F1

اگر در DFT ترتیب را به شیوه زیر مرتب کنیم آنگاه محاسبات به شدت کاسته می‌شود

مثال: فرض کنید $f=\{f_i\}_{i=0}^{16}$ یک تابع باشد که بر اساس ضرایب سری خود [illegible] محاسبه شود

$$nC_K = f_0\omega_k^0 + f_8\omega_k^8 + f_4\omega_k^4 + f_{12}\omega_k^{12} + f_2\omega_k^2 + f_{10}\omega_k^{10} + f_6\omega_k^6 + f_{14}\omega_k^{14}$$

$$+ f_1\omega_k^1 + f_9\omega_k^9 + f_5\omega_k^5 + f_{13}\omega_k^{13} + f_3\omega_k^3 + f_{11}\omega_k^{11} + f_7\omega_k^7 + f_{15}\omega_k^{15}$$

$$= (f_0 + f_8\omega_k^8) + \omega_k^4(f_4 + f_{12}\omega_k^8) + \omega_k^2(f_2 + f_{10}\omega^8) + \omega_k^6(f_6 + f_{14}\omega_k^8)$$

A

$$\omega_k^1(f_1 + f_9\omega_k^8) + \omega_k^5(f_5 + f_{13}\omega_k^8) + \omega_k^3(f_3 + f_{11}\omega_k^8) + \omega_k^7(f_7 + f_{15}\omega_k^8)$$

B)

$$F_2 = \left[(f_0+f_8\omega_K^8) + \omega_K^4(f_4+f_{12}\omega_K^8)\right] + \omega_K^2\left[(f_2+f_{10}\omega_K^8) + \omega_K^4(f_6+f_{14}\omega_K^8)\right]$$

$$+\,\omega_K^1\left[(f_1+f_9\omega_K^8) + \omega_K^4(f_5+f_{13}\omega_K^8)\right] + \omega_K^3\left[(f_3+f_{11}\omega_K^8) + \omega_K^4(f_7+f_{15}\omega_K^8)\right]$$

C)

$$= \left(\overbrace{\left[(f_0+f_8\omega_K^8) + \omega_K^4(f_4+f_{12}\omega_K^8)\right]}^{X_{4k}} + \omega_K^2 \overbrace{\left[(f_2+f_{10}\omega_K^8) + \omega_K^4(f_6+f_{14}\omega_K^8)\right]}^{X_{4k+2}} \right)$$

$$+\,\omega_K^1 \left(\overbrace{\left[(f_1+f_9\omega_K^8) + \omega_K^4(f_5+f_{13}\omega_K^8)\right]}^{X_{4k+1}} + \omega_K^2 \overbrace{\left[(f_3+f_{11}\omega_K^8) + \omega_K^4(f_7+f_{15}\omega_K^8)\right]}^{X_{4k+3}} \right)$$

F3

اگر بخواهیم ضرایب C_K ، $K=0,\ldots,15$ را بدست آوریم

نیاز به این است که هر دو پرانتز مرحله آخر ( ) را داشته باشیم

پس با قاعده زیر

$$nC_K = \left(\qquad \right) + \omega_k^1 \left(\qquad \right)$$

به 16 ضرب نیاز داریم اما برای محاسبه هر پرانتز ( )

لازم است که عبارت زیر محاسبه شود

$$\left[\qquad \right] + \omega_k^2 \left[\qquad \right]$$

اگر براکت ها موجود نشد آنگاه به تعداد

$$\omega_k^2 \qquad k=0,\ldots,15$$

ضرب نیاز است اما $\omega = e^{\frac{-2\pi j}{16}}$ پس

$$\omega_k^2 = e^{\frac{-2\pi j}{16} \times 2 \times k} = e^{\frac{-2\pi j}{8} k}$$

$$= \begin{cases} e^{-\frac{2\pi j}{8} k} & k=0,\ldots,7 \\ e^{-\frac{2\pi j}{8} s} \times \cancel{e^{\frac{-2\pi j}{8} 8}}^{\,1} & s=k-8 \quad k>7 \end{cases}$$

F 4

پس تعداد ω_k^2 های مجزا دقیقا 8 تا است پس

هر ( ) به 8 ضرب نیاز دارد در مجموع به 16 ضرب نیاز است

پرانتز بیرونی

اما صرفاً به شرطی که [ ] براکت ها موجود باشد اما برای

محاسبه [ ] به 8 جایگاه عبارت زیر نیازمندیم

$$(\qquad) + \omega_k^4 (\qquad)$$

اگر ( ) پرانتز ها موجود باشد آنگاه به تعداد.

$$\omega_k^4 \qquad k = 0, 1, 2, \ldots, 15$$

ضرب نیاز است اما

$$\omega_k^4 = \begin{cases} e^{-\frac{2\pi j}{4} t} & t = 0, 1, 2, 3 \\ e^{-\frac{2\pi j t}{4}} \times \overbrace{e^{-\frac{2\pi j}{4}(4s)}}^{1} & \end{cases}$$

$$k = 4s + t$$
$$k \geq 4$$

پس تعداد این ω_k^4 4 تا است

F5

پس برای محاسبه پرانتزهای آخر لازم است که ۴ براکت محاسبه

شود در هر براکت به ۴ ضرب نیاز دارد اما پرانتزها موجود بود

یعنی 16 مرحله اما برای محاسبه براکت نیاز محاسبه پرانتزهای درونی

داریم که هر پرانتز درونی از رابطه زیر استفاده می‌کند

$$f_m + \omega_k^8 f_{m+8}$$

تعداد این پرانتزها 8 تاست اما در هر پرانتز به تعداد

$$\omega_k^8 \qquad k = 0, \ldots 15$$

ضرب نیاز داریم

$$e^{-\frac{2\pi j}{16} 8 \times k} = e^{-\pi j k} = \begin{cases} 1 & k \text{ زوج} \\ -1 & k \text{ فرد} \end{cases}$$

پس هر پرانتز دو ضرب رخ می‌دهد برای kهای مختلف

در نهایت 16 ضرب

F6

$nC_k = (\qquad) + \omega_k^4 (\qquad)$ مرحله آخر ۱۶ ضرب

$\underbrace{[\quad] + \omega_k^2 [\quad]}_{\text{8 ضرب}} \to$ ۱۶ ضرب چون دو پرانتز بیرونی است

$(\quad) + \underbrace{\omega_k^4 (\quad)}_{\text{4 ضرب}}$ ۱۶ ضرب چون ۴ براکت لازم است

$\underbrace{f_m + \omega_k^8 f_{m+8}}_{\text{دو ضرب}}$ ۱۶ ضرب چون 8 پرانتز درونی لازم است

پس در مجموع 16×4 ضرب نیاز داریم یعنی

$2^4 \times 4$

و با کمک استقرای ریاضی برای $\{f_i\}_{i=1}^{2^r = n}$ نیاز به

$2^r \times r = n \log n$

ضرب نیاز است

حال به بررسی الگوریتم می‌پردازیم. با ورش وجه می‌رویم

اگر پارامترهای درونی را داشته باشیم می‌توانیم براکت‌ها را

اگر براکت‌ها را داشته باشیم می‌توان پارامترهای بیرونی

را محاسبه کنیم اما دقت در اندیس‌ها هم یک راهنمای مفید است

پس به صورت بازگشتی این محاسبات انجام می‌شود

پارامترهای بیرونی باعث افراز شدن $\{f_i\}_{i=0}^{15}$ به دو دسته زیر

$$\{f_{2k}\}_{k=0}^{8} \quad , \quad \{f_{2k+1}\}_{k=0}^{8}$$

ورود پارامترهای بیرونی که شامل زوج‌ها است به این افراز مجدد انجام

می‌دهد یعنی

$$\{f_{2k}\}_{k=0}^{7} \longrightarrow \begin{cases} \{f_{4k}\}_{k=0}^{3} \\ \{f_{4k+2}\}_{k=0}^{3} \end{cases}$$

F8

۵۹

و پرانتز که شامل فردهاست به شکل زیر عمل می کند

$$\{f_{2k+1}\}_{k=0}^{7} \to \begin{cases} \{f_{4k+1}\}_{k=0}^{3} \\ \{f_{4k+3}\}_{k=0}^{3} \end{cases}$$

یعنی ~~[illegible]~~ و در کل به شرح زیر است

$$\{f_0, \ldots, f_{15}\}$$

$$\{f_0, f_2, f_4, f_6, f_8, f_{10}, f_{12}, f_{14}\} \qquad \{f_1, f_3, f_5, f_7, f_9, f_{11}, f_{13}, f_{15}\}$$

$$\Big(\{f_0, f_4, f_8, f_{12}\} \quad \{f_2, f_6, f_{10}, f_{14}\}\Big)$$

$$\big[\{f_0, f_8\}, \{f_4, f_{12}\}\big] \qquad \big[\{f_2, f_{10}\}, \{f_6, f_{14}\}\big]$$

$$\{f_0\}, \{f_8\} \quad \{f_4\} \{f_{12}\} \quad \ldots$$

نقطه پایه‌ی الگوریتم

F9 ۲۵

```
function Y = FFT1(f)
   n = Length(f); w = exp(-2*pi*1i/n);
   if n==1
      y = f;
   else
      Yzoj = FFT(f(1:2:n));
      Yfard = FFT(f(2:2:n));
      for k = 1:n/2
         t = k-1;   **
         Y(k) = Yzoj(k) + (w^t)*Yfard(k);
         Y(k+n/2) = Yzoj(k) + (w^(t+n/2))*Yfard(k);
      end
   end
end
```

* در متلب چون اندیس از یک
شروع می‌شود تا آخر
اندیس‌ها یک واحد شیفت
پیدا کند که در ** این
افزایش اصلاح می‌شود.

درنهایت Y را بر
n تقسیم می‌کنیم

F9 ۳۶

```
function
Y = FFT1(f)
    n= Length(f);  w= exp(-2*pi*1i/n);
    if n==1
        y=f ;
    else
        yzoj = FFT(f(1:2:n));
        yfard = FFT(f(2:2:n));       *
        for k= 1:n/2
            t=k-1;   **
            Y(k)= yzoj(k) + (w^ t) * yfard(k);
            Y(k+n/2) = yzoj(k) + (w^(t+n/2)) * yfard(k);
        end
    end
end
```

* در متلب چون اندیس از یک شروع می‌شود تا ریز اندیس ها یک واحد شیفت پیدا کند مسئله در ** این افزایش اصلاح می‌شود.

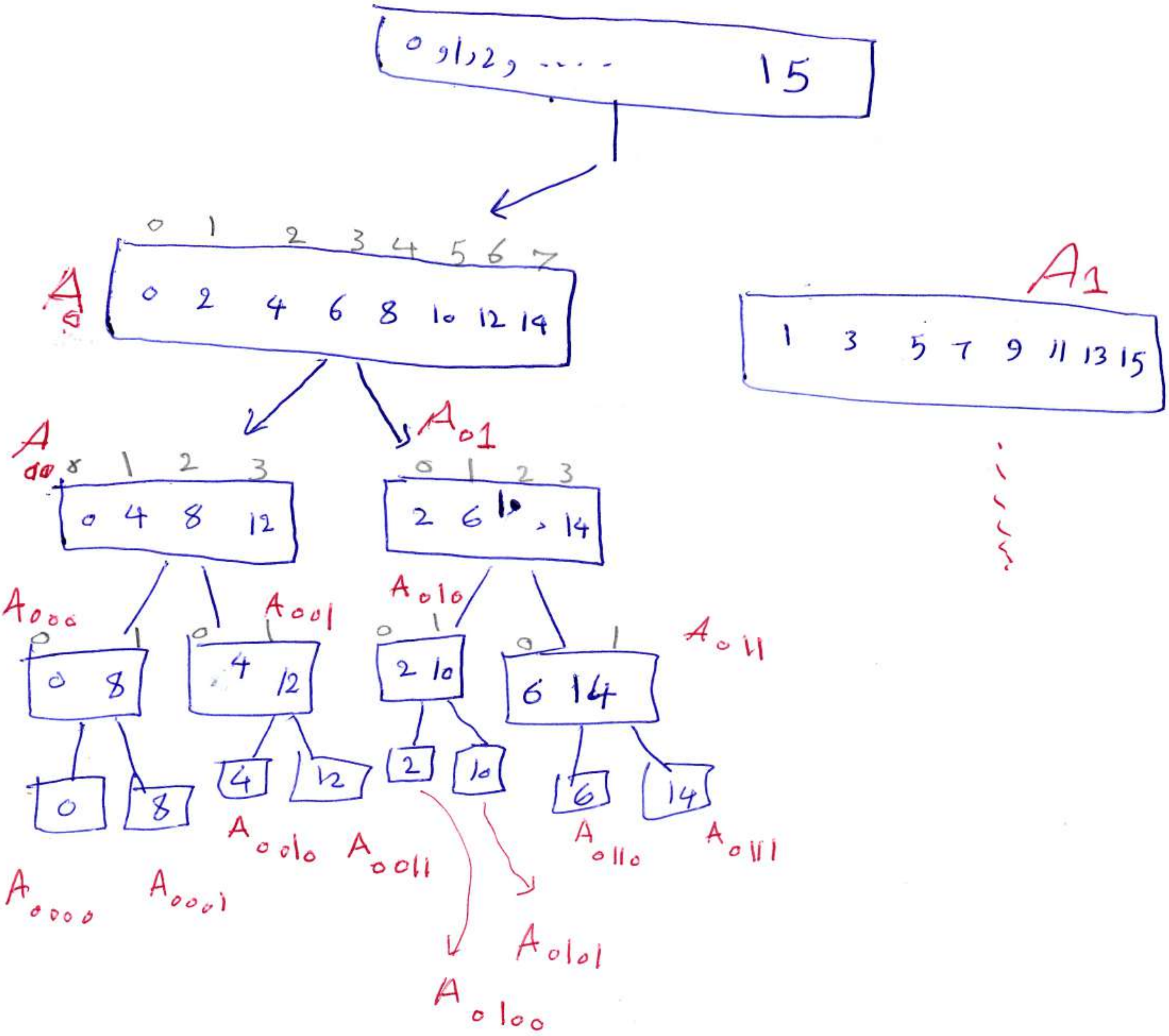
در نهایت Y را بر n تقسیم می‌کنیم

F10

دقت کنیم که این الگوریتم بازگشتی دو بار خودش را فراخوانی خواهد کرد

و خروجی آن بردار روبرو است برای راحتی از f ها

صرف نظر کنیم



F11

ابتدا تابع را با یک بردار 16 تایی $f=[0,1,2,\dots,14,15]$

فراخوانی می‌کنیم این مرحله را A نامیدیم چون $\underset{16}{n} \neq 1$ پس خط 6

لازم اجرا می‌شود و مجدد FFT با عناصر زوج اندیس f یعنی

$$f_0=[0,2,4,6,8,10,12,14]$$

فراخوانی می‌شود این فراخوانی را A_0 نامیدیم چون $n=8\neq 1$

پس لازم خط 6 مجدداً می‌رسیم و مجدد FFT با عناصر زوج اندیس f_0 یعنی

$$f_{00}=[0,4,8,12]$$

فراخوانی می‌شود این فراخوانی را A_{00} نامیدیم چون $n=4\neq 1$

پس لازم به خط 6 مجدداً می‌رسیم و مجدد FFT با عناصر زوج اندیس F_{00} یعنی

$$f_{000}=[0,8]$$

فراخوانی می‌شود این فراخوانی را A_{000} نامیدیم $n=2\neq 1$

پس لازم به خط 6 می‌رسیم و مجدداً FFT با عناصر زوج اندیس F_{000}

یعنی $f_{0000}=[0]$

F12

فراخوان ورودی این فراخوان را A_{0000} بنامیم چون

$n=1$ پس قسمت else اجرا می‌شود و خروجی فراخوان

A_{0000} همان ورودی f_{0000} یعنی [0] می‌باشد

یعنی $y_{zoj}=[0]$ در خط 6 فراخوانی A_{000}

حال در فراخوان A_{000} به خط 7 می‌رسیم در خط 7 می‌رود

اندیس

FFT با عناصر فرد یعنی f_{000} فراخوان می‌شود

$$[8]=f_{0001}$$

این فراخوان را A_{0001} بنامیم چون $n=1$ پس قسمت else اجرا می‌شود

و خروجی همان $[8]=f_{0001}$ می‌شود یعنی $y_{fard}=[8]$ پس

خط 7 در فراخوان A_{000} نیز به اتمام می‌رسد

حال خطوط 8 تا 12 که یک حلقه است برای $n=2$ اجرا می‌شود

$$y(1)=y_{zoj}+w^0 y_{fard} \qquad w_0=e^{-\frac{2\pi j}{2}}$$

$$y(2)=y+w^1 y_{fard}$$

F13

پس خروجی فراخوان A_{000} یک بردار است به شکل

$$[y_1^{000} , y_2^{000}]_{000} = y_{zoj}^{000}$$

حال خط ۶ فراخوان A_{00} با اتمام مورد وارد خط 7

الگوریتم می‌شویم که مجدد یک فراخوان FFT است با عناصر اندیس فرد f_{00} یعنی $f_{001} = [4 , 12]$. بقیه همان شیوه

که کافیست یعنی A_{001} سپس A_{0010} و بعد A_{0011}

که خروجی آن می‌شود

$$[y_1^{001} , y_2^{001}]_{001} = y_{ford}^{001}$$

با اتمام خط 7 فراخوان A_{00} وارد خطوط 8 تا 12

که یک حلقه است می‌شویم

این حلقه منجر به یک خروجی ۴ تایی می‌شود $w = e^{\frac{-2\pi j}{4}}$

$$[y_1^{00} \quad y_2^{00} \quad y_3^{00} \quad y_4^{00}]_{00} = y_{zoj}^{00}$$

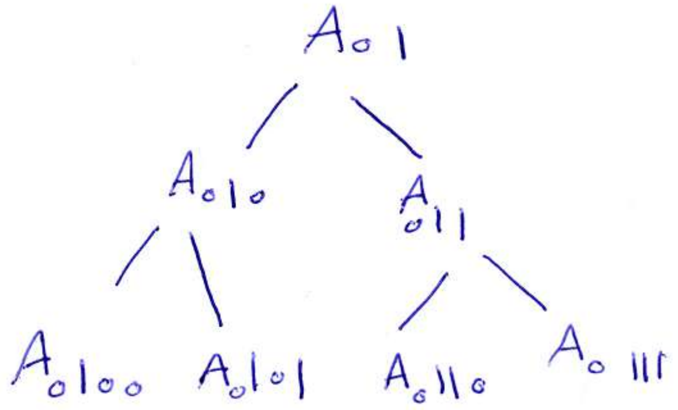
F14

پس خط 6 فراخوان A_0 که فراخوان A_{00} بود به اتمام می‌رسد

وارد خط 7 می‌شویم، بنابراین شبیه خط 6 که A_{00} و

A_{000} و A_{001} که هر کدام نیز دو فراخوان مجدد داشتند

عمل می‌شود و فرایندهای



به همان شیوه که ذکر آن رفت انجام می‌شود و خروجی

$$[y_1^{01}, y_2^{01}, y_3^{01}, y_4^{01}]_{01} = y_{ford}^{01}$$

به دست می‌آید حال خط 7 فراخوان A_0 به اتمام رسید

به خطوط 8 تا 12 که حلقه است وارد می‌شویم

F15

خروجی این حلقه به شرح زیر است $\omega = e^{\frac{-2\pi j}{8}}$

$[y^0_1, y^0_2, y^0_3, y^0_4, y^0_5, y^0_6, y^0_7, y^0_8] = y^0_{zoj}$

حال به خط 6 الگوریتم A می‌رسد و وارد خط 7

الگوریتم 3 شویم عین آنچه که از ابتدای A0 گفته شد

عمل می‌شود و فقط به جای اولین عدد 1 را در f های A ها

قرار می‌دهیم که خروجی آن

$[y^1_1, y^1_2, y^1_3, y^1_4, y^1_5, y^1_6, y^1_7, y^1_8]$

وارد حلقه 8 تا 12 الگوریتم A می‌شویم که خروجی آن

$\omega = e^{\frac{-2\pi j}{16}}$

$[c_0, c_1, c_2, c_3, \ldots, c_{15}]$

FFT این الگوریتم بازگشتی محاسبه ی ضرایب سری

فوریه برای یک تابع گسسته [سیگنال دیجیتال]

در 1960 ابداع شد که موجب انقلابی در حوزه سیگنال

گردید

پیچیدگی DFT = $O(n^2)$

و پیچیدگی FFT $O(n\log_2 n)$

بنابراین FFT با نرخ $\frac{n}{\log_2 n}$ سریعتر از DFT است

مثال اگر $n = 1024$ یعنی سیگنال با 1024 نقطه رشته به

دارا باشد یا تصویر [مثلا سطر یک تصویر که 1024 پیکسل دارد]

آنگاه FFT نسبت به DFT $\frac{1024}{\log_2 1024} \cong 100$

یعنی FFT صد برابر سریعتر از DFT با این n عمل می کند