

بسم الله الرحمن الرحيم



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

مبانی برنامه سازی به زبان C / C++

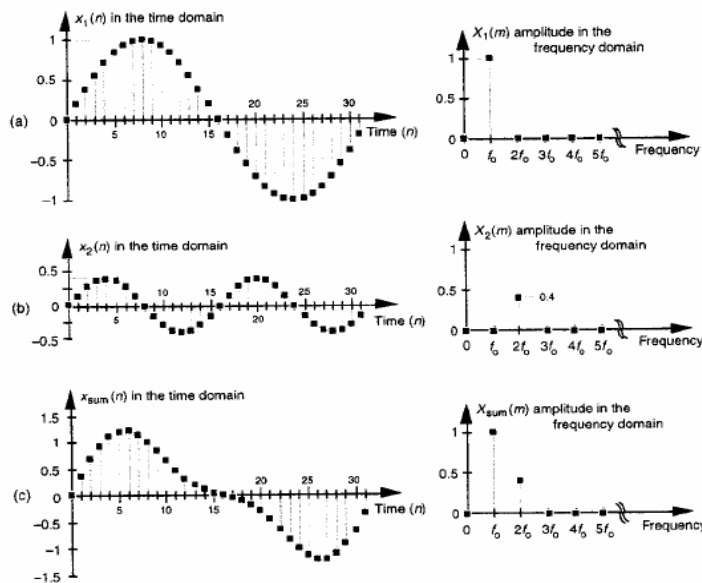
پروژه پایانی

# تبدیل فوریه گسسته سریع

۴-۳ نفر

## • توضیحات :

همواره یک سیگنال در حوزه زمان را می توان در حوزه فرکانس نیز نمایش داد. ( در دیدگاه مهندسی کار با حوزه فرکانس بر حوزه زمان ارجحیت دارد. ) برای مثال سیگنال  $x(n) = \sin(2 * \pi * f_0 * n * t_s)$  را در نظر بگیرید. شکل فرکانسی آن به صورت زیر خواهد بود: به تغییر فرکانس و جمع دو سیگنال توجه کنید :



Time- and frequency-domain graphical representations: (a) sinewave of frequency  $f_0$ ; (b) reduced amplitude sinewave of frequency  $2f_0$ ; (c) sum of the two sinewaves.

برای یک سیگنال پیوسته در زمان تبدیل فوریه آن به صورت زیر تعریف می شود :

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt$$

( به شباهت تبدیل فوق به تبدیل لاپلاس توجه کنید. این تبدیل دقیقاً همان رفتارها را دنبال می کند. )  
مدل گسسته تبدیل فوق به شکل زیر است :

$$X(m) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nm/N}$$

طبیعتاً تبدیل معکوس رابطه فوق به صورت زیر خواهد بود.

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)e^{j2\pi mn/N}$$

نوشتن یک برنامه برای محاسبه تبدیل فوریه از روی سری آسان ولی بسیار کند است. به همین دلیل روش های (بسیار) سریعتری برای محاسبه این تبدیل وجود دارد. ایده اصلی این روش بر مجزاسازی اندیس های فرد و زوج استوار است. بر اساس DFT داریم:

$$X(m) = \sum_{n=0}^{(N/2)-1} x(2n)e^{-j2\pi(2n)m/N} + \sum_{n=0}^{(N/2)-1} x(2n+1)e^{-j2\pi(2n+1)m/N}$$

در این صورت خواهیم داشت :

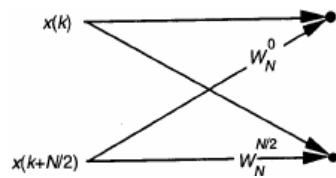
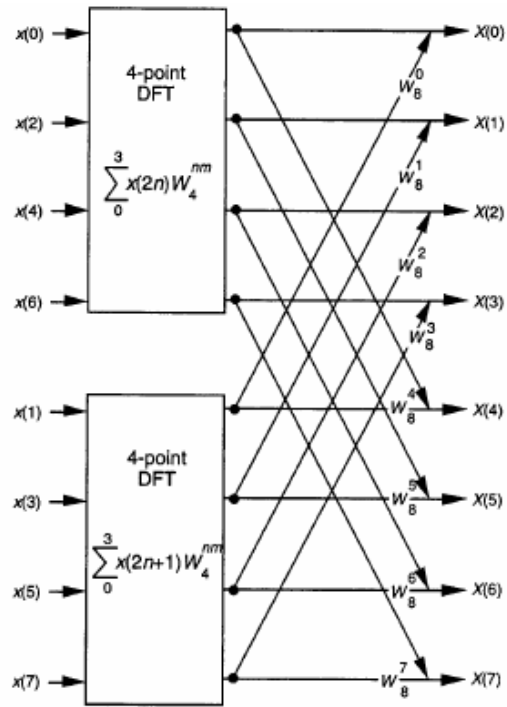
$$\begin{aligned} X(m) &= \sum_{n=0}^{(N/2)-1} x(2n)e^{-j2\pi(2n)m/N} + e^{-j2\pi m/N} \sum_{n=0}^{(N/2)-1} x(2n+1)e^{-j2\pi(2n)m/N} \\ &= \sum_{n=0}^{(N/2)-1} x(2n)W_N^{2nm} + W_N^m \sum_{n=0}^{(N/2)-1} x(2n+1)W_N^{2nm} & W_N = e^{-j2\pi/N} \\ &= \sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nm} + W_N^m \sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nm} & W_N^2 = e^{-j2\pi 2/(N)} = e^{-j2\pi/(N/2)} \end{aligned}$$

و در نهایت :

$$\begin{aligned} X(m) &= \sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nm} + W_N^m \sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nm} \\ X(m+N/2) &= \sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nm} - W_N^m \sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nm} \end{aligned}$$

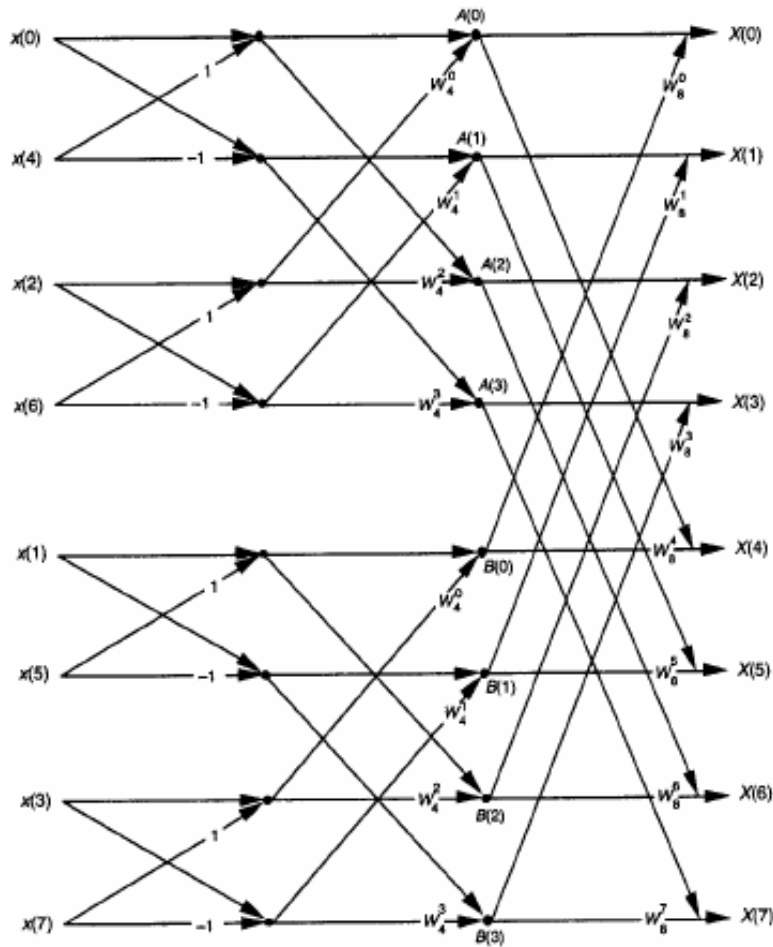
یک فرم بازگشتی ساده، توجه کنید که دیگر از محاسبه exp در این مرحله خبری نیست.

به عنوان مثال ۸ نقطه را می توان از روی ۴ DFT نقطه به صورت زیر محاسبه کرد. ( پروانه!)

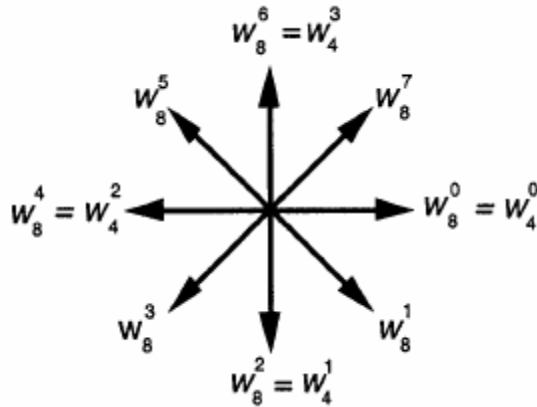


با بلوک های پروانه ای زیر :

و اگر بازگشتی را ادامه دهیم به ساختار زیر می‌رسیم:



برای پیاده‌سازی چنین ساختاری با توجه به تکرار  $W$  ها توصیه می‌کنیم در ابتدا همه  $W$  ها را محاسبه کرده و بعد از آنها استفاده کنید.



البته پیاده‌سازی‌های دیگری برای FFT مانند Radix-4 که شکل عرف آن است و یا Constant-Geometry (مناسب برای پیاده‌سازی‌های سخت‌افزاری FFT بر روی ASIC یا FPGA) نیز وجود دارد.

### شبیه سازی Spice:

در یک سیستم ( مدار ) پاسخ سیستم به سیگنال ضربه معرف کلیه مشخصه های سیستم است. چرا که با پیش آن در سیگنال ورودی می توان سیگنال خروجی را بدست آورد. به عبارت دیگر برای هر سیستم رابطه زیر در حوزه زمان برقرار است:

$$y(t) = x(t) * h(t)$$

اگر تبدیل فوریه عبارت بالا را بنویسیم خواهیم داشت :

$$Y(s)|_{s=j\omega} = (X(s) \times H(s))|_{s=j\omega}$$

( توجه کنید که انتگرال کانولوشن به ضرب عادی تبدیل شده است. ).  $H(s)$  که همان پاسخ فرکانسی ضربه است بر اساس امپدانس المان های مدار به آسانی قابل محاسبه است. به این صورت که به جای  $R$  المانی با امپدانس  $R$ ، به جای خازن المانی با امپدانس  $1/CS$  و به جای سلف از المانی با امپدانس  $LS$  استفاده می کنیم. سپس بر اساس سری یا موازی بودن المان ها سیستم فانکشن یا همان  $H(S)$  را بدست آورده و به ازای  $s=j\omega$  در درایه های تبدیل فوریه سیگنال ورودی ضرب می کنیم تا پاسخ  $Y(s)$  را بدست آوریم. تبدیل معکوس فوریه این سیگنال پاسخ سیستم به سیگنال ورودی می باشد.

- **ورودی/خروجی:**

ورودی خود را از `standard input` بخوانید و خروجی تان را بر روی `standard output` بنویسید.<sup>1</sup>

- **۱. ورودی برنامه:**

ورودی برنامه شما عدد  $n$  تعداد داده های نمونه از سیگنال،  $f$  فرکانس آن و  $n$  عدد نمونه های سیگنال می باشد.  
در هنگام تحویل پروژه به برنامه شما یک سیگنال صدا به مدت ۵ ثانیه که در فرکانس  $8192 \text{ Hz}$  نمونه برداری شده داده خواهد شد.

- **۲. خروجی های برنامه:**

- برنامه شما می بایست DFT سیگنال ورودی حوزه زمان را محاسبه کرده و بر اساس فرکانس بر روی نمودار نمایش دهد.
- یک فیلتر FIR از نوع `low pass filter` یا `band pass filter` را تعریف کرده و بر روی سیگنال فوق اعمال نماید.
- از سیگنال حوزه فرکانس ساخته شده تبدیل گسسته معکوس فوریه بگیرد. برای اینکار نیازی نیست از ایده های الگوریتم های `Fast Fourier Transform` استفاده کنید.
- شبیه سازی `spice`: یک مدار شامل عناصری مانند `R`, `L`, `C` را بر روی سیگنال فوق اعمال کنید و خروجی را نشان دهید.

توجه کنید که برای همه این اعمال در برنامه شما می بایست در زمان مناسبی پاسخگو باشد.

- **توجه بسیار مهم:**

برنامه شما باید در زمان تحویل کمپایل و اجرا شود و بتواند برای تست های ورودی خروجی صحیح ایجاد کند. کد منبع پروژه خودتان را به همراه مستندات مربوط به آن را در غالب یک فایل فشرده<sup>۲</sup> به نام `DFT` تا زمان تحویل به آدرس `ce153c@gmail.com` ارسال نمایید. فایل `subject` نامه خود را برابر با `Project2-dft` (`8510xxxx-8510xxxx-8510xxxx`) قرار دهید که در آن `xxxx` چهار رقم انتهایی شماره دانشجویی اعضای تیم شما می باشد.  
لطفاً توجه کنید که تحویل پروژه ها به صورت حضوری می باشد.

---

<sup>1</sup> برای این کار از `header file` های `stdio.h` (برای زبان C) و فایل `iostream` (برای زبان C++) استفاده کنید.

<sup>2</sup> با فرمت `rar`, `zip` یا `tar`

• **مراجع** می توانید از مراجع زیر برای طراحی و پیاده سازی برنامه تان کمک بگیرید.

1. **Openheim** et all, *Signals & Systems* .
2. **Openheim** et all, *Disrete-Time signal processing*.
3. **Jorg Arndt**, *Algorithms for Programmers*
4. **Stergiopoulos** (Editor), *Advanced Signal Processing Handbook Theory And Implementation For Radar, Sonar, And Medical Imaging Real-Time*
5. **Poularikas** , *Handbook of Formulas and Tables for Signal Processing*
6. **Proakis & Manolakis** , *Digital Signal Processing - Principles, Algorithms & Applications*
7. **Smith**, *Guide to Digital Signal Processing*

دستور *fft* در *Matlab* برای محاسبه تبدیل فوریه به کار می رود. می توانید از آن برای تست درستی برنامه تان استفاده کنید. به علاوه مطلب! در شاخه `toolbox\matlab\audiovideo` تعدادی فایل `sample` نگهداری می کند که با دستور `sample=load('-mat', 'laughter.mat')` قابل بارگذاری می باشند. این ها تعدادی فایل صوتی هستند که در فرکانس  $8192 \text{ Hz}$  نمونه برداری شده اند. برای دستیابی به داده ها از `sample.y` استفاده کنید. برای انتگرال کانوولوشن ( برای فیلتر کردن ) می توانید از `conv` را اجرا کنید و در نهایت با `Plot` یا `stem` می توانید نمودار را بکشید. بد نیست نگاهی به جعبه ابزار DSP مطلب هم بیندازید.

موفق باشید.

احمدیان