

Final Project

Frequency Analysis System



Department of Electrical Engineering
National Cheng Kung University

Video Link : <https://youtu.be/myyFNzLfjWI>

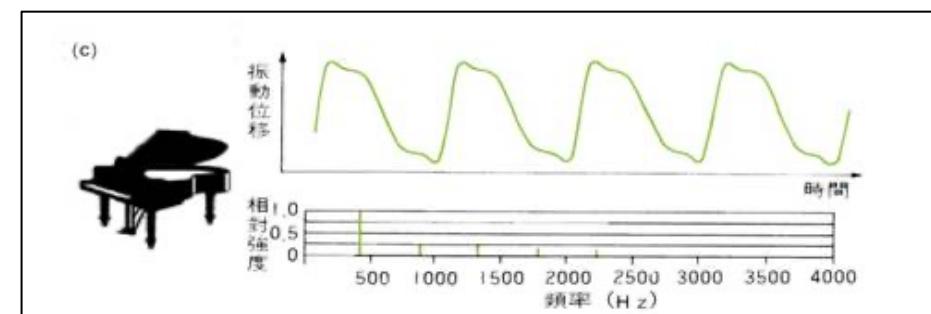
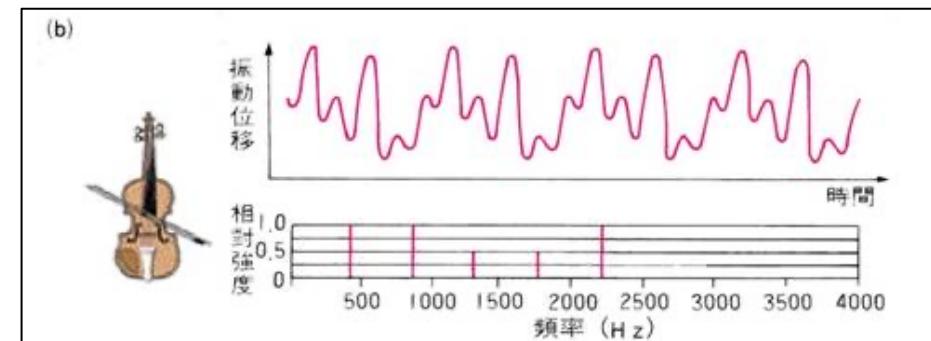
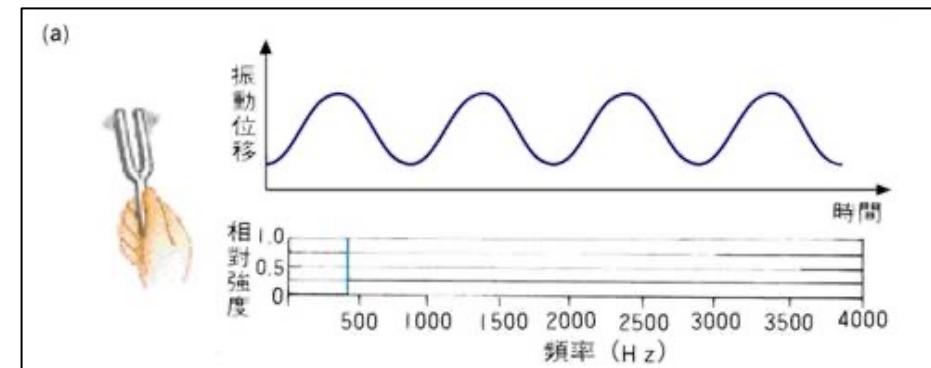
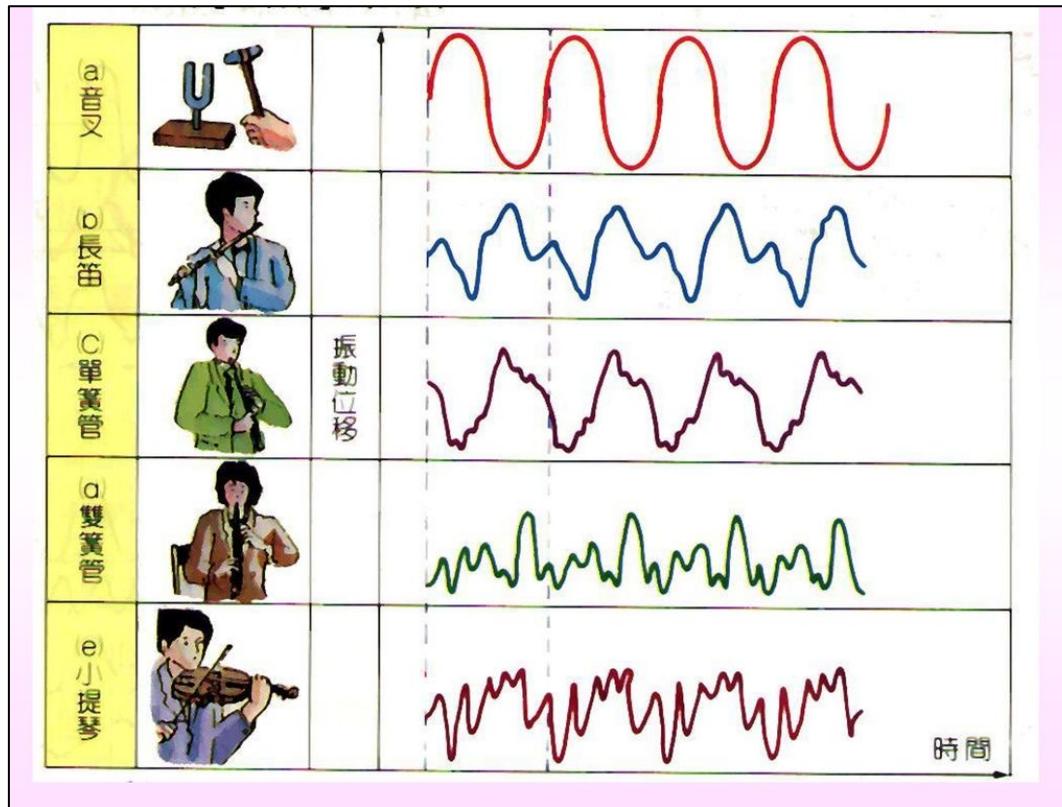
Reference

- 圖解傅立葉分析
 - <https://hackmd.io/@sysprog/fourier-transform>
- FIR Lowpass Filter Example
 - https://www.keil.com/pack/doc\CMSIS/DSP/html/group_FIRLPF.html
- 從影片中提取音訊資料，然後應用傅立葉對音訊降噪（python）
 - <https://www.gushiciku.cn/pl/p7uE/zh-tw>

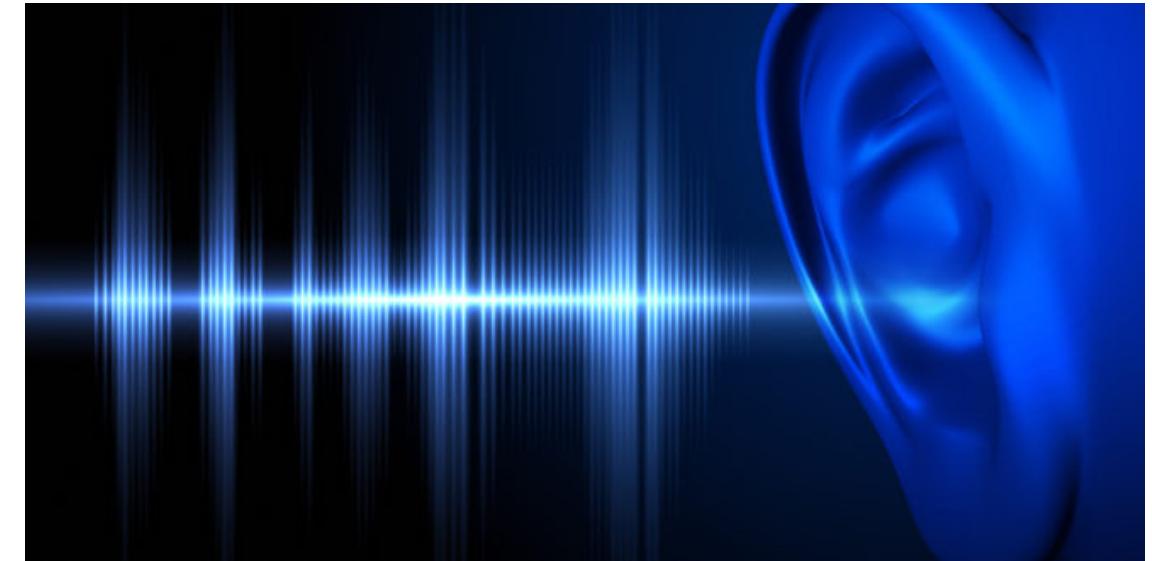
Timbre (音色)

- Why human ear can distinguish when different instruments play the same pitch(音高)?

基頻 & 泛音



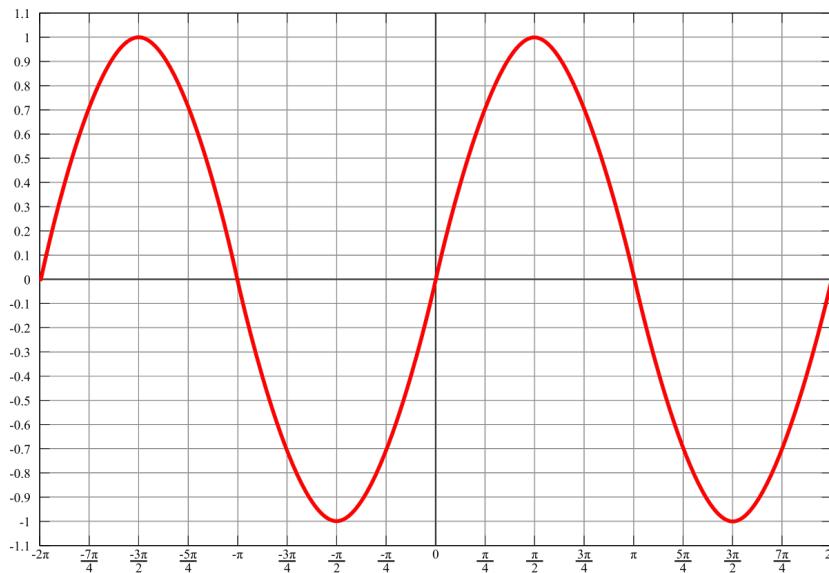
What's Sound ?



- Multiple waves of different frequencies with different amplitudes & different phases are interleaved(交織) and transmitted through the medium(介質) to your ears

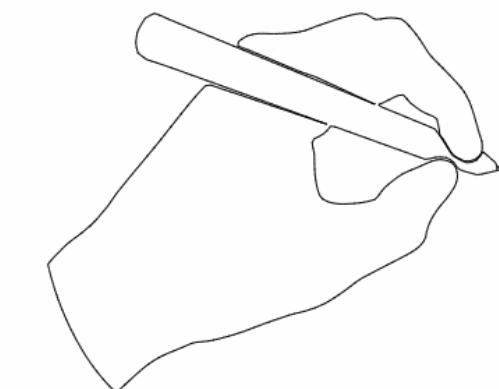
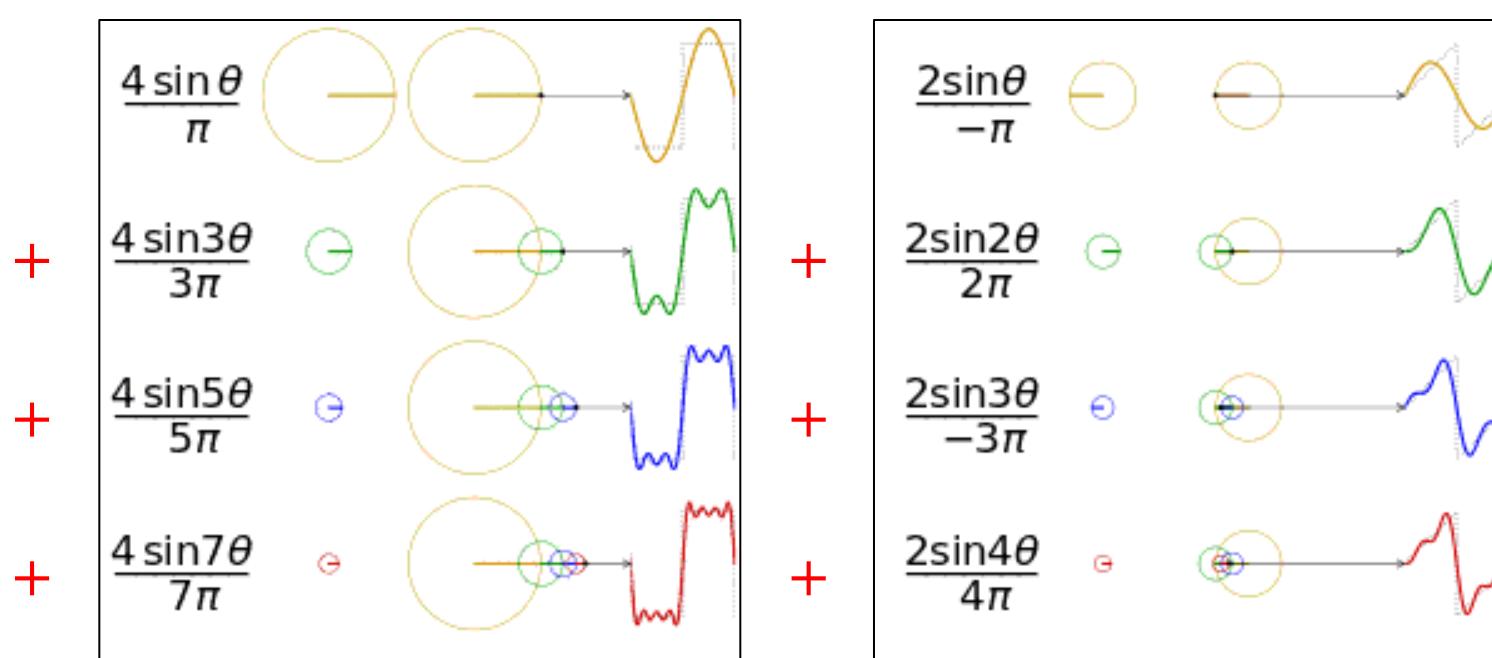
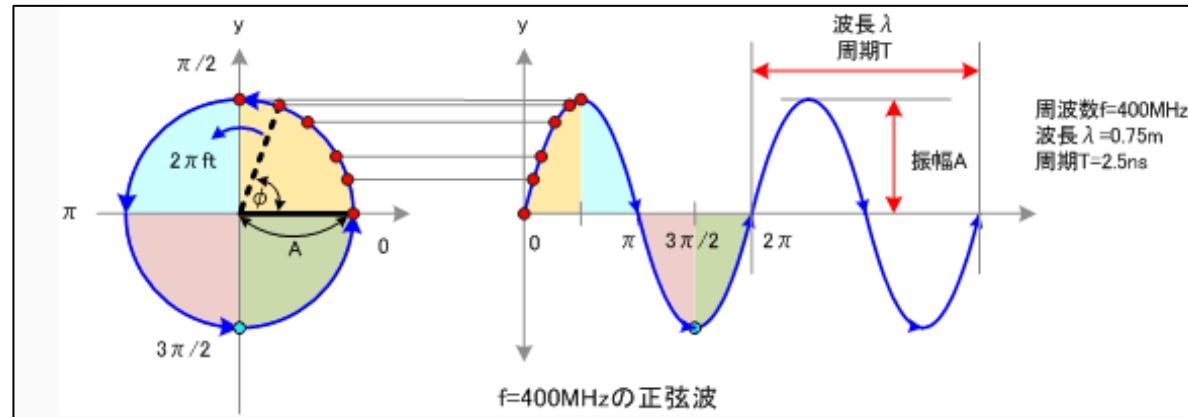
History - Fourier

- 傅立葉 ([Joseph Fourier](#), 1768-1830) 生於法國藝術和歷史之都 Auxerre，因研究熱傳導理論名於世，他提出的傅立葉級數方法是工程數學的重要工具。
- 1807 年，時年 39 歲的傅立葉於法國科學學會上展示一篇論文《固體中的熱傳導》(On the Propagation of Heat in Solid Bodies)，包含當時極具爭議的論斷：
「任何連續周期信號可由一組適當的正弦曲線組合而成」

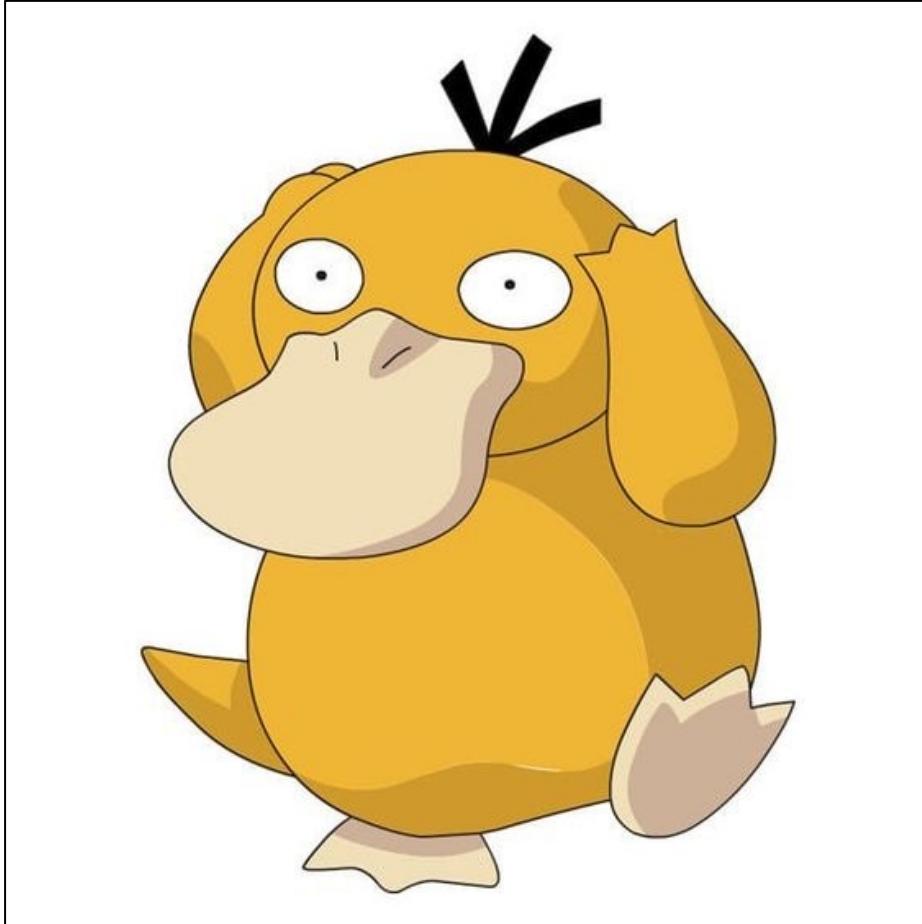


Sine Wave

The Effect of different frequency combination

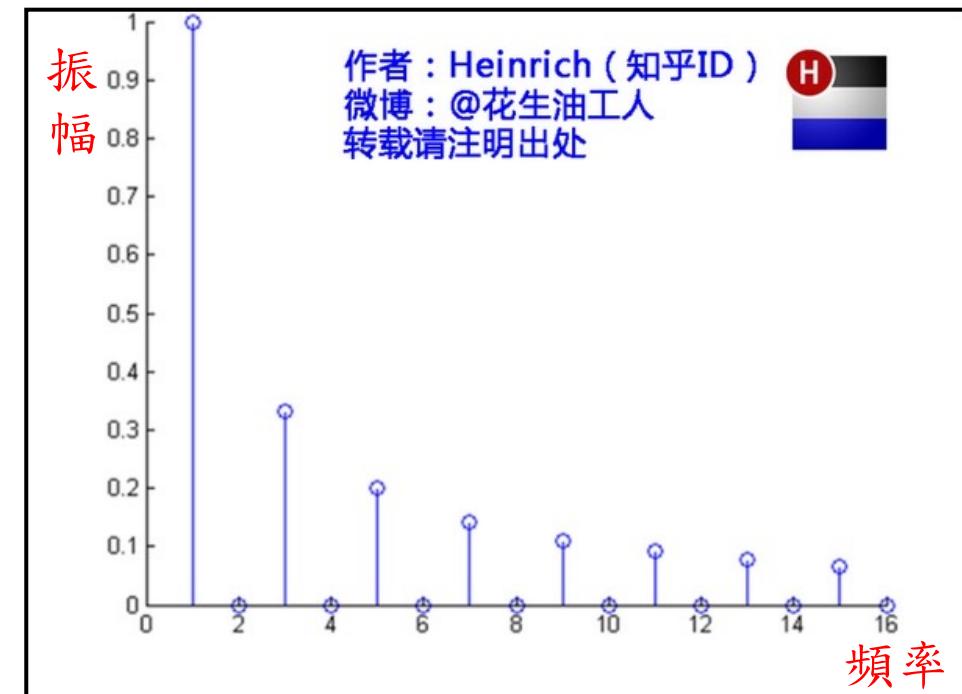
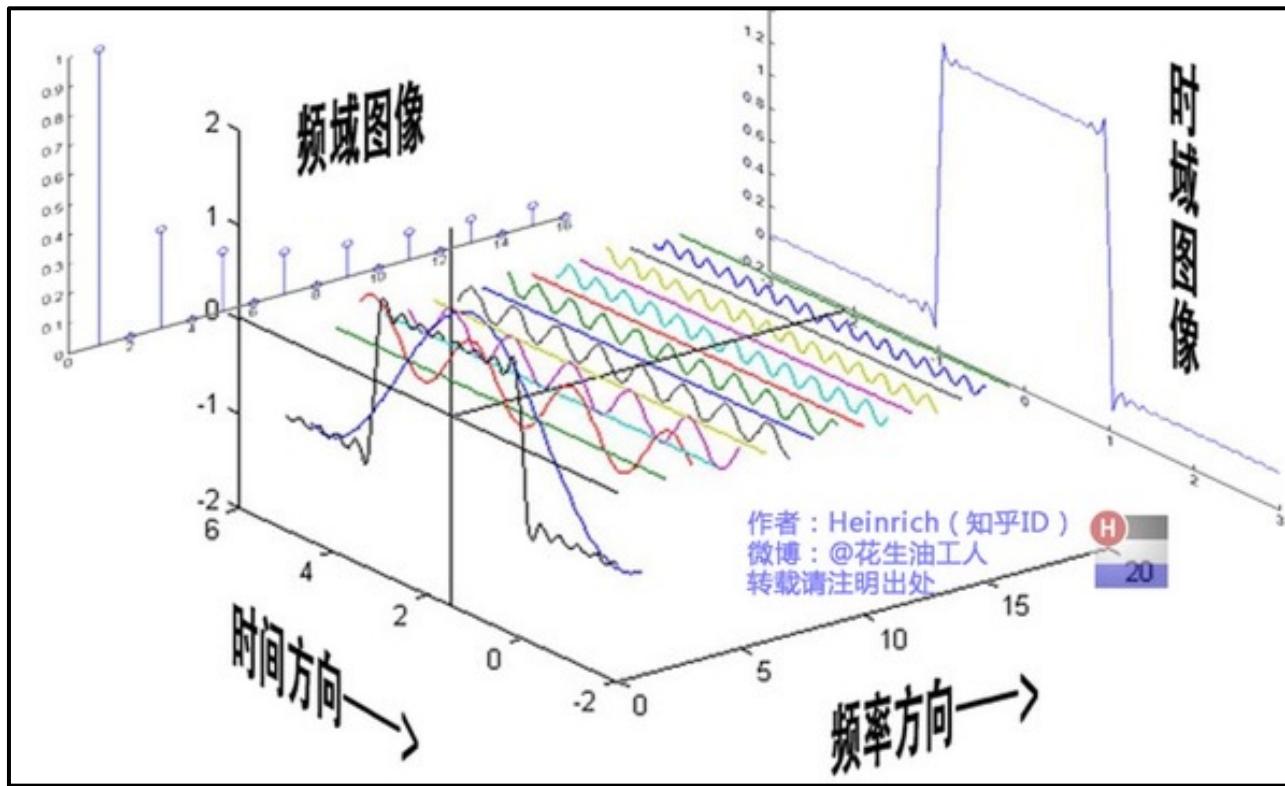


The Effect of different frequency combination



Time Domain & Frequency Domain

- Multiple waves of different frequencies with different amplitudes & different phases are interleaved(交織)

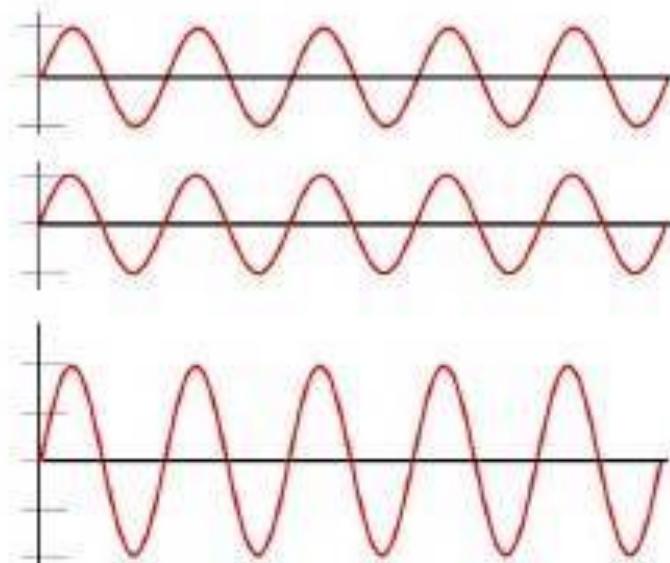


Phase

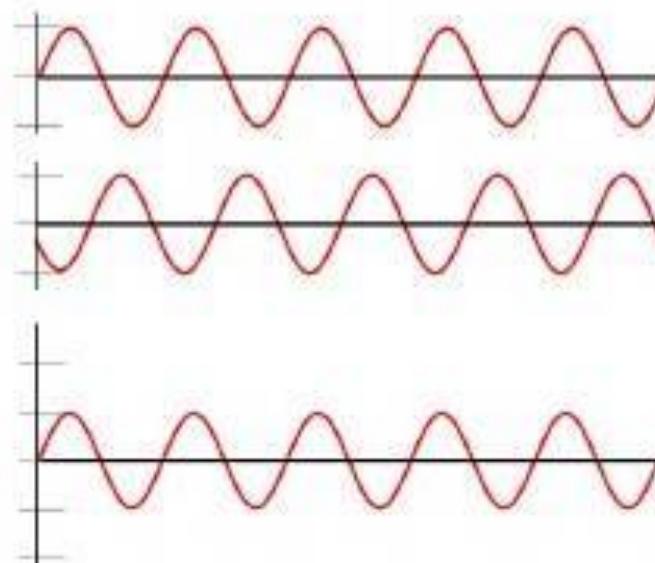
- In addition to amplitude, phase also needs to be considered

完全建設性干涉

0° phase shift = Constructive

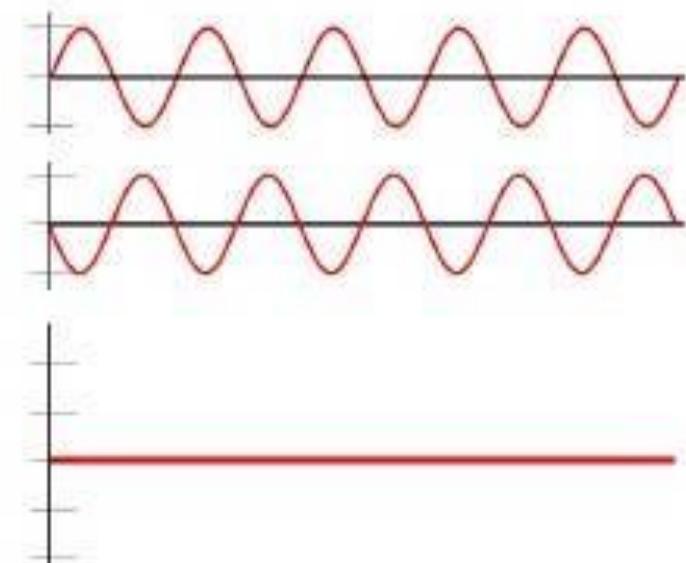


151° phase shift



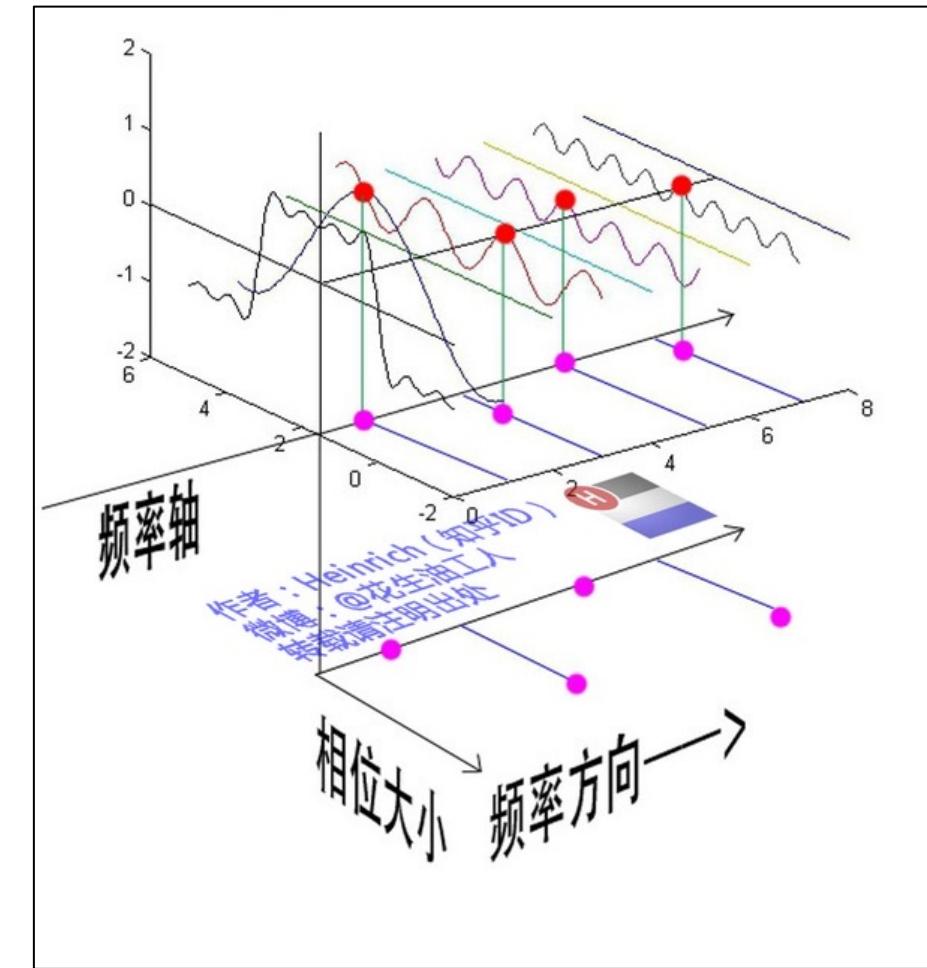
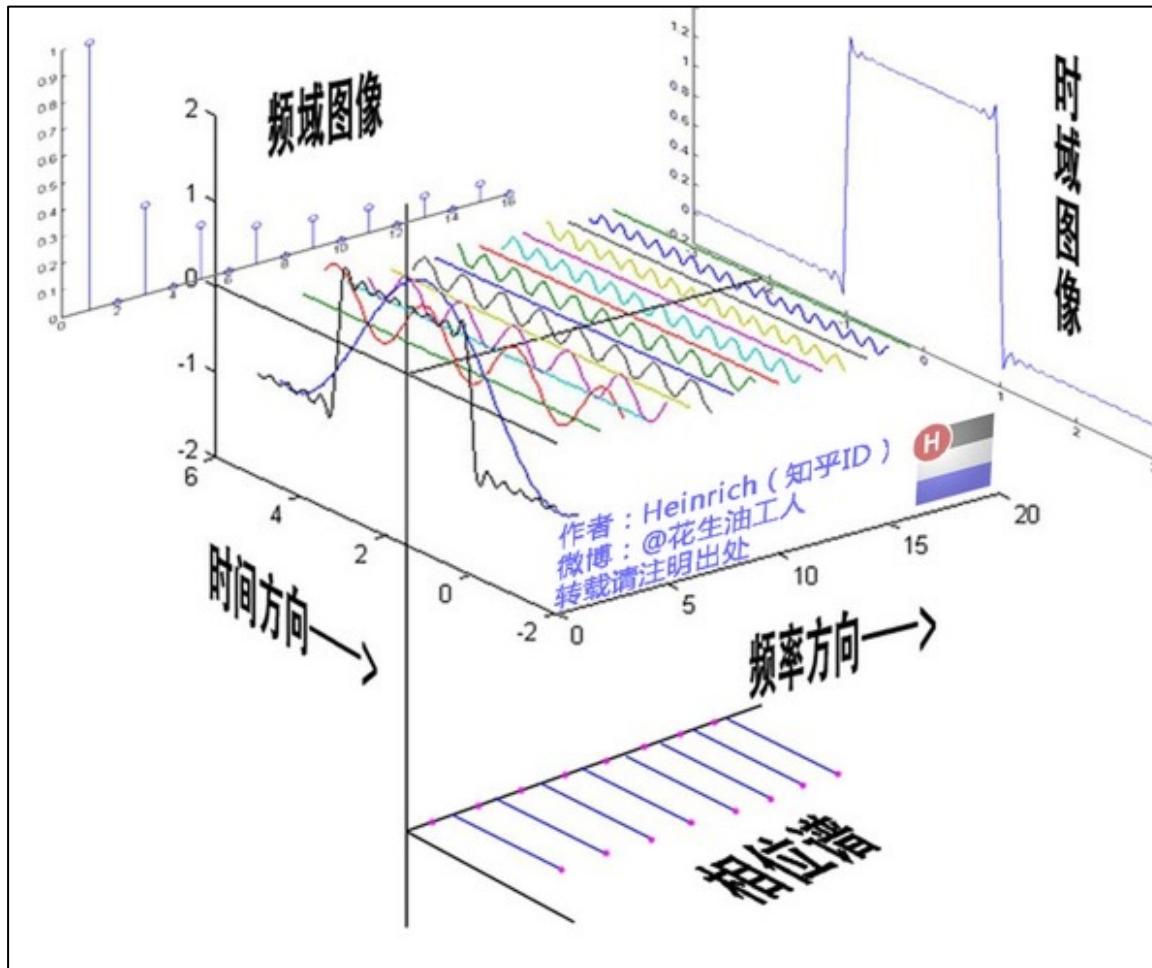
完全破壊性干涉

180° phase shift = Destructive



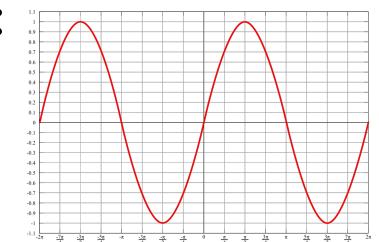
Phase

- In addition to amplitude, phase also needs to be considered



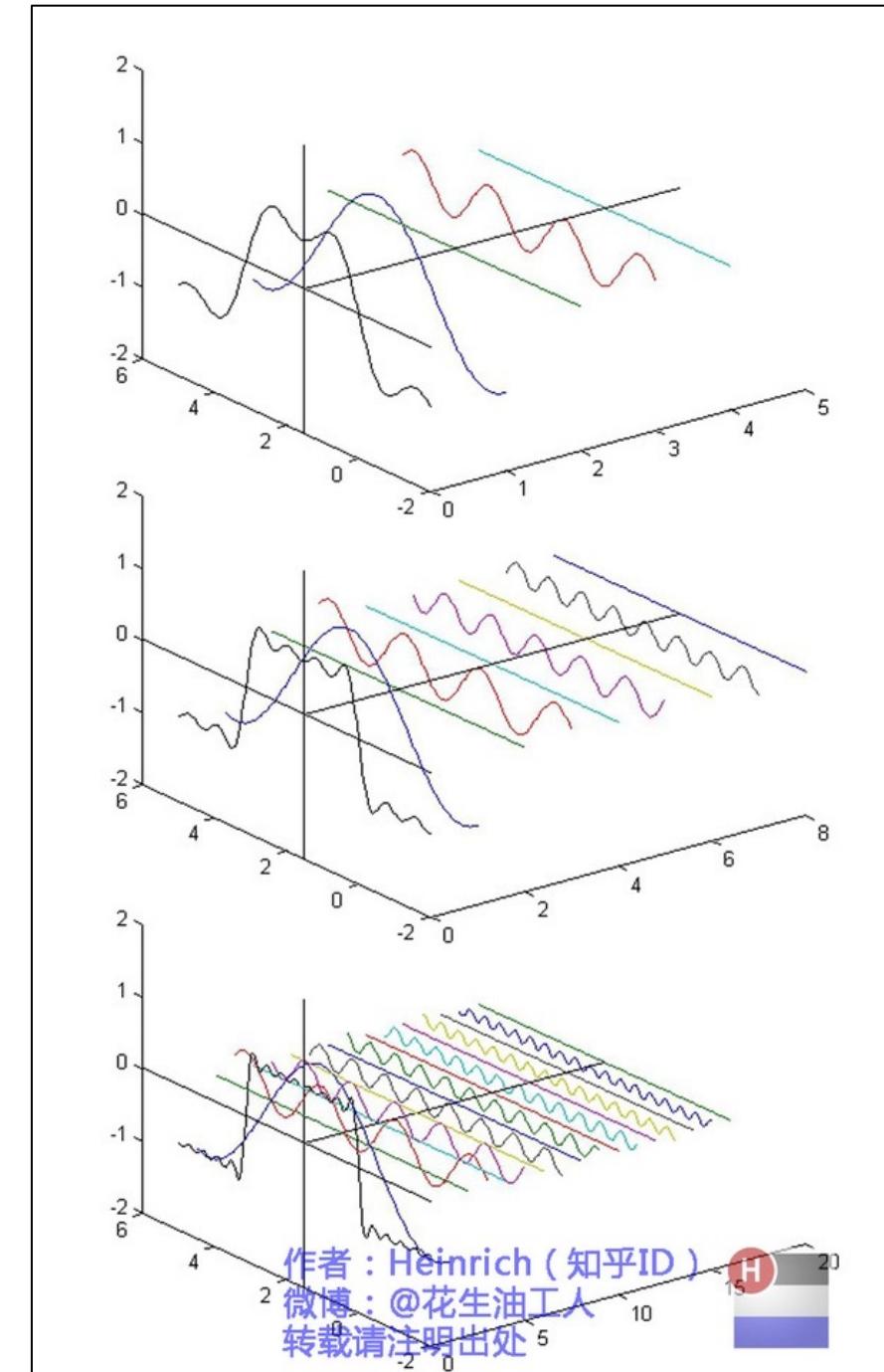
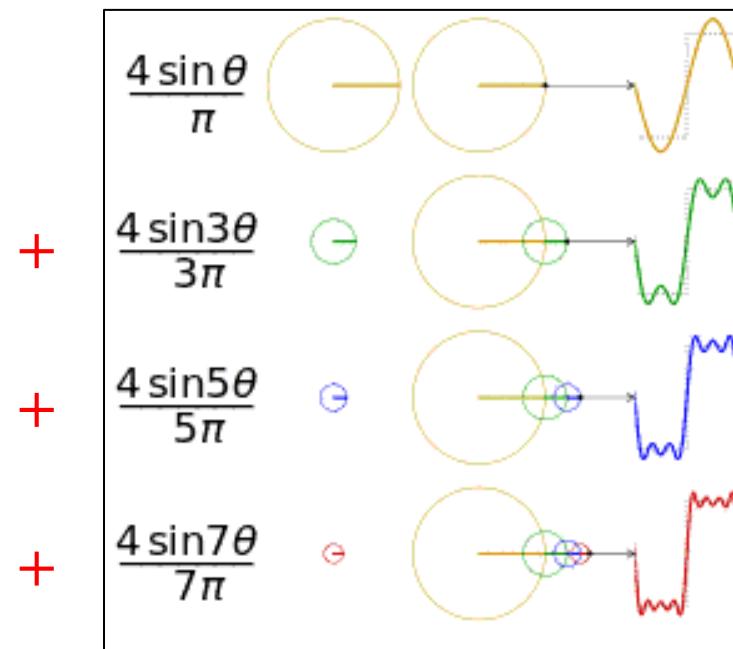
History - Fourier

- 傅立葉 ([Joseph Fourier](#), 1768-1830) 生於法國藝術和歷史之都 Auxerre，因研究熱傳導理論名於世，他提出的傅立葉級數方法是工程數學的重要工具。
- 1807 年，時年 39 歲的傅立葉於法國科學學會上展示一篇論文《固體中的熱傳導》(On the Propagation of Heat in Solid Bodies)，包含當時極具爭議的論斷：
「任何連續周期信號可由一組適當的正弦曲線組合而成」
- 58 歲的拉普拉斯 ([Laplace](#)) 贊成傅立葉的觀點，但 71 歲的 **拉格朗日** ([Lagrange](#), 地位有如終生職的院士) 反對，後者**反對的理由是「正弦曲線無法組合成一個帶有稜角的信號」**。屈服於朗格朗日的威望，該論文直到朗格朗日去世後的第 15 年，經過 Poisson, Cauchy, 一直到 Dirichlet 等人的論證，爭議才落幕，傅立葉的成果得以發表。
- 之後的科學家證明：傅立葉和拉格朗日**兩人的觀點都正確**：
 - 有限數量的正弦曲線的確無法組合成一個帶有稜角的信號**
 - 無限數量的正弦曲線的組合從能量的角度可以無限逼近帶有稜角的信號**



Square Wave

- As shown in the figure, as the frequency increases, it will get closer and closer to a square wave



Question :

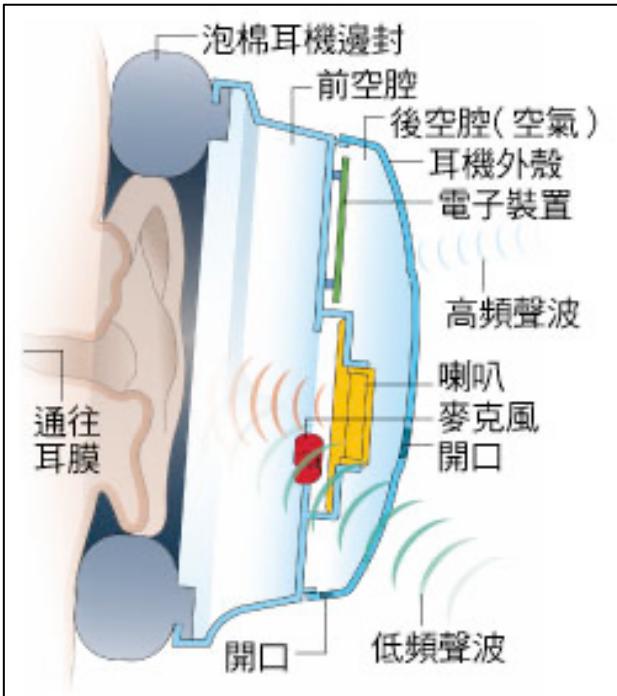
What's the benefit of the knowledge of any signal / sound / wave can be combined by different Sine Waves?

Answer :

We can handle a lot of things conveniently if we can decompose the waves of different frequencies from the original wave

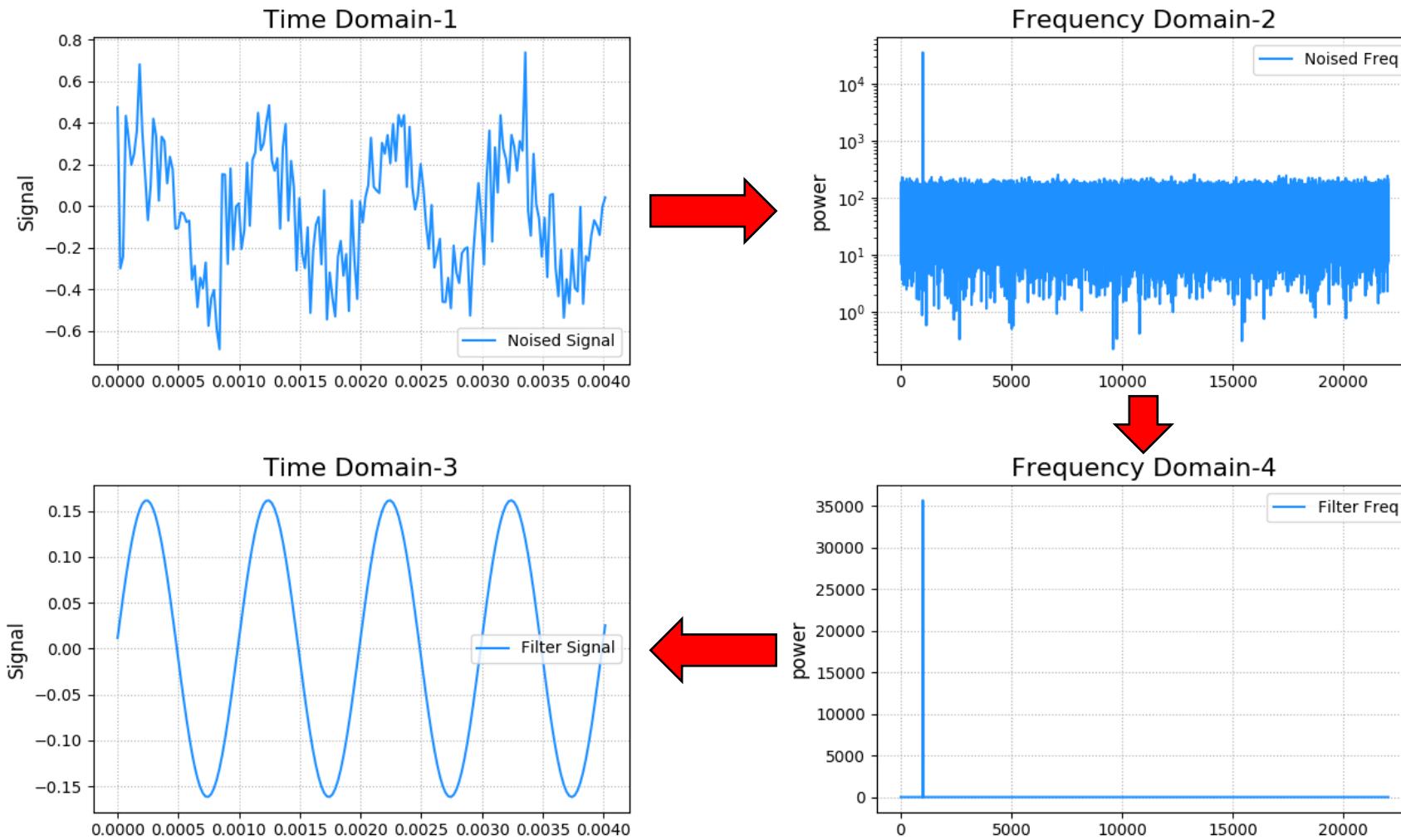
Application

主動式降噪耳機

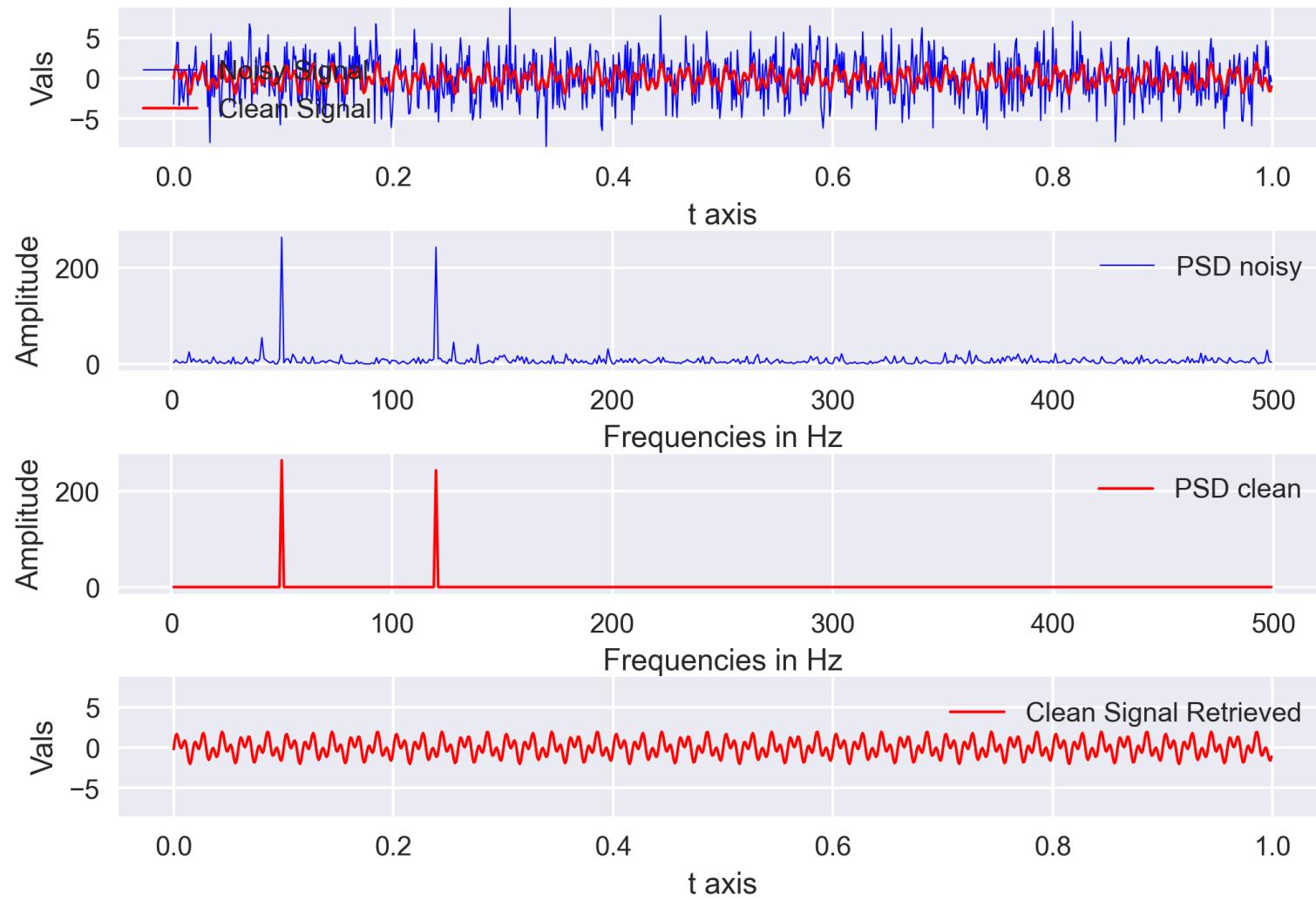


- 耳機裡內建的麥克風可記錄你周圍環境的噪音，並在所有頻譜裡測量噪音的頻率，之後這些噪音的頻率會被反轉，然後混合到你的聲音內容裡頭，這樣能抵消你身邊嬰兒的哭聲或公路上的噪音。
- 降噪的原理是「**反相**」而非「**反向**」，利用波在週期內部的對稱性，製造出一個振幅相同，但相位相反的波動，實現相互的抵消。聲波本質是機械振動，而且人耳的聽力範圍也有限（ $20\text{Hz} \sim 20\text{KHz}$ ），在此頻率範圍中，無論透過耳機還是音響，產生降噪所需的波動，進而讓波動相消。

Application



Application



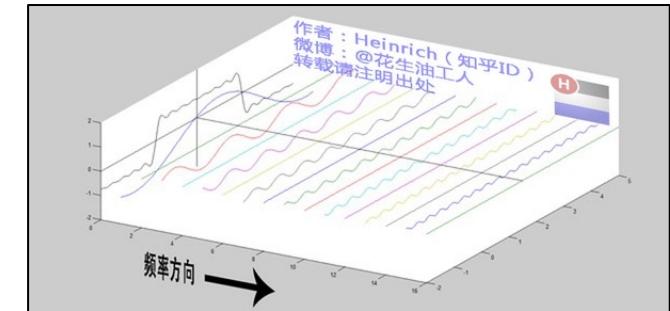
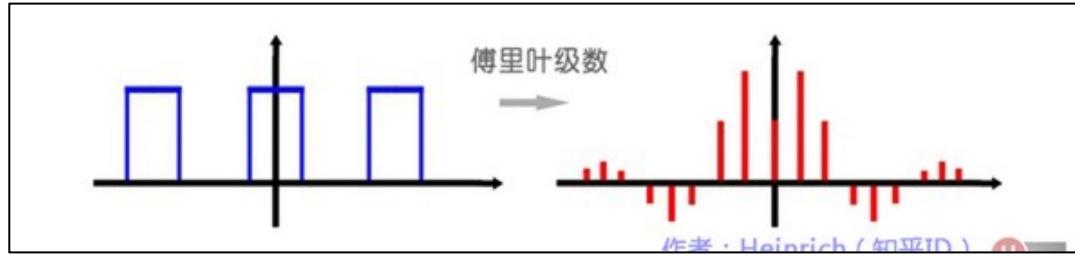
But how to decompose ?

Decomposition Method - Fourier Analysis

- **Fourier Series**

- 將一個時域週期的信號分解成無限多分開的離散的弦波

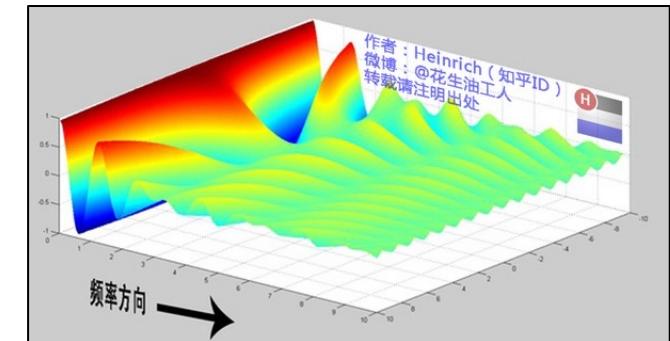
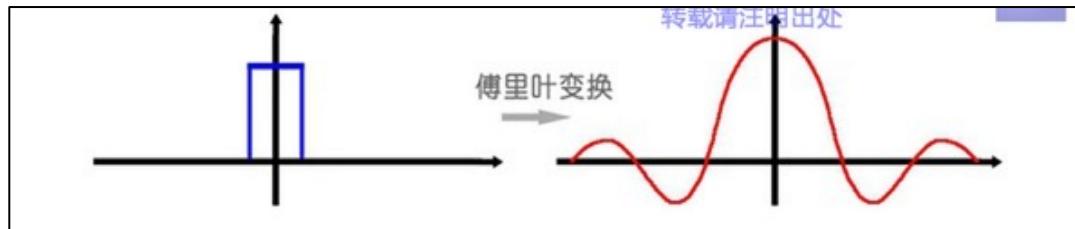
週期



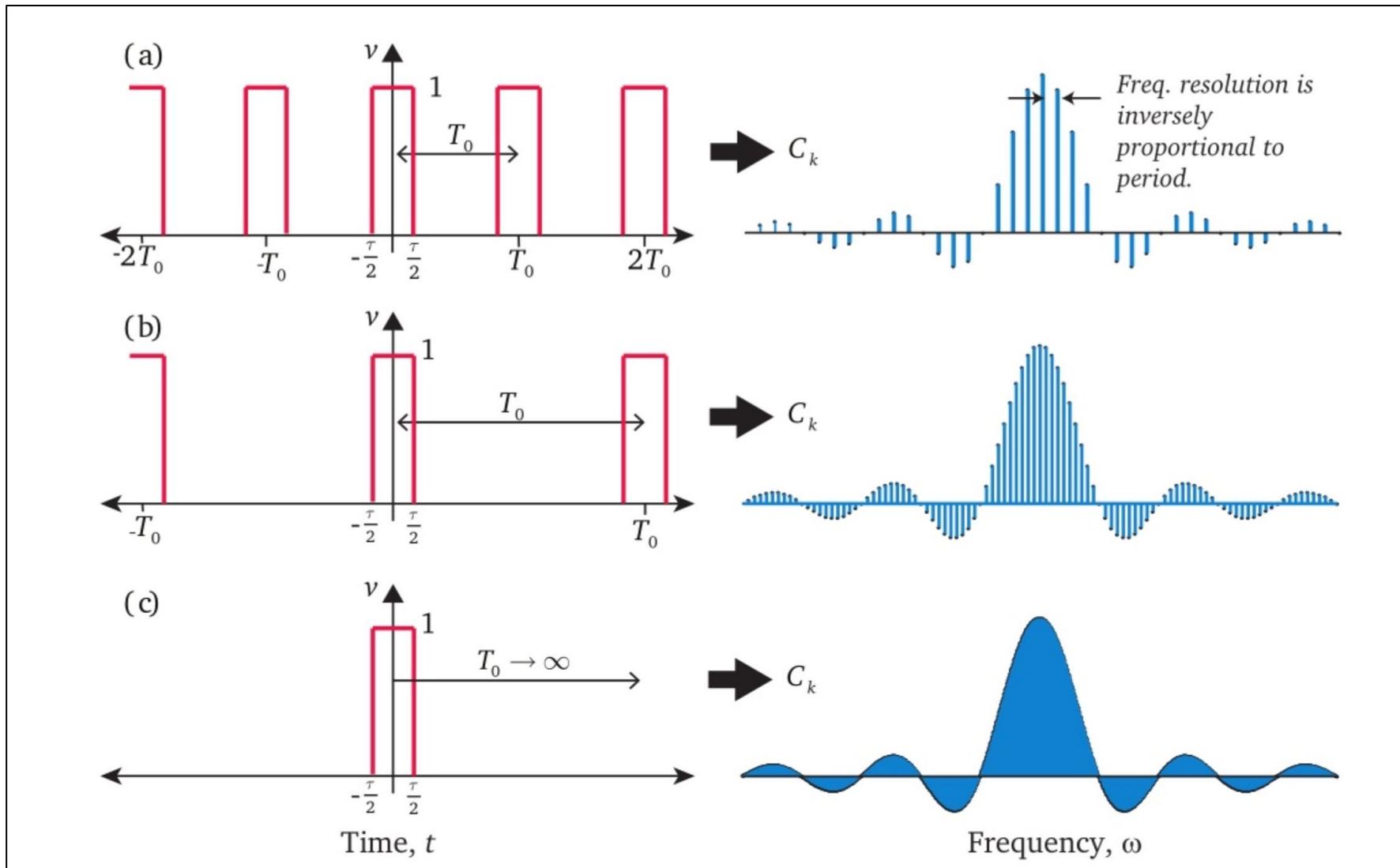
- **Fourier Transform**

- 將一個時域非週期的連續信號轉換為一個在頻域非週期的連續信號
- 「時域非週期」可視為「時域週期，但週期是無限大」

非週期

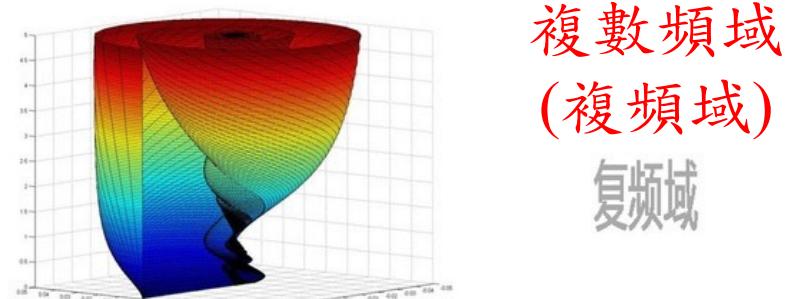
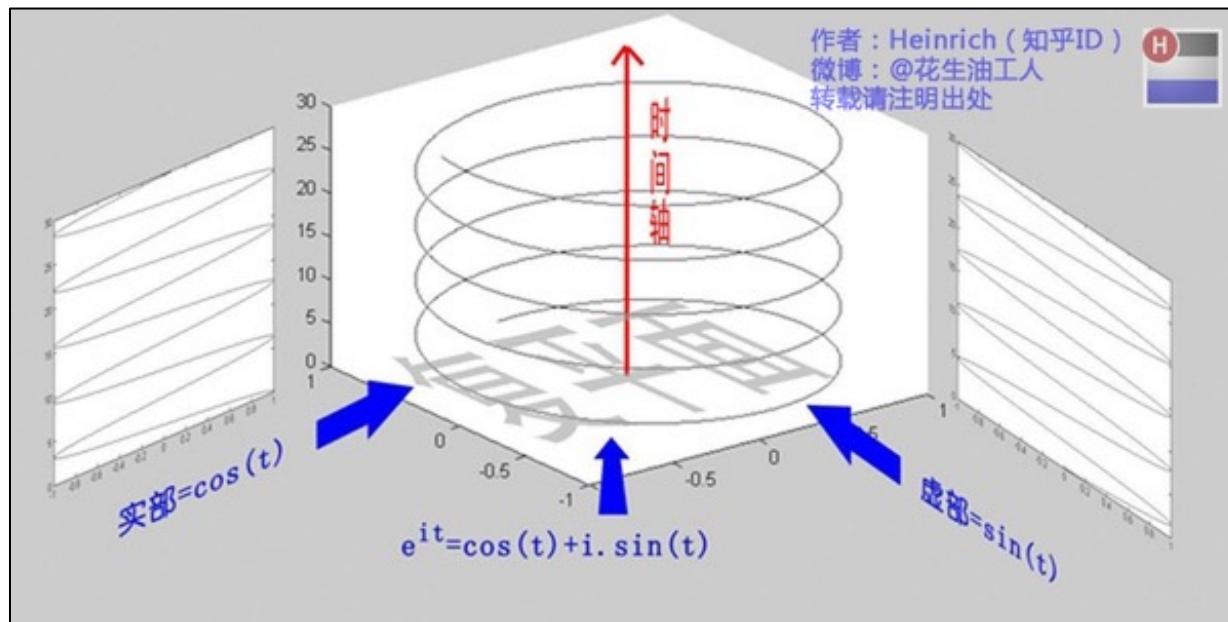


Decomposition Method - Fourier Analysis

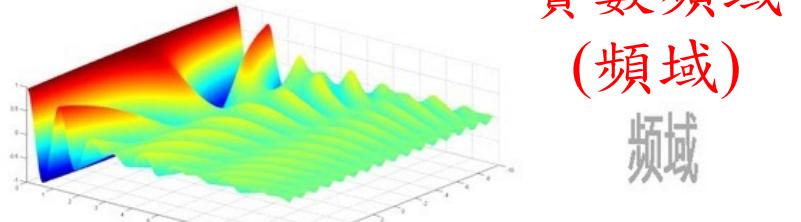


Helix (螺旋)

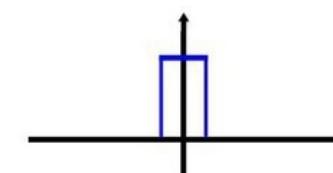
- 弦波的疊加，也可以理解為螺旋線的疊加在**實數空間**的投影。



↓ 投影到实数空间



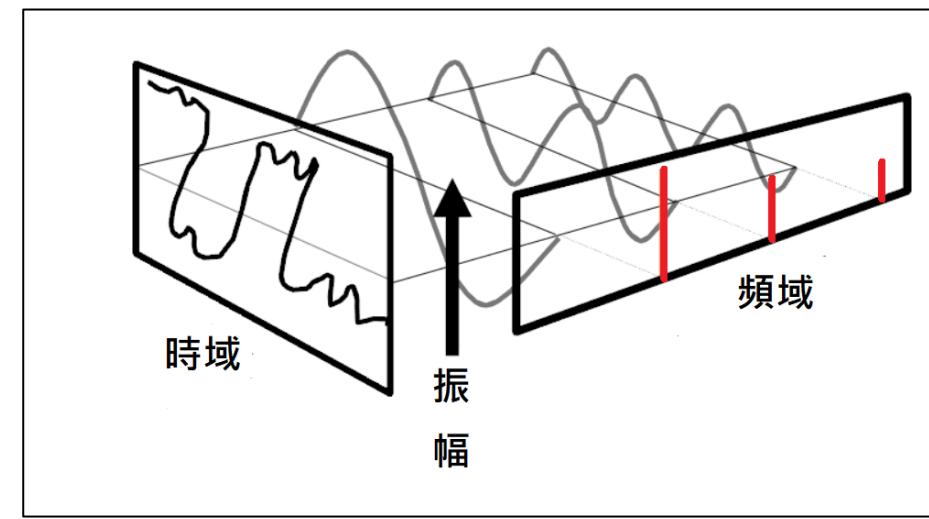
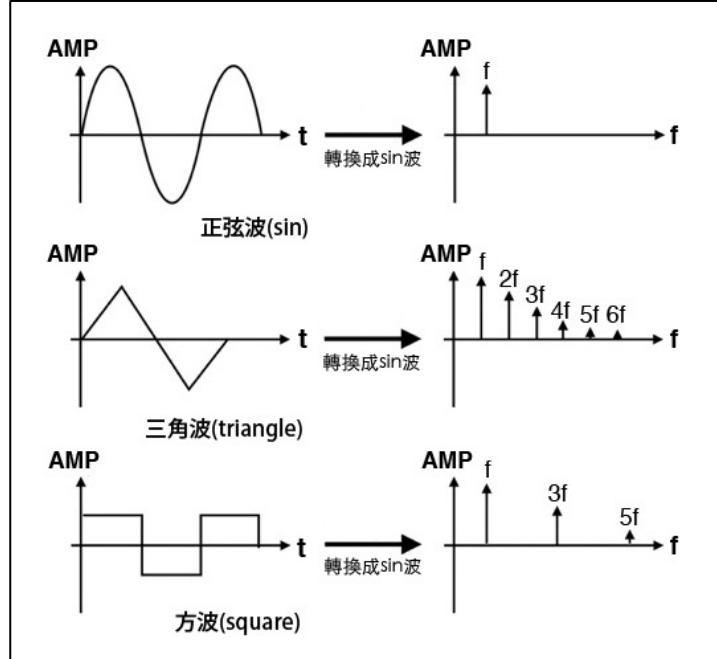
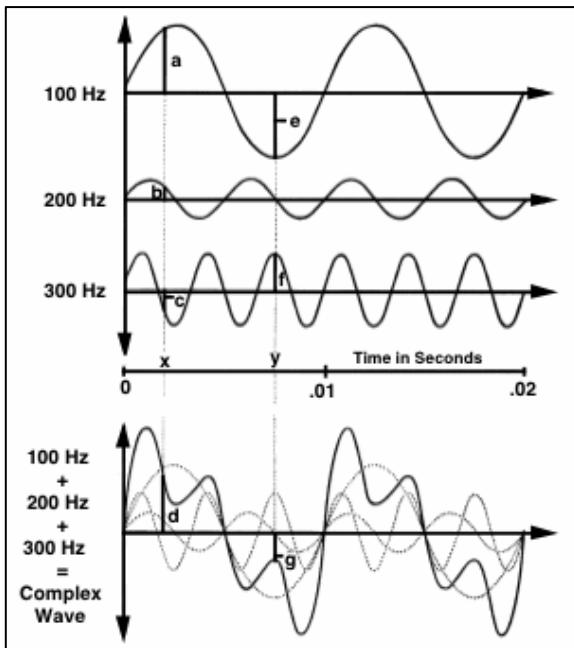
↓ 各频率成分累积



作者 : Heinrich (知乎ID)
微博 : @花生油工人
转载请注明出处

Conclusion

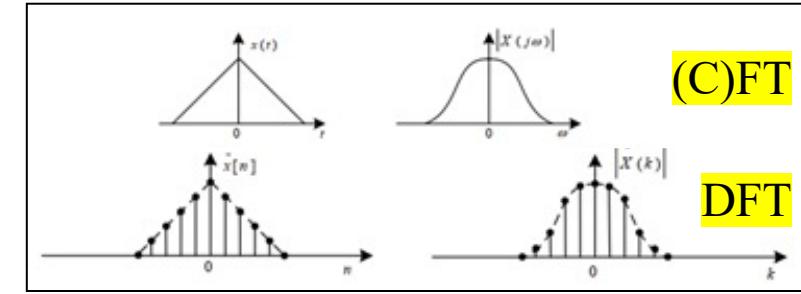
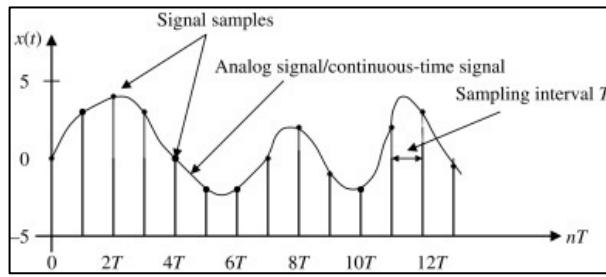
- Fourier Transform is a mathematical transform
- Transform signal from time domain into frequency domain
- We can do a lot of things conveniently in frequency domain compared to time domain



Digital System can't handle continuous signal

- 數位系統只有 1/0 是離散系統，無法處理連續訊號
- 所以須將 (連續)傅立葉轉換 改為 離散傅立葉轉換 (Discrete Fourier Transform)

- 其時域及頻域皆為離散
- Sampling
(採樣)



- 實際應用中會採用 DFT 的升級版 「快速傅立葉轉換」 (Fast Fourier Transform)
 - 計算複雜度可以從 $O(n^2)$ (DFT) 降低到 $O(n \log n)$ (FFT)， n 為資料大小
 - 1994年美國數學家吉爾伯特·斯特朗把 FFT 描述為
「我們一生中最重要的數值演算法」
 - 它還被 IEEE 科學與工程計算期刊列入 20 世紀十大演算法

Final Project - Frequency Analysis System

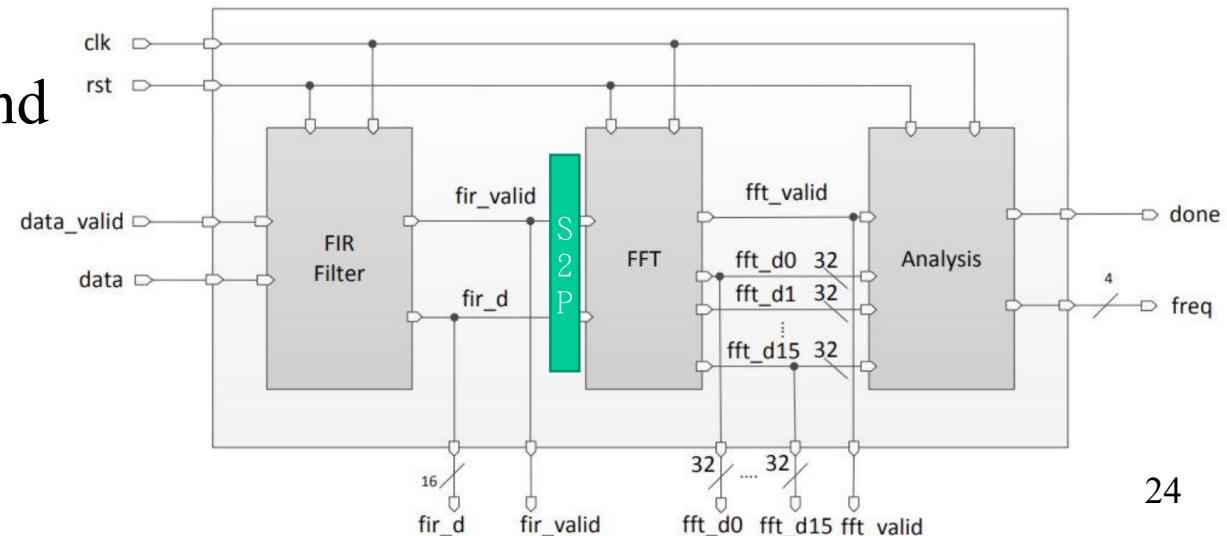
- Given discrete time domain signal for a period of time
- Please find the main frequency band
 - main frequency band :
the frequency whose sum of square of the real part data and imaginary part data is maximum

3 steps

1. Use low-pass filter to filter noise which frequency higher than cut-off frequency
2. Use FFT to transform signal from time domain into frequency domain
3. Analysis all frequencies and find the main frequency band

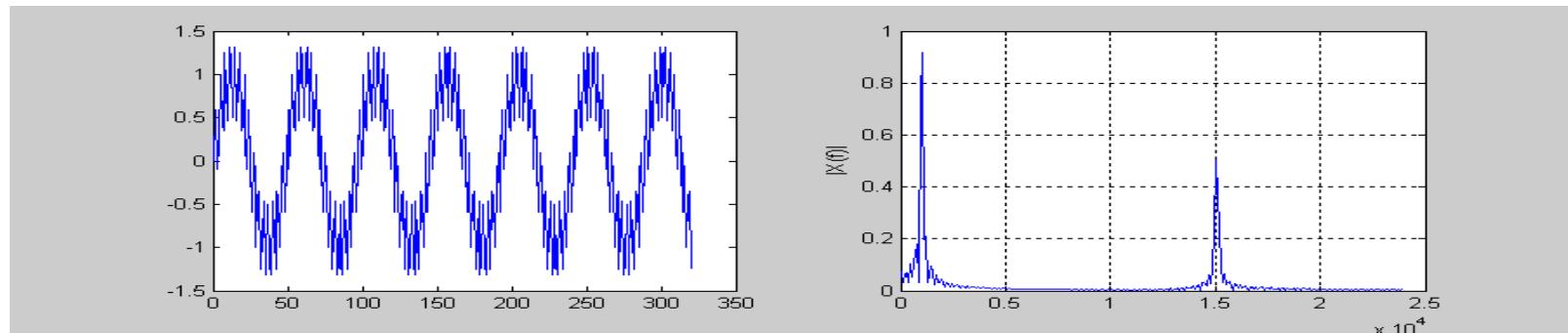
System Block Diagram

- System structure : 4 components
 1. FIR Filter (Finite Impulse Response Filter)
 - Filter out the high frequency noise (限定欲處理的頻率範圍)
 2. S2P (Serial to Parallel)
 - Transfer the serial outputs which output from FIR Filter to FFT in parallel
 3. FFT (Fast Fourier Transform)
 - Time domain -> Frequency domain
 4. Analysis
 - Find the main frequency band

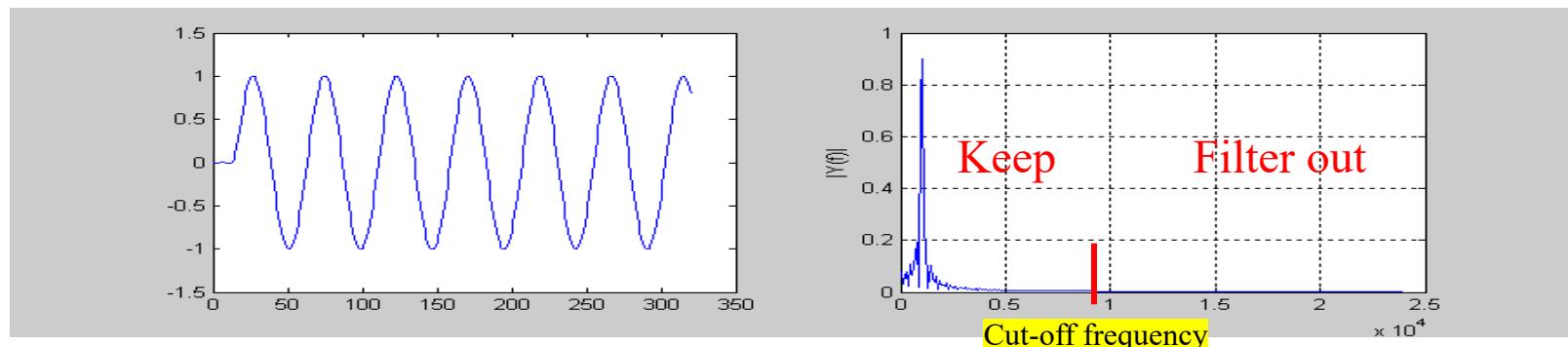


Component 1 : FIR Filter

- We want just handle the specific frequency scope(限定欲處理的頻率範圍)
- If the signal is mixed with higher frequencies, it will produce glitches, which are noise



- We can use a low-pass filter to filter out high frequency noise above the cut-off frequency(截止頻率) (Only pass the frequencies below the cut-off frequency)



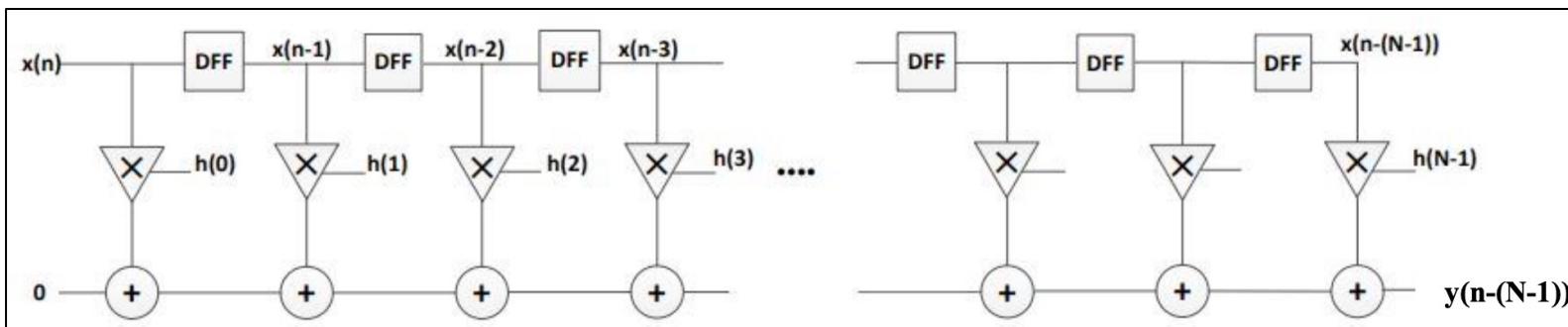
Component 1 : FIR Filter

- There are many kind of filters, this time we use FIR filter (Finite Impulse Response)
- FIR Filter Equation

$$y(n - (N - 1)) = \sum_{k=0}^{N-1} h(k)x(n - k)$$

- If you are interested, you can google it.
- You can think of it just as a Verilog practice without going too deep into how it works !!!

- FIR Filter hardware architecture



N=32

Component 1 : FIR Filter

- The coefficients of the filter are fixed
 - You can check “FIR_coefficient.dat”

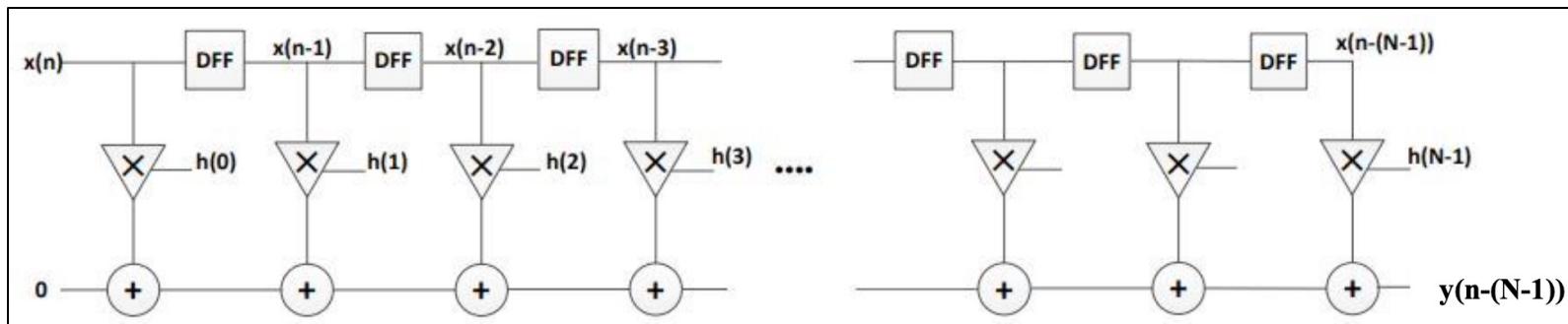
```

1 parameter signed [19:0] FIR_C00 = 20'hFFF9E ; //The FIR_coefficient
2 parameter signed [19:0] FIR_C01 = 20'hFFF86 ; //The FIR_coefficient
3 parameter signed [19:0] FIR_C02 = 20'hFFFA7 ; //The FIR_coefficient
4 parameter signed [19:0] FIR_C03 = 20'h0003B ; //The FIR_coefficient
5 parameter signed [19:0] FIR_C04 = 20'h0014B ; //The FIR_coefficient
6 parameter signed [19:0] FIR_C05 = 20'h0024A ; //The FIR_coefficient
7 parameter signed [19:0] FIR_C06 = 20'h00222 ; //The FIR_coefficient
8 parameter signed [19:0] FIR_C07 = 20'hFFFE4 ; //The FIR_coefficient
9 parameter signed [19:0] FIR_C08 = 20'hFFBC5 ; //The FIR_coefficient
10 parameter signed [19:0] FIR_C09 = 20'hFF76A ; //The FIR_coefficient

```

- FIR Filter hardware architecture

Low-pass Filter Coefficient (h)			
h(0)	-0.001505748051548	h(16)	0.229154203266836
h(1)	-0.001868548463782	h(17)	0.186019113110601
h(2)	-0.001366448872269	h(18)	0.115911664195512
h(3)	9.086560980884849e-04	h(19)	0.043365841112764
h(4)	0.005060978234550	h(20)	-0.009960448073993
h(5)	0.008948776436908	h(21)	-0.033966171510252
h(6)	0.008342764987706	h(22)	-0.032084609166509
h(7)	-4.333516292017973e-04	h(23)	-0.016526671675409
h(8)	-0.016526671675409	h(24)	-4.333516292017973e-04
h(9)	-0.032084609166509	h(25)	0.008342764987706
h(10)	-0.033966171510252	h(26)	0.008948776436908
h(11)	-0.009960448073993	h(27)	0.005060978234550
h(12)	0.043365841112764	h(28)	9.086560980884849e-04
h(13)	0.115911664195512	h(29)	-0.001366448872269
h(14)	0.186019113110601	h(30)	-0.001868548463782
h(15)	0.229154203266836	h(31)	-0.001505748051548

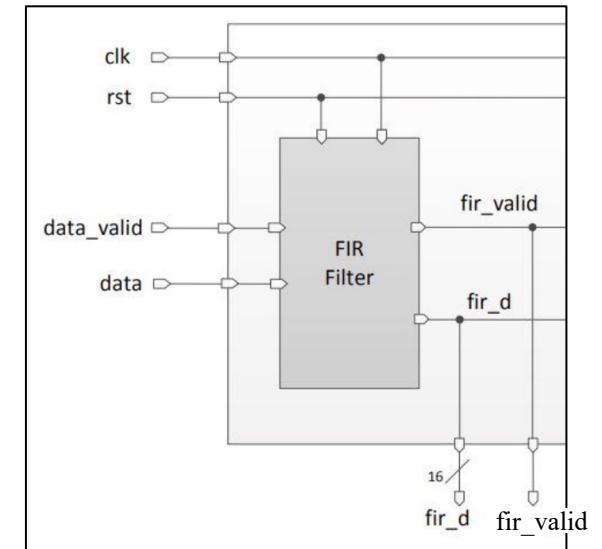
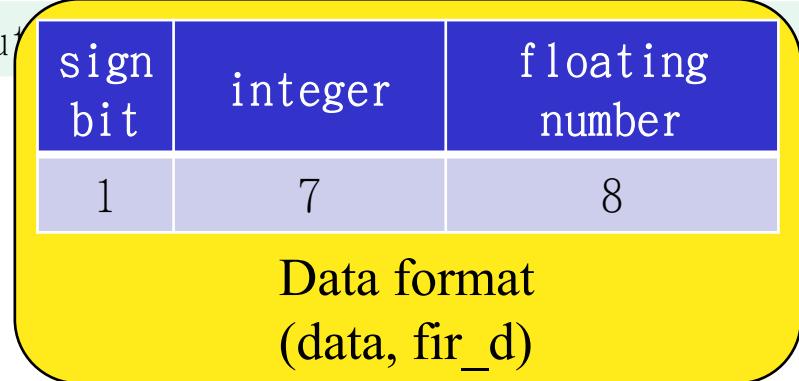


N=32

Component 1 : FIR Filter

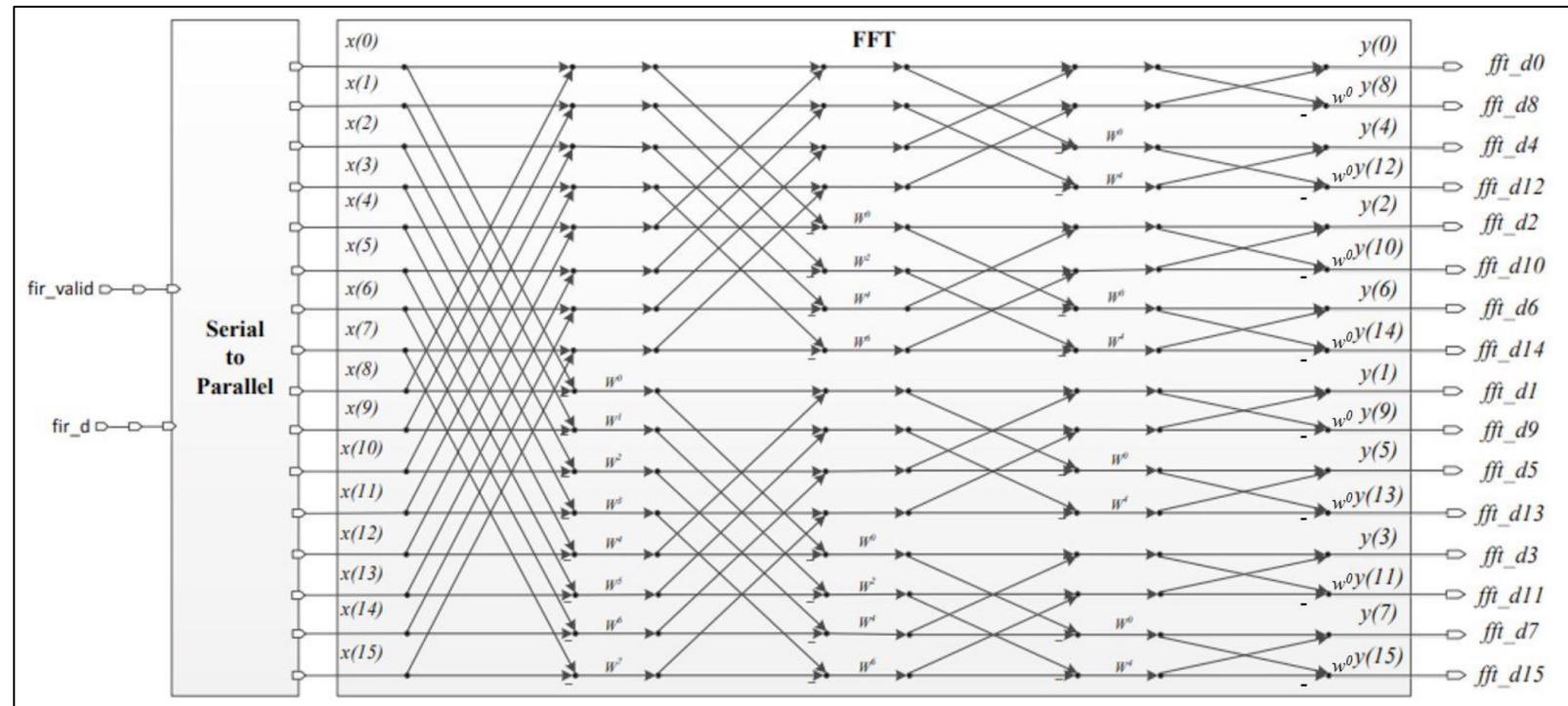
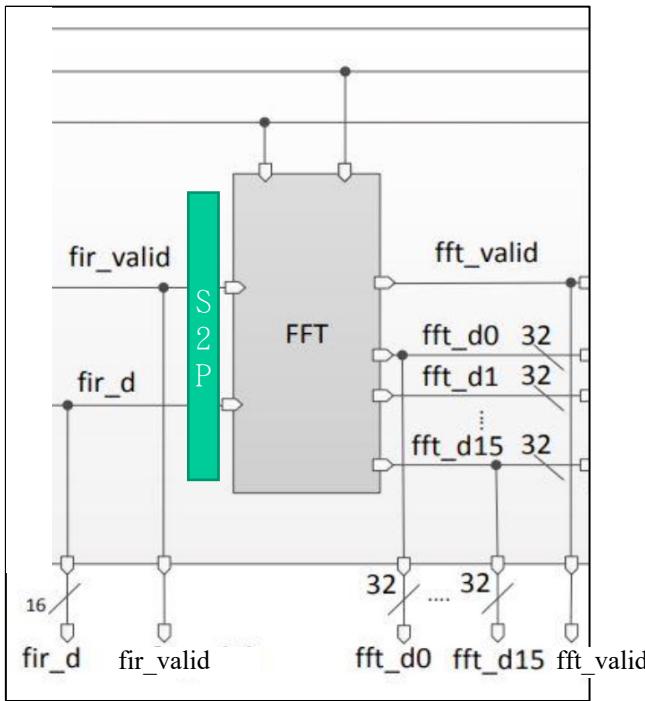
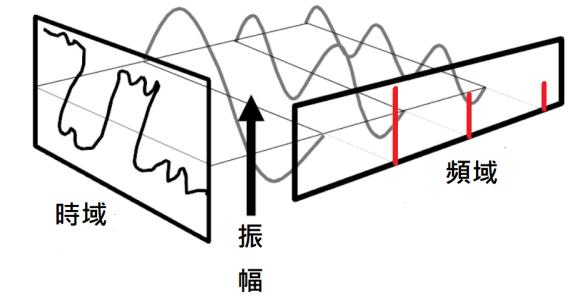
- I/O

Name	I/O	Width	Description
clk	I	1	System clock signal. This system is synchronized with the positive edge of the clock.
rst	I	1	Active-high asynchronous reset signal.
data_valid	I	1	When the host is ready to send data, this signal will be set as high.
data	I	16	Time domain signal from the host.
fir_valid	O	1	Data valid signal of FIR filter.
fir_d	O	16	Output



Component 2 + 3 : S2P + FFT

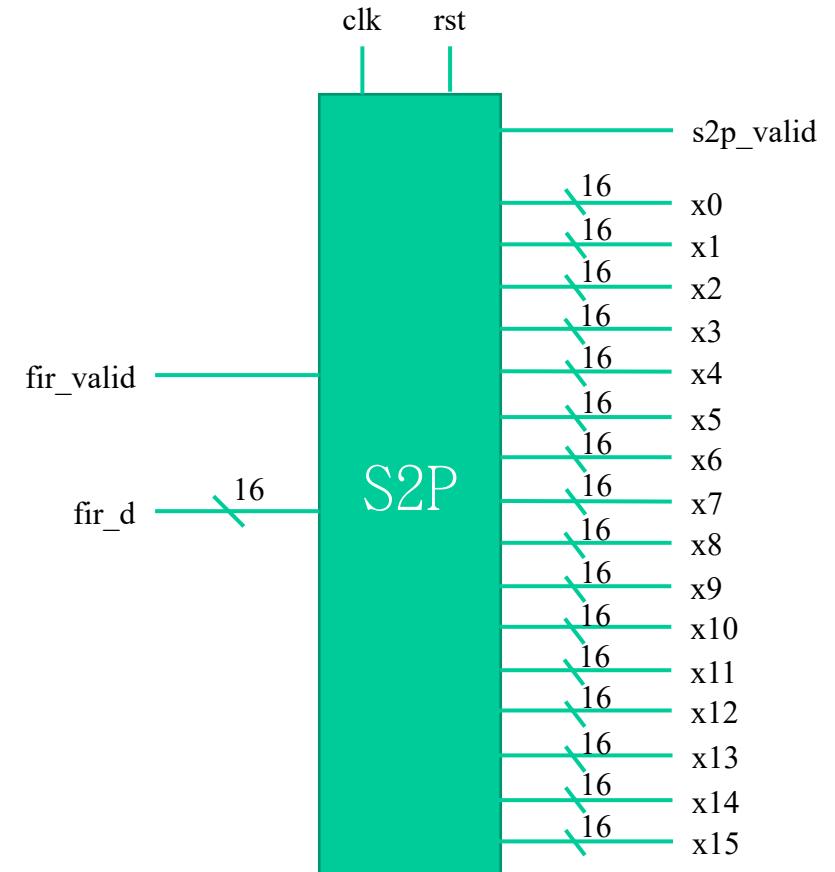
- Transform signal from time domain to frequency domain
- 2 parts : Serial to Parallel + FFT



Component 2 : S2P

- Serial to Parallel (S2P)

- Collect 16 serial FIR filter outputs and output them to FFT in parallel
- Steps
 1. Wait for collecting **new** 16 data
 2. Output them to FFT in parallel
 3. Invalid all 16 data
 4. Go back to step 1

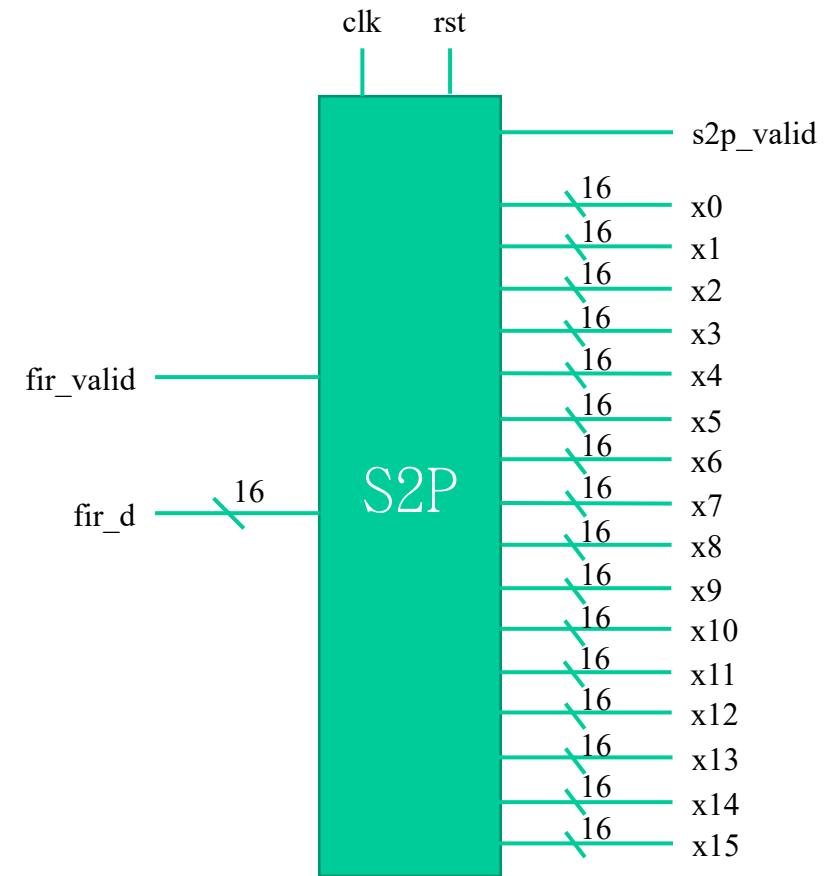


Component 2 : S2P

- Serial to Parallel (S2P)

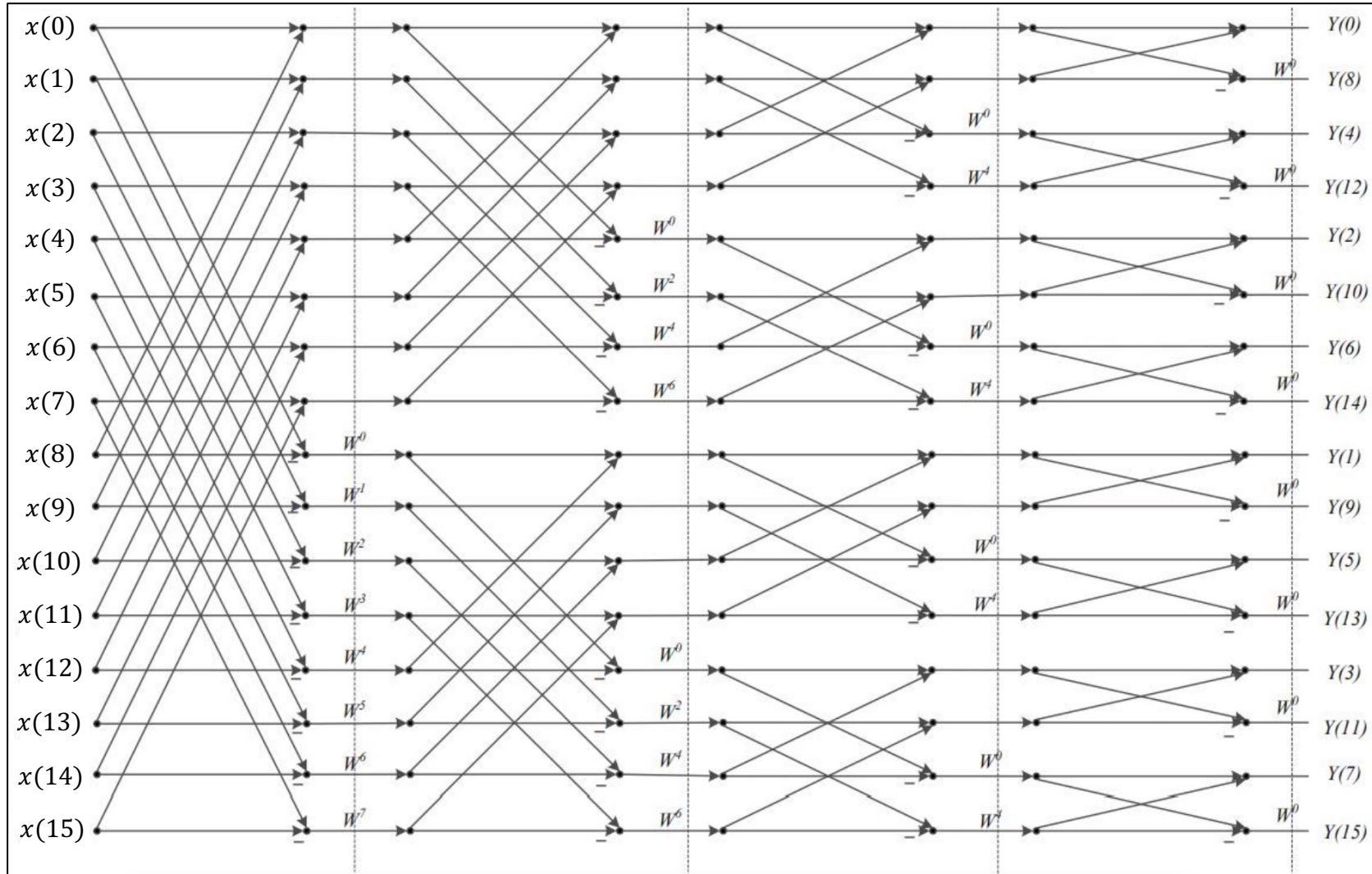
- I/O

Name	I/O	Width
clk	I	1
rst	I	1
fir_valid	I	1
fir_d	I	16
s2p_valid	O	1
x0	O	16
x1	O	16
...	O	16
x15	O	16

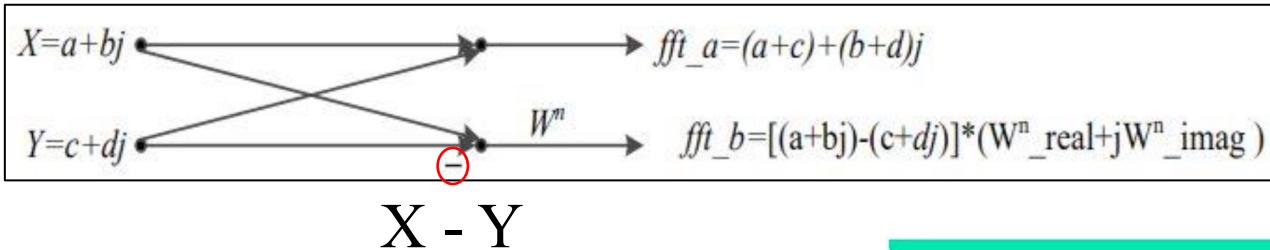


Component 3 : FFT

- FFT hardware architecture



Component 3 : FFT



- $W^n = FFT \text{ coefficient}$
 - A complex number
 - $W^n = W_{real}^n + jW_{imag}^n$
- $fft_a = X + Y$
- $fft_b = (X - Y) \times W^n$
 - $[(a + bj) - (c + dj)] \times (W_{real}^n + jW_{imag}^n)$

Real part of fft_b	$(a - c) \times W_{real}^n + (d - b) \times W_{imag}^n$
imaginary part of fft_b	$(a - c) \times W_{imag}^n + (b - d) \times W_{real}^n$

W^n	
W^0	$1.000 + 0.000j$
W^1	$0.923879532511287 - 0.382683432365090j$
W^2	$0.707106781186548 - 0.707106781186548j$
W^3	$0.382683432365090 - 0.923879532511287j$
W^4	$0.000 - 1.0000j$
W^5	$-0.382683432365090 - 0.923879532511287j$
W^6	$-0.707106781186548 - 0.707106781186547j$
W^7	$-0.923879532511287 - 0.382683432365089j$

Coefficient are stored in

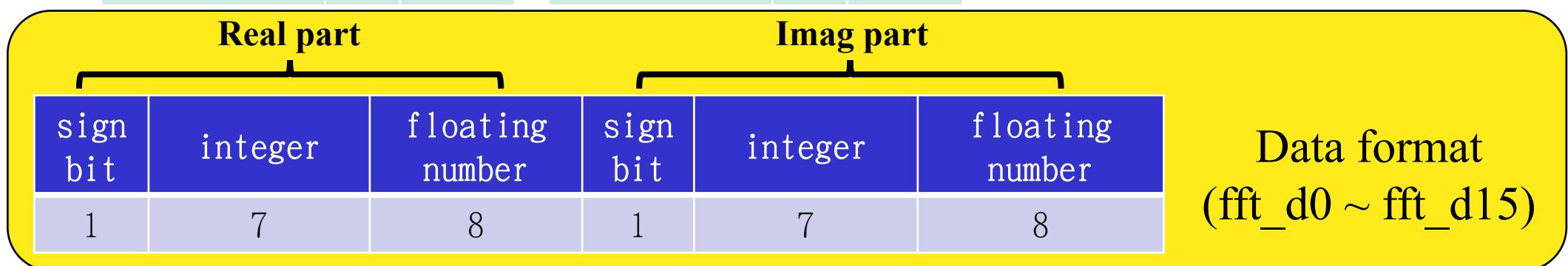
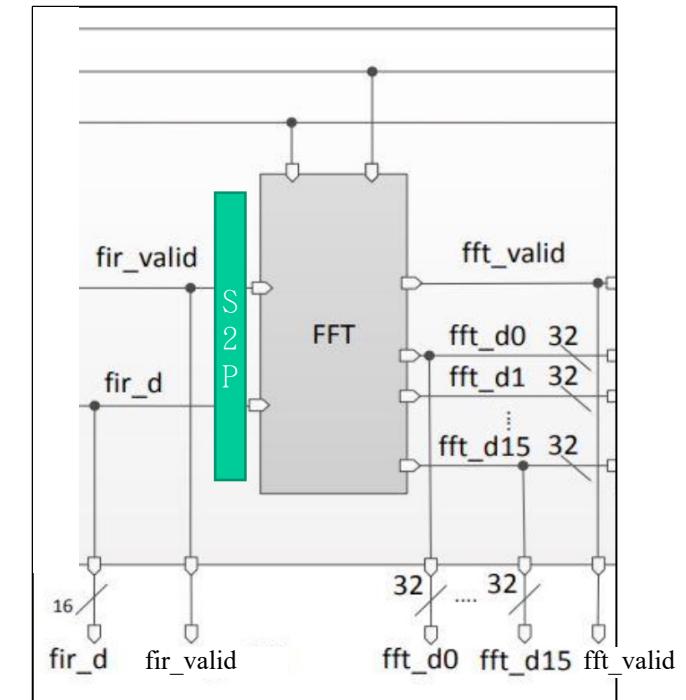
- FFT_coefficient.dat

Component 3 : FFT

- FFT I/O

Name	I/O	Width
clk	I	1
rst	I	1
s2p_valid	I	1
x0	I	16
x1	I	16
...	I	16
x15	I	16

Name	I/O	Width
fft_valid	0	1
fft_d0	0	32
fft_d1	0	32
...	0	32
fft_d15	0	32



Component 4 : Analysis

- Find out the main frequency band
 - Main frequency band : the frequency whose sum of square of the real part data and imaginary part data is maximum

For example

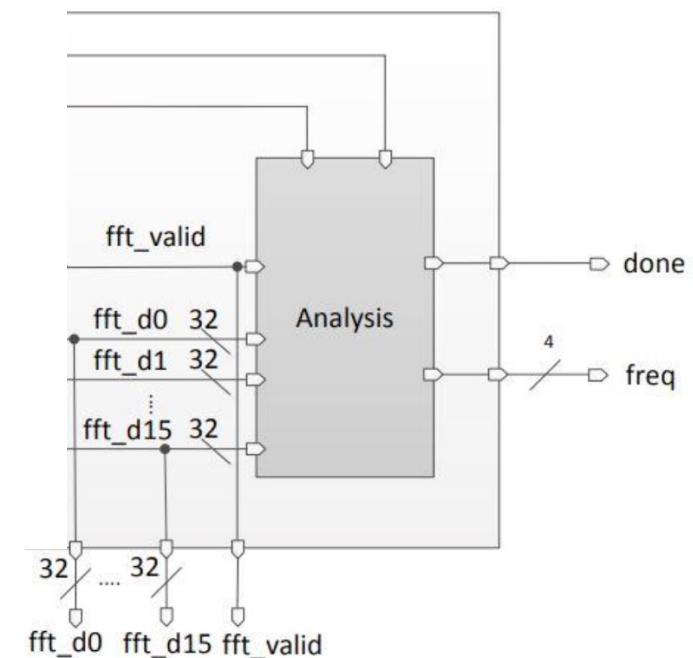
$$\text{fft_d}\textcolor{red}{n} = a_n + b_n j, \quad \textcolor{red}{n} = 0 \sim 15$$

Calculate $fb_n = a_n^2 + b_n^2$ for each n

The max fb_n is the Main frequency band

If fb_3 is the Main frequency band

Then output freq = 4'b0011

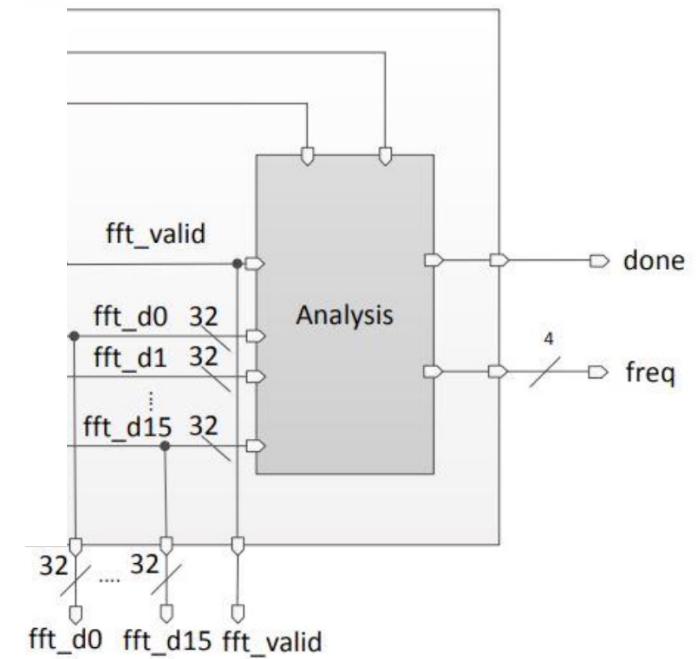


Component 4 : Analysis

- Find out the main frequency band
 - Main frequency band** : the frequency whose sum of square of the real part data and imaginary part data is maximum
- I/O

Name	I/ O	Width
clk	I	1
rst	I	1
fft_valid	I	1
fft_d0	I	32
fft_d1	I	32
...	I	32
fft_d15	I	32

Name	I/ O	Width
done	0	1
freq	0	4



Verilog Module Format

1. FAS.v

- Module name : FAS
- Description : System top file. Include all other files and connect them

2. FIR.v

- Module name : FIR

3. S2P.v

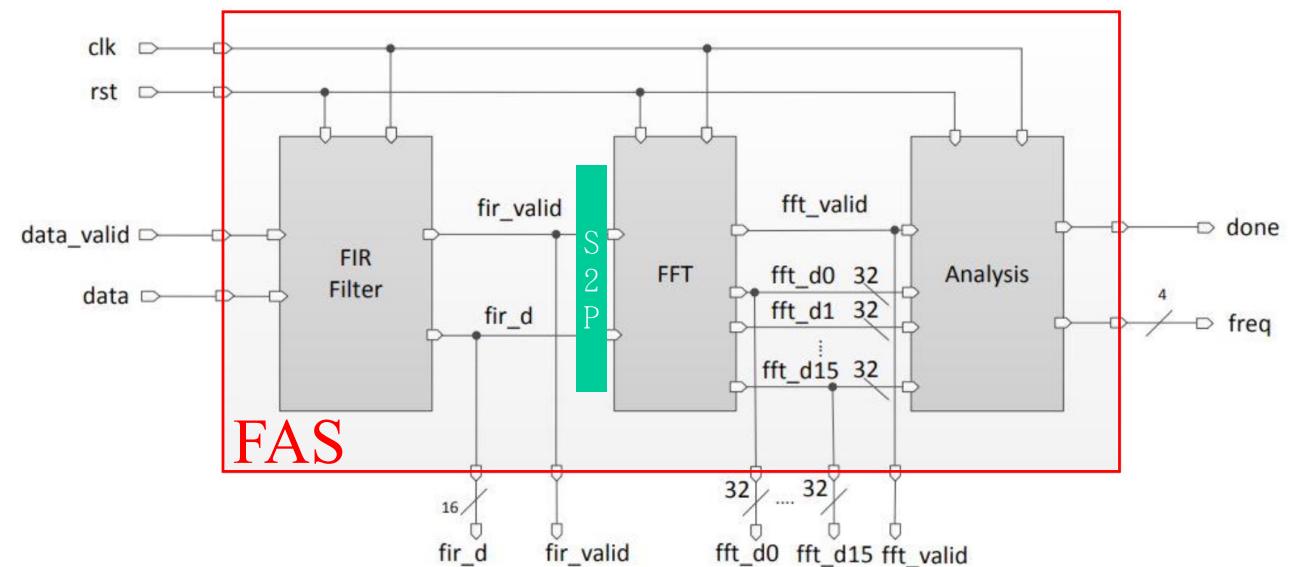
- Module name : S2P

4. FFT.v

- Module name : FFT

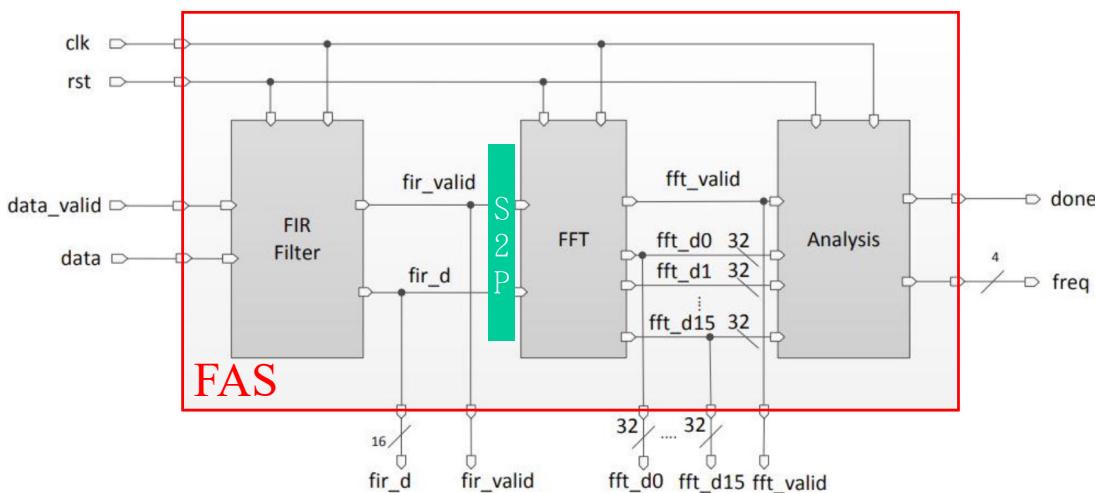
5. Analysis.v

- Module name : Analysis



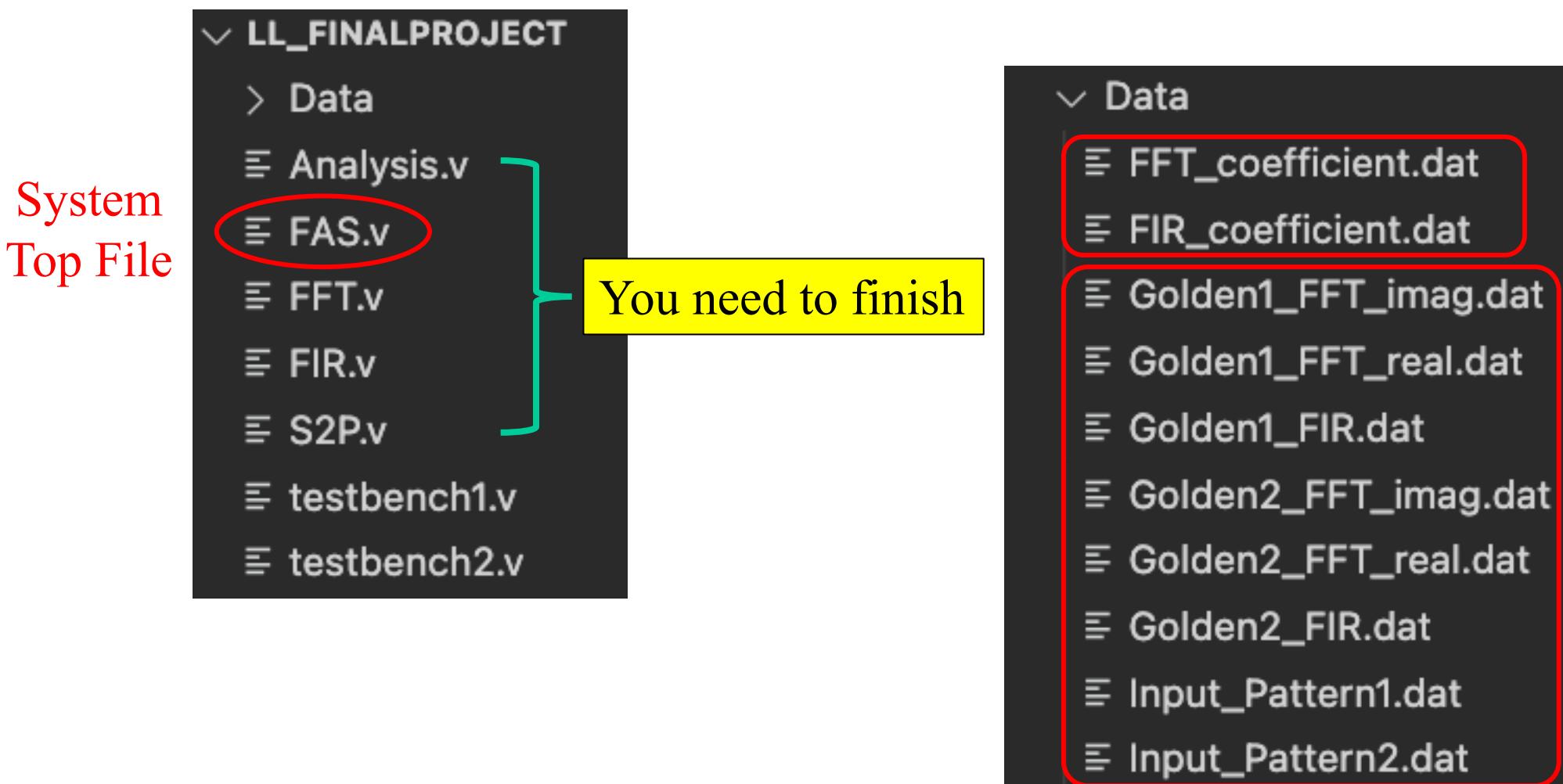
Verilog Module Format

- FAS.v – System I/O port



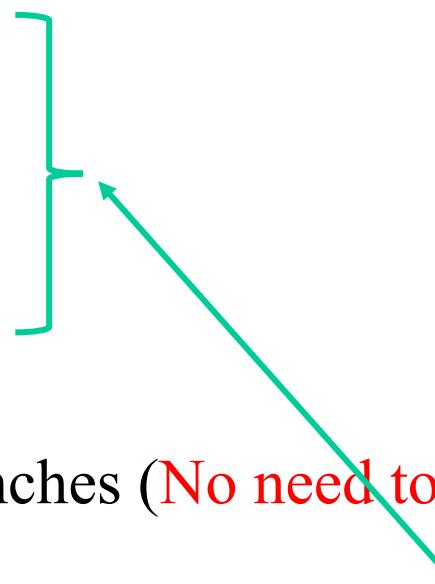
Name	I/O	Width	Description
clk	I	1	System clock signal. This system is synchronized with the positive edge of the clock.
rst	I	1	Active-high async reset signal.
data_valid	I	1	When the host is ready to send data, this signal will be set as high.
data	I	16	Time domain signal from the host.
fir_valid	0	1	Data valid signal of FIR filter.
fir_d	0	16	Output data of FIR filter.
fft_valid	0	1	Data valid signal of FFT.
fft_d0~fft_d15	0	32	Output data of FFT.

Project Folder Structure



Project Requirements

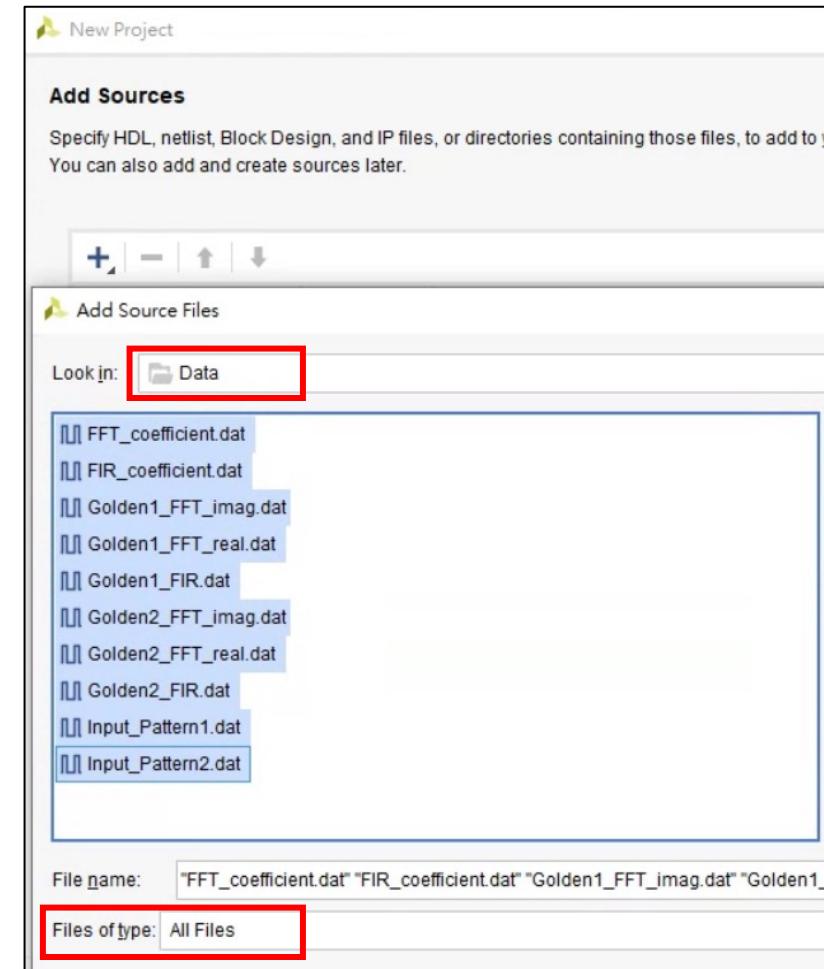
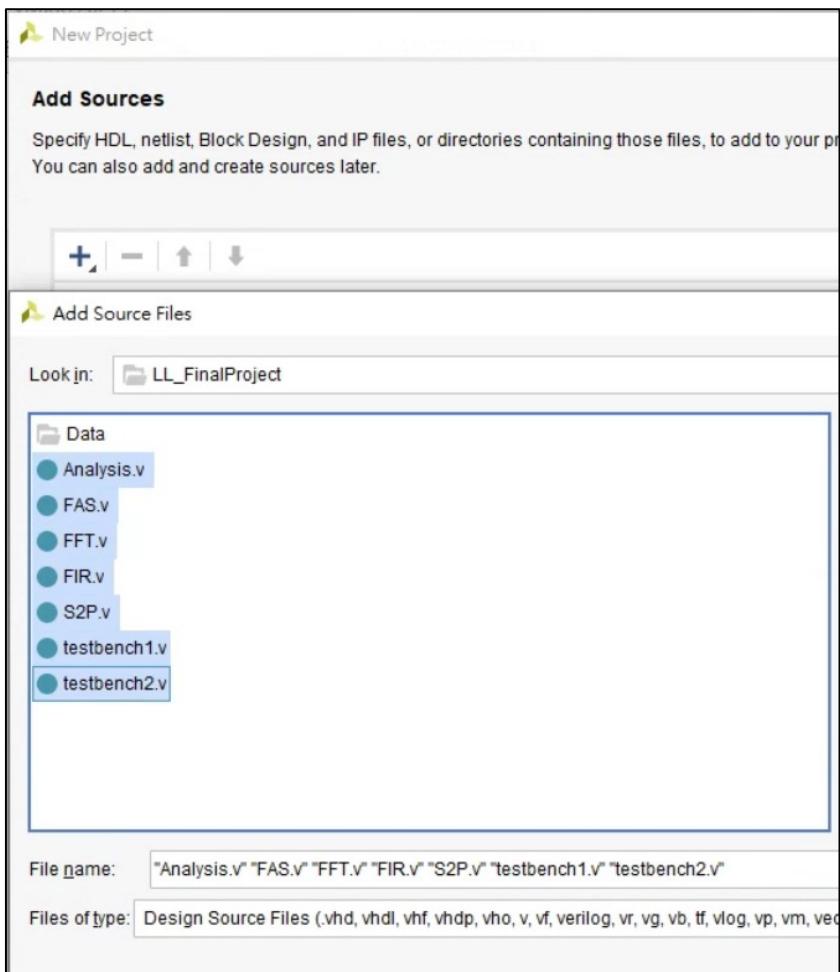
- Use Verilog to implement Frequency Analysis System described in this ppt
 - FAS.v
 - FIR.v
 - S2P.v
 - FFT.v
 - Analysis.v
- Pass all 2 testbenches (**No need to use board, we only do simulation this time**)
- Describe how you design each component in the report



Simulation Hints

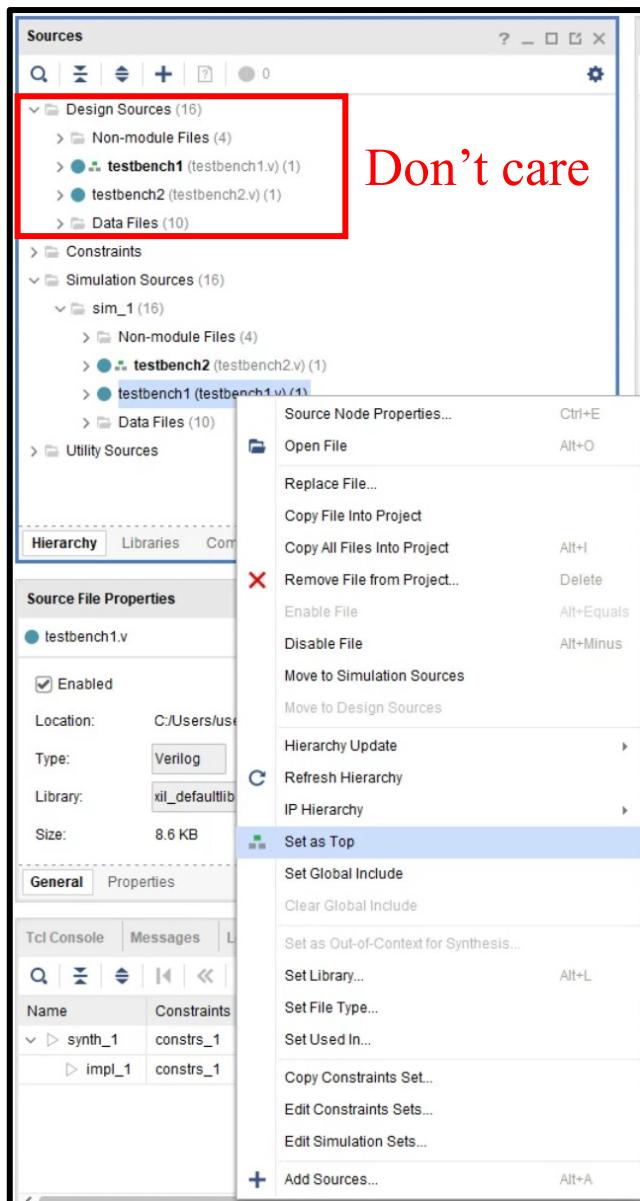
Simulation Hints – Add Sources

- Add all files

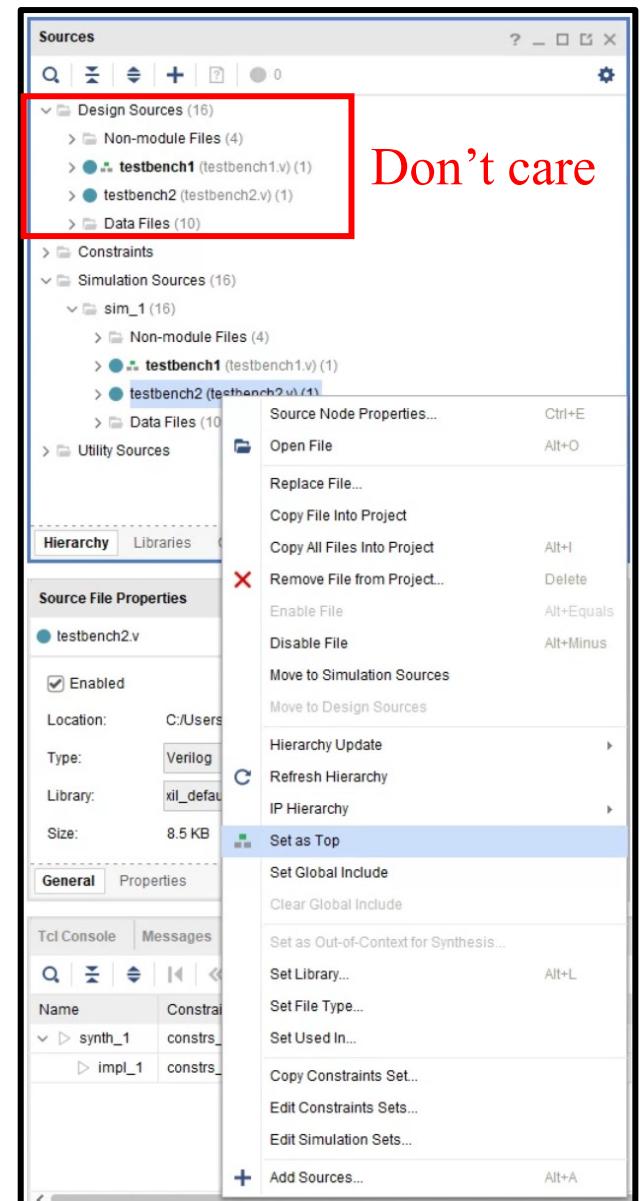


Simulation Hints – Switch between 2 testbenches

- Set the corresponding testbench as Top
- Then you can run the simulation with the testbench you chose



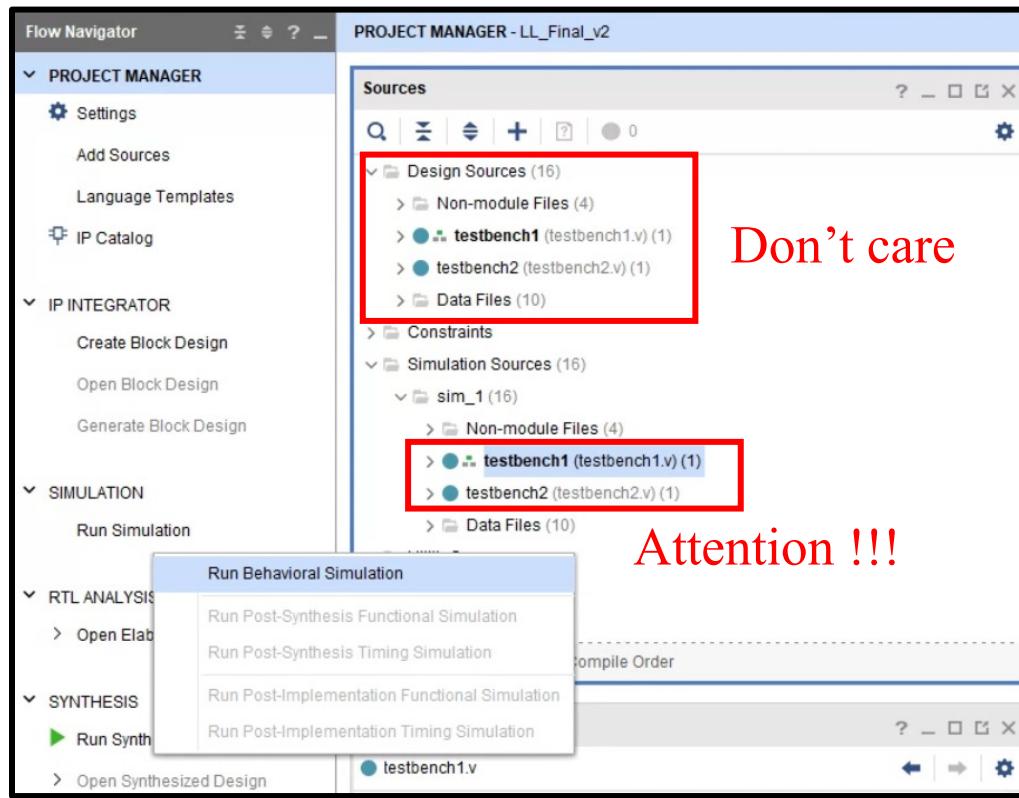
Don't care



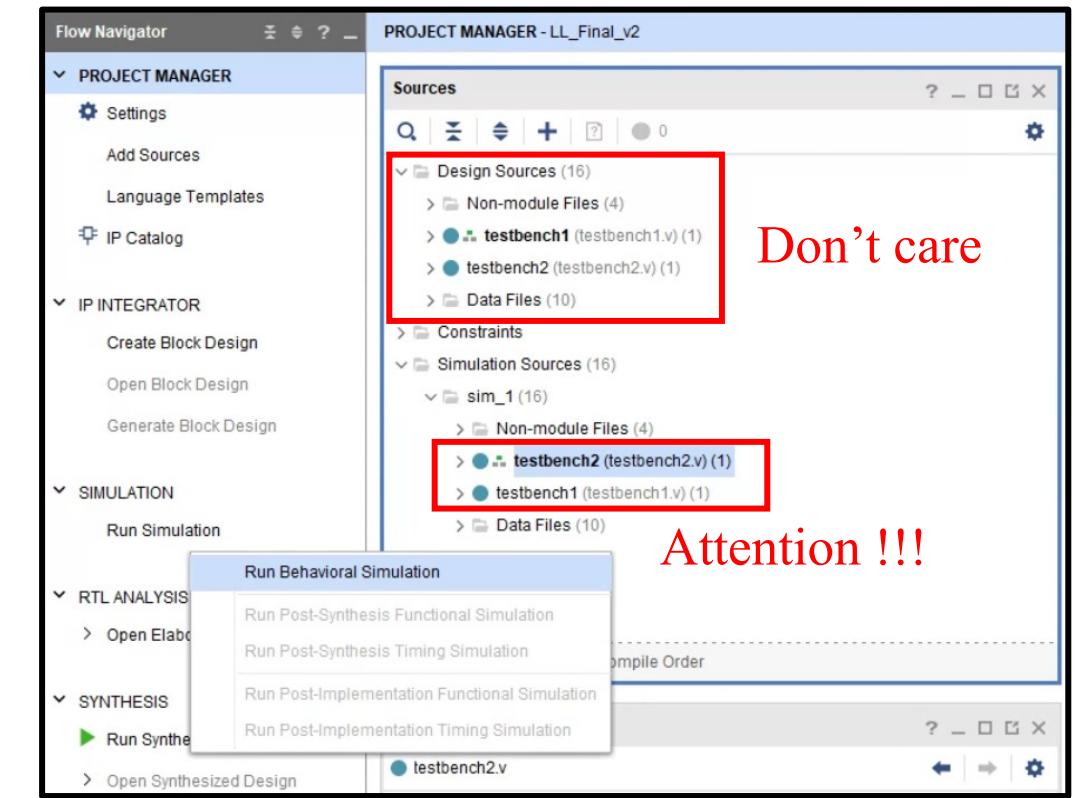
Don't care

Simulation Hints – Run Simulation

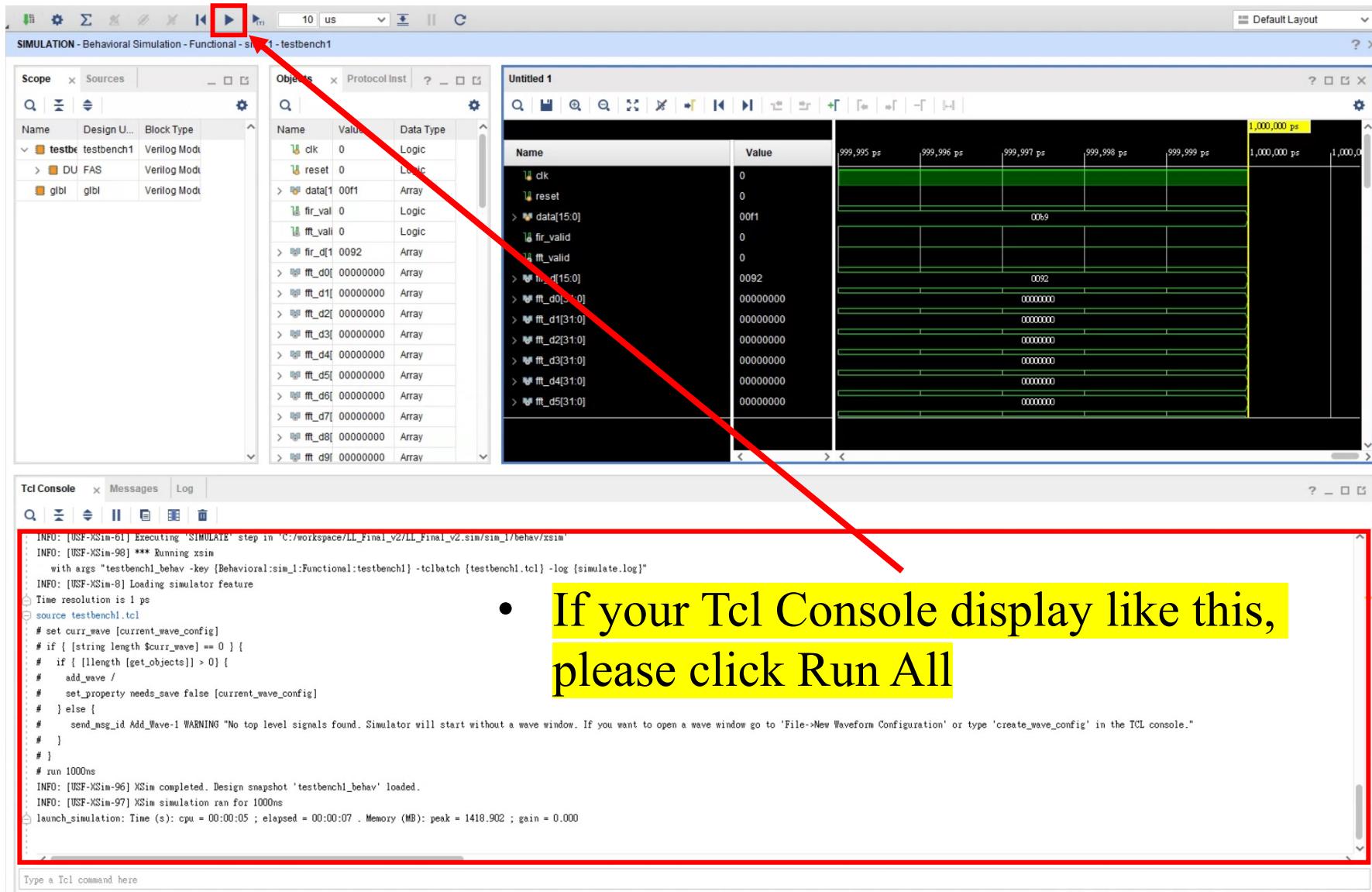
Run Testbench 1



Run Testbench 2



Simulation Hints – Run All



Simulation Hints – Successful Results

