Efikasno izračunavanje Diskretne Furijeove transformacije primenom split-radiks algoritama

Nemanja Stefan Perović
Mentor: prof. dr. Miodrag Popović

***Sadržaj* — U ovom radu dati su rezultati merenja efikasnosti izvršavanja split-radiks i modifikovanog split-radiks algoritama u različitim uslovima. Vršeno je poredjenje njihove efikasnosti kako medjusobno, tako i u odnosu na radiks 2 algoritam. Istovremeno je data analiza njihove efikasnosti u zavisnosti od matematičke kompleksnosti. Dobijeni rezultati dokazuju da smanjenje broja aritmetičkih operacija, kao prednost split-radiksa u odnosu na ranije nastale algoritme, uslovljava njegovu veću efikasnost. Medjutim, smanjeni broj aritmetičkih operacija kod modifikovanog split-radiksa, prilikom testiranja, nije iskazan odgovarajućim smanjenjem vremena izvršavanja.**

***Ključne reči* — Algoritam, Diskretna Furijeova transformacija, matematička kompleksnost, modifikovani split-radiks, split-radiks, vreme izvršavanja.**

I. Uvod

U drugoj polovini dvadesetog veka došlo je do razvoja algoritama za brzo izračunavanje Diskretne Furijeove transformacije (DFT), koji su još poznati pod nazivom nazivom Brze Furijeove transformacije (Fast Furier Transform - FFT) [1].

Split-radiks algoritam (*split-radix algorithm*) predstavlja kombinaciju radiks 2 i radiks 4 algoritama, a u literaturi je još poznat i pod nazivom algoritam sa dvostrukim radiksom [2]. Nastao je osamdesetih godina prošloga veka i u širokoj je upotrebi sve do danas, zbog manjeg broja operacija množenja i sabiranja. Njegovom pojavom uglavnom su izašli iz upotrebe svi ranije nastali algoritmi.

Pre nekoliko godina, došlo je do pojave modifikovanog split-radiks algoritma [3]. Njegova osnovna prednost u odnosu na klasični split-radiks je manji broj matematičkih operacija i to otprilike za 5 do 6 %.

U II delu ovog rada se analizira matematička kompleksnost split-radiksa i modifikovanog split-radiksa, a III deo rada sadrži prikaz i poredjenje vremena izvršavanja u različitim uslovima ovih algoritama, kako medjusobno, tako i sa radiks 2 algoritmom. Na kraju, u IV delu se na osnovu rezultata dobijenih testiranjem izvode zaključci o efikasnosti ovih algoritama.

II. Matematička kompleksnost

A. Kompleksnost split- radiksa

Osnova split-radiks algoritma je njegova dekompozicija na radiks 2 i radiks 4. Na taj način se smanjuje broj matematičkih operacija neophodnih za izračunavanje, čime se ubrzava izvršavanje programa [4].

Aritmetička kompleksnost split-radiksa izračunava se postavljanjem i rešavanjem diferencnih jednačina na osnovu „šeme realizacije“ split-radiksa (izgled leptira). Tako se dobija relacija (1) kojom se izračunava ukupan broj operacija množenja [5] za izračunavanje split-radiks algoritma sekvence dužine N podataka:

$$M(N) = \frac{4}{3} N \log_2 N - \frac{38}{9} N + \frac{2}{9}(-1)^{\log_2 N} + 6. \quad (1)$$

Na sličan način dobija se broj operacija sabiranja:

$$A(N) = \frac{8}{3} N \log_2 N - \frac{16}{9} N - \frac{2}{9}(-1)^{\log_2 N} + 2. \quad (2)$$

B. Kompleksnost modifikovanog split-radiksa

Modifikovani split-radiks predstavlja kombinaciju konjugovanog FFT algoritma (*conjugate-pair split-radix FFT)* [6-9] i split-radiks algoritma, koji imaju istu matematičku kompleksnost, odnosno broj operacija sabiranja i množenja. Tako dobijeni modifikovani split-radiks ima isti broj operacija sabiranja, kao i obični split-radiks, dok je broj operacija množenja smanjen. Ta ušteda se postiže dekompozicijom ulazne sekvence podataka na manje sekvence, korišćenjem s i t parametara i korišćenjem četiri potprograma za izračunavanje.

Osnov za smanjivanje broja aritmetičkih operacija u odnosu na konjugovani FFT algoritam je preskaliranje rotacionih faktora. Ono se vrši množenjem (skaliranjem) rotacionih faktora odgovarajućim parametrima, i to s parametrima. Izračunavanje skalirajućih parametara $s_{N,k}$ vrši se uvodenjem pomoćne promenljive k4 = k mod N/4 u izraz koji glasi:

$$s_{N,k} = \begin{cases} 1, N \le 4 \\ s_{N/4,k_4} \cos(2\pi k_4 / N), k_4 \le N/8 \\ s_{N/4,k_4} \sin(2\pi k_4 / N), drugde \end{cases} \quad (3)$$

Ušteda od dve operacije množenja, u odnosu na množenje rotacionog faktora i kompleksnog podatka, dobija se množenjem t parametra i istog kompleksnog podatka. Izraz za izračunavanje t paremetara glasi:

$$t_{n,k} = W_N^k s_{N/4,k} / s_{N,k}, \quad (4)$$

gde izraz $s_{N/4,k} / s_{N,k}$ uvek ima oblik ili sec (1/cos) ili cosec (1/sin). Iz ove činjenice na kraju proizilazi da je $t_{N,k}$ uvek u nekom od sledećih oblika: ±1±itg ili ±ictg±1.

Modifikovani split-radiks algoritam se softverski realizuje korišćenjem četiri potprograma, koji se pri radu medjusobno „pozivaju“. U radu [3] dati su samo prototipi ovih potprograma. Oni su autoru poslužili kao osnova za razvijanje kompletnih potprograma modifikovanog split-

Nemanja Stefan Perović, Beograd, Srbija (telefon: 381-63-1922-367, email: n.s.perovic@gmail.com)
Miodrag Popović, Elektrotehnčki fakultet, Beograd, Srbija (telefon: 381-11-3370-089, email: pop@etf. rs)

radiks algoritma za potrebe ovog rada, odnosno za testiranje vremena izvršavanja algoritama.

Pri odredjivanju broja operacija množenja, svakom potprogramu odgovara po jedna diferencna jednačina. Stoga, kao rezultat imamo sistem od četiri jednačine, čijim rešavanjem dobijamo broj „uštedjenih" operacija množenja. Konačno, ukupan broj operacija modifikovanog split-radiksa izračunava se kao razlika ukupnog broja operacija konjugovanog FFT algoritma i broja „uštedjenih operacija", što je predstavljeno relacijom (5):

$$\frac{34}{9}N\log_2 N - \frac{124}{27}N - 2\log_2 N - \frac{2}{9}(-1)^{\log_2 N}\log_2 N + \frac{16}{27}(-1)^{\log_2 N} + 8. \quad (5)$$

III. Poredjenje vremena izvršavanja algoritama

Radi poredjenja efikasnosti split-radiksa i modifikovanog split-radiksa, kako medjusobno, tako i u odnosu na radiks 2, kao i utvrdjivanja veze izmedju vremena izvršavanja i odgovarajuće matematičke kompleksnosti, izvršena su merenja vremena njihovog izvršavanja u različitim uslovima:

-vršena su na dva različita kompjutera kako bi se sagledao uticaj različitih hardverskih karakteristika na brzinu izvršavanja algoritama. Tako, prvi kompjuter ima procesor sa taktom 2,8 GHz i keš memorijom od 3 MB, RAM memoriju veličine 4 GB, magistralu podataka širine 32 bajta i učestanosti 800 MHz (u nastavku teksta: 1. kompjuter). Drugi kompjuter ima procesor sa taktom 2 GHz i keš memorijom od 1 MB, RAM memoriju veličine 512 MB, magistralu podataka širine 32 bajta i učestanosti 133 MHz (u nastavku teksta: 2. kompjuter);

-istovremeno, korišćeni su različiti programski kompajleri, koji omogućavaju generisanje i testiranje programa, što je omogućilo i sagledavanje njihovog uticaja na brzinu izvršavanja algoritama. Na 1. kompjuteru je instaliran programski paket VISUAL STUDIO 6, a na 2. kompjuteru je instaliran programski paket VISUAL STUDIO 2005;

-vršena su pri pojedinačnom izvršavanju i izvršavanju algoritama sa višestrukim ponavljanjima (iteracijama), kako bi se minimizovao uticaj vremenskog kvanta prisutan kod pojedinačnog izvršavanja;

-vršena su pri izvršavanju algoritama sa višestrukim ponavljanjima sa pokretanjem exe fajlova, kako bi se eliminisao uticaj različitih kompajlera na rezultate testiranja, dok se mogao pratiti uticaj različitih hardvera;

-takodje, vršena su sa i bez prethodnog izračunavanja rotacionih faktora, radi sagledavanja njihovog uticaja na brzinu izvršavanja;

-koristila se ulazna sekvenca od N podataka, čije su vrednosti: 0, 1, 2, ..., N-1. Na taj način, promenom veličine ulazne sekvence mogao se pratiti uticaj različite količine ulaznih podataka na brzinu izvršavanja algoritama.

Vremena su izražena u milisekundama (ms), a u radu će biti predstavljene srednje vrednosti vremena dobijenih pri testiranju. Programi za izvršavanje odgovarajućih algoritama, odnosno kodovi koji su korišćeni pri testiranju, pisani su u programskom jeziku C.

Rezultati testiranja biće prikazani tabelarno, a zbog uštede prostora kolone u tabeli biće označene rimskim brojevima, i to:

I – radiks 2 bez prethodnog izračunavanja rotacionih faktora, **II** – radiks 2 sa prethodnim izračunavanjem rotacionih faktora;

III – split-radiks bez prethodnog izračunavanja rotacionih faktora, **IV** – split-radiks sa prethodnim izračunavanjem rotacionih faktora;

V – modifikovani split-radiks sa predhodnim izračunavanjem *s* i *t* parametrima (bez prethodnog izračunavanja rotacionih faktora), **VI** – modifikovani split-radiks sa predhodnim izračunavanjem *s* i *t* parametara i rotacionog faktora.

A. Pojedinačno izvršavanje algoritama

Kao što se vidi, u slučaju pojedinačnog izvršavanja algoritama na 1. kompjuteru (VISUAL STUDIO 6) nema razlike izmedju vremena izvršavanja algoritama radiks 2 i split-radiksa, posmatrano u obe verzije: sa i bez prethodno izračunatih rotacionih faktora (tabela 1.), dok na 2. kompjuteru (VISUAL STUDIO 2005) rezultati testiranja, naročito pri većim sekvencama ulaznih podataka, dokazuju prednost split-radiksa u pogledu brzine u odnosu na radiks 2, koja je značajnija ukoliko su prethodno izračunati rotacioni faktori (tabela 2.). Modifikovani split-radiks u obe verzije (na različitim kompjuterima i primenom različitih kompajlera) pokazuje manju brzinu, ne samo u odnosu na split-radiks, nego i u odnosu na radiks 2. Uticaj prethodnog izračunavanja rotacionih faktora je neznatan u okviru iste vrste algoritama.

Tabela 1. Prikaz srednjih vremena izvršavanja algoritama na 1. kompjuteru

N	I	II	III	IV	V	VI
4096	7,5	7,5	7,5	7,5	7,5	7,5
8192	7,5	7,5	7,5	7,5	7,5	7,5
16384	7,5	7,5	7,5	7,5	23	23
32768	23	23	23	23	54	54
65536	54	54	54	54	117	117

Tabela 2. Prikaz srednjih vremena izvršavanja algoritama na 2. kompjuteru

N	I	II	III	IV	V	VI
4096	7,5	7,5	7,5	7,5	15	15
8192	7,5	7,5	7,5	7,5	31	31
16384	32	31	23	21	70	70
32768	74	68	68	68	146	141
65536	262	259	214	191	306	305

Na osnovu analize ovih rezultata zaključuje se da je nemogućnost odredjivanja tačnog odnosa vremena izvršavanja različitih algoritama pri pojedinačnom izvršavanju, naročito pri kraćim sekvencama, posledica postojanja vremenskog kvanta od oko 15 ms, koji prouzrokuje nepreciznost merenja vremena izvršavanja algoritama. Pri tome, kod izvršavanja algoritama na 1. kompjuteru, čije su hardverske i softverske performanse bolje, odnosno koji ima veću brzinu i učestanost u odnosu na 2. kompjuter, uticaj vremenskog kvanta je dominantniji zbog smanjenog ukupnog vremena izvršavanja algoritma.

B. Izvršavanje algoritama sa višestrukim ponavljanjima

Programi za izvršena testiranja, a čiji će rezultati ovde biti prikazani, dobijaju se dodavanjem jedne for petlje na već postojeće programe u dodatku. Na taj način odredjeni algoritam moguće je izvršiti više puta. Inicijalno su razmatrani slučajevi sa 20, 50, 100 ponavljanja (iteracija). Kako razlike u vremenima postaju sve izraženije sa većim

brojem iteracija, prezentovan je slučaj sa 100 iteracija. Pokretanja programa se i ovde vrše iz odgovarajućih VISUAL STUDIO-a, na različitim kompjuterima.

Rezultati testiranja na 1. kompjuteru, pri više iteracija, pokazuju uticaj prethodnog izračunavanja rotacionih faktora u okviru iste vrste algoritama, i to u smislu smanjenja vremena izvršavanja. Dokazana je konstantna prednost u brzini split-radiksa u odnosu na radiks 2, ali samo u slučajevima sa prethodno izračunatim rotacionim faktorima, koja je sve značajnija sa porastom sekvenci ulaznih podataka (tabela 3.).

TABELA 3. PRIKAZ SREDNJIH VREMENA IZVRŠAVANJA ALGORITAMA PRI 100 ITERACIJA NA 1. KOMPJUTERU

N	I	II	III	IV	V	VI
4096	258	242	281	250	609	625
8192	554	539	601	539	1289	1289
16384	1195	1171	1289	1164	2711	2711
32768	2554	2496	2426	2468	5711	5703
65536	5440	5328	5789	5366	/	/

Kod 2. kompjutera rezultati testiranja pokazuju da se, u slučaju prethodnog izračunavanja rotacionih faktora, brzina izvršavanja algoritam radiks 2 povećava, dok se brzina izvršavanja split-radiksa smanjuje. Pri tome, uočava se značajna prednost split-radiksa u brzini izvršavanja u odnosu na radiks 2, kako u slučajevima bez prethodnog izračunavanja rotacionih faktora, a još više u slučajevima sa prethodno izračunatim rotacionim faktorima (tabela 4.).

TABELA 4. PRIKAZ SREDNJIH VREMENA IZVRŠAVANJA ALGORITAMA PRI 100 ITERACIJA NA 2. KOMPJUTERU

N	I	II	III	IV	V	VI
4096	609	562	601	562	1414	1468
8192	1437	1534	1344	1293	3027	3000
16384	3043	3242	2906	2788	42497	42429
32768	6537	7191	6239	6146	156707	158188
65536	23949	24615	18961	18822	/	/

Modifikovani split-radiks konstantno ostaje najsporiji, a ne može se uočiti odredjena zakonitost uticaja prethodnog izračunavanja rotacionih faktora na brzinu njegovog izvršavanja.

Kao što se vidi, primenom više iteracija u testiranju vremenski kvant procentualno gubi značaj, naročito pri dužim ulaznim sekvencama, tako da se dobijaju precizniji podaci vremena izvršavanja algoritama. Samim tim dolaze do izražaja i njihove razlike.

Uopšteno se može uočiti pravilo da je brzina izvršavanja klasičnog split-radiksa, sa prethodno izračunatim rotacionim faktorima, značajno veća od brzine izvršavanja radiksa 2 – uvek kada su u pitanju duže sekvence ulazanih podataka. Ovakvi rezultati su logična posledica razlike izmedju split-radiksa i radiksa 2 u broju aritmetičkih operacija. Naime, ta razlika u broju aritmetičkih operacija postaje sve veća sa povećanjem broja ulaznih podataka, a što se dalje direktno odražava na vreme izvršavanja.

Iako bi bilo logično očekivati istu zakonitost i kod modifikovanog split-radiksa, obzirom na smanjeni broj aritmetičkih operacija, ona nije utvrdjena. Naime, pokazalo se da je prilikom svih testiranja (na različitim kompjuterima i upotrebom različitih kompajlera) modifikovani split-radiks pokazuje manju brzinu, ne samo u odnosu na split-radiks, nego i u odnosu na radiks 2. Njegova prednost u smanjenju broja matematičkih operacija kod izračunavanja DFT definitivno ne može biti iskazana odgovarajućim smanjenjem brzine izvršavanja.

C. Izvršavanje algoritama sa višestrukim ponavljanjima pri pokretanju exe fajlova

Nadalje će biti analizirana vremena izvršavanja exe fajlova kreiranih na 1. komjuteru (VISUAL STUDIO 6), a izvršavanih na 1. kompjuteru (VISUAL STUDIO 6) i 2. kompjuteru (VISUAL STUDIO 2005). Pri tome su napravljene minimalne izmene u odnosu na prethodne verzije programskih algoritama, koje se sastoje u tome da se broj članova ulazne sekvence i broj iteracija (koji je i ovde 100) unose tek po pokretanju programa. Na taj način ostvaruje se mogućnost ispitivanja uticaja hardvera na brzine izvršavanja algoritama.

Exe fajlovi kreirani na 1. kompjuteru (VISUAL STUDIO 6), a testirani na različitim kompjuterima daju različite brzine izvršavanja algoritama, kao posledica različitih hardvera, memorija i ostalih performansi.

Na osnovu rezultata iz exe-fajlova kreiranih i testiranih na 1. kompjuteru (tabela 5.) može se uočiti da, neočekivano, split-radiks nema prednost u pogledu brzine, odnosno sporiji je od radiksa 2. Uticaj prethodnog izračunavanja rotacionih faktora je neznatan, i to kod radiksa 2 neznatno ubrzava izvršavanje algoritma, dok kod split-radiksa usporava.

TABELA 5. PRIKAZ SREDNJIH VREMENA IZVRŠAVANJA EXE FAILOVA NA 1. KOMPJUTERU

N	I	II	III	IV	V	VI
4096	258	250	282	282	610	615
8192	557	541	610	610	1302	1302
16384	1296	1165	1302	1321	2734	2742
32768	2555	2493	2776	2791	5750	5760
65536	5432	5336	5896	5992	/	/

Na osnovu rezultata iz exe-fajlova kreiranih na 1. kompjuteru, a testiranih na 2. kompjuteru (tabela 6.), može se uočiti da klasičan split-radiks ima veću brzinu od radiksa 2. Prethodno izračunavanje rotacionih faktora ima neznatan i promenljiv uticaj na izvršavanje i radiksa 2 i split-radiksa.

TABELA 6. PRIKAZ SREDNJIH VREMENA IZVRŠAVANJA EXE FAILOVA NA 2. KOMPJUTERU

N	I	II	III	IV	V	VI
4096	396	398	383	375	1099	1088
8192	992	984	914	896	2302	2297
16384	2446	2132	1979	1953	43887	43907
32768	4621	4699	4230	4297	142870	140696
65536	18784	19054	15338	15368	/	/

Na osnovu analize prezentiranih rezultata možemo zaključiti da je uticaj karakteristika hardvera na vreme izvršavanja algoritama toliko veliki da dovodi u pitanje prednost u pogledu brzine izvršavanja split–radiksa u odnosu na radiks 2.

Naime, u konkretnom slučaju 1. kompjuter ima hardver veće brzine, odnosno učestanosti tako da se neutrališe osnovna prednost split-radiksa u smanjenom broju aritmetičkih operacija. Osnovni uzrok tome su različite strukture potprograma za izvršavanje radiksa 2 i split-radiksa. Poredjenjem se može zaključiti da je potprogram za izračunavanje radiksa 2 jednostavniji u odnosu na onaj za izračunavanje klasičnog split-radiksa. Naime, pored instrukcija koje se odnose na aritmetičke operacije obuhvaćene kompleksnošću algoritama, u potprogramima postoji i niz instrukcija koje ne služe za direktna izračunavanja. Upravo one mogu da utiču na vreme izvršavanja istih exe fajlova za algoritme radiks 2 i split-radiks na različitim kompjuterima.

IV. ZAKLJUČAK

Obzirom da se prednost split-radiksa u odnosu na prethodne algoritme sastoji u smanjenju broja aritmetičkih operacija, logično je očekivati smanjenje vremena njegovog izvršavanja, kao najznačajniju praktičnu posledicu unapredjenja. Rezultati testiranja izvršenih u pripremi ovog rada uglavnom potvrdjuju ovu prednost. To je naročito slučaj pri izvršavanju split-radiksa sa većim brojem iteracija, sa prethodno izračunatim rotacionim faktorima i ako su sekvence ulaznih podataka veće dužine. Kao što je već ukazano, primenom više iteracija, vremenski kvant srazmerno gubi značaj, naročito pri dužim vremenskim sekvencama, tako da se dobijaju precizniji podaci vremena izvršavanja algoritama, pa samim tim dolaze do izražaja i razlike izmedju algoritama. Takodje, prednost split-radiksa u broju aritmetičkih operacija postaje sve veća sa povećanjem broja ulaznih podataka, što se direktno odražava na veću brzinu izvršavanja ovog algoritma.

Saglasno tome, rezultati testiranja nisu potvrdili prednost split-radiksa u slučajevima pojedinačnog izvršavanja, naročito pri manjim dužinama ulaznih sekvenci. Razlog za to je u nemogućnosti odredjivanja preciznog vremena izvršavanja, usled postojanja za date uslove značajnog vremenskog kvanta.

Medjutim, rezultati testiranja pokazuju značajan uticaj koji na brzinu izvršavanja split-radiksa imaju programski kompajleri, kao i kompjuterski hardveri, na kojima je algoritam izvršavan. Tako, pri radu na kompjuterima sa boljim hardverskim karakeristikama i boljim kompajlerima svi algoritmi se generalno brže izvršavaju. Hardveri veće brzine, odnosno učestanosti neutrališu osnovnu prednost split-radiksa koja se sastoji u smanjenom broju aritmetičkih operacija. Ovo je posledica činjenice da u potprogramima za izvršavanje algoritama, pored instrukcija u okviru aritmetičke kompleksnosti, egzistira i niz instrukcija koje ne služe za direktna računanja (instrukcije grananja, instrukcije sabiranja...), a koje takodje utiču na konačno vreme izvršavanja.

Takodje, kod izvršavanja algoritama na kompjuteru, čije su hardverske i softverske performanse bolje, odnono koji ima veću brzinu i učestanost, uticaj vremenskog kvanta je dominantniji zbog smanjenog ukupnog vremena izvršavanja algoritma. Samim tim, smanjuje se preciznost odredjivanja vremena izvršavanja algoritama, odnosno izražavanja medjusobnih razlika algoritama.

Iako bi, obzirom na smanjeni broj aritmetičkih operacija, bilo logično očekivati istu zakonitost u pogledu smanjenog vremena izvršavanja modifikovanog split-radiksa u odnosu na klasični split-radiks algoritam, ona nije utvrdjena. Prilikom svih testiranja izvršavanja algoritama za izračunavanje DFT, obavljenih za potrebe ovog rada (na različitim kompjuterima i upotrebom različitih kompajlera, prilikom pojedinačnog izvršavanja i izvršavanja sa više iteracija) modifikovani split-radiks pokazuje manju brzinu, ne samo u odnosu na split-radiks, nego i u odnosu na radiks 2. To se može objasniti različitim uticajima:

-dok se kod klasičnog split-radiksa radi samo sa običnim rotacionim faktorima, čije izračunavanje se sastoji od odredjivanja sinusa i kosinusa izračunatog argumenta, a što definitivno dosta kraće traje, kod modifikovanog split-radiksa je zastupljeno računanje s i t parametara i rotacionih faktora, što usporava vreme izvršavanja, naročito pri višestrukim iteracijama, gde se ovi faktori svaki put iznova izračunavaju;

-takodje, prilikom rekurzivnog izračunavanja kod modifikovanog split-radiksa, neophodno je često izvršavanje alokacija memorije, što sigurno utiče na dužinu vremena izvršavanja. Isto tako, iz analize funkcija za izračunavanje modifikovanog split-radiksa, proizilazi da dekompozicija ulazne sekvence (njeno deljenje na manje podsekvence) zahteva odredjeno vreme. Ove dodatne alokacije i dekompozicije nisu postojale kod klasičnog split-radiksa i radiksa 2, što sigurno utiče na veće vreme izršavanja;

-primetno je i da svaka funkcija za izračunavanje modifikovanog split-radiksa na svom početku proverava dužinu ulazne sekvence podataka (odnosno da li je sekvenca dužine jedan, dva ili veće dužine) i u zavisnosti od toga generiše odgovarajuće izlazne podatke. To je prouzrokovano rekurzivnim načinom izvršavanja algoritma. Nesumnjivo, i ove provere iziskuju odredjeno vreme, pa tako doprinose dužem vremenu izvršavanja.

LITERATURA

[1] Dr. Miodrag Popović, *Digitalna obrada signala*, Akademska misao, Beograd, 1994.

[2] P. Duhamel and H. Hollmann, "Split-radix FFT algorithm", *Electronics Letter*, vol. 20, no. 1, pp. 14–16, January 1984.

[3] Steven G. Johnson and Matteo Frigo, "A modified split-radix FFT with fewer arithmetic operations", *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 111-119, January 2007.

[4] P. Duhamel, "Implementation of "split-radix" FFT algorithms for complex, real, and real-symmetric data", *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 34, no. 1, pp. 285–295, February 1986.

[5] H. V. Sorensen, M. T. Heideman and C. S. Burrus, "On computing the split-radix FFT", *IEEE Transactions on Acoustics., Speech, Signal Processing*, vol. 34, no. 1, pp. 152–156, February 1986.

[6] I. Kamar and Y. Elcherif, "Conjugate pair fast Fourier transform", *Electronics Letter*, vol. 25, no. 5, pp. 324–325, March 1989.

[7] R. A. Gopinath, "Comment: Conjugate pair fast Fourier transform", *Electroics Letter*, vol. 25, no. 16, pp. 1084, August 1989.

[8] H.-S. Qian and Z.-J. Zhao, "Comment: Conjugate pair fast Fourier transform", *Electronics Letter*, vol. 26, no. 8, pp. 541–542, April 1990.

[9] A. M. Krot and H. B. Minervina, "Comment: Conjugate pair fast Fourier transform", *Electronics Letter*, vol. 28, no. 12, pp. 1143–1144, June 1992.

ABSTRACT

In this paper, measured results of efficiencies of split-radix and modified split-radix algorithms executions are presented. Their efficiencies were compared with each other and with efficiency of radix 2 algorithm. At a same time, analysis of their efficiencies depending mathematical complexity is presented. Obtained results shows that reducing a number of arithmetic operations, as a split-radix advantage to previous algorithms, conditions its greater effectiveness. But, reduced number of arithmetic operations of modified split-radix, did not show appropriate execution time reduction during testing.

EFFECTIVE COMPUTING OF DISCRETE FOURIER TRANSFORM USING SPLIT-RADIX ALGORITHMS

Author: Nemanja Stefan Perović

Mentor: prof. Miodrag Popović