ლაბორატორია 6

DFT და სწრაფი ფურიე გარდაქმნა (FFT)

1 შესავალი

ამ კვირაში განვაგრძობთ დისკრეტული ფურიე გარდამნის (DFT) შესწავლას და სწრაფი ფურიე გარდაქმნის (FFT) MATLAB-ში დანერგვაზე ვიმუშავებთ.

2 DFT-ს ანალიზის გაგრძელება

დისკრეტული ფურიე გარდამნა (DFT) და შებრუნებული დისკრეტული ფურიე გარდაქმნქმნა (IDFT) განსაზღვრულია, როგორც

(DFT) $$X_N[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \tag{1}$$

(IDFT) $$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_N[k]e^{j2\pi kn/N}\,. \tag{2}$$

2.1 სიხშირული დიაპაზონის წანაცვლება

განვიხილოთ DFT-ს განტოლების წარმოდგენა, რომელიც შედარებით უფრო ინტუიტიურია.

პირველ რიგში, შექმენით $N = 20$ სიგრძის ჰემინგის ფანჯარა x, MATLAB-ის ბრძანების, x=hamming(20), გამოყენებით. წინა კვირაში შექმნილი ფუნქციის, DFTsum, გამოყენებით გამოთვალეთ x-ის DFT (20 წერტილი). ააგეთ DFT-ს მაგნიტუდის გრაფიკი, $|X_{20}(k)|$. (ყურადღება მიაქციეთ, რომ DFT ინდექსი k ნულიდან იწყება არა ერთიდან!)

DFT-ის ასეთ გრაფიკს ორი ნაკლი აქვს: DFT-ს მნიშვნელობები, სიხშირე ω-ს ნაცვლად, ინდექსი k-ს მიმართაა წარმოდგენილი; ასევე, DFT-ს სიხშირის ანათვალები, $[-\pi, \pi]$ შუალედის ნაცვლად (როგორც ეს ტრადიციულ DTFT-ის გრაფიკის შემთხვევაშია), $[0,2\pi]$ შუალედზეა განლაგებული. ამის „გამოსასწორებლად“ და გრაფიკის სტანდარტულ $[-\pi, \pi]$ შუალედზე წარმოსადგენად სიხშირეების ვექტორების რადიან/ანათვალი (rad/sample) ერთეულებში გამოთვლა და შემდეგ ვექტორის „მოტრიალებაა“ საჭირო.

პირველ რიგში, განვიხილოთ სიხშირეების ვექტორი w რადიან/ანათვალი ერთეულებში. w-ს ელემენტები შესაბამისი DFT ანათვალების, $X(k)$, სიხშირეებია და შემდეგი ფორმულით გამოითვლება

$$\omega = \frac{2\pi k}{N}, \quad k = 0,1,\ldots,N-1. \tag{3}$$

სიხშირეების $[-\pi, \pi]$ შუალედში განლაგებისთვის $\omega \geq \pi$ სიხშირეები $\omega - 2\pi$ მნიშვნელობებზე გადაგვაქვს. ამისთვის შემდეგი მატლაბის კოდის გამოყენებაა შესაძლებელი: w(w>=pi) = w(w>=pi)-2*pi.

შედეგად მიღებული ორი ვექტორი, X და w, სწორი წევრებისგან შედგებიან, თუმცა მათი გადალაგებაა საჭირო: ვექტორების პირველ და მეორე ნახევრებს ადგილი უნდა შევუცვალოთ. MATLAB-ში ამისთვის ბრძანება, fftshift, ვიყენებთ.

დაწერეთ MATLAB-ის ფუნქცია DTFT ანათვალების და შესაბამისი სიხშირეების ($[-\pi, \pi]$ შუალედში) გამოსათვლელად. გამოიყენეთ სინტაქსი: [X,w]=DTFTsamples(x), სადაც x N-სიგრძის ვექტორია, X DTFT ანათვალების ვექტორია, w კი შესაბამისი წრიული სიხშირეების ვექტორი. თქვენმა ფუნქციამ DTFTsamples, თქვენი DFTsum ფუნქცია და MATLAB-ის fftshift უნდა გამოიყენოს.

გამოიყენეთ DTFTsamples $N = 20$ სიგრძის ჰემინგის ფანჯრის DTFT-ს გამოსათვლელად. ააგეთ DTFT-ს მაგნიტუდის გრაფიკი, სიხშირის ანათვალების მიმართ.

2.2 Zero-Padding

DTFT ანათვალებს შორის სივრცეს DFT-ის სიგრძე განსაზღვრავს. ეს გასაკვირ შედეგს იძლევა, როდესაც ანათვალების რაოდენობა მცირეა. ამ ეფექტრის დემონსტრირებისთვის, განიხილეთ სასრული სიგრძის სიგნალი

$$x[n] = \begin{cases} \sin(0.1\pi n) & 0 \leq n \leq 49 \\ 0 & სხვა \end{cases} \tag{4}$$

თქვენი მიზანია გამოთვალოთ $x[n]$-ს DTFT ანათვალები $N = 50$ და $N = 200$ წერტილიანი DFT-ის გამოყენებით. $N = 200$ შემთხვევაში გამოთვლილი ანათვალების უმეტესობა ნულის ტოლი იქნება, რადგან $x[n] = 0, n \geq 50$. ან მეთოდს zero-padding (ნულებით შევსება) ეწოდება.

$N = 50$ და $N = 200$ შემთხვევებისთვის:

- გამოთვალეთ x ვექტორი.
- DTFTsamples გამოყენებით გამოთვალეთ $X(k)$-ს ანათვალები.
- ააგეთ DTFT ანათვალების მაგნიტუდის გრაფიკი.
- რომელი გრაფიკი წააგავს მეტად DTFT გრაფიკს?

- განიხილეთ რატომ განსხვავდებიან გრაფიკები.

3 სწრაფი ფურიე გარდაქმნის (FFT) ალგორითმი

უკვე ვნახეთ, რომ DFT-ს გამოთვლა საკმაოდ გამოთვლითად ინტენსიური ოპერაციაა. 1965 წელს Cooley და Tukey-მ გამოაქვეყნეს ალგორითმის, რომელიც DTF-ს ეფექტიანი გამოთვლისთვის გამოიყენება. ამ ალგორითმის სხვადასხვა ფორმას სწრაფი ფურიე გარდაქმნა (FFT) ეწოდება. მრავალი მათგანი სხვადასხვა მათემატიკოსის მიერ ბევრად ადრინდელ ნამუშევრებში გვხდევა (ყველაზე ძველი მათგანი გაუსს ეკუთვნის). თუმცა, Cooley და Tukey-ს ნაშრომია (მისი დროულობის გამო), რომელმაც სიგნალების დამუშავებაში რევოლუცია გამოიწვია.

FFT ახალი გარდაქმნა არ არის. ის უბრალოდ DFT-ის გამოთვლის ეფექტიანი მეთოდია. როგორც უკვე იხილეთ, DFT-ის იმპლემენტაციის გამოთვლითი ღირებულება შემავალი სიგნალის სიგრძის კვადრატის პროპორციულად იზრდება ($O(N^2)$). სწრაფი ფურიე გარდაქმნა გამოთვლას ორი მარტივი, თუმცა მნიშვნელოვანი პრინციპის გამოყენებით ამარტივებს. პიველია, „დაყავი და იბატონე“, რომელიც პრობლემის დანაწევრებას აღნიშნავს. მეორე, რეკურსიაა, რომლის გამოყენებითაც „დაყავი და იბატონე“ პრინციპი მეორდება, სანამ ამოცანა ამოიხსნება.

განიხილეთ DFT-ის განმარტების განტოლება იმ დაშვებით, რომ N ლუწია ($N/2$ მთელი რიცხვია):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \tag{5}$$

$x[n]$ სიგნალის N-წერტილიანი DFT-ს ასაღნიშნად ასევე გამოვიყენებთ ნოტაციას

$$X[k] = DFT_N[x[n]]$$

განტოლება (5) შეგვიძლია ორ ჯამად გადავწეროთ. ერთი ჯამი ლუწი n-ის წევრებს შეიცავს, მეორე კი კენტი n-ის წევრებს.

$$X[k] = \sum_{\substack{n=0 \\ n\ \text{ლუწი}}}^{N-1} x[n]e^{-j2\pi kn/N} + \sum_{\substack{n=0 \\ n\ \text{კენტი}}}^{N-1} x[n]e^{-j2\pi kn/N}. \tag{6}$$

ლუწი და კენტი წევრების წერილობით გარჩევის ნაცვლად შეგვიძლია ისინი ჯამებში ცვლადების ცვლილებით გავარჩიოთ. პირველ ჯამში $n \rightarrow 2m$ ჩავანაცვლებთ. m-ის 0-დან $N/2-1$-მდე დაჯამებით, $n = 2m$ ყველა ლუწ მნიშვნელობას მიიღებს 0-დან $N-2$-მდე. ანალოგიურად, მეორე ჯამში გვაქვს $n \rightarrow 2m+1$. ამ შემთხვევაში ჯამის ზღვრებია 0 და $N/2-1$. $n = 2m+1$ ყველა კენტ მნიშვნელობას მიიღებს 0-დან $N-1$-მდე. ასე, შეგვიძლია ჩავწეროთ

$$X[k] = \sum_{m=0}^{N/2-1} x[2m]e^{-j2\pi k2m/N} + \sum_{m=0}^{N/2-1} x[2m+1]e^{-j2\pi k(2m+1)/N}. \tag{7}$$

შემდეგი ნაბიჯი კომპლექსური ექსპონენციალების ექსპონენტების დამუშავებაა

$$X[k] = \sum_{m=0}^{N/2-1} x[2m]e^{-j2\pi km/(N/2)} + e^{-j2\pi k/N} \sum_{m=0}^{N/2-1} x[2m+1]e^{-j2\pi km/(N/2)}. \tag{8}$$

(8)-ში პირველი ჯამის და DFT-ის განმარტების (5) შედარებით, ნახვთ, რომ მათ ზუსტად ერთი და იგივე ფორმა აქვთ, გარდა იმისა, რომ N, $N/2$-ით არის ჩანაცვლებული. შესაბამისად პირველი ჯამი საწყისი მიმდევრობის ლუწ-ინდექსიანი წევრების $N/2$ წერტილიანი DFT-ია. ასევე, (8)-ის მეორე ჯამი საწყისი მიმდევრობის კენტ-ინდექსიანი წევრების $N/2$ წერტილიანი DFT-ია.

შეგვიძლია (8) ამ ინტერპრეტაციის მიხედვით გადავწეროთ. პირველ რიგში, განვსაზღვროთ ორი $N/2$ სიგრძის მიმდევრობა

$$\begin{aligned} g[n] &= x[2n] \\ h[n] &= x[2n+1], \end{aligned} \tag{9}$$

სადაც $n = 0,1,\ldots,N/2-1$. კენტი და ლუწი წევრების ასეთ დანაცწევრებას ***დროითი დეციმაცია*** ეწოდება. ამ განმარტების გამოყენებით სიგნალის, $x[n]$, N-წერტილიანი DFT-ს ვწერთ, როგორც

$$X[k] = G(k) + e^{-\frac{j2\pi k}{N}} H(k), \tag{10}$$

სადაც $G(k)$ და $H(k)$ სიგნალის ლუწი და კენტი წევრების $N/2$-DFT-ია.

$$\begin{aligned} G[k] &= DTF_{N/2}[g[n]] \\ H[k] &= DTF_{N/2}\big[h[n]\big]. \end{aligned} \tag{11}$$

განტოლება (10) N-წეტილიან DFT-სთან შედარებით ნაკლებ გამოთვლას საჭიროებს, თუმცა მისი კიდევ უფრო გამარტივებაა შესაძლებელი. თითოეული $N/2$-წერტილიანი DFT პერიოდულია $N/2$ პერიოდით. ეს ნიშნავს, რომ სიგნალების $G[k]$ და $H[k]$ მხოლოდ

$N/2$ k-ს მნიშვნელობისთვის გამოთვლაა საჭირო, N-ის ნაცვლად. ასევე, კომპლექსური ექსპონენციალისთვის, $e^{-\frac{j2\pi k}{N}}$, გამოვიყენებთ თვისებას $e^{-\frac{j2\pi k}{N}} = e^{-\frac{j2\pi k+N/2}{N}}$.

- ვიზუალურად აჩვენეთ რატომაა ეს თვისება ჭეშმარიტი.

ამ ორი თვისების გამოყენებით გაერთიანებით შეგვიძლია N-DFT-ის მარტივი გამოსახულება ჩავწეროთ:

$$\begin{gathered} X[k] = G(k) + W_N^k H(k), \\ X[k+N/2] = G(k) - W_N^k H(k), \\ k = 0,1,\dots,N/2-1 \end{gathered} \tag{12}$$

სადაც კომპლექსური მუდმივა $W_N^k = e^{-\frac{j2\pi k}{N}}$.

Figure 1-ზე გამოსახულია (12)-ის გრაფიკული ინტერპრეტაცია. დიაგრამის მარცხენა მხარეს სიგნალი ლუწ და კენტ ქვე-სიმრავლეებად არის დანაწევრებული. თითოეული ქვე-სიმრავლის $N/2$-DFT-ს გამოთვლის შემდეგ, კენტი DFT-ის შედეგს შესაბამის მუდმივაზე ვამრავლებთ. გამომავალი სიგნალის პირველი ნახევარი ორი ტოტის მიმატებით, მეორე ნახევარი კი მათი გამოკლებით მიიღება. ასეთი სტრიმ-გრაფები ხშირად გამოიყენება სწრაფი ფურიე გარდაქმნის ალგორითმის აღწერისთვის.

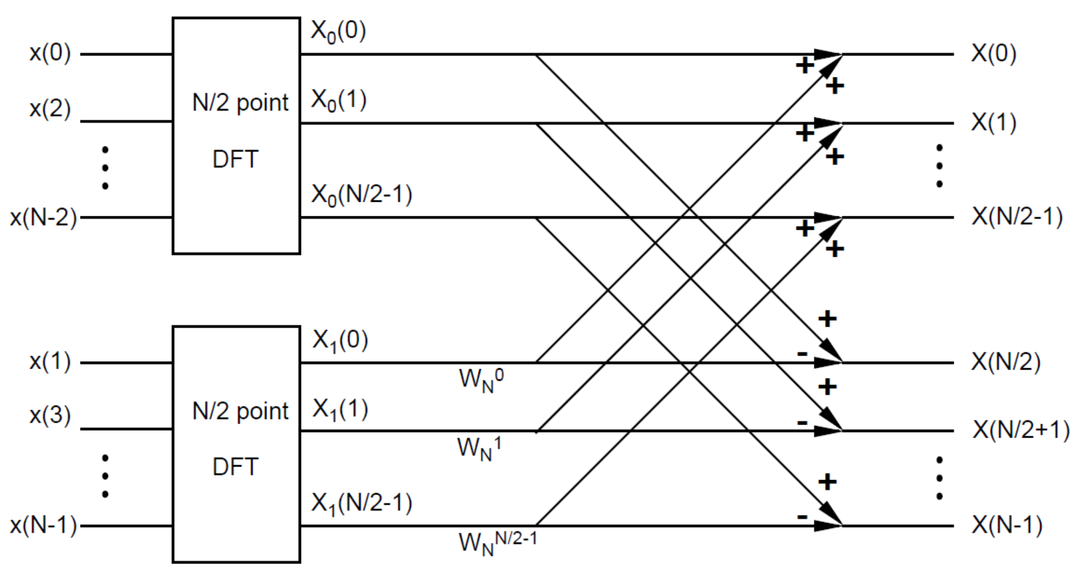

Figure 1 N-DFT გამოთვლა ორი N/2-DFT-ის გამოყენებით

3.1 „დაყავი და იბატონე“: DFT ალგორითმის ინპლემენტაცია

ლაბის ამ ნაწილში თქვენ DFT-ის იმპლემენტაციას განტოლება (12)-ის და Figure 1-ზე გამოსახული სქემის მეშვეობით მოახდენთ.

დაწერეთ MATLAB-ის ფუნქცია X = dcDFT(x), სადაც არგუმენტი x N-სიგრძის ვექტორია და X მისი DFT. თქვენი შექმნილი dcDFT ფუნქციის გამოყენებით:

- დაანაწევრეთ x-ის ელემენტები ლუწ და კენტ წევრებად. ვექტორის ლუწი წევრების მისაღებად შეგიძლიათ გამოიყენოთ ბრძანება x(2:2:N).
- გამოიყენეთ DFTsum ფუნქცია ორი $N/2$-წერტილიანი DFT-ს გამოსათვლელად.
- გაამრავლეთ შესაბამისი წევრები $W_N^k = e^{-\frac{j2\pi k}{N}}$ მუდმივაზე.
- შეაერთეთ ორი DFT X-ის მისაღებად.
- გამოცადეთ dcDFT შემდეგი სიგნალების DFT-ის გამოთვლით
  1. $x[n] = \delta[n], \qquad N = 10$
  2. $x[n] = 1, \qquad N = 10$
  3. $x[n] = e^{j2\pi n/10}, \qquad N = 10$
- გამოთვალეთ ამ გზით N-წერტილიანი DFT-ის გამოთვლისთვის საჭირო გამრავლებების რაოდენობა. (ნამდვილ ან კომპლექსურ რიცხვზე გამრავლება შეგიძლიათ ერთ გამრავლებად ჩათვალოთ). გამრავლებების დათვლისთვის Figure 1-ზე გამოსახული სქემაც გამოიყენეთ.

3.2 რეკურსიული ალგორითმი

FFT ალგორითმის უკან მდგომი მეორე ფუნდამენტური პრინციპი რეკურსიაა. დავუშვათ, რომ $N/2$ ასევე ლუწია. ამ შემთხვევაში შეგვიძლია თითოეული $N/2$-DFT-ს გამოთვლისთვის კვლავ ***დროითი დეციმაცია*** გამოვიყენოთ. ეს პროცესი Figure 2-ზეა ნაჩვენები. ამ შემთხვევაში W_N^k მამრავლებზე გადამრავლება ორ სხვადასხვა ეტაპზე გვიწევს.

რამდენჯერ გვიწევს შემავალი სიგნალის დეციმაციის პროცესის გამეორება? როდესაც N 2-ის ხარისხია, $N = 2^p, p \in \mathbb{Z}$, შეგვიძლია დეციმაციის პროცესის იქამდე გამეორება, სანამ მიღებული ქვე-სიმრავლეები მხოლოდ ორი წერტილისგან შედგება. ამ შემთხვევაში (განტოლება (5)-დან) შეგვიძლია ვნახოთ, რომ ორ წერტილიანი DFT სიგნალის მნიშვნელობების უბრალო ჯამი და სხვაობაა:

$$\begin{aligned} X[0] &= x[0] + x[1] \\ X[1] &= x[0] - x[1] \end{aligned} \tag{13}$$

Figure 3-ზე რვა წერტილიანი DFT ალგორითმის სრული სტრიმ გრაფაა გამოსახული. რამდენი გამრავლებაა საჭირო ამ შემთხვევაში?

დაწერეთ სამი MATLAB-ის ფუნქცია 2,4 და 8-წერტილიანი FFT-ს გამოსათვლელად

X = FFT2(x)

X = FFT4(x)

X = FFT8(x)

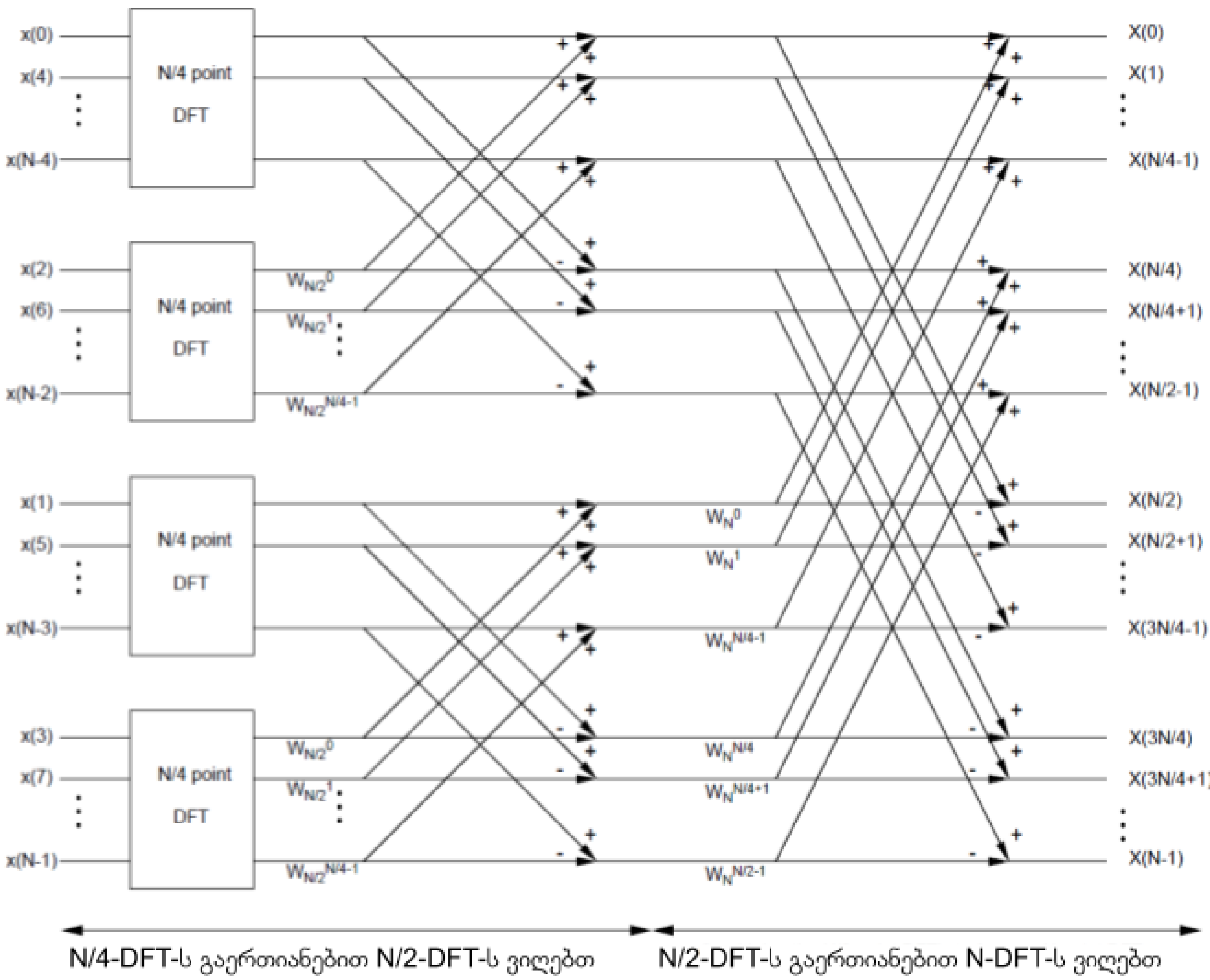

Figure 2 8-წერტილიანი FFT ალგორითმი

FFT2 2-წერტილიან DFT-ს პრდაპირ (13)-ის მიხედვით გამოთვლის. FFT4 და FFT8 ფუნქციებმა შესაბამისი FFT „დაყავი და იბატონე" სტრატეგიის გამოყენებით უნდა გამოთვალონ. (FFT8 ფუნქცია FFT4-ს, ხოლო FFT4 კი FFT2-ს გამოიყენებს).

- გამოცადეთ თქვენი FFT8 ფუნქცია შემდეგი სიგნალების DFT-ს გამოთვლისთვის და შეადარეთ მიღებული შედეგები წინა ნაწილში მიღებულ შედეგებს.
  1. $x[n] = \delta[n], \qquad N = 10$
  2. $x[n] = 1, \qquad N = 10$
  3. $x[n] = e^{j2\pi n/10}, \qquad N = 10$

- გამოთვალეთ ამ გზით N-წერტილიანი DFT-ის გამოთვლისთვის საჭირო გამრავლებების რაოდენობა.
- გამოიყვანეთ $N = 2^p$ წერტილიან FFT-ს შემთხვევაში გამოყენებული გამრავლებების რაოდენობის ფორმულა.
- შეადარეთ თქვენი ფორმულის მიხედვით მიღებული გამრავლებების რაოდენობა პირდაპირი იმპლემენტაციის რაოდენობას $p = 10$ შემთხვევისთვის.

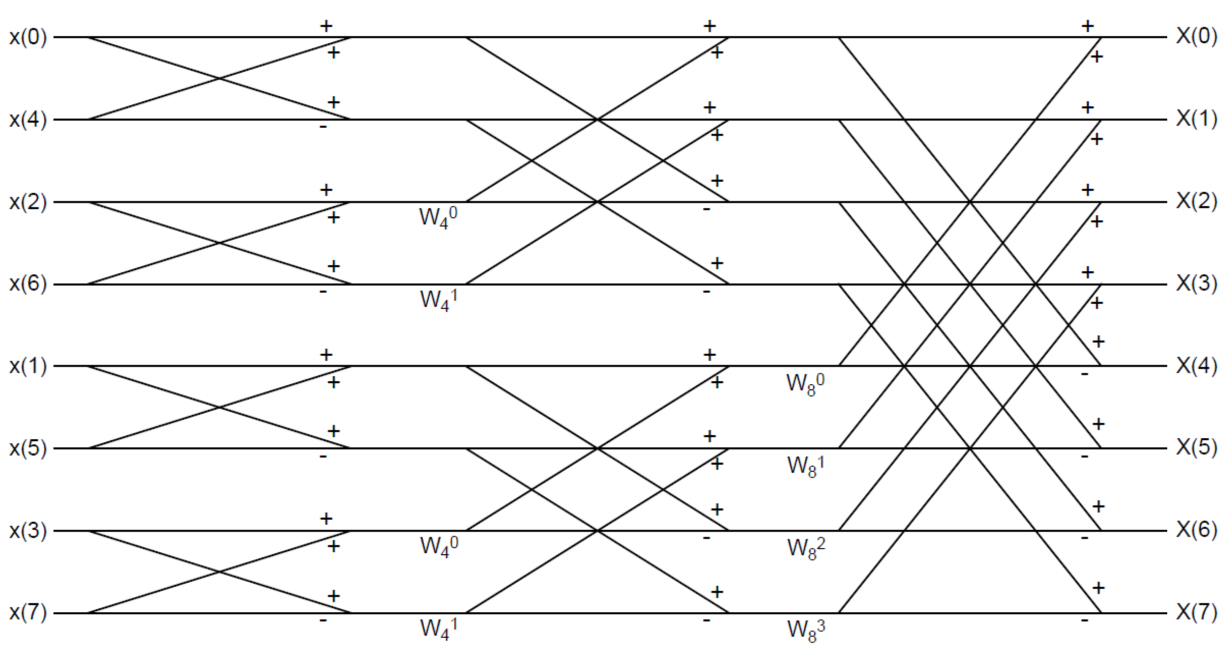

Figure 3 8-წერტილიანი FFT

FFT4 და FFT8 ფუნქციების სწორი იმპლემენტაციის შემთხვევაში მათ თითქმის იდენტური ფორმა უნდა ჰქონდეთ. ერთადერთი განსხვავება შემავალი სიგნალების სიგრძე და $N/2$-DFT-ის გამოსათვლელად გამოძახებული ფუნქცია უნდა იყოს. რა თქმა უნდა, ამ შემთხვევაში სხვადასხვა DFT სიგრძისთვის სხვადასხვა ფუნქციის გამოყენება საჭირო არ არის. ამის გაუმჯობესება რეკურსიული ფუნქციით არის შესაძლებელი. რეკურსიულ ფუნქციას შეწყვეტის პირობა უნდა ჰქონდეს, რომ მან დაუსრულებლად არ განარგძოს თავისი გამოძახება.

დაწერეთ რეკურსიული FFT ფუნქცია X = fft_rec(x), რომელიც FFT ალგორითმის ერთ ეტაპს შეასრულებს 2-ის ხარისხი სიგრძის სიგნალისთვის. ფუნქციის მონახაზი შემდეგია:

1. განსაზღვრეთ შემომავალი სიგნალის სიგრძე.
2. თუ $N = 2$, ფუნქციამ 2-წერტილიანი FFT უნდა გამოთვალოს განტოლება (13)-ის მიხედვით და მნიშვნელობები დააბრუნოს.
3. თუ $N > 2$, ფუნქციამ უნდა მოახდინოს სიგნალის დეციმაცია, N/2-წერტილიანი DFT-ს გამოთვლა და მიღებული შედეგის კომბინირება. N/2-წერტილიანი DFT-ს გამოთვლისთვის ფუნქცია fft_rec-ს გამოიძახებს.

ამ ფუნქციის უმეტესობა ამ ლაბში დაწერილ ფუნქციების მსგავსი იქნება.

- გამოცადეთ fft_rec ფუნქცია ზემოთ მოცემულ სამ სატესტო 8-წერტილიან სიგნალზე და შეადარეთ ის FFT8 ფუნქციით მიღებულ შედეგს.