Digitalna obrada signala

Efikasno izračunavanje DFT

$$X(\tfrac{2\pi}{N}k) = X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad k = 0,1,2,\ldots,N-1$$

Direktna primena za sve odmerke

Jedno kompleksno množenje = 4 realna množenja
Jedno kompleksno sabiranje = 2 realna sabiranja
Jedno W = dve trigonometrijske funkcije sin i cos

1. Broj izračunavanja trigonometrijskih funkcija - $2N^2$
2. Broj realnih množenja - $4N^2$
3. Broj realnih sabiranja - $2N(2N-1) \approx 4N^2$

Cilj -> smanjiti potreban broj izračunavanja

Digitalna obrada signala

Efikasno izračunavanje DFT

sin i cos?

$$e^{-j2\pi k(n+1)/N} = e^{-j2\pi kn/N}e^{-j2\pi k/N}$$

$$e^{-j2\pi k/N} = \cos(2\pi k/N) - j\sin(2\pi k/N)$$

$$\cos[2\pi k(n+1)/N] = \cos(2\pi kn/N)\cos(2\pi k/N) - \sin(2\pi kn/N)\sin(2\pi k/N)$$

$$\sin[2\pi k(n+1)/N] = \cos(2\pi kn/N)\sin(2\pi k/N) + \sin(2\pi kn/N)\cos(2\pi k/N)$$

$\cos(0) = 1$ i $\sin(0) = 0$

$2N^2$ izračunavanja trigonometrijskih funkcija zamenjuje sa $2N$ izračunavanja

ali dodatak

$4N^2$ realnih množenja i $2N^2$ realnih sabiranja

Digitalna obrada signala

Efikasno izračunavanje DFT – Gercelov algoritam

Treba izračunati za mali broj odmeraka << N

$$W_N^{-kN} = e^{j2kN\pi/N} = 1$$

$$X[k] = W_N^{-kN}\sum_{m=0}^{N-1} x[m]W_N^{km} = \sum_{m=0}^{N-1} x[m]W_N^{-k(N-m)}, \quad k = 0,1,2,\dots,N-1$$

što ima oblik konvolucije sekvence $x[n]$ sa sekvencom $W_N^{-kn}u[n]$

$y_k[n]$ kao konvoluciju konačne sekvence $x[n]$ i sekvence $W_N^{-kn}u[n]$

$$y_k[n] = \sum_{m=0}^{N-1} x[m]W_N^{-k(n-m)}$$

$$\boxed{X[k] = y_k[n]\big|_{n=N} = y_k[N]}$$

tj. vrednost DFT na učestanosti $\omega_k = 2\pi k/N$ predstavlja odziv diskretnog sistema čiji je impulsni odziv $W_N^{-kn}u[n]$ na pobudu $x[n]$, a koji je izračunat u trenutku $n = N$.

Digitalna obrada signala

Efikasno izračunavanje DFT – Gercelov algoritam

$$W_N^{-kn}u[n]$$

$$y_k[n] = W_N^{-k}y_k[n-1] + x[n], \quad y_k[-1] = 0$$

$$y_k[n-1] = W_N^{-k}y_k[n-2] + x[n-1]$$

$$y_k[n] = 2\cos(2\pi k/N)\, y_k[n-1] - y_k[n-2] + x[n] - W_N^{k}x[n-1]$$

$$\boxed{\begin{gathered} v_k[n] = 2\cos(2\pi k/N)\, v_k[n-1] - v_k[n-2] + x[n] \\ y_k[n] = v_k[n] - W_N^{k}v_k[n-1] \end{gathered}}$$

FFT

Ako se ulazna sekvenca podeli na dve parcijalne sekvence, i za svaku od njih odredi DFT direktnom metodom, ukupan broj operacija množenja iznosi $2*[4*(N/2)^2]=2N^2$, odnosno upola je manji nego za direktno izračunavanje DFT cele ulazne sekvence. Naravno, dobijeni rezultati u oba slučaja nisu identični, pa je zato potrebno izvršiti dodatne aritmetičke operacije nad parcijalnim DFT da bi se dobio ispravan rezultat. Ukoliko je broj dodatnih množenja znatno manji od $2N^2$ dobija se značajna ušteda u broju potrebnih množenja, ako je dužina sekvence velika. Princip dekompozicije se može dalje primeniti na parcijalne sekvence, sve dok se ne dođe do sekvence koja se ne može više deliti. Na principu dekompozicije su zasnovani svi tzv. brzi algoritmi za izračunavanje DFT, poznati pod opštim nazivom *Brza Furijeova Transformacija* (engl. *Fast Fourier Transform*).

FFT ALGORITAM ZA $N = 2^p$ SA PREUREĐIVANJEM U VREMENU

$\boxed{N = 2^p}$ ako treba i dodavanjem nula

$$X(\tfrac{2\pi}{N}k) = X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad k = 0,1,2,\ldots,N-1$$

Simetrija i periodičnost

$$W_N^{k(N-n)} = W_N^{-kn} = \left(W_N^{kn}\right)^*$$
$$W_N^{k+N/2} = -W_N^k$$
$$W_N^{kn} = W_N^{k(N+n)} = W_N^{n(k+N)}$$

Na primer ako uočimo bilo koji x sa parnim n i koeficijent kojim se mnozi, njegov sledbenik x sa neparnim n+1, se uvek množi sa “istim” koeficijentom “rotiranim” vrednošću

W_N^k i za taj harmonik je konstantno

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{kn}=\underbrace{\sum_{n=2i}x[n]W_N^{kn}}_{\text{parni}}+\underbrace{\sum_{n=2i+1}x[n]W_N^{kn}}_{\text{neparni}},\quad k=0,1,2,\ldots,N-1$$

$$W_N^2=e^{-j4\pi/N}=e^{-j2\pi/(N/2)}=W_{N/2}$$

$$W_N^{2ik}=W_{N/2}^{ik}\ \text{ i }\ W_N^{(2i+1)k}=W_N^kW_{N/2}^{ik}$$

$$X[k]=\sum_{i=0}^{N/2-1}x_{10}[i]W_{N/2}^{ki}+W_N^k\sum_{i=0}^{N/2-1}x_{11}[i]W_{N/2}^{ki},\quad k=0,1,2,\ldots,N/2-1$$

Iz originalne sekvence sa parnim n

Iz originalne sekvence sa neparnim n

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT

$$X[k]=\sum_{i=0}^{N/2-1}x_{10}[i]W_{N/2}^{ki}+W_N^k\sum_{i=0}^{N/2-1}x_{11}[i]W_{N/2}^{ki},\quad k=0,1,2,\ldots,N/2-1$$

$$X[k]=X_{10}[k]+W_N^kX_{11}[k],\quad k=0,1,2,\ldots,N/2-1$$

$$X[k+N/2]=X_{10}[k+N/2]+W_N^{k+N/2}X_{11}[k+N/2],\quad k=0,1,2,\ldots,N/2-1$$

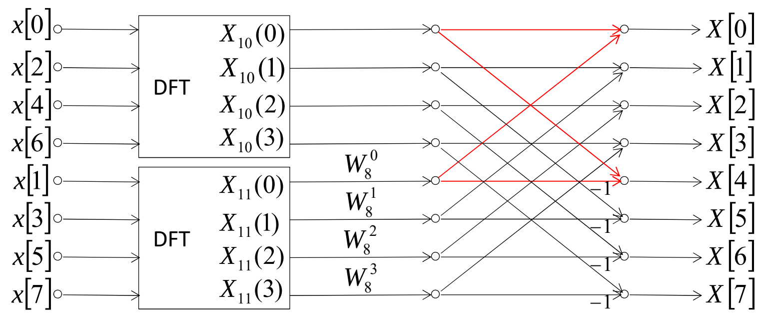
$$X[k+N/2]=X_{10}[k]-W_N^kX_{11}[k],\quad k=0,1,2,\ldots,N/2-1$$

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT

$$X[k] = X_{10}[k] + W_N^k X_{11}[k], \quad k = 0,1,2,\ldots,N/2-1$$

$$X[k+N/2] = X_{10}[k] - W_N^k X_{11}[k], \quad k = 0,1,2,\ldots,N/2-1$$



Strelice putanja. Strelice ulaze u čvor -> sabiranje. Koeficijent na strelici -> množenje.

Leptir operacija $X[k]$ i $X[k+N/2]$ na osnovu $X_{10}[k]$ i $X_{11}[k]$

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT

Direktno - >

1. Broj izračunavanja trigonometrijskih funkcija - $2N^2$
2. Broj realnih množenja - $4N^2$
3. Broj realnih sabiranja - $2N(2N-1) \approx 4N^2$

Podelom

Pošto je za izračunavanje svih DFT odbiraka sekvence od $N/2$ tačaka potrebno $4(N/2)^2 = N^2$ množenja, ukupno je potrebno $2N^2$ množenja za izračunavanje sekvenci $X_{10}[k]$ i $X_{11}[k]$ i dodatnih $4(N/2) = 2N$ množenja za kombinovanje sekvenci $X_{10}[k]$ i $X_{11}[k]$.

Dakle, za izračunavanje DFT podelom sekvence na parni i neparni deo ukupno je potrebno $2N(N+1)$ realnih množenja, što je manje od $4N^2$ ako je $N > 2$. Pošto je po pretpostavci dužina ulazne sekvence $N = 2^P$, sekvence $x_{10}[n]$ i $x_{11}[n]$ imaju paran broj elemenata pa se mogu, u cilju daljeg smanjenja broja operacija, ponovo podeliti na parcijalne sekvence dužine $N/4$ po osnovu parnosti indeksa. Tako se dobija:

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT

$$X_{10}[k]=\sum_{n=0}^{N/4-1}x_{10}[2n]W_{N/2}^{2kn}+\sum_{n=0}^{N/4-1}x_{10}[2n+1]W_{N/2}^{k(2n+1)}=$$

$$=\sum_{n=0}^{N/4-1}x_{20}[n]W_{N/4}^{kn}+W_N^{2k}\sum_{n=0}^{N/4-1}x_{21}[n]W_{N/4}^{kn},\quad k=0,1,2,\ldots,N/4-1$$

$$X_{11}[k]=\sum_{n=0}^{N/4-1}x_{22}[n]W_{N/4}^{kn}+W_N^{2k}\sum_{n=0}^{N/4-1}x_{23}[n]W_{N/4}^{kn},\quad k=0,1,2,\ldots,N/4-1$$

$$\begin{aligned}x_{20}[n]&=x_{10}[2n], & x_{21}[n]&=x_{10}[2n+1]\\ x_{22}[n]&=x_{11}[2n], & x_{23}[n]&=x_{11}[2n+1]\end{aligned}\qquad n=0,1,2,\ldots,N/4-1$$

$$X_{10}[k]=X_{20}[k]+W_N^{2k}X_{21}[k],\quad X_{10}[k+N/4]=X_{20}[k]-W_N^{2k}X_{21}[k]$$

$$X_{11}[k]=X_{22}[k]+W_N^{2k}X_{23}[k],\quad X_{11}[k+N/4]=X_{22}[k]-W_N^{2k}X_{23}[k]$$

$$k=0,1,2,\ldots,N/4-1$$

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT

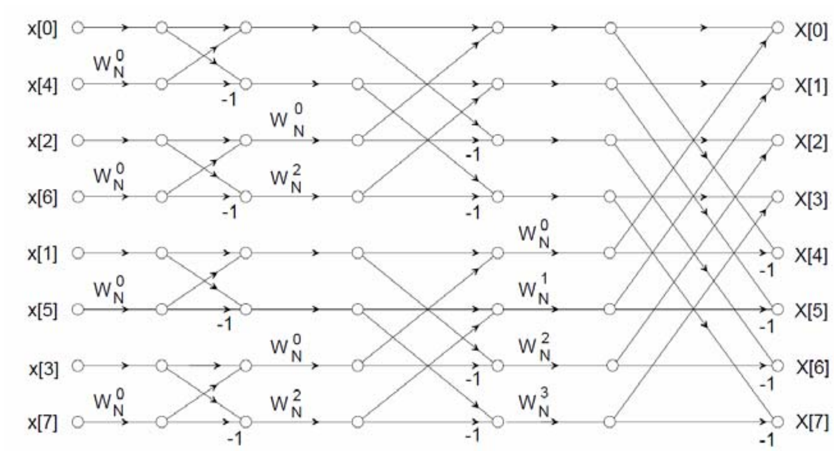
Podelom ulazne sekvence na četiri parcijalne sekvence dalje se smanjuje broj potrebnih aritmetičkih operacija. Pošto je $N=2^p$, postupak podele se može nastaviti. U opštem slučaju, posle m koraka dekompozicije ulazna sekvenca podeljena je na parcijalne sekvence:

$$\begin{aligned}x_{m0}[n]&=x_{(m-1)0}[2n]\\ x_{m1}[n]&=x_{(m-1)0}[2n+1]\\ x_{m2}[n]&=x_{(m-1)1}[2n]\\ x_{m3}[n]&=x_{(m-1)1}[2n+1]\\ &\ldots\ldots\ldots\ldots\end{aligned}\qquad n=0,1,\ldots,N/2^m-1$$

$$\begin{aligned}X_{(m-1)0}[k]&=X_{m0}[k]+W_N^{2^{m-1}k}X_{m1}[k]\\ X_{(m-1)0}[k+N/2^m]&=X_{m0}[k]-W_N^{2^{m-1}k}X_{m1}[k]\\ X_{(m-1)1}[k]&=X_{m2}[k]+W_N^{2^{m-1}k}X_{m3}[k]\\ X_{(m-1)1}[k+N/2^m]&=X_{m2}[k]-W_N^{2^{m-1}k}X_{m3}[k]\\ &\ldots\ldots\ldots\ldots\end{aligned}\qquad k=0,1,\ldots,N/2^m-1$$

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT



Ukupno $p = \log_2 N$ stepena

Ukupno $N_B = \frac{N}{2}\log_2 N$ leptir operacija

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT

$$N_M = 2N\log_2 N$$

$$N_A = 3N\log_2 N$$

$$N_T = N\log_2 N$$

N=4096

49152/33554432=0.00146

		DFT			DIT FFT		
p	N	N_T	N_M	N_A	N_T	N_M	N_A
2	4	32	64	48	8	16	24
3	8	128	256	224	24	48	72
4	16	512	1024	960	64	128	192
5	32	2048	4096	3968	160	320	480
6	64	8192	16384	16128	384	768	1152
7	128	32768	65536	65024	896	1792	2688
8	256	131072	262144	261120	2048	4096	6144
9	512	524288	1048576	1046528	4608	9216	13824
10	1024	2097152	4194304	4190208	10240	20480	30720
11	2048	8388608	16777216	16769024	22528	45056	67584
12	4096	33554432	67108864	67092480	49152	98304	147456

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija – Bit inverzija

Preuređenje podataka - Bit inverzija

Redosled sekvence: 0 1 2 3 4 5 6 7

Redosled koji je potreban: 0 4 2 6 1 5 3 7

preuređenje

0 1 2 3 4 5 6 7

0 4 2 6 1 5 3 7

i tu postoji simetrija

	000	001	010	011	100	101	110	111
1	000	010	100	110	001	011	101	111
2	000	100	010	110	001	101	011	111
	000	001	010	011	100	101	110	111

nove pozicije

$$b_{n-1}b_{n-2}b_{n-3}\ldots b_2b_1b_0 \rightarrow b_0b_1b_2\ldots b_{n-3}b_{n-2}b_{n-1}$$

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija – Bit inverzija

$$b_{n-1}b_{n-2}b_{n-3}\ldots b_2b_1b_0$$

$$N = 2^n$$

$$if\,(b_0 = 1) \quad then \quad novapozicija = N/2 + nesto$$
$$else \quad novapozicija = 0 + nesto$$
$$novapozicija = b_0 * N/2 + nesto$$
$$nesto = funkcija(b_{n-1}b_{n-2}b_{n-3}\ldots b_2b_1)$$

$$if\,(b_1 = 1) \quad then \quad novapozicija = novapozicija + N/4 + nesto^1$$
$$else \quad novapozicija = novapozicija + 0 + nesto^1$$
$$novapozicija = novapozicija + b_1 * N/4 + nesto^1 = b_0 * N/2 + b_1 * N/4 + nesto^1$$
$$nesto^1 = funkcija(b_{n-1}b_{n-2}b_{n-3}\ldots b_2)$$

$\ldots$

$$novapozicija = b_0 * N/2 + b_1 * N/4 + b_2 * N/8 + \ldots = b_0 * 2^{n-1} + b_1 * 2^{n-2} + b_3 * 2^{n-3} + \ldots$$

Pošto se u opisanom algoritmu preuređuje ulazna sekvenca, tj, preuređivanje se vrši u vremenskom domenu, ovakav FFT algoritam naziva se *FFT algoritam sa preuređivanjem u vremenu*. U literaturi na engleskom jeziku ovaj algoritam naziva *decimation-in-time* (*DIT*) *algoritam*, pa se i u našoj terminologiji odomaćio naziv *decimacija* (*desetkovanje*) *u vremenu*. Kako se u poslednje vreme u literaturi iz obrade signala pod decimacijom uglavnom podrazumeva redukcija broja odbiraka, pravilnije je i pogodnije koristiti naziv algoritam sa preuređivanjem u vremenu, ili kraće, DIT algoritam.

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija – In place

POTREBNA MEMORIJA ZA IZRAČUNAVANJE DFT

Ulazna sekvenca kompleksna
Broj potrebnih lokacija N+N
Izlazna je "sigurno" kompleksna
Broj potrebnih lokacija N+N
Sve zajedno 4N

Leptir operacija u m tom stepenu dekompozicije

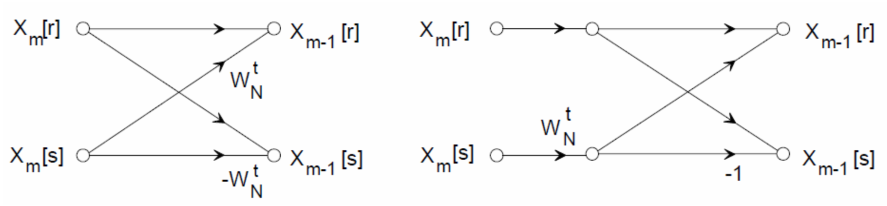
$$X_{m-1}[r] = X_m[r] + W_N^t X_m[s]$$

$$X_{m-1}[s] = X_m[r] - W_N^t X_m[s]$$

$$s = r + N/2^m$$

$$t = -2^{m-1}k$$

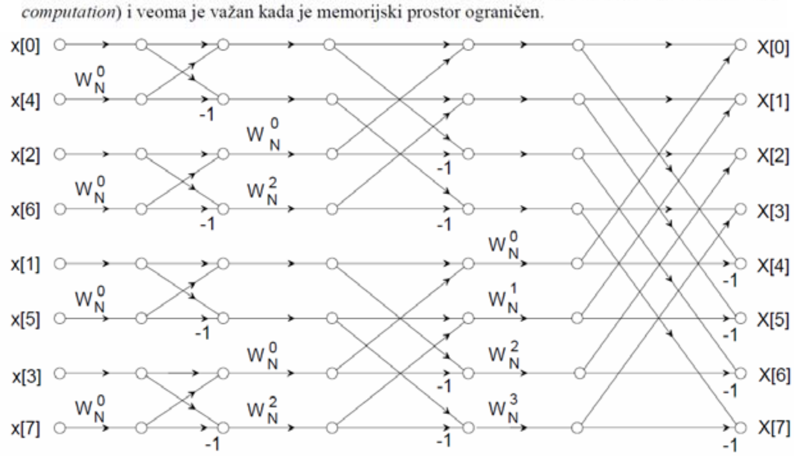
$$k = 0,1,\ldots,N/2^m - 1$$



Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija – In place

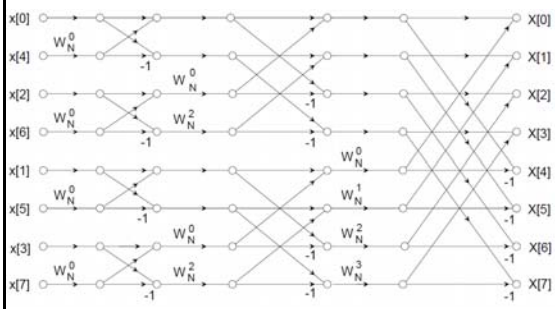
Dakle, u bilo kom stepenu izvršavanja algoritma, podaci čiji su indeksi r i s potrebni su za izračunavanje samo dva nova podatka na istim lokacijama. Ako se izračunavanje obavlja kolonu po kolonu sa slike 5.6, $2N$ memorijskih lokacija u kojima se nalazi ulazna sekvenca može se koristiti za smeštanje međurezultata i izlazne DFT sekvence. Ustvari, potrebna je još samo jedna dodatna memorijska lokacija za smeštanje međurezultata kod izvršavanja svake od leptir operacija. Ovakav efikasan postupak za korišćenje memorije naziva se *izračunavanje u mestu* (engl. *in-place computation*) i veoma je važan kada je memorijski prostor ograničen.



Slika 5.6 Kompletan dijagram toka FFT algoritma sa preuređivanjem ulazne sekvence za $N = 8$.

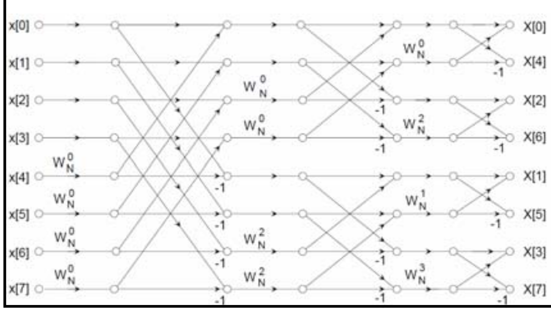
Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT



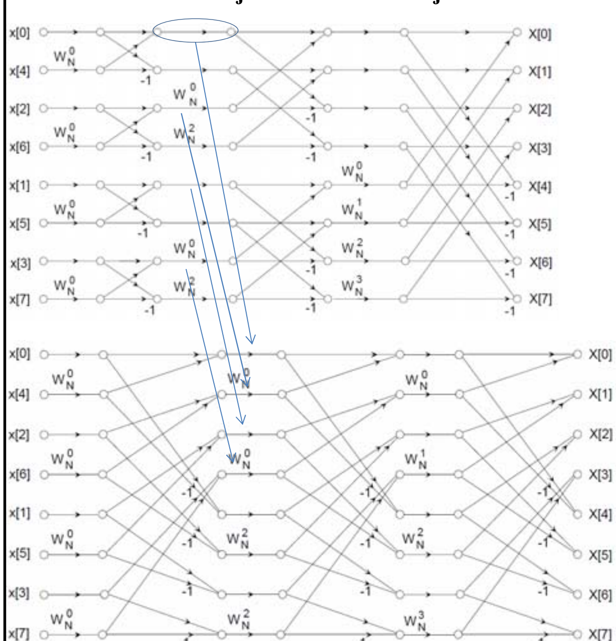
Modifikacije DIT algoritma

Povlačimo celu horizontalnu liniju na gore, tako da ulazni podatak zauzme svoju pravu poziciju. Ne diramo čvorove, koeficijente ....



Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT



Modifikacije DIT algoritma

Ista geometrija leptira.
In place

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIF

FFT ALGORITAM $N = 2^p$ SA PREUREĐIVANJEM U FREKVENCIJSKOM DOMENU

Algoritmi za brzo izračunavanje DFT koji su zasnovani na dekompoziciji izlazne sekvence poznati su pod opštim nazivom *algoritmi sa preuređivanjem u frekvencijskom domenu* (engl. *decimation-in-frequency - DIF*). U izvođenju osnovnog DIF algoritma obično se polazi od sekvence dužine $N = 2^p$ jer je tada proces dekompozicije najjednostavniji. Proces dekompozicije izlazne sekvence vrši se njenom podelom prema parnosti indeksa DFT odbiraka. Odbirci sa parnim indeksima dati su izrazom:

$$X[2k] = \sum_{n=0}^{N-1} x[n] W_N^{2kn} = \sum_{n=0}^{N/2-1} x[n] W_N^{2kn} + \sum_{n=N/2}^{N-1} x[n] W_N^{2kn} =$$

$$= \sum_{n=0}^{N/2-1} x[n] W_N^{2kn} + \sum_{n=0}^{N/2-1} x[n+N/2] W_N^{2k(n+N/2)}, \quad k = 0,1,2,\ldots,N/2-1$$

$$W_N^{2k(n+N/2)} = W_N^{2kn} W_N^{kN} = W_N^{2kn} = W_{N/2}^{kn}$$

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIF

$$X[2k] = \sum_{n=0}^{N/2-1} \left(x[n] + x[n+N/2]\right) W_{N/2}^{kn}, \quad k = 0,1,2,\ldots,N/2-1$$

$$X[2k+1] = \sum_{n=0}^{N-1} x[n] W_N^{n(2k+1)} = \sum_{n=0}^{N/2-1} \left(x[n] - x[n+N/2]\right) W_N^{n} W_{N/2}^{kn}, \quad k = 0,1,2,\ldots,N/2-1$$

Pomocni nizovi

$$x_{10}[n] = x[n] + x[n+N/2]$$

$$x_{11}[n] = \left(x[n] - x[n+N/2]\right) W_N^{n}$$

$$X[2k] = \sum_{n=0}^{N/2-1} x_{10}[n] W_{N/2}^{kn}, \quad k = 0,1,2,\ldots,N/2-1$$

$$X[2k+1] = \sum_{n=0}^{N/2-1} x_{11}[n] W_{N/2}^{kn}, \quad k = 0,1,2,\ldots,N/2-1$$

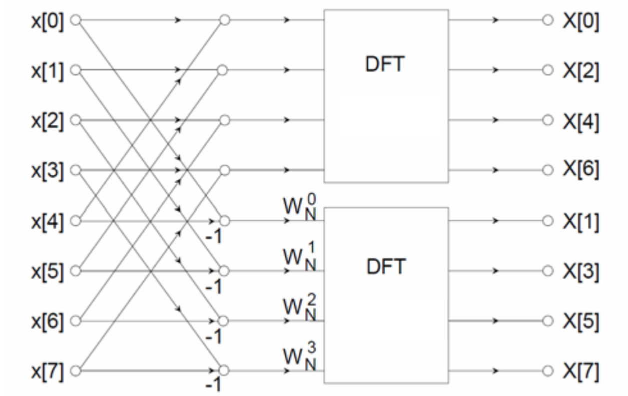
Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIF

Napravimo pomocne nizove pa uradimo DFT u N/2 tačaka

$$x_{10}[n] = x[n] + x[n + N/2]$$

$$x_{11}[n] = \left(x[n] - x[n + N/2]\right)W_N^n$$



Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIF

Kao i u slučaju DIT algoritma, proces dekompozicije se može dalje nastaviti, jer je i $N/2$ paran broj zbog uslova $N = 2^p$. Posle m koraka dekompozicije, dobija se:

$$X[2^m k] = \sum_{n=0}^{N/2^m - 1} x_{m0}[n] W_{N/2^m}^{kn}$$

$$X[2^m k + 2^{m-1}] = \sum_{n=0}^{N/2^m - 1} x_{m1}[n] W_{N/2^m}^{kn} \qquad k = 0,1,\ldots, N/2^m - 1$$

. . . .

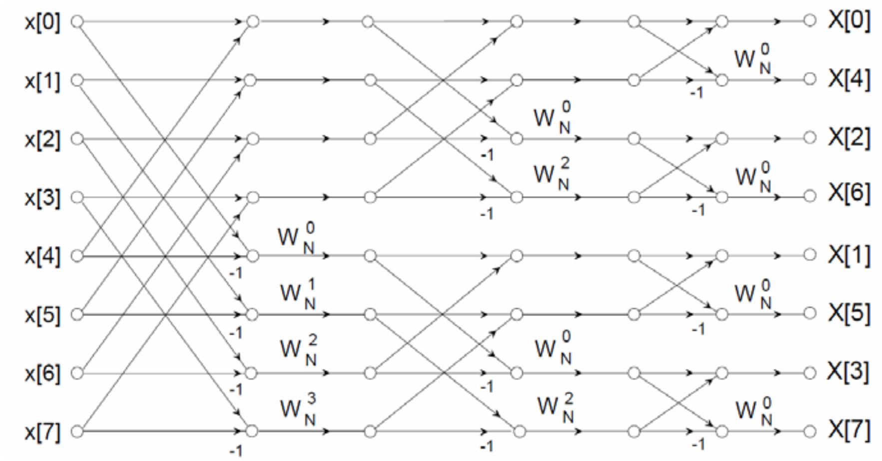
$$x_{m0}[n] = x_{(m-1)0}[n] + x_{(m-1)0}[n + N/2^m]$$

$$x_{m1}[n] = \left(x_{(m-1)0}[n] - x_{(m-1)0}[n + N/2^m]\right) W_N^{2^{m-1} n} \qquad n = 0,1,\ldots, N/2^m - 1$$

. . . .

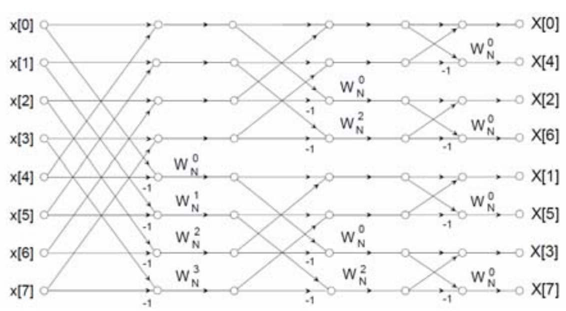
Digitalna obrada signala

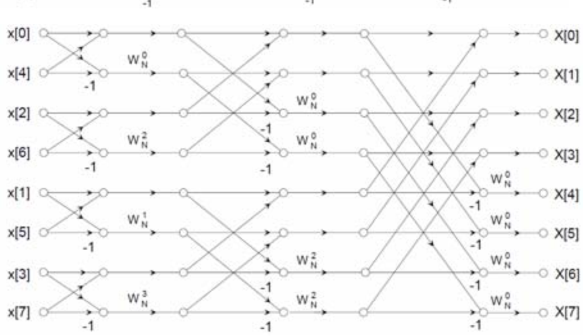
Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIF



Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIF

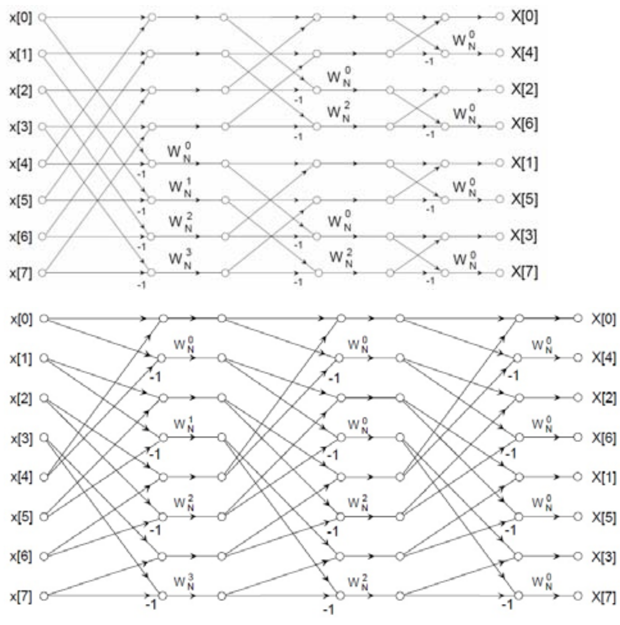




Modifikacije DIF algoritma

Digitalna obrada signala

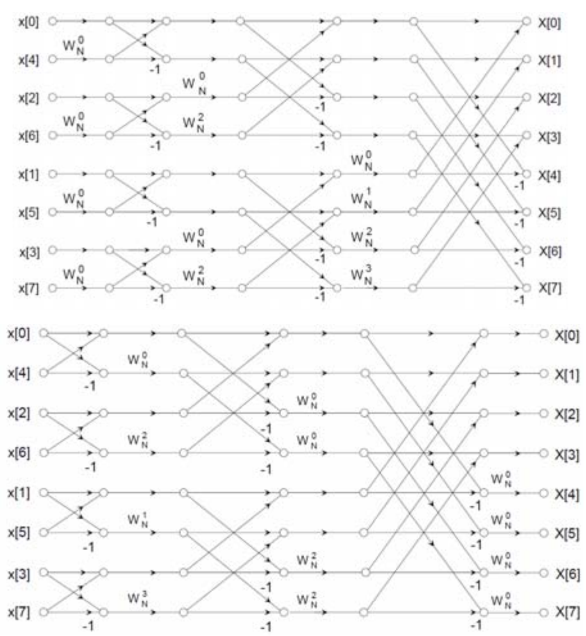
Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIF



Modifikacije DIF algoritma

Digitalna obrada signala

Efikasno izračunavanje DFT – Brza Furijeova transformacija - DIT vs DIF



DIT

DIF