan od mogućih modela je da adaptivni filter ima jedan ulaz kojim se pobuđuje nepoznati sistem i adaptivni filter koji je vezan paralelno sa nepoznatim sistemom (Sl. 2). Nepoznati sistem je određen svojom strukturom koja se u opštem slučaju može opisati vektorom parametara $h[n]$.

Zadatak identifikacije može biti procena vrednosti parametara nepoznatog sistema, minimizacija razlike izlaznih signala nepoznatog sistema i adaptivnog filtra, i slično. Ukoliko je poznata struktura nepoznatog sistema zadatak identifikacije se svodi na procenu vrednosti parametara sistema. Bez obzira na odabranu metodu izbora parametara adaptivnog filtra, procenjena vrednost parametara $\hat{h}[n]$ treba da najviše odgovarati konkretnom zadatku. Signal greške je definisan sledećom jednačinom:

$$e[n] = d[n] - y[n] \quad (1)$$

Izjednačavanjem Viner-Hopfovih jednačina sa nulom [3], dobija se optimalno rešenje za vektor parametara:

$$\begin{aligned} \nabla = 2R\hat{h} - 2P &= 0 \\ h_{opt} &= R^{-1}D, \end{aligned} \quad (2)$$

gde h_{opt} predstavlja vektor optimalnih vrednosti parametra FIR filtra, odnosno one vrednosti vektora parametara \hat{h} za koje se postiže minimum MSE kriterijumske funkcije, J_{min} .

III. ADAPTIVNI ALGORITMI MINIMALNE SREDNJE KVADRATNE VREDNOSTI

LMS algoritam (Least Mean Square) [4] je algoritam minimalne srednje kvadratne vrednosti koji pripada metodi najbržeg spusta, ali koristi posebnu estimaciju gradijenta, odnosno uzima trenutnu vrednost kvadrata signala greške $e^2(k)$ umesto matematičkog očekivanja MSE u (12) kroz sledeće korake algoritma:

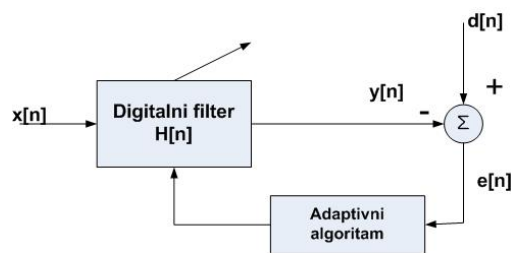
Korak1. Inicijalizacija matrice (Lx1) koeficijenata filtra:

$$h = x = 0 = [0, 0, \dots, 0]^T. \quad (3)$$

Korak2. Priprihvatanje novog para ulaznih signala:

$$\{x[n], d[n]\}, \quad (4)$$

gde $x[n]$ predstavlja slučajni ulazni signal, a $d[n]$ referentni signal.



Slika 2. Osnovna konfiguracija sistema sa adaptivnim filtrom

Korak3. Izračunavanje izlaznog signala adaptivnog filtra:

$$y[n] = \hat{h}[n]^T x[n]. \quad (5)$$

Korak4. Izračunavanje signala greške:

$$e[n] = d[n] - y[n]. \quad (6)$$

Korak5. Ažuriranje koeficijenata adaptivnog filtra u granicama skalarnog parametra μ :

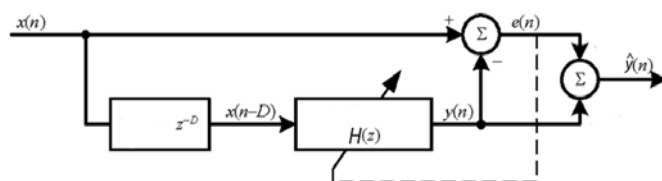
$$h[n+1] = h[n] + \mu e[n] x[n], \quad (7)$$

$$0 < \mu < \frac{2}{\lambda_{max}}. \quad (8)$$

Uvođenjem postupka protočne obrade signala teži se povećanju brzine konvergiranja vektora parametara filtra ka optimalnim vrednostima.

Analizom hardverske implementacije standardnog LMS algoritma u FPGA [4] programabilnom logičkom kolu, dolazi se do zaključka da postoji problem niske radne učestanosti, koja je posledica činjenice da operacije množenja i sabiranja moraju da se obave u jednom ciklusu takta pre nego što mogu da budu ažurirani koeficijenti filtra. Zbog toga su razvijene različite metode za poboljšanje performansi, odnosno za povećanje propusnog opsega adaptivnog LMS filtra.

U slučaju primene protočne arhitekture, može da se odredi optimalan broj stepena obrade LMS FIR filtra (Sl. 3).



Slika 3. Osnovna struktura DLMS FIR filtra

- Za (bxb) množača neophodno je $\log_2(b)$ stepena
- Za blok sabirača neophodno je $\log_2(L)$ stepena
- Za množenje pri ažuriranju koeficijenata potrebno je $\log_2(b)$ stepena
- Dodaje se jedan stepen za izračunavanje greške

Na osnovu prethodnog, sledi da je za maksimalan propusni opseg neophodno

$$D_{opt} = 2\log_2(b) + \log_2(L) + 1, \quad (9)$$

stepena obrade u protočnoj arhitekturi, pod uslovom da je skalarni parameter μ (koji je stepen broja dva) konstantan i da nije neophodan dodatni stepen za njegovo izračunavanje. Međutim ukoliko se primenjuje NLMS (eng. *Normalised LMS*), μ nije više konstantan i u zavisnosti od broja bita odbiraka mora da se uvede još stepena obrade.

Implementacija LMS filtra sa protočnom obradom nije jednostavna, jer ima povratnu spregu [5]. U protočnoj obradi je neophodno obezbediti da koeficijenti filtra teže istim optimalnim vrednostima kao i u slučaju kada protočna obrada nije primenjena.

Neke od metode koje se mogu primeniti za realizacije protočne obrade kod adaptivnih LMS filtara su:

- LMS sa kašnjenjem (DLMS)
- LMS sa protočnom arhitekturom i propagacijom
- LMS sa transpozicijom

U **LMS algoritmu sa kašnjenjem (DLMS)** polazi se od pretpostavke da se gradijent greške:

$$\nabla[n] = e[n]x[n] \quad (10)$$

ne menja mnogo ukoliko se zakasni ažuriranje koeficijenata za nekoliko uzoraka, odnosno ako važi:

$$\nabla[n] \approx \nabla[n-D] \quad (11)$$

Može se pokazati da ovo važi sve dok je kašnjenje manje od reda filtra, odnosno dužine filtra i da ažuriranje neće da smanji brzinu konvergencije. Prilikom implementacije DLMS filtra u FPGA kolu neophodno je uvođenje dodatnih stepena zbog implementacije množača i ažuriranja koeficijenata. Ako se sa D_1 označi kašnjenje filtra u putanji za izračunavanje, a sa D_2 kašnjenje u putanji za ažuriranje koeficijenata LMS algoritma, dobija se:

$$e[n-D_1] = d[n-D_1] - h[n-D_1]^T x[n-D_1], \quad (12)$$

$$h[n+1] = h[n-D_1-D_2] + \mu e[n-D_1-D_2]x[n-D_1-D_2]. \quad (13)$$

LMS sa protočnom arhitekturom i propagacijom je metoda koja se primenjuje kod filtara sa malim redom filtra, kod kojih je nepoželjno menjanje konvergencijskog svojstva filtra uvođenjem kašnjenja prilikom ažuriranja koeficijenata filtra, što je prikazano DLMS algoritmom. Da bi se to izbeglo mora se otkloniti promena koja nastaje u funkciji signala greške kod DLMS algoritma u (12) u odnosu na funkciju signala greške kod LMS algoritma. Neophodno je izračunati korekcionu konstantu $\Lambda[n]$ kojom se otklanja razlika između signala greške kod primene LMS algoritma (6) i signala greške kod primene DLMS algoritma (12):

$$\Lambda[n] = e^{LMS}[n] - e^{DLMS}[n-D]. \quad (14)$$

Korigovana funkcija signala greške kod DLMS algoritma:

$$e^{\bar{DLMS}}[n-D] = d[n-D] - x^T[n-D]h[n-D_1] - \Lambda[n]. \quad (15)$$

Prilikom izračunavanja korigovane funkcije signala greške u (15) uvodi se dodatnih 2D operacija množenja, što utiče na kompleksnost algoritma. Metodom LMS protočne arhitekture sa propagacijom mogu se izbeći dodatne operacije množenja prilikom ažuriranja koeficijenata filtra, otklanjanjem kašnjenja D_2 :

$$h[n+1] = h[n-D_1] + \mu \sum_{k=0}^{D_2-1} e[n-D_1-k]x[n-D_1-k]. \quad (16)$$

LMS sa transpozicijom otklanja kompletno kašnjenje u bloku sabirača. Na ovaj način smanjuje se neophodan broj stepena za protočnu obradu signala za $\log_2(L)$ stepena, pri čemu se mora voditi računa o granicama μ .

IV. IMPLEMENTACIJA DLMS ALGORITMA

Način hardverske implementacije protočne obrade biće opisan na primeru DLMS algoritma. DLMS FIR adaptivni filter implementiran je u FPGA logičkom kolu firme Altera, u Quartus okruženju, čija je osnovna struktura prikazana na (Sl.3). Takođe, kao što se vidi na Sl. 3, izlaz svakog bloka linije za kašnjenje je ulaz adaptivnog filtra. Na osnovu izraza (9) može da se pokaže da je optimalan broj stanja u protočnoj obradi $D_{opt} = 2x3 + 1 + 1 = 8$.

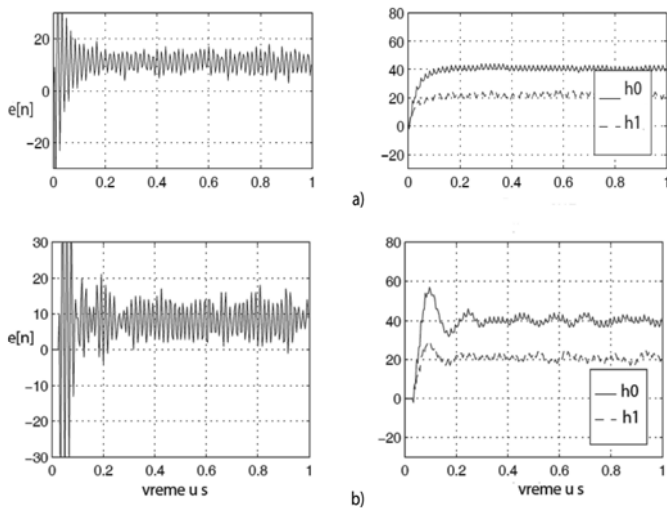
S druge strane, primenom protočne obrade (Sl. 1), broj stanja može da se svede na 6 [6].

U cilju hardverske implementacije DLMS algoritma za adaptivni FIR filter izvršen je VHDL opis dizajna [7], [8] na osnovu strukture prikazane na Sl. 3. Pri realizaciji VHDL opisa (eng. *VHSIC hardware description language*) definisani su sledeći koraci:

- Korak1.* Specifikacija širine ulaznog podatka ($n = 8$ bita), širine množača ($2*n=16$ bita) i dužine filtra ($L=2$)
- Korak2.* Učitavanje ulaznog signala $x(n)$ i referentnog signala $d(n)$ 8-bitne širine
- Korak3.* Određivanje koeficijenata filtra h_0 i h_1
- Korak4.* Određivanje proizvoda: $p(i) = h(i)x(i)$
- Korak5.* Određivanje izlaza adaptivnog filtra
 $y = p(0) + p(1)$
- Korak6.* Određivanje signala greške
 $e = d - y$
- Korak7.* Određivanje narednog proizvoda (*Korak4* protočne obrade)

V. ANALIZA PERFORMANSI DIZAJNA

Da bi se analizale karakteristike adaptivnog DLMS FIR filtra sa protočnom obradom izvršena je simulacija u Matlab (Sl. 4) i Quartus II okruženju. Rezultati simulacije prikazani su na Sl. 4 i Sl. 5 respektivno.



Slika 4. Izlaz i koeficijenti filtra dobijeni kao rezultat simulacije pri testiranju a) LMS i b) DLMS algoritma

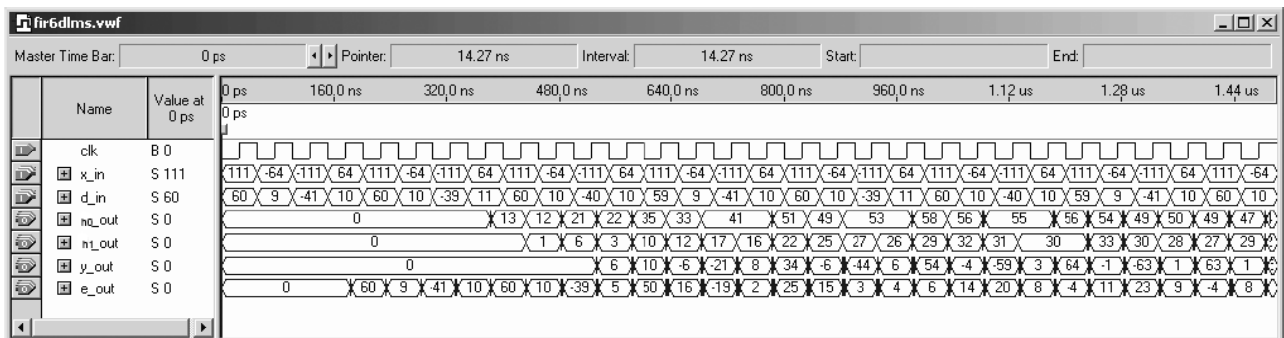
Cilj simulacije u okruženju Matlab (Sl. 4) bila je samo predikcija konvergencije koeficijenata adaptivnog FIR filtra ka optimalnim vrednostima u slučaju primene LMS i DLMS algoritma, zbog čega ostali filtri nisu simulirani.

Kompletna simulacija dizajna u Quartus II okruženju (Sl. 5) je izvršena na osnovu implementacije dizajna VHDL opisom [9] i [10] u FPGA logičkom kolu. Simulacijom su analizirani vrednosti izlaznog signala y i signala greške e na osnovu referentnog signala d i ulaznog signala x .

Dužina reči od 8-bitna je dovoljna za neke praktične primene, mada se u praksi najčešće koriste dužine reči od minimalno 10 bita. Primenljivost rešenja adaptivnog filtra, čije je rešenje opisano u ovom radu zavisi pre svega od tehničkih zahteva konkretne aplikacije. Međutim rezultati testiranja predstavljeni u ovom rada mogli bi da pomognu pri donošenju odluke da li filter sa ovakvim performansama zadovoljava zahteve dizajna u kome treba da bude implementiran ili ne. Moguće je proširiti reč, ali to bi uticalo za zauzeće resursa čipa.

U Tabeli I prikazani su rezultati dobijeni implementacijom različitih algoritama za optimizaciju propusnog opsega primenom protočne obrade signala kod adaptivnih LMS filtara. Prikazano je zauzeće resursa za filter drugog reda izraženo preko ukupnog broja logičkih elemenata FPGA kola potrebnih za implementaciju, broj primenjenih množača $9x9$ i maksimalne radne frekvencije implementiranog dizajna.

Maksimalna radna frekvencija predstavlja frekvenciju na kojoj programabilna logika implementirana određenom metodom u izabranom FPGA kolu radi pouzdano. Svaka frekvencija iznad definisane maksimalne radne frekvencije date u Tabeli I može da izazove nekorektan rad implementiranog dizajna, zbog čega se primenjuju različite metode implementiranja dizajna sa ciljem povećanja maksimalne radne frekvencije. Testiranje je urađeno pod pretpostavkom da je dizajn implementiran u Alterinom Cyclone II EP2K35F672C6 FPGA kolu koje poseduje hardverske množače i 33.216 logičkih elemenata [11]. U fazi testiranja dizajna utvrđeno je da adaptivni DLMS filter dostiže optimalne vrednosti koeficijenata nakon 30 iteracija pri čemu ostvaruje maksimalnu brzinu koja je približno tri puta veća od brzine adaptivnog LMS filtra. Na osnovu Tabele I takođe sledi da ostali implementirani algoritmi daju mogućnost još značajnijeg povećanja brzine, ali po cenu većeg zauzeća resursa, pri čemu treba primetiti da je najveća brzina ostvarena realizacijom algoritma za protočnu obradu sa optimalnim brojem stepena obrade.



Slika 5. Rezultati simulacije DLMS algoritma u Quartus okruženju

TABELA I. Metode realizacije LMS filtera

D	Logički elementi	Množeći 9x9	MHz	Metoda
0	51	4	78.68	LMS algoritam
1	131	4	190.88	DLMS algoritam
3	85	4	143.15	LMS sa protočnom arhitekturom i propagacijom
6	140	4	175.27	LMS sa transpozicijom
8	184	4	378.12	Optimalan broj stepena

VI. ZAKLJUČAK

Eksperimentalnom analizom primene različitih metoda protočne obrade signala, implementacijom dizajna u programabilnom logičkom kolu FPGA tipa utvrđeno je da je moguće višestruko unapređivanje brzine adaptivnih LMS filtera. Protočna obrada može da bude primenjena samo u delu obrade koji se odnosi na ažuriranje koeficijenata ili se željena optimizacija brzine postiže optimizacijom broja stanja. Na osnovu Tabele I može da se zaključi da je u odnosu na originalan LMS algoritam moguće obezbediti pet puta veću brzinu, pri čemu je zauzeće resursa u odnosu na ukupan broj logičkih elemenata u izabranom FPGA kolu ispod 1%. Jedino ograničenje u broju stepena obrade u protočnoj arhitekturi predstavlja parametar μ , koji u slučaju velikog broja stepena obrade mora da se modifikuje kako bi se obezbedila stabilnost rada filtra.

VII. LITERATURA

- [1] C. Cowan, P. Grant: “*Adaptive Filters*”, Englewood cliffs, Prentice Hall, New Jersey, 1985.
- [2] Branko D. Kovačević, Zoran Đ. Banjac, Milan M. Milosavljević: “Adaptivni digitalni filtri”, Akademska misao, Beograd, 2005.

- [3] Divna Mičić, Dragana Prokin, Gabrijela Dimić, “Implementacija LMS algoritma u programabilnoj logici” INFOTEH-JAHORINA Vol. 13, March 2014.
- [4] Ž. Zečević, B. Krstajić, „Povećanje brzine konvergencije FxLMS algoritma u uslovima bijelog šuma“, *TELFOR* 2011. Beograd, Nov. 2011
- [5] Srđan S. Brkić, Dajana M. Lazarević, Predrag N. Ivaniš, “FPGA implementacija sum-product algoritma za dekodovanje LDPC kodova” INFOTEH-JAHORINA Vol. 12, March 2013.
- [6] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Florida State University, 2001.
- [7] C. S. Burrus, “Block implementation of digital filters,” *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 697-701, Nov. 1971.
- [8] M. Petrović, A. Smiljanić, Programiranje Alterinih FPGA čipova, Akademska misao, Beograd, 2008.
- [9] V. A. Pedroni, *Circuit design with VHDL*, Massachusetts Institute of Technology, 2004.
- [10] Z. Jiang and A. N. Willson, “Efficient digital filtering architectures using pipelining/interleaving,” *IEEE Trans. Circuits Syst.*, vol. 44, no. 2, pp. 110-118, February 1994.
- [11] http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf

ABSTRACT

In this paper we present a design flow of the pipelined adaptive FIR filter implemented as DLMS algorithm (eng. *Delayed Least Mean Square*), which is used to minimize the error signal by determining the coefficients of the adaptive filter. Proposed design in comparison with standard implementation of LMS algorithm iteratively improves bandwidth which affects the speed increase with increasing utilization of FPGA logic resources.

IMPLEMENTATION OF THE ADAPTIVE FILTER USING PIPELINING-INTERLEAVING METHOD

Divna Mičić, Dragana Prokin, Gabrijela Dimić