

# 无线通信实验在线开放课程

主讲人：吴光 博士

广东省教学质量工程建设项目



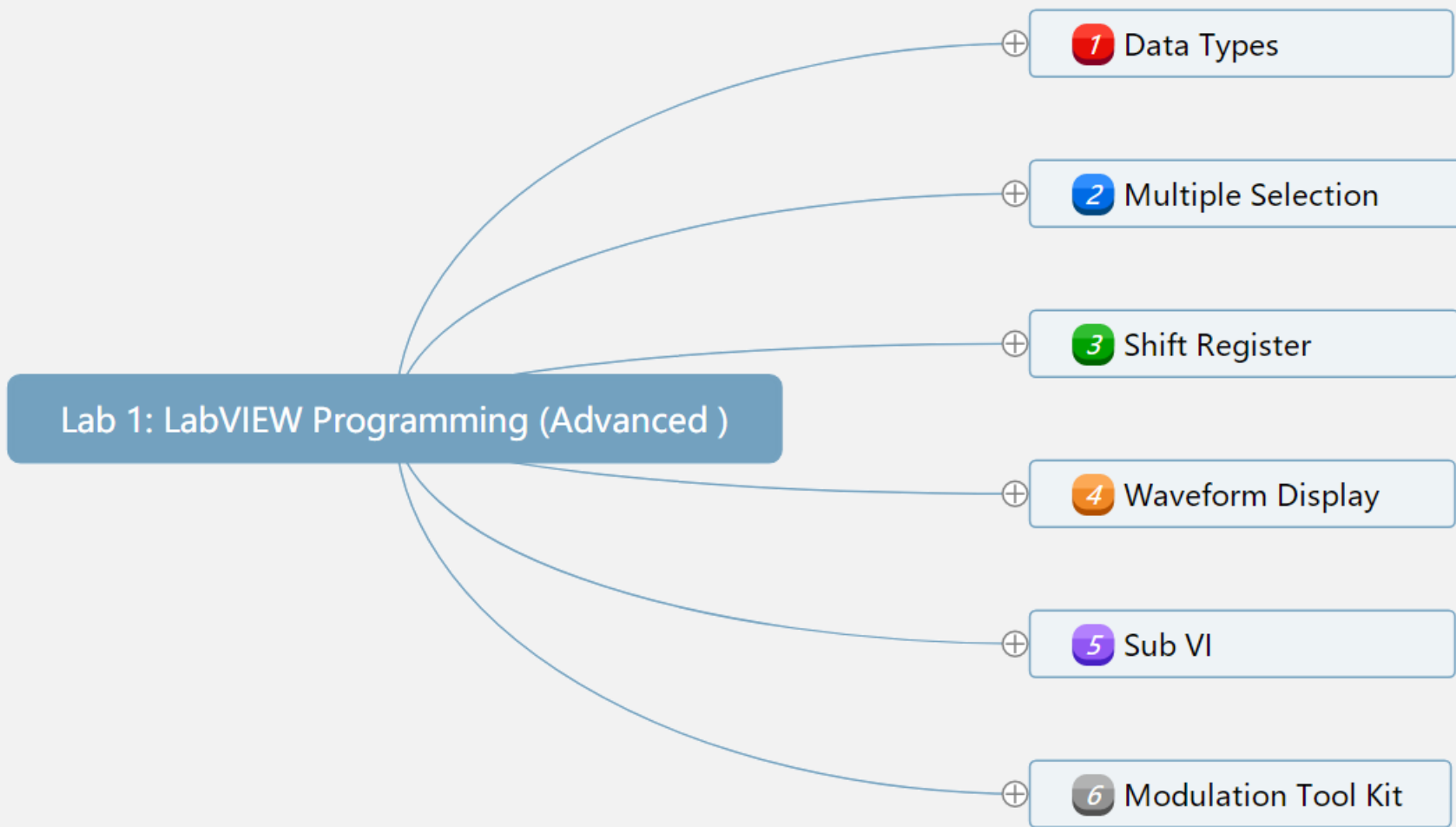


# LabVIEW 通信编程

(Advanced)

主讲人：吴光 博士

Email: [wug@sustech.edu.cn](mailto:wug@sustech.edu.cn)



# Test 1



IF (The input number N is an **even integer**)

$$\text{Sum}=2+4+6+\dots+N;$$

IF (The input number N is an **odd integer**)

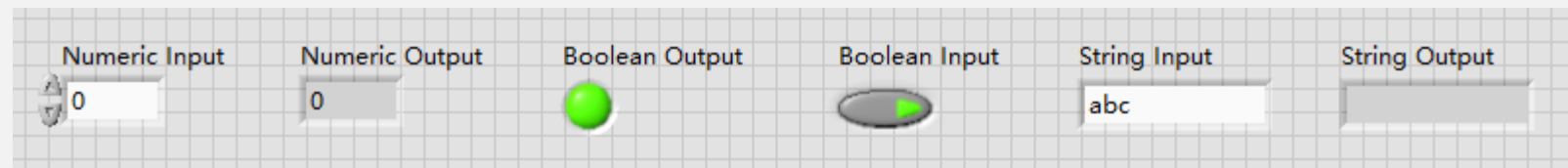
$$\text{Sum}=1+3+5+\dots+N;$$

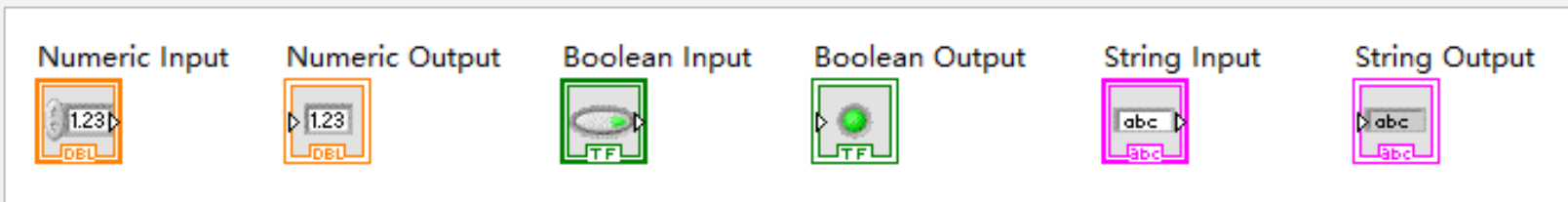


# Exercise: Test 1

# Basic Data Type

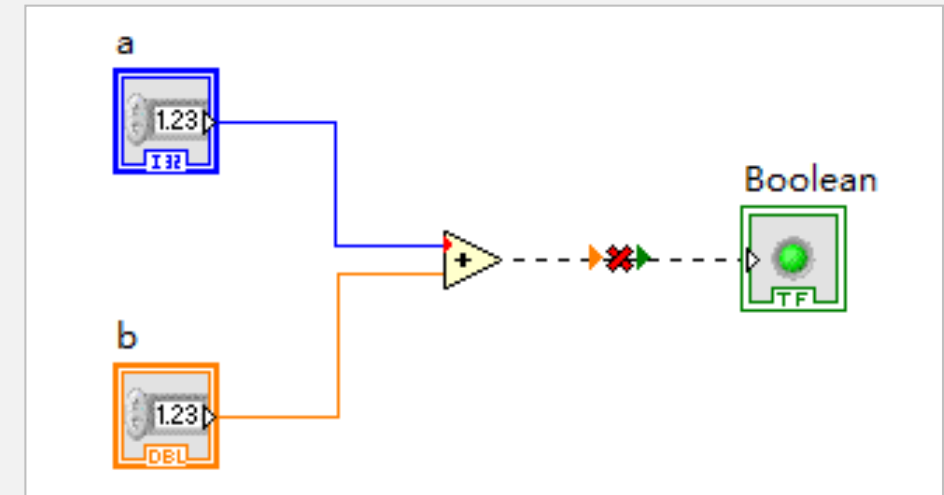
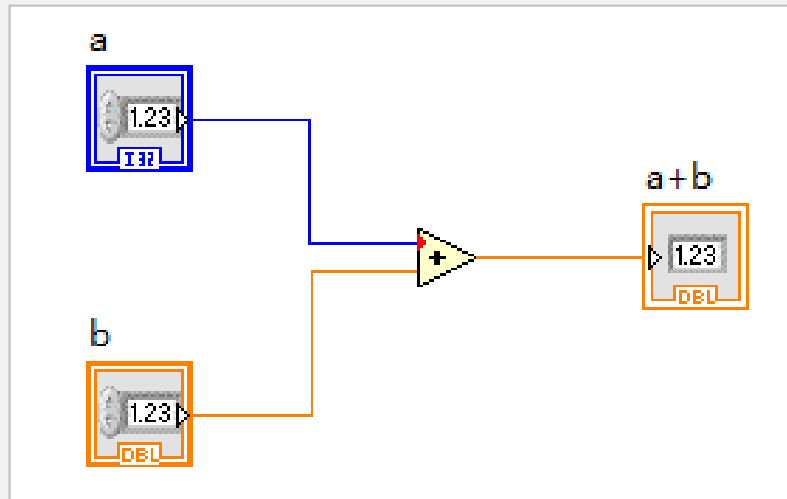
Basic Data Type:





Type	Color	Scalar	1D-Array	2D-Array
Int	Blue			
DBL	Orange			
Boolean	Green			
String	Pink			

# Type Mismatch

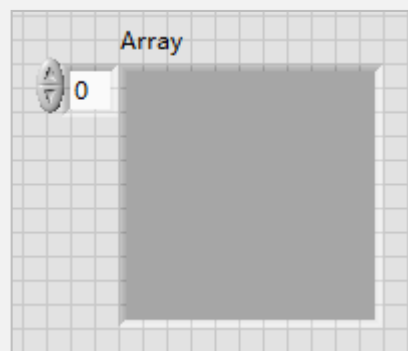
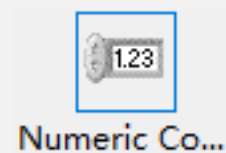
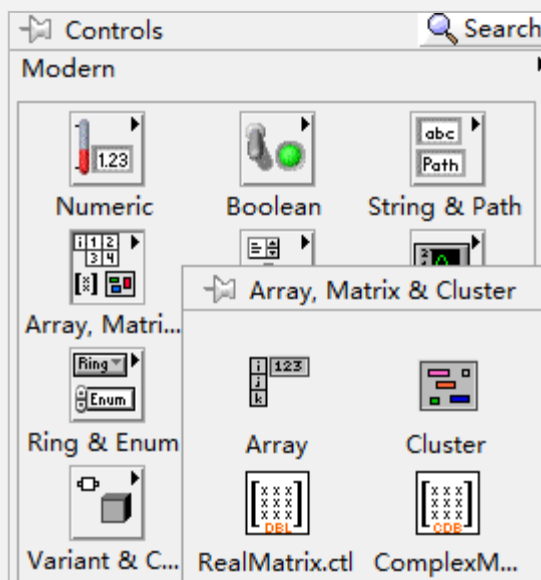
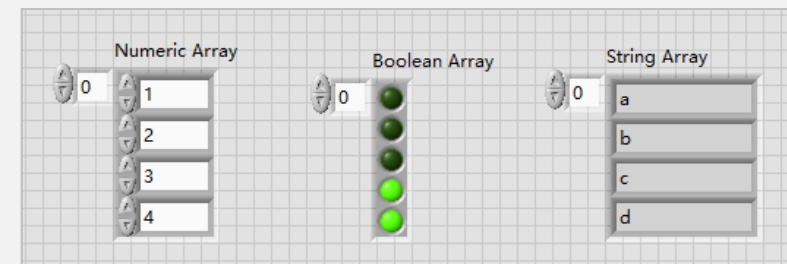


You have connected two terminals of different types.  
The type of the source is double [64-bit real (~15 digit precision)].  
The type of the sink is boolean (TRUE or FALSE).

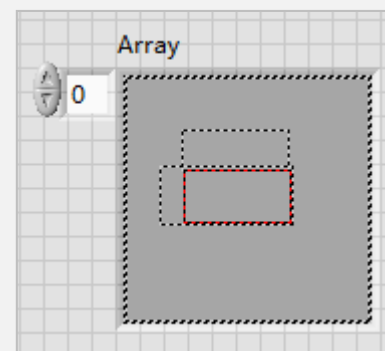


# 1-Dimension Array

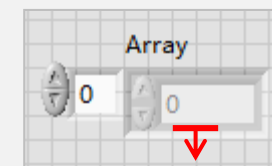
- Array
  - Multidimensional collections of **like data**
  - Vectors, matrices, array of booleans, etc.



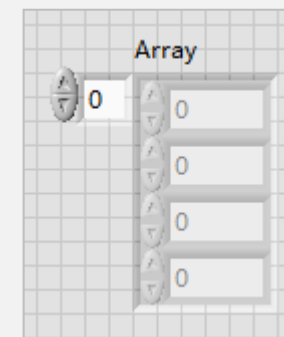
Step 1



Step 2



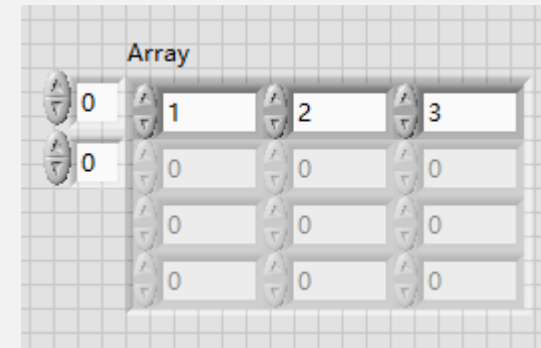
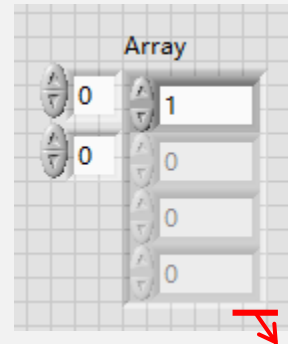
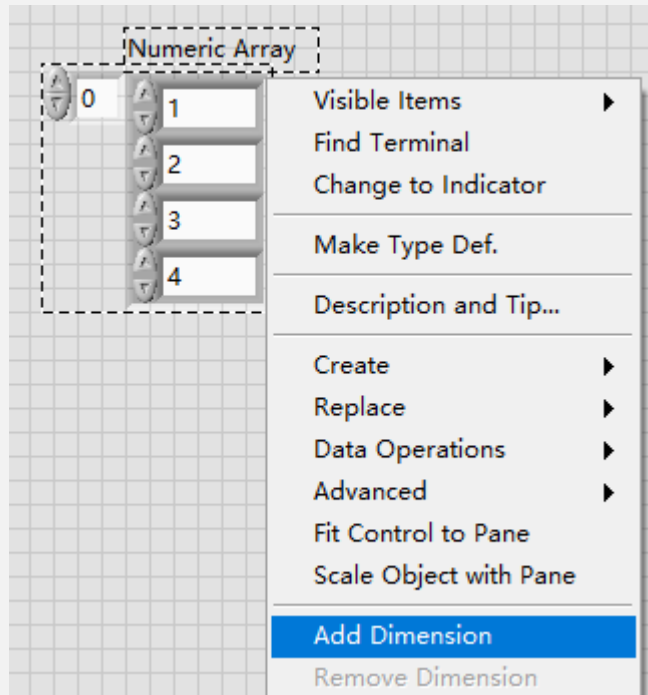
Step 3





# Exercise: Build 1D Array

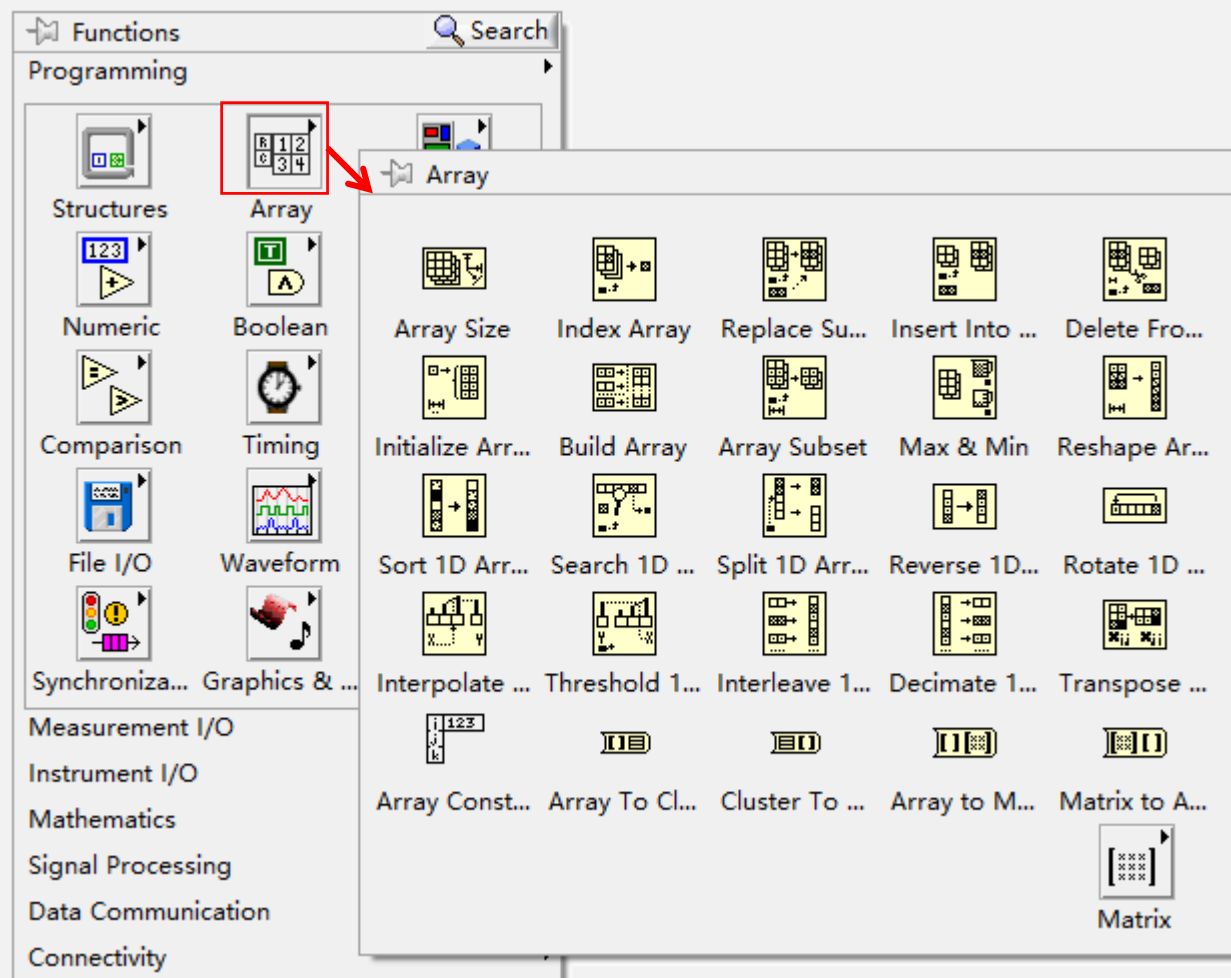
# 2-Dimension Array





# Exercise: Build 2D Array

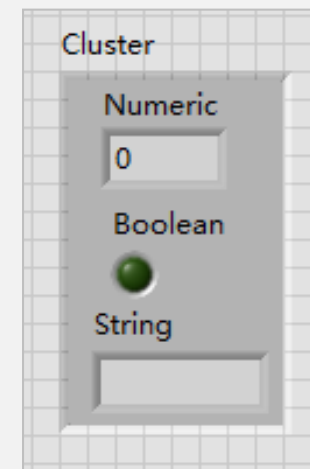
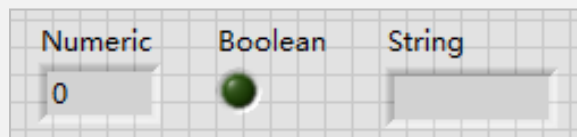
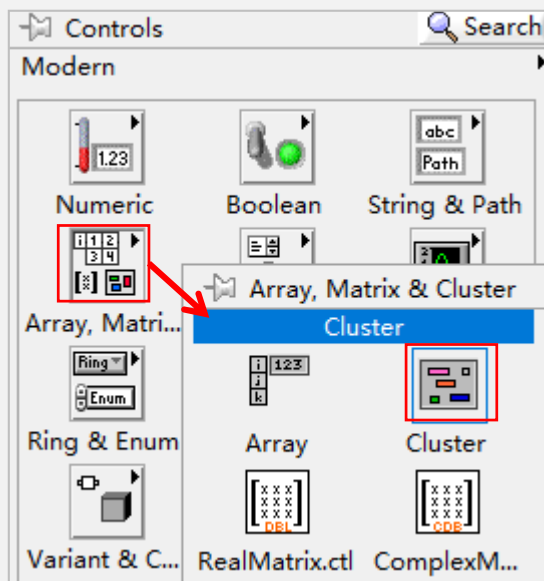
# Array functions



# Cluster

- Cluster

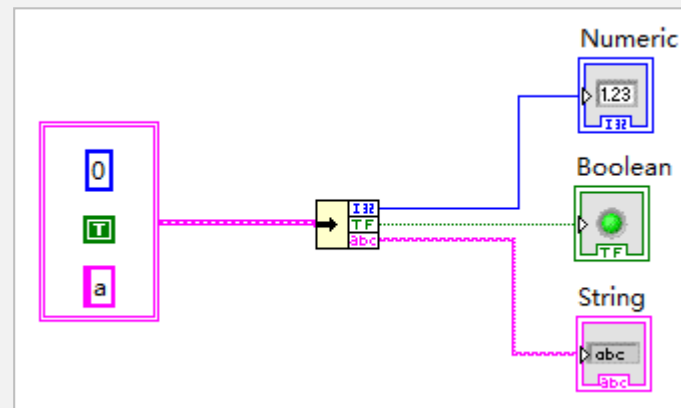
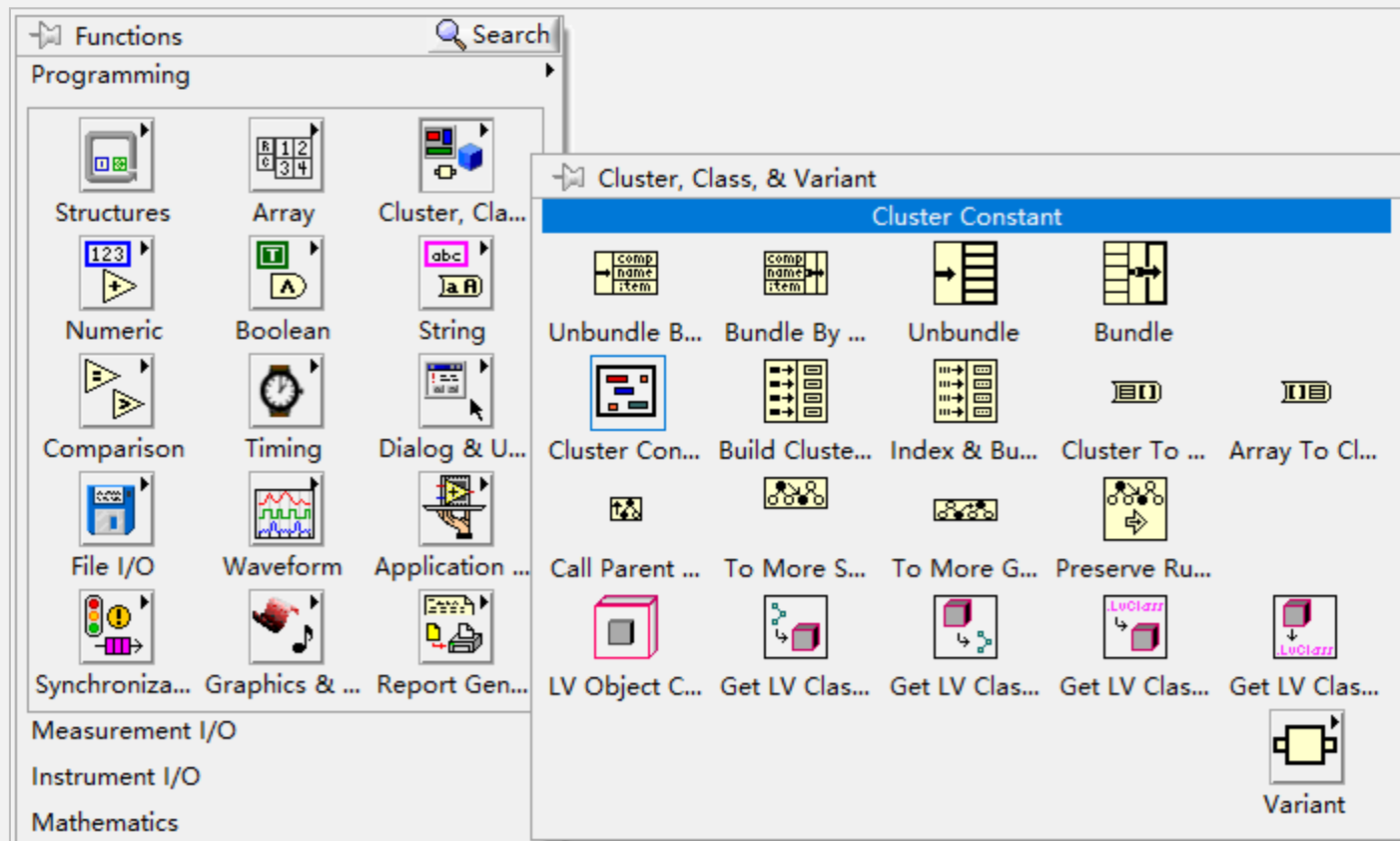
- Collections of **unlike data** used for conveniently transporting the data from one place to another.
- Similar to the idea of a Struct in C or Matlab.





# Exercise: Build a Cluster

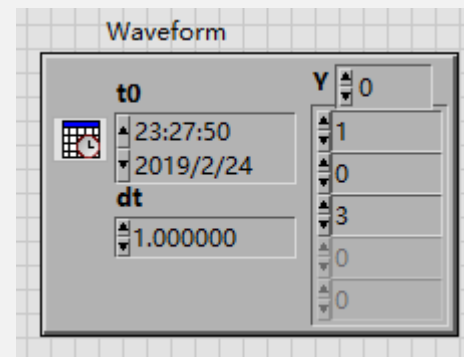
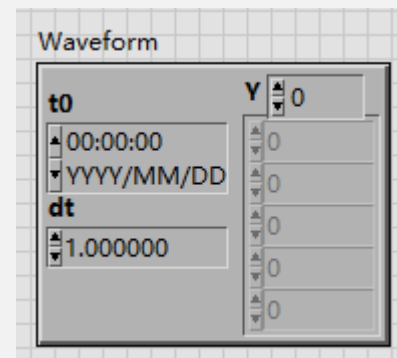
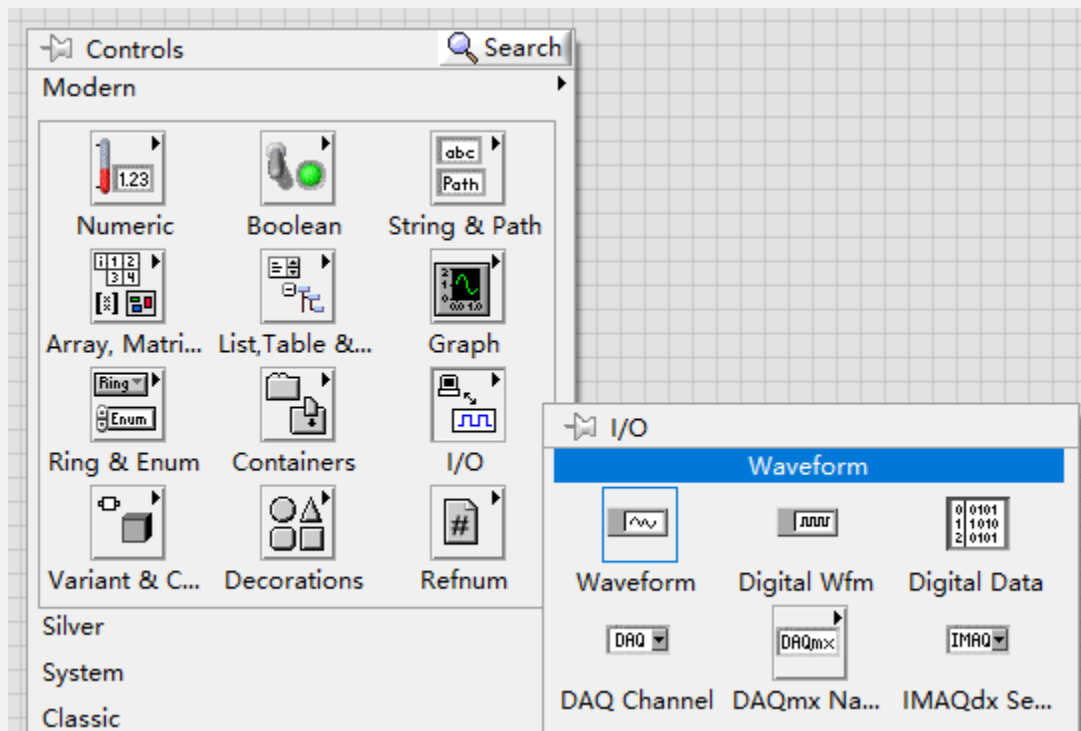
# Cluster functions





# Waveform

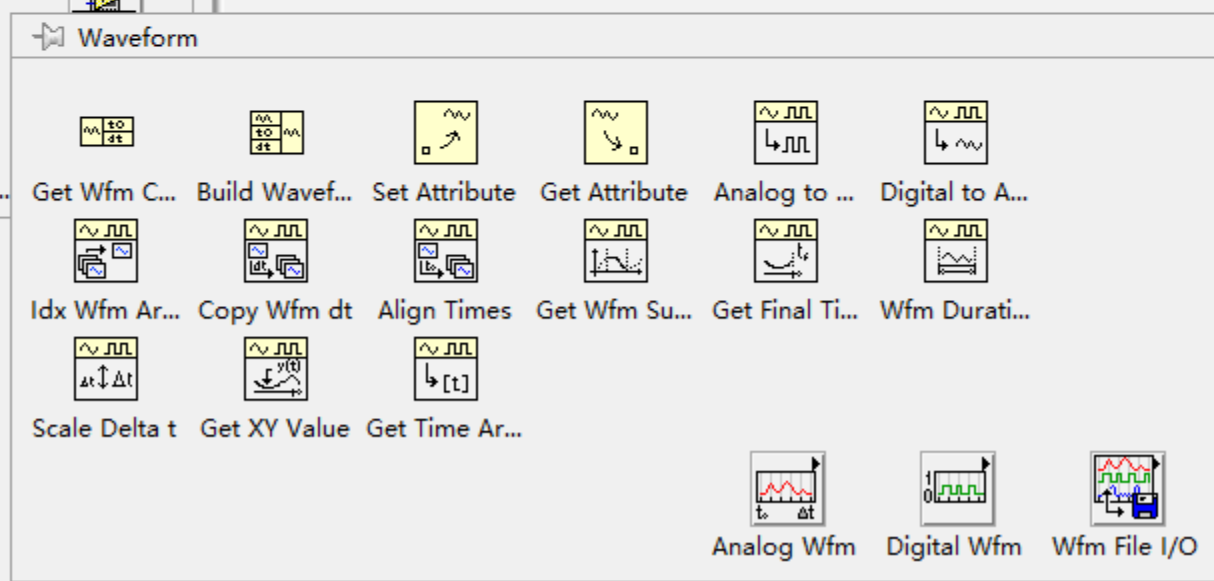
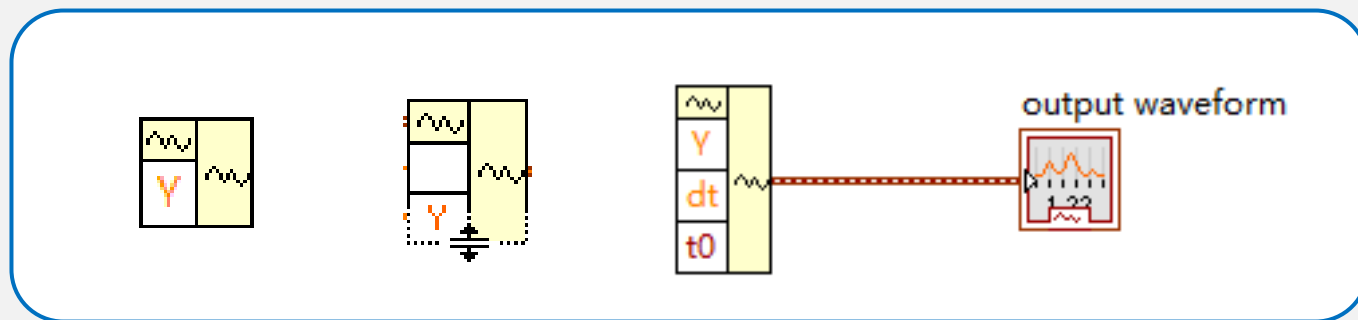
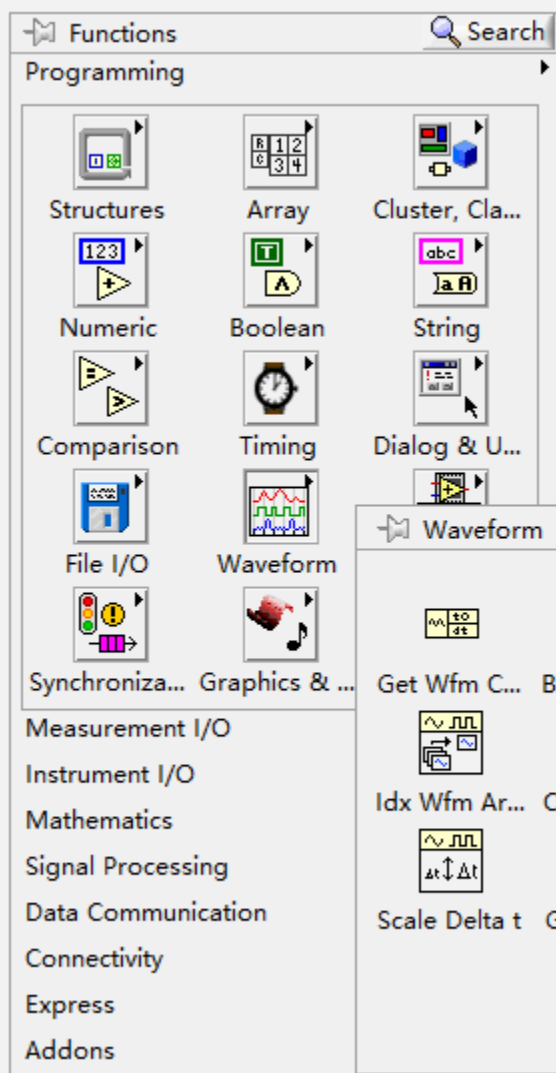
- **t0**: Specifies the start time of the waveform.
- **dt**: Specifies the time interval in seconds between data points in the waveform.
- **Y**: Specifies the data values of the waveform.



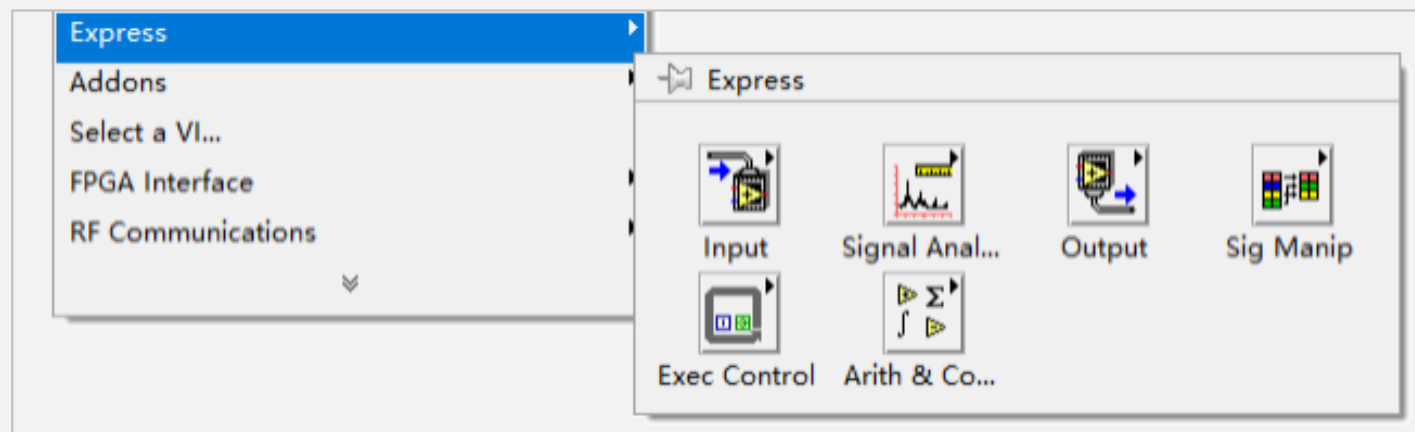
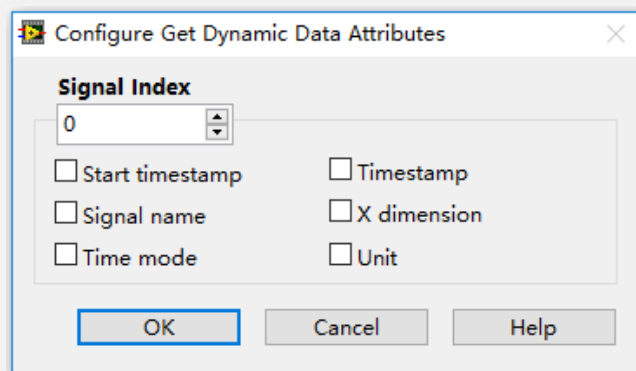
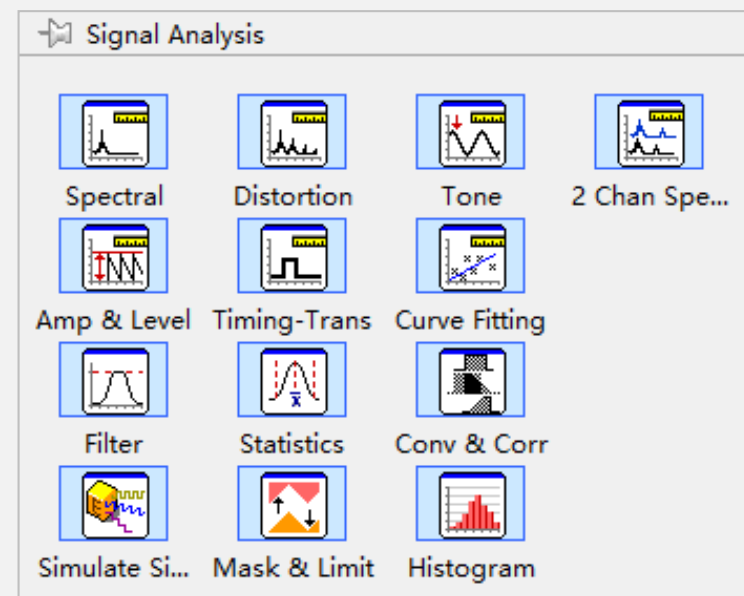
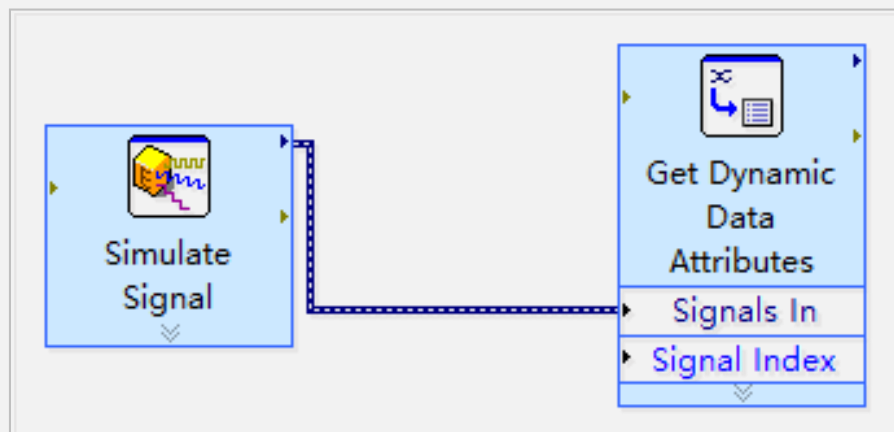


# Exercise: Build a Waveform

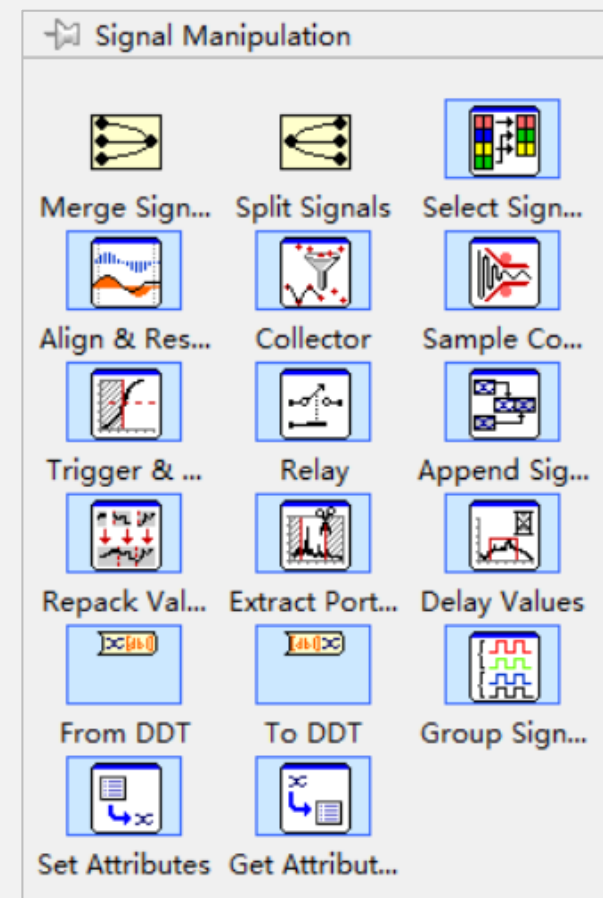
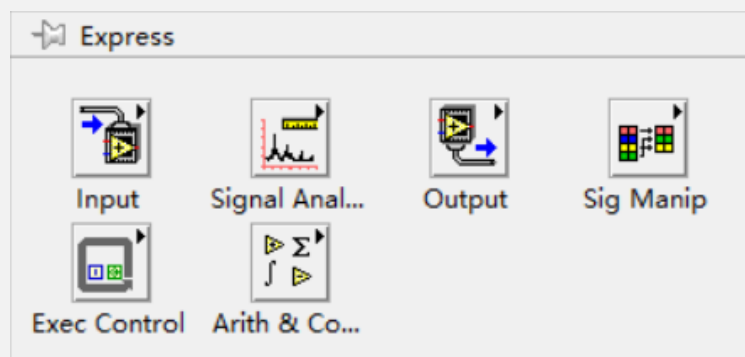
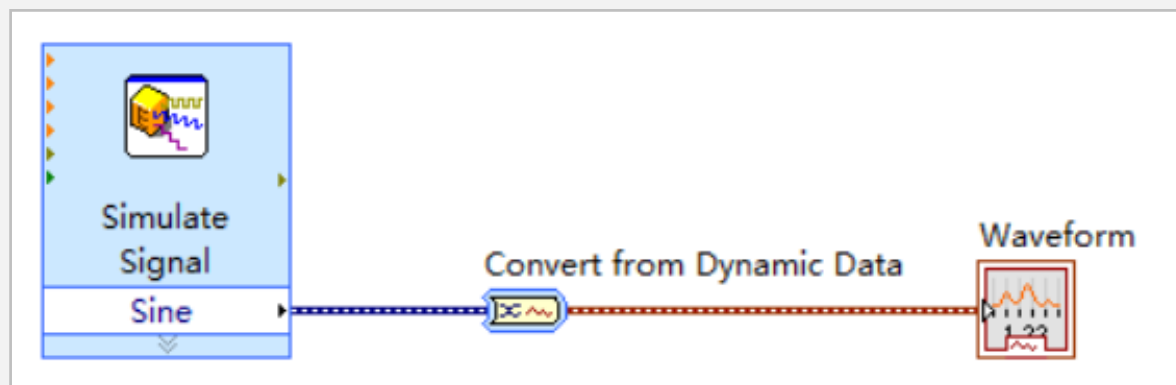
# Waveform functions



# Dynamic Data

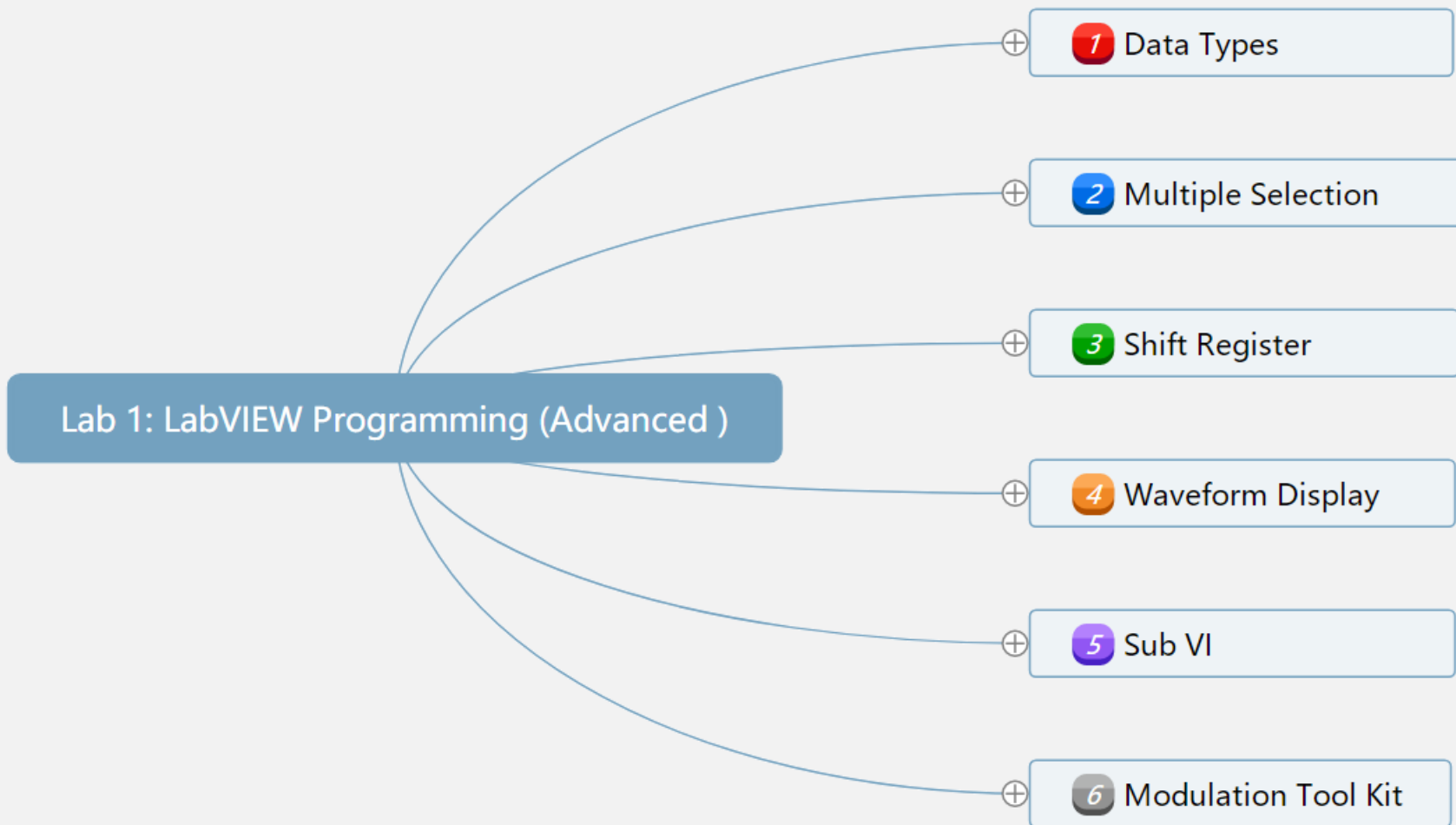


# Dynamic Data



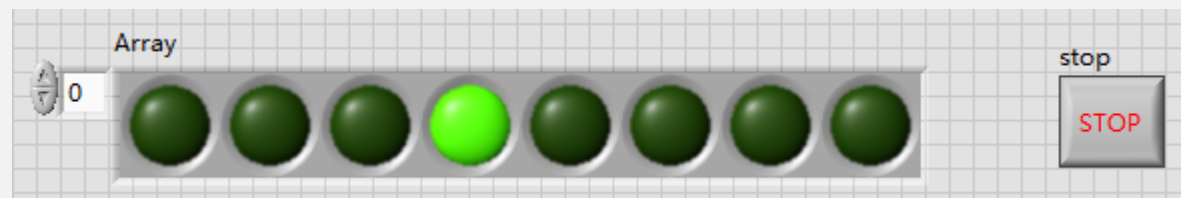


# I have an ability to understand:





# Demo: Blinking LED



State 1



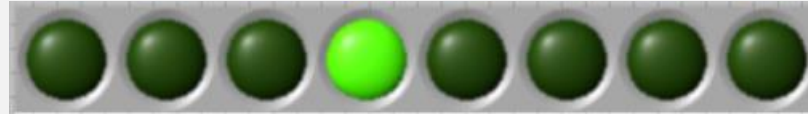
State 2



State 3



State 4



State 5



State 6



State 7



State 8

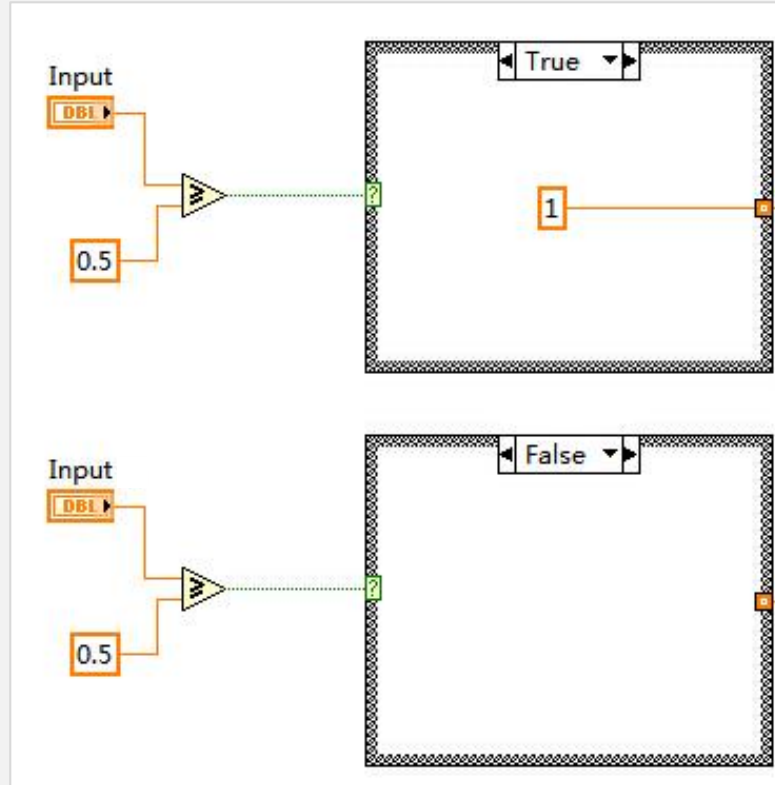




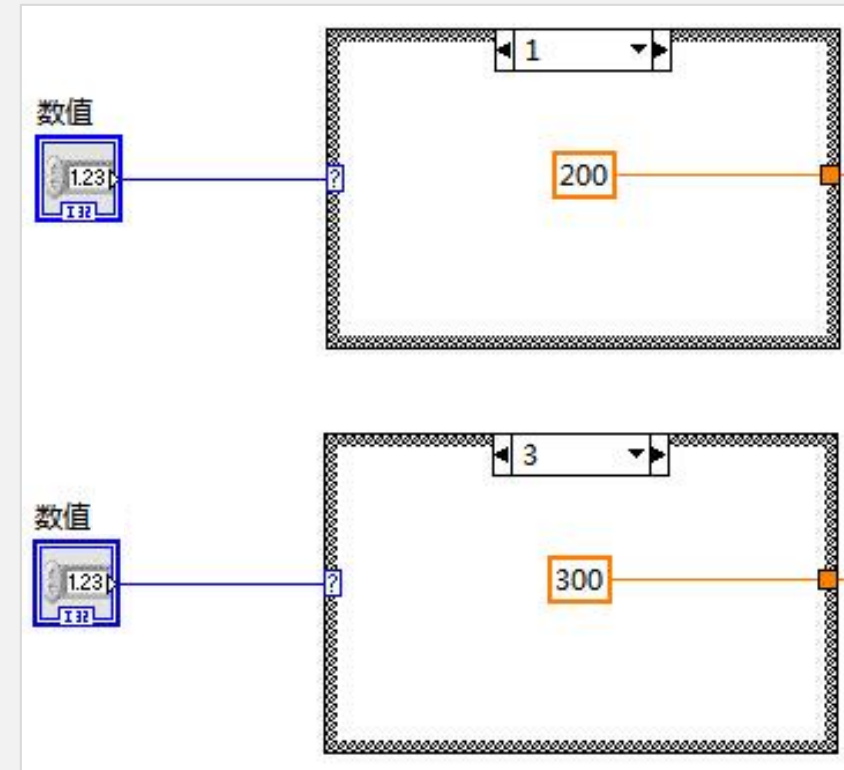


# Exercise: Blinking LED

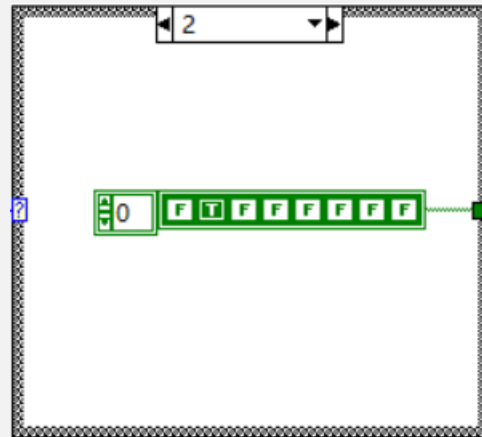
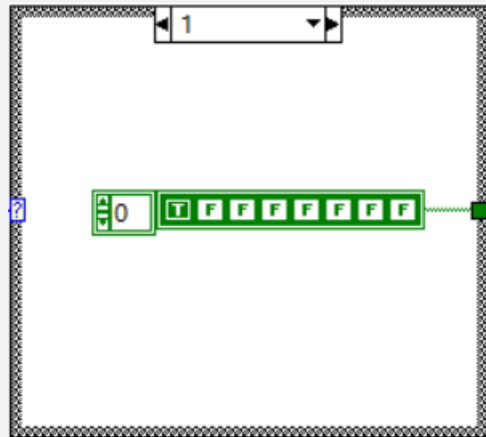
# Input of Case Selector



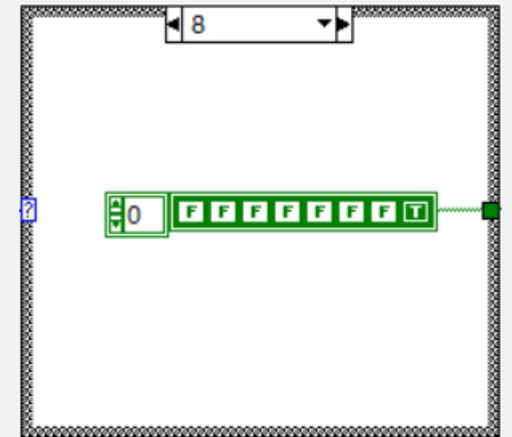
(a) Bool Input

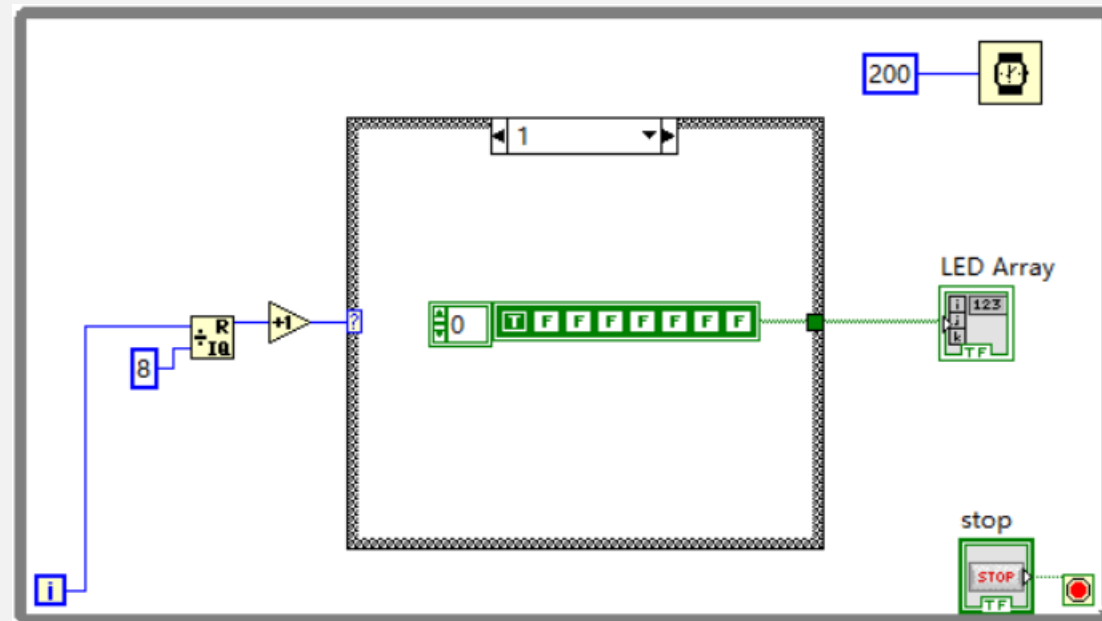


(b) Numeric Input



.....







# Nested-If Structure

```
IF (Input ≤ 1)
```

```
    Output1 = 1;
```

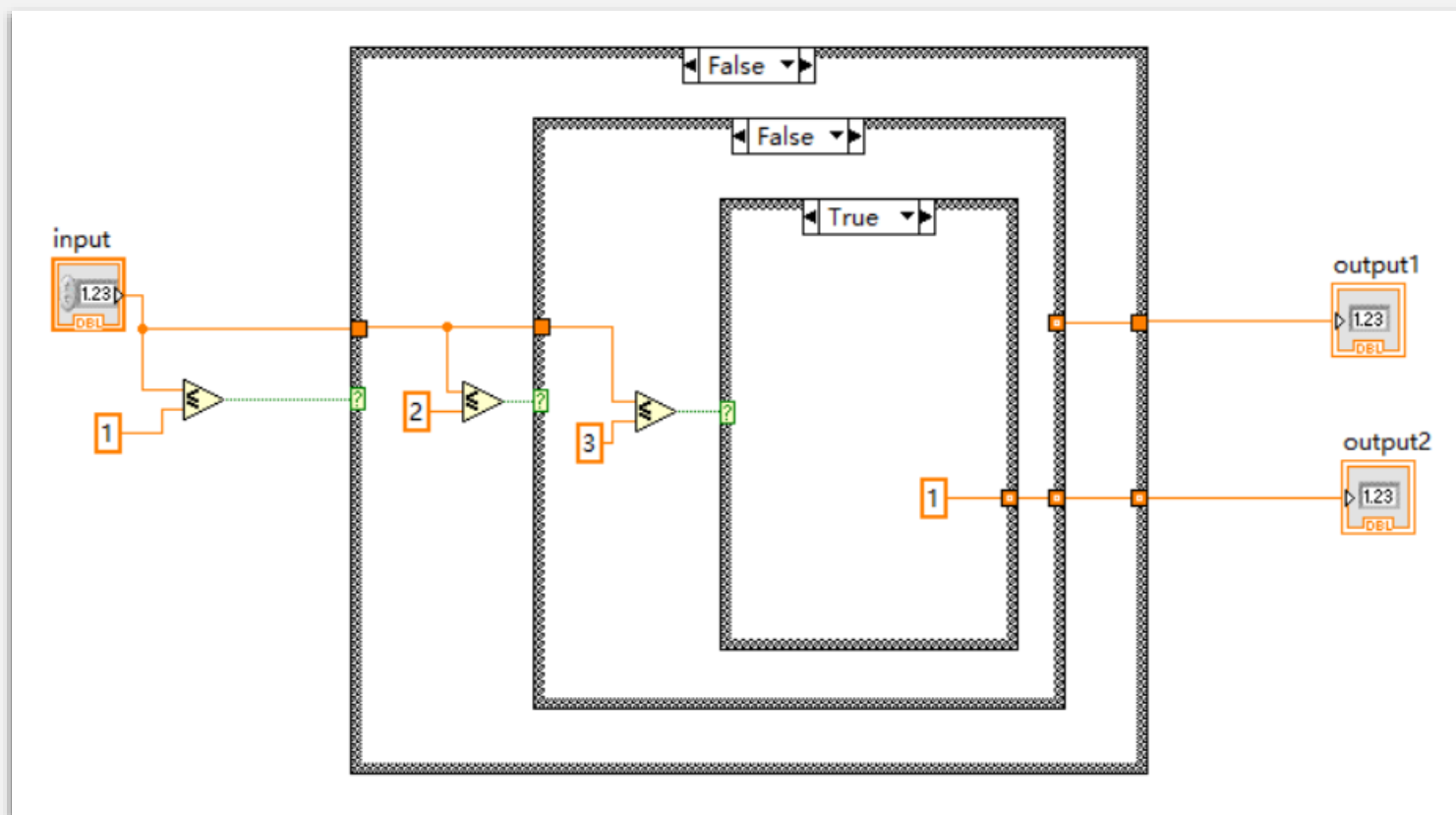
```
        Elseif (Input > 1 && Input ≤ 2)
```

```
            Output1 = 2;
```

```
                Elseif (Input > 2 && Input ≤ 3)
```

```
                    Output2 = 1;
```

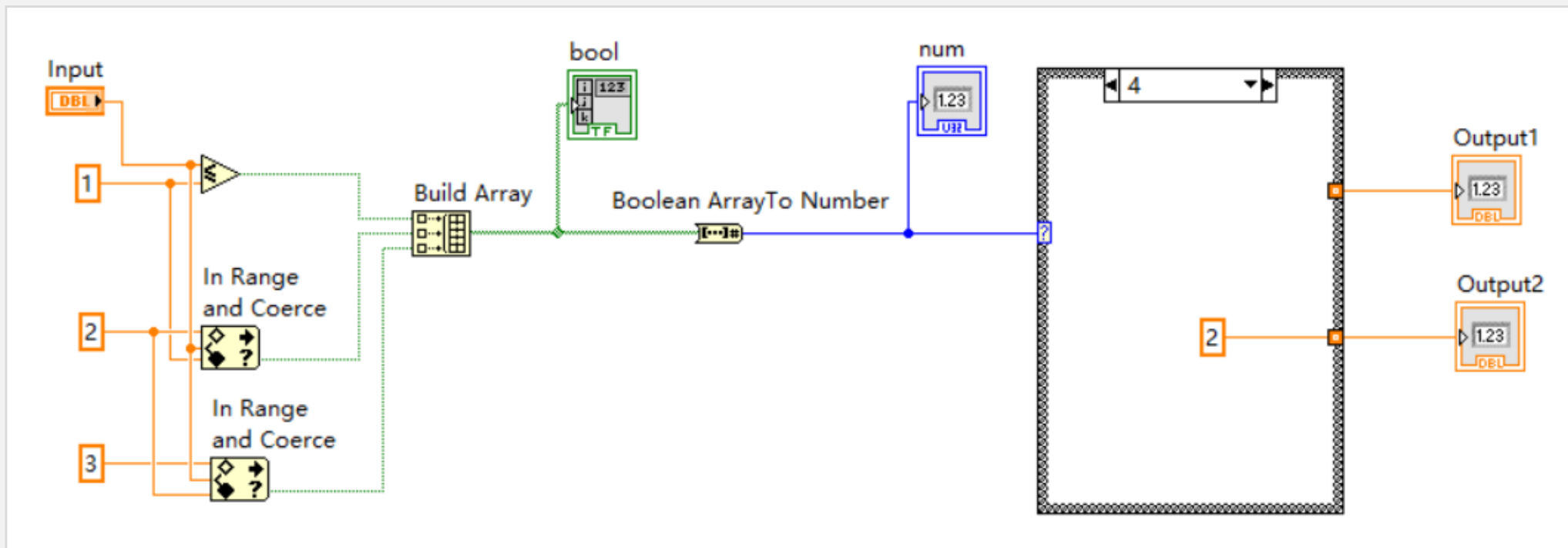
# Solution 1





input		bool		num		output
$\text{Input} \leq 1$	→	001	→	1	→	Output1=1; Output2=0;
$1 < \text{Input} \leq 2$	→	010	→	2	→	Output1=2; Output2=0;
$2 < \text{Input} \leq 3$	→	100	→	4	→	Output1=0; Output2=1;
其他	→	000	→	0	→	Output1=0; Output2=0;

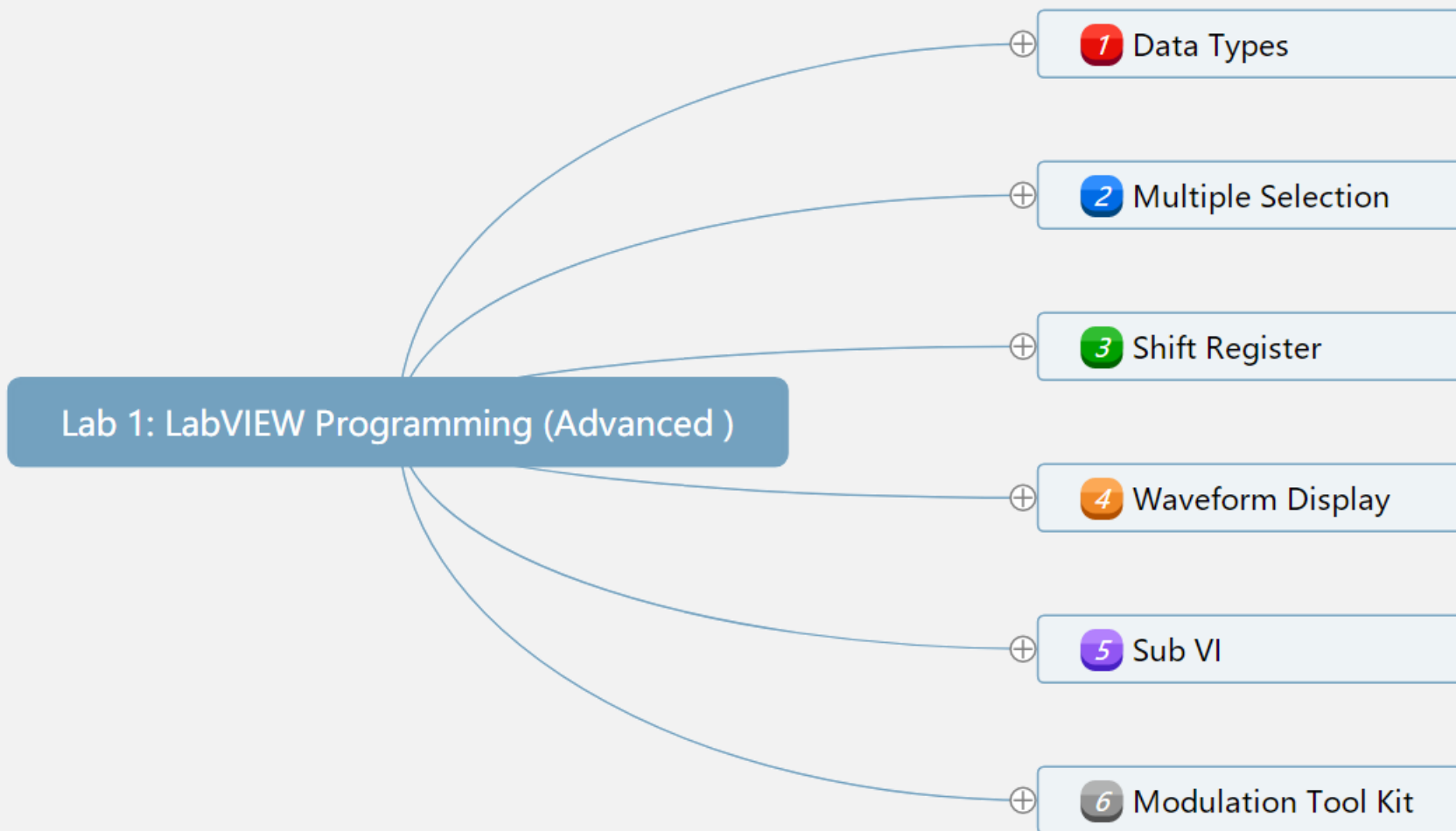
# Solution 2



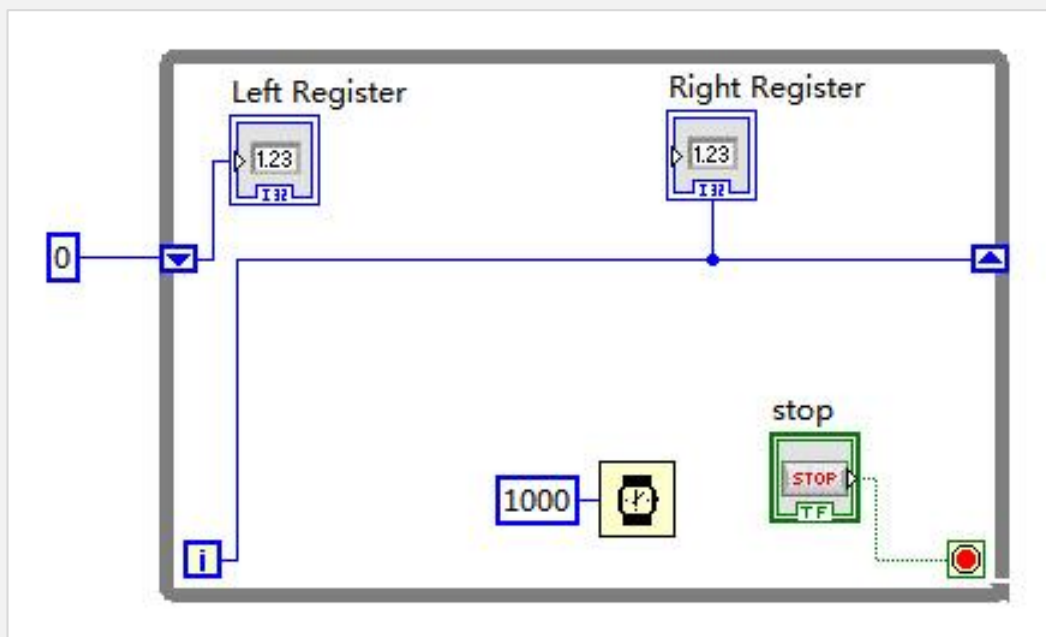




# I have an ability to understand:

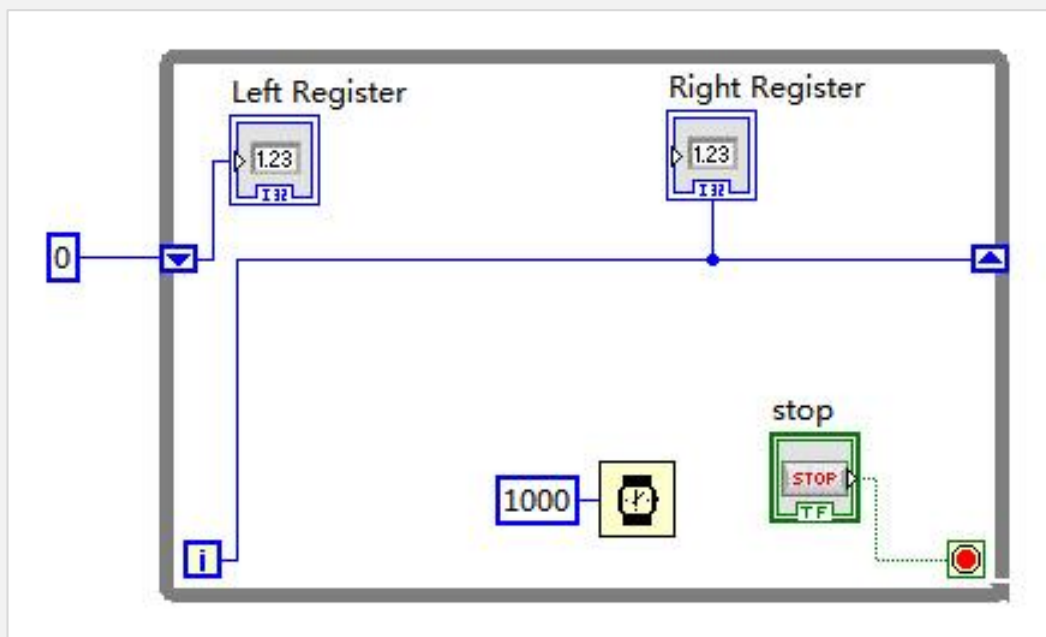


# Introduction to Shift Register



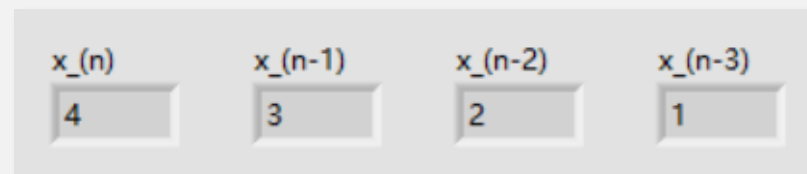
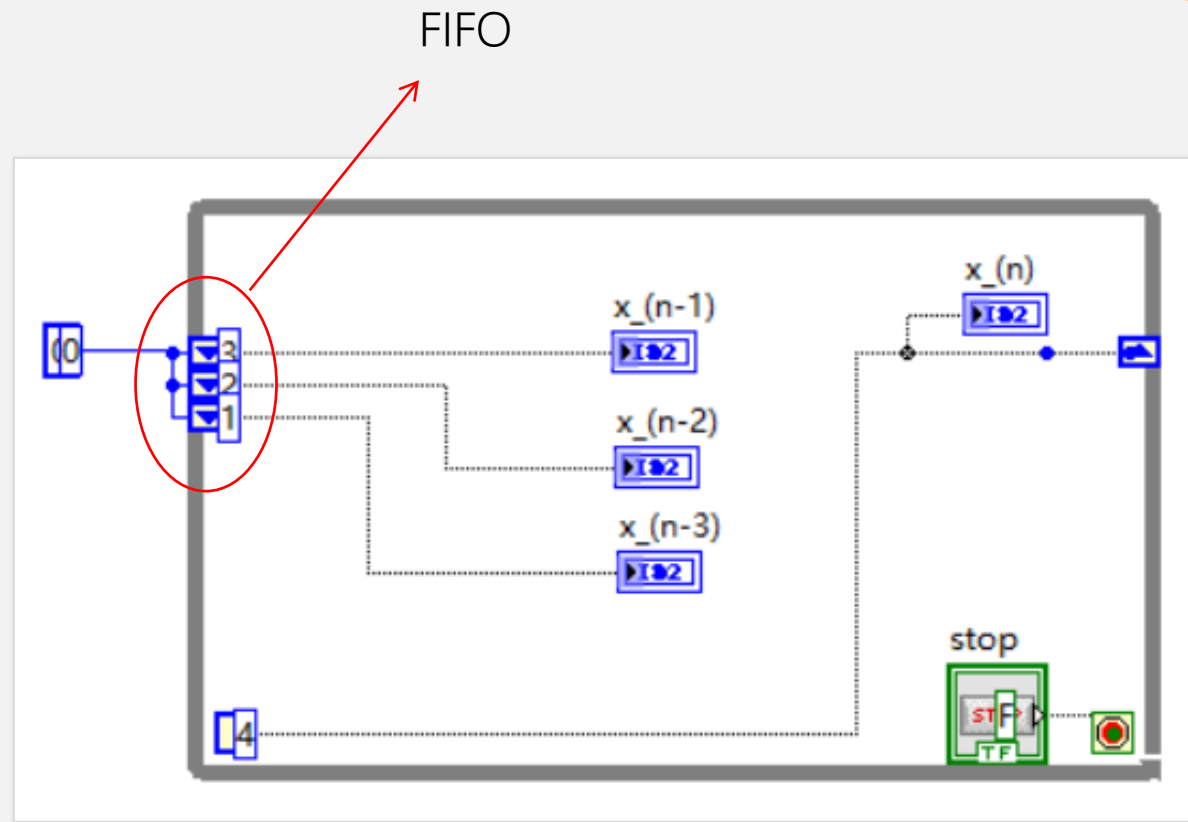
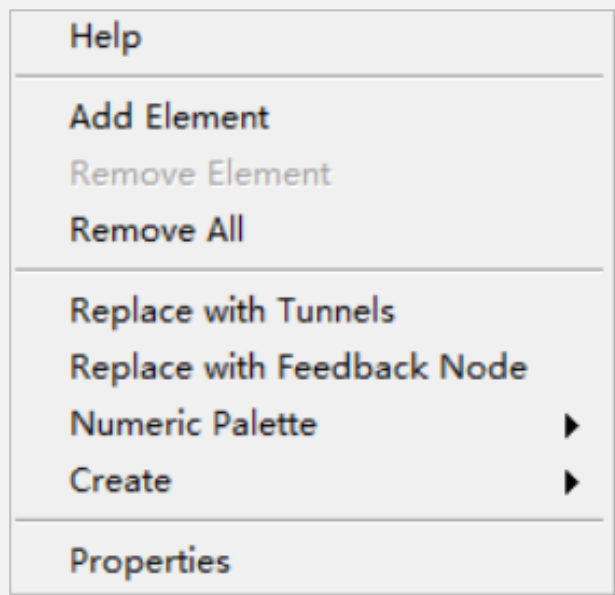
- Visible Items ▶
- Help
- Examples
- Description and Tip...
- Breakpoint ▶
- Structures Palette ▶
- ✓ Auto Grow
- Exclude from Diagram Cleanup
- Conditional Terminal
- Configure Iteration Parallelism...
- Replace with While Loop
- Remove For Loop
- Add Shift Register
- Properties

# Introduction to Shift Register

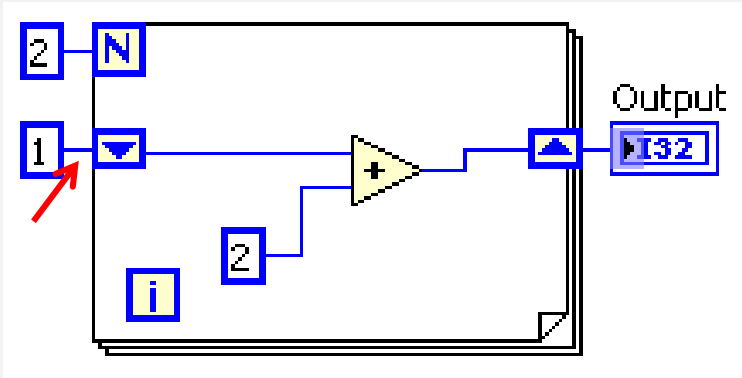
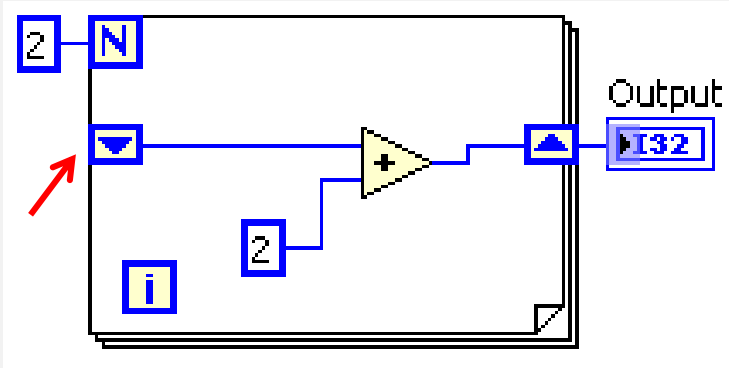


i=0	Left Register	Right Register
	0	0
i=1	Left Register	Right Register
	0	1
i=2	Left Register	Right Register
	1	2
i=3	Left Register	Right Register
	2	3

# Shift Register



# Initialization of Shift Register

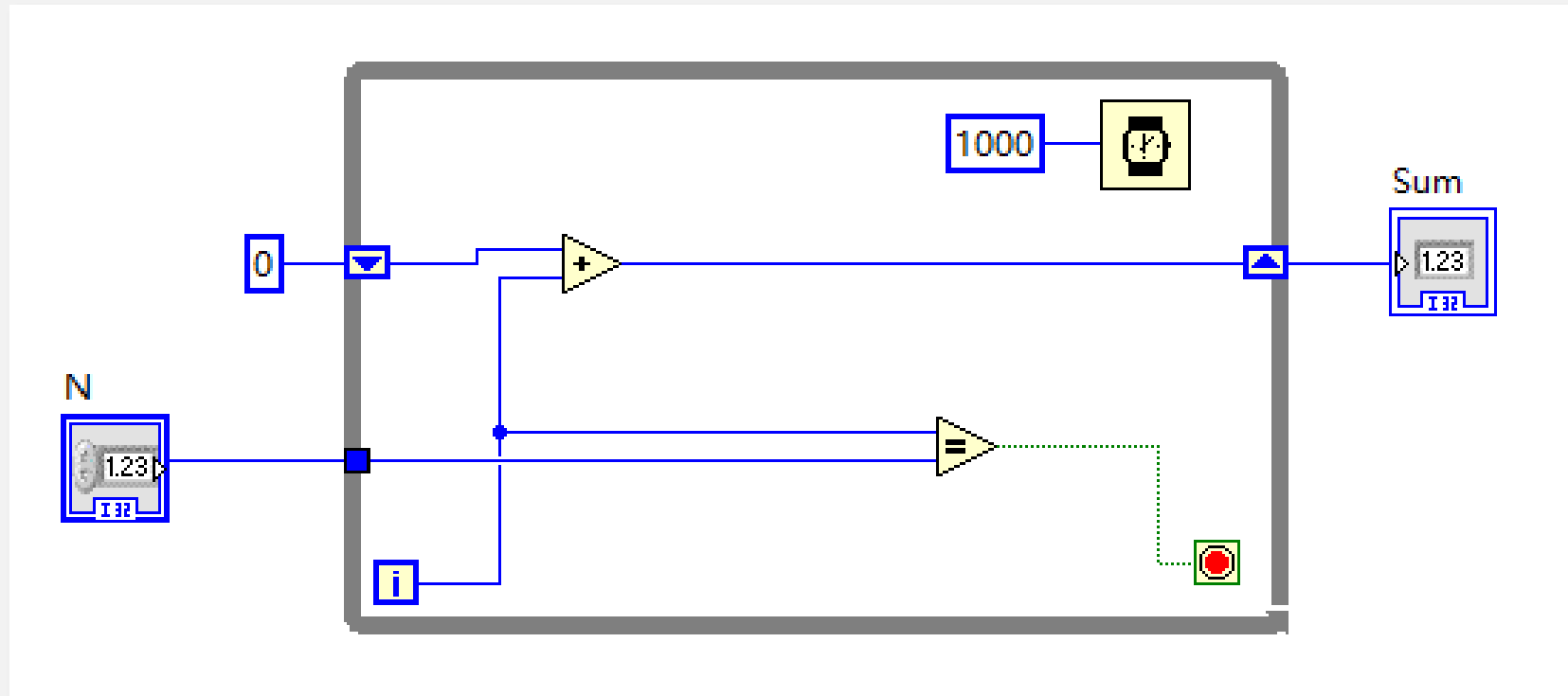
Block Diagram	Firstly	Next
	Output = 5	Output = 5
	Output = 4	Output = 8




# Demo: Shift Register Accumulator

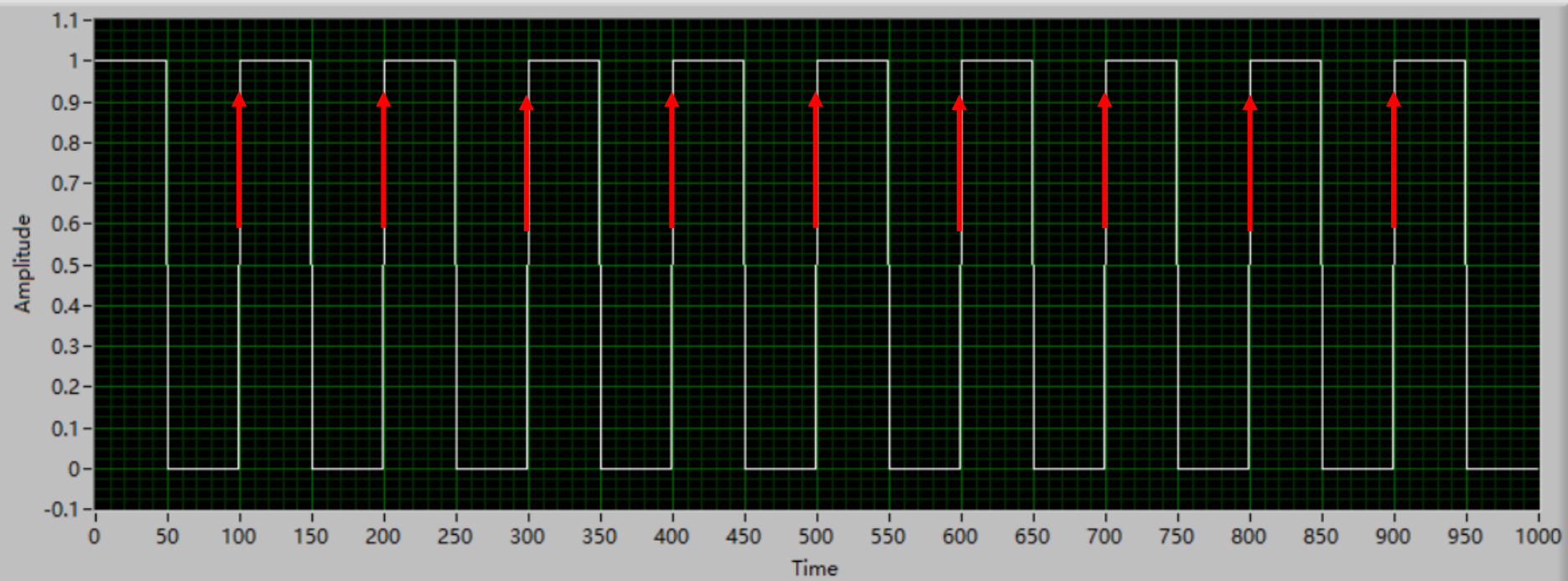
Create a VI to output the sum of numbers from 1 to the number input.

$$\text{Sum}(N) = \text{Sum}(N - 1) + N$$

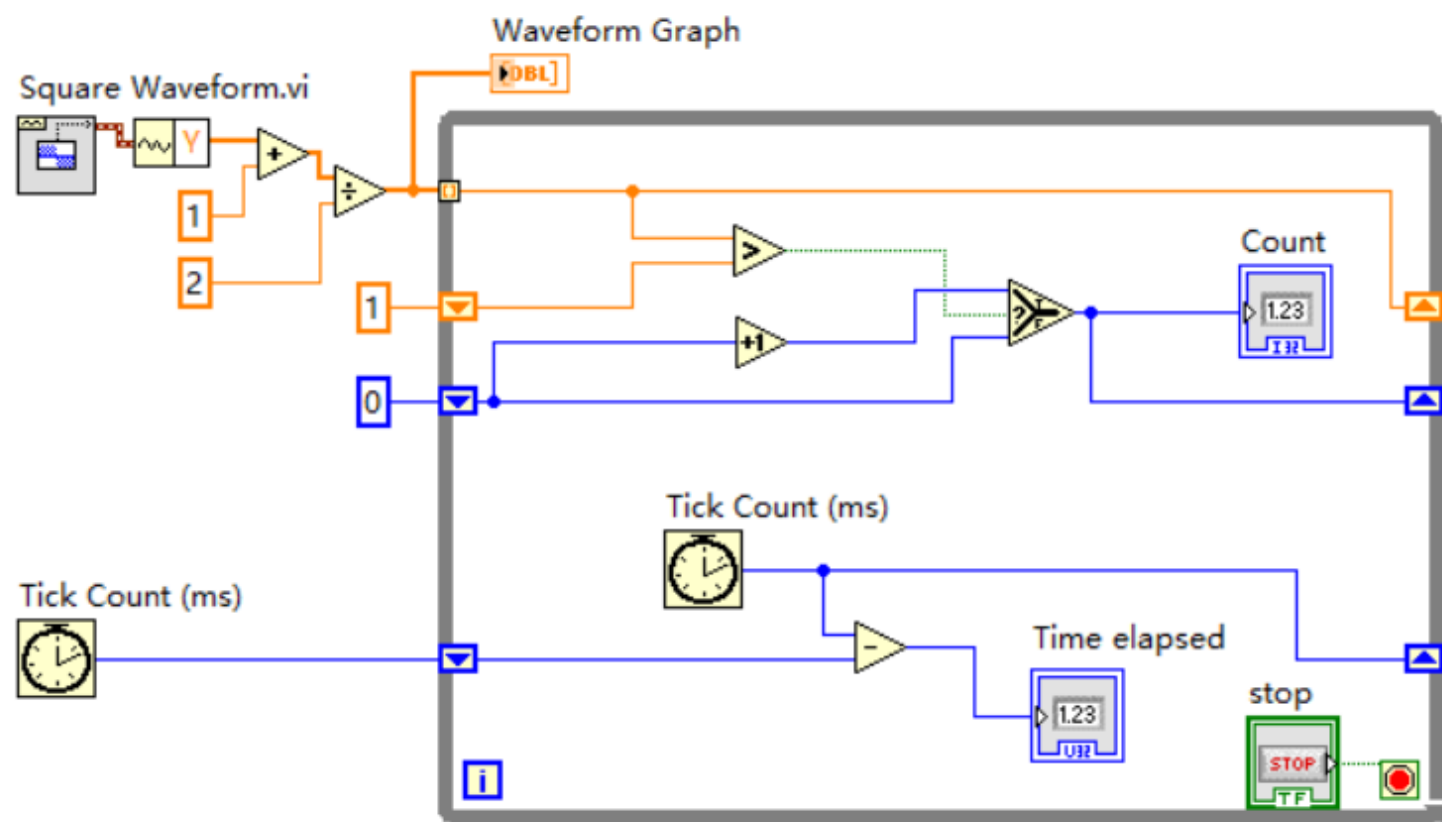


Waveform Graph

Plot 0 

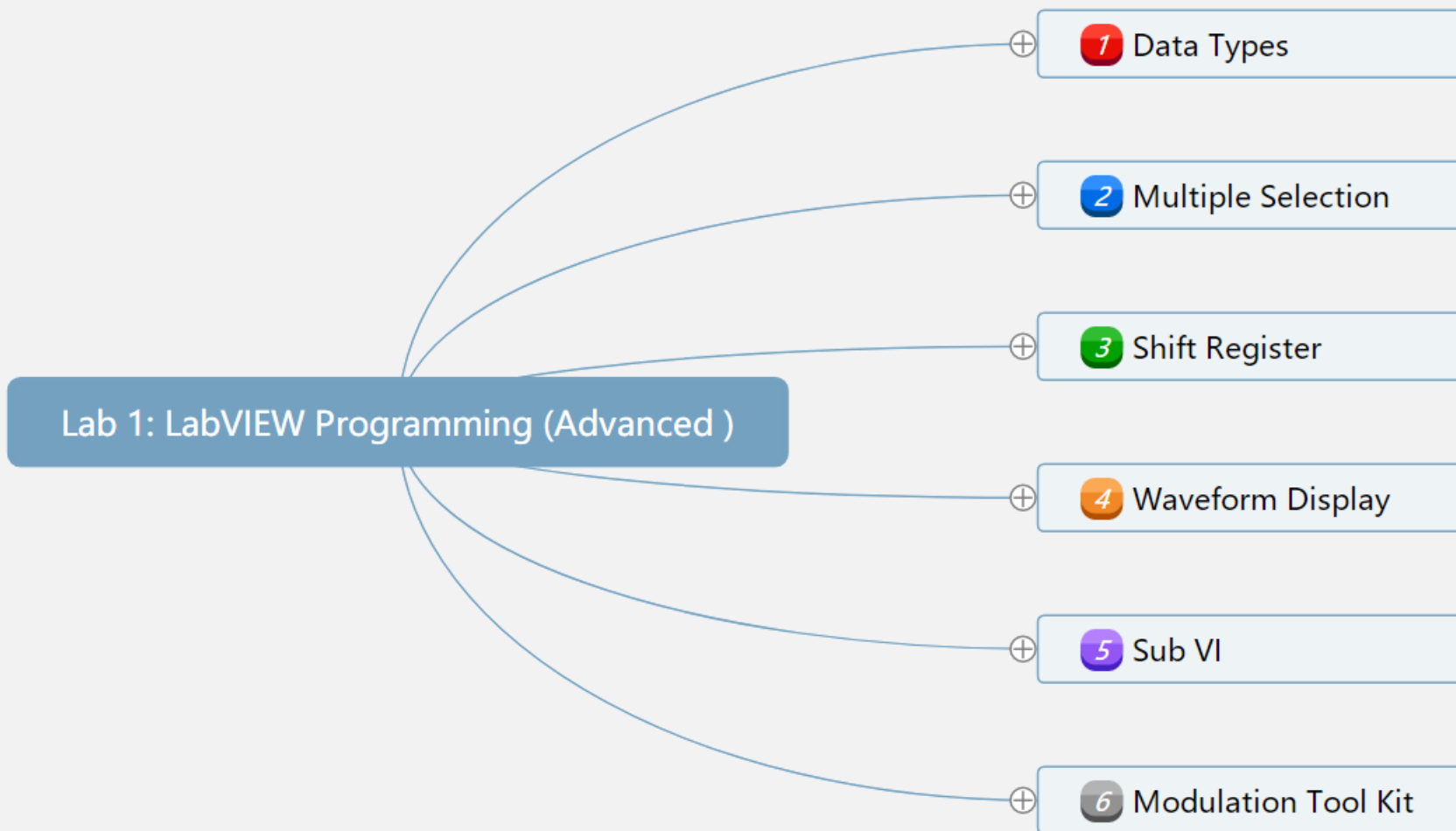








# I have an ability to understand:

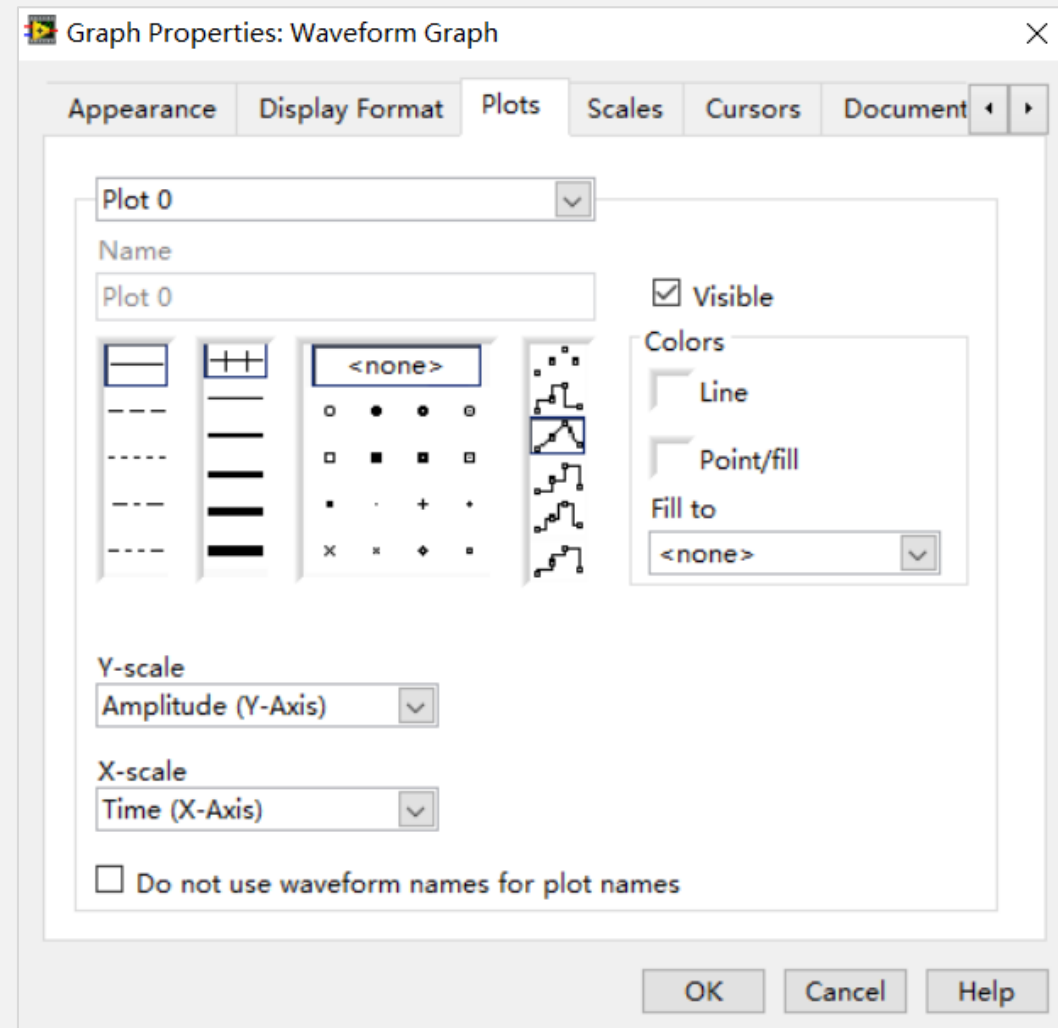
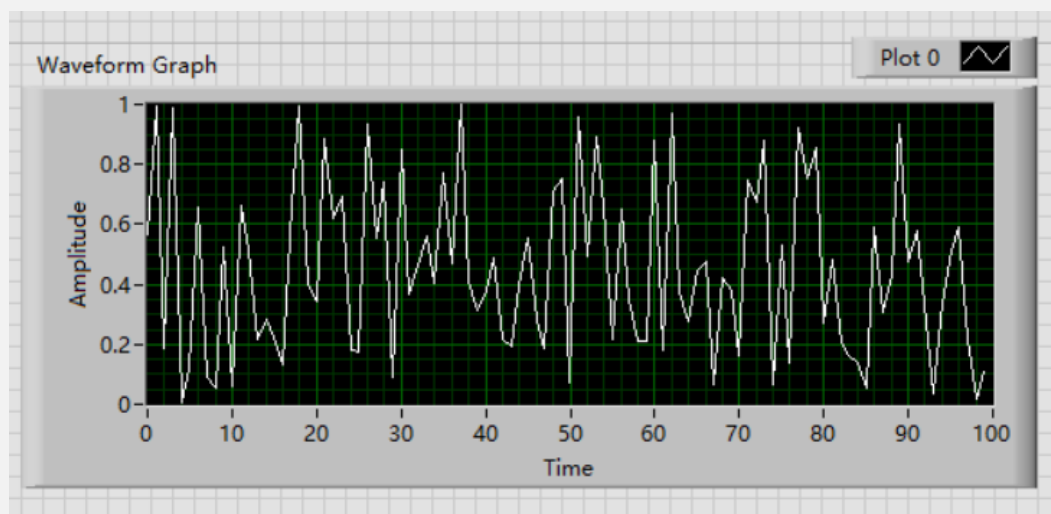
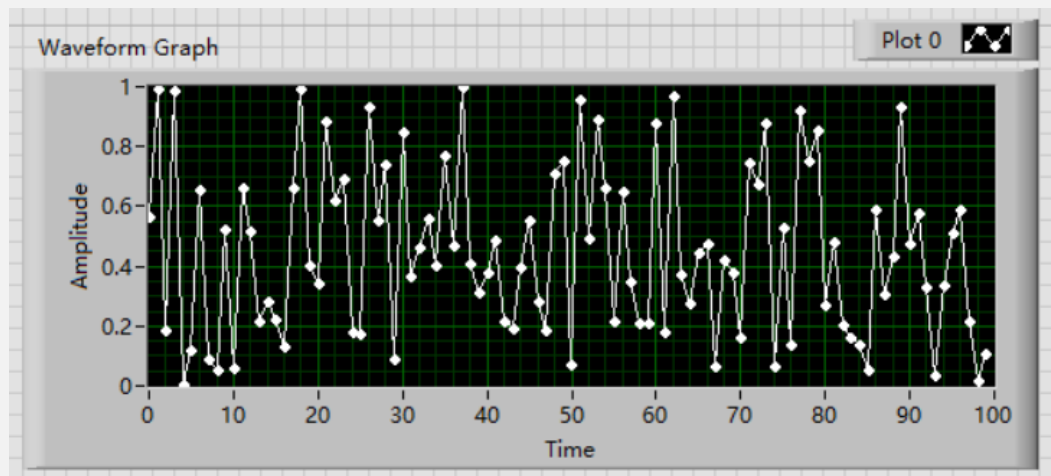




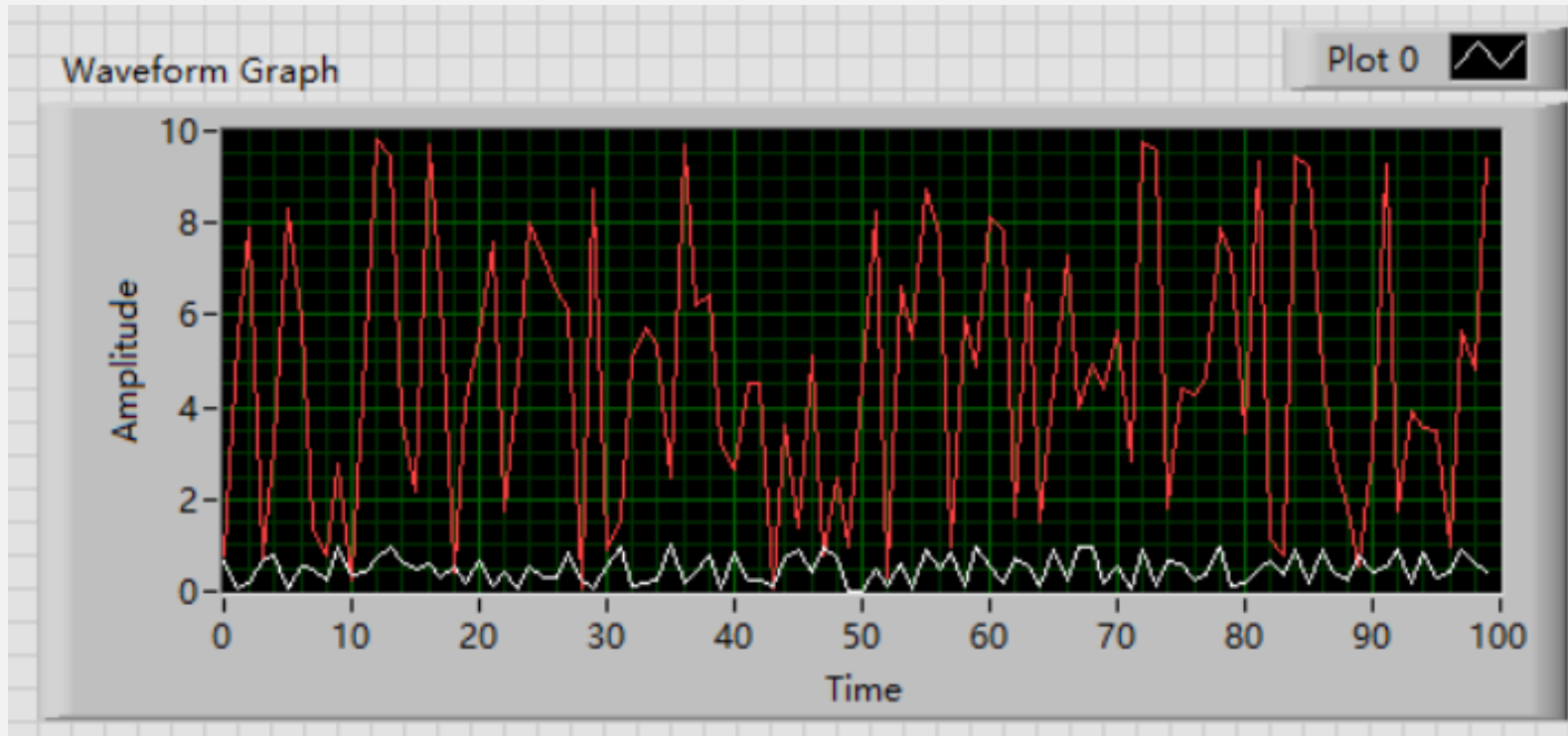
# Demo: Waveform Display



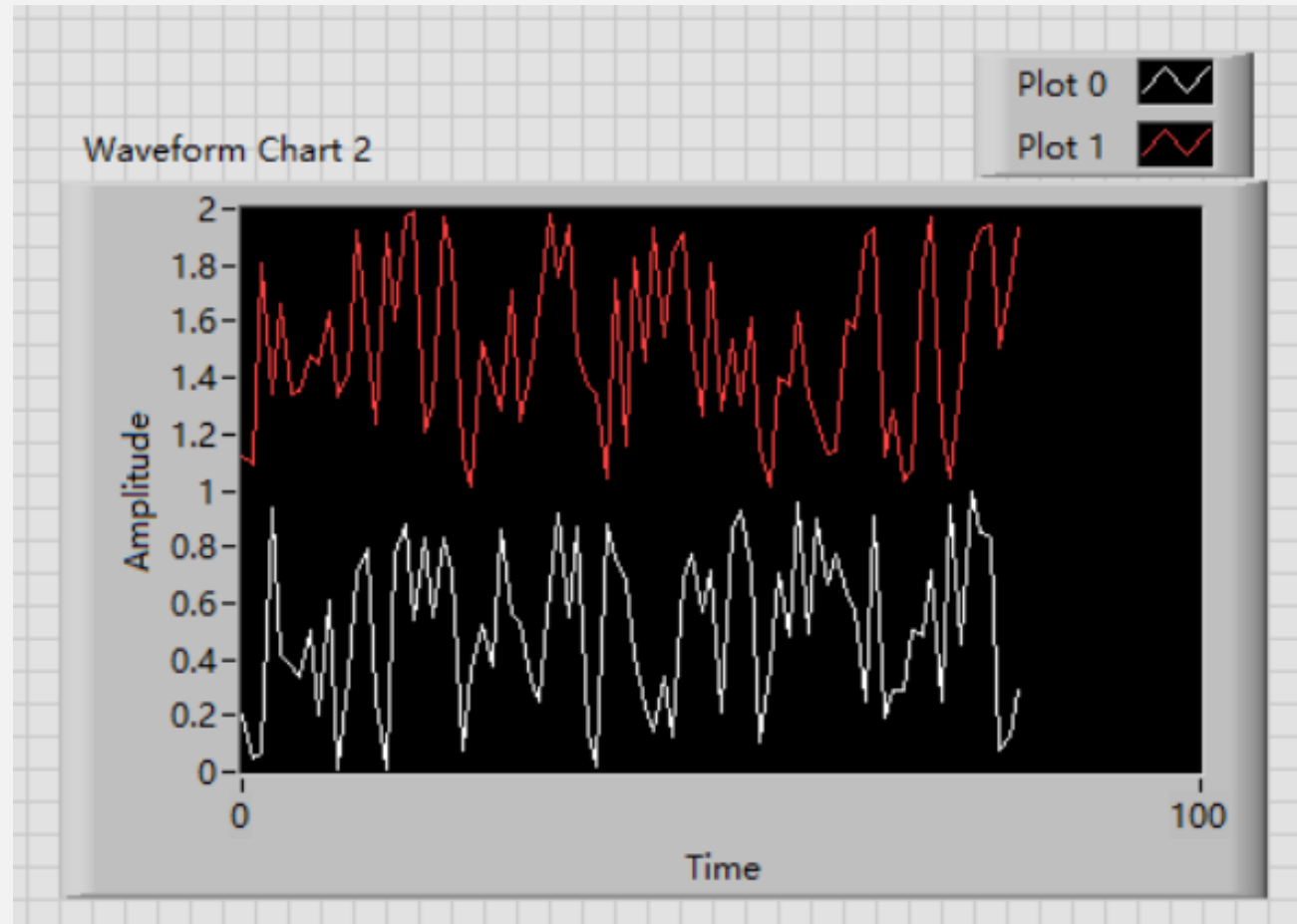
# Demo: Random Noise Display



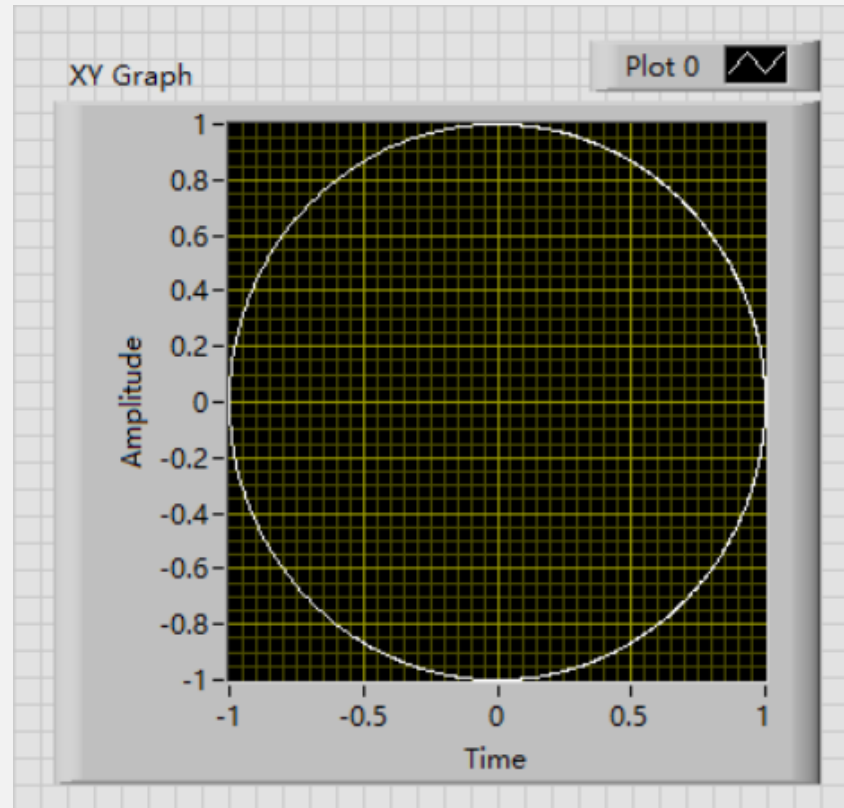
# Demo: Multi-Curve Display



# Demo Real Time Display

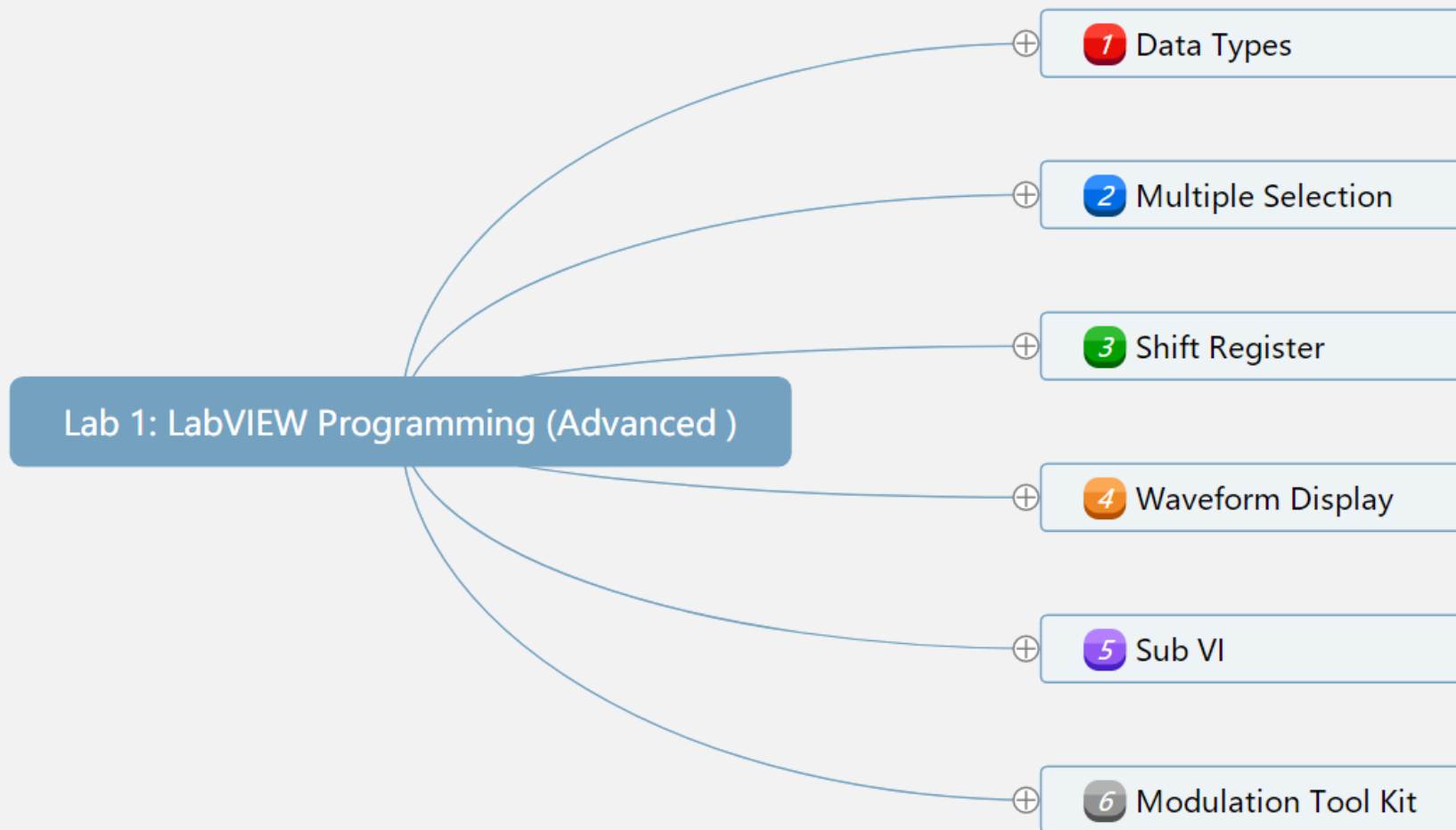


# Demo: XY Graph

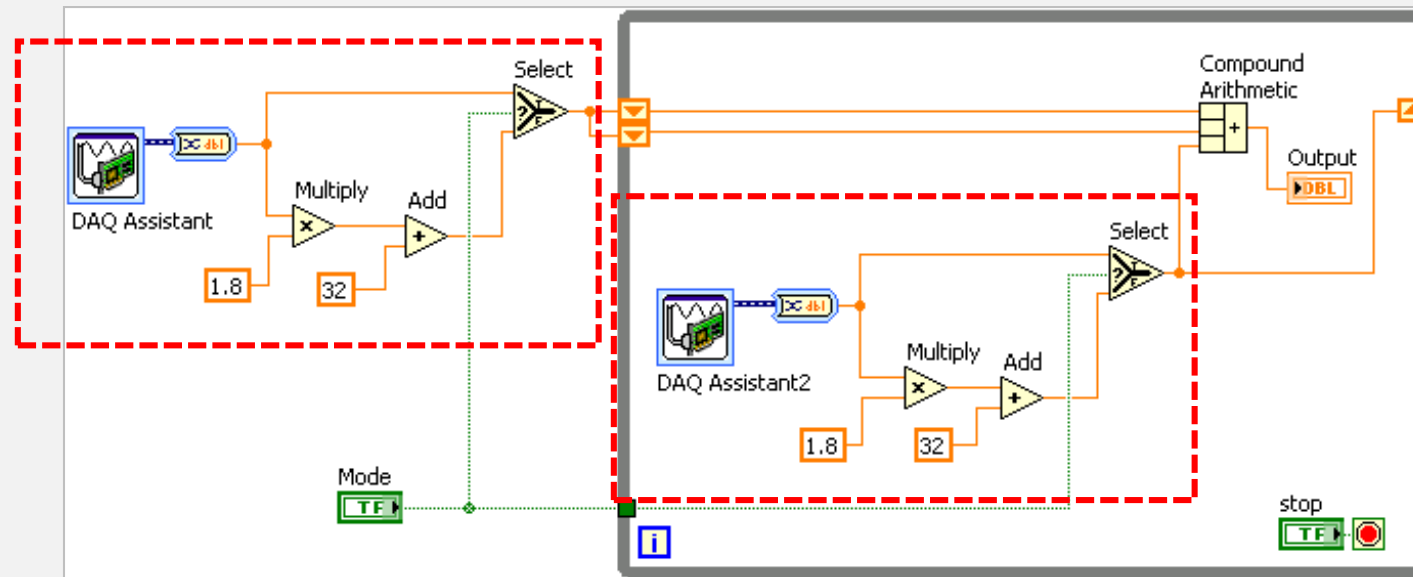




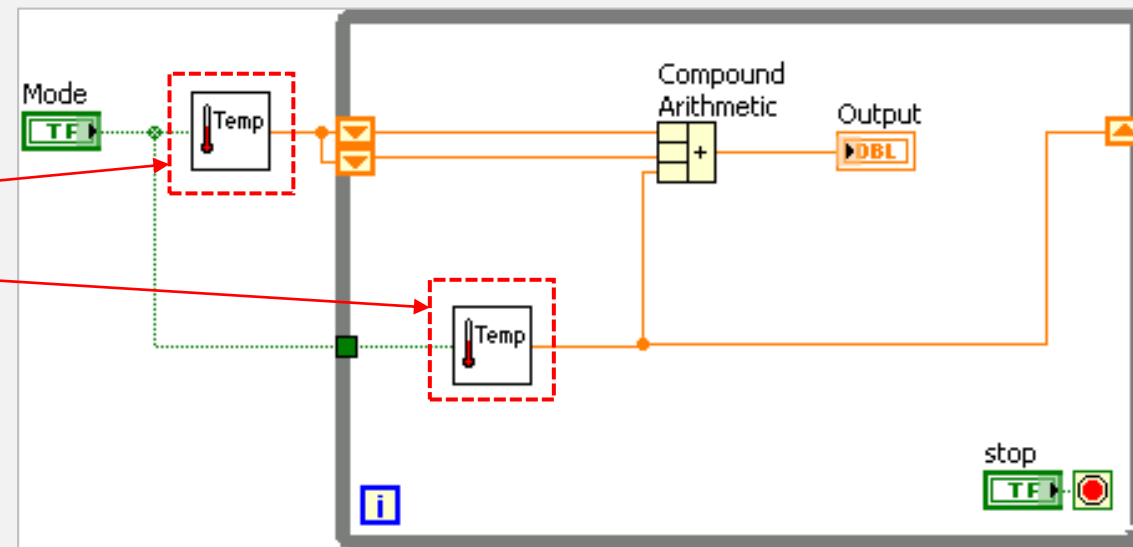
# I have an ability to understand:



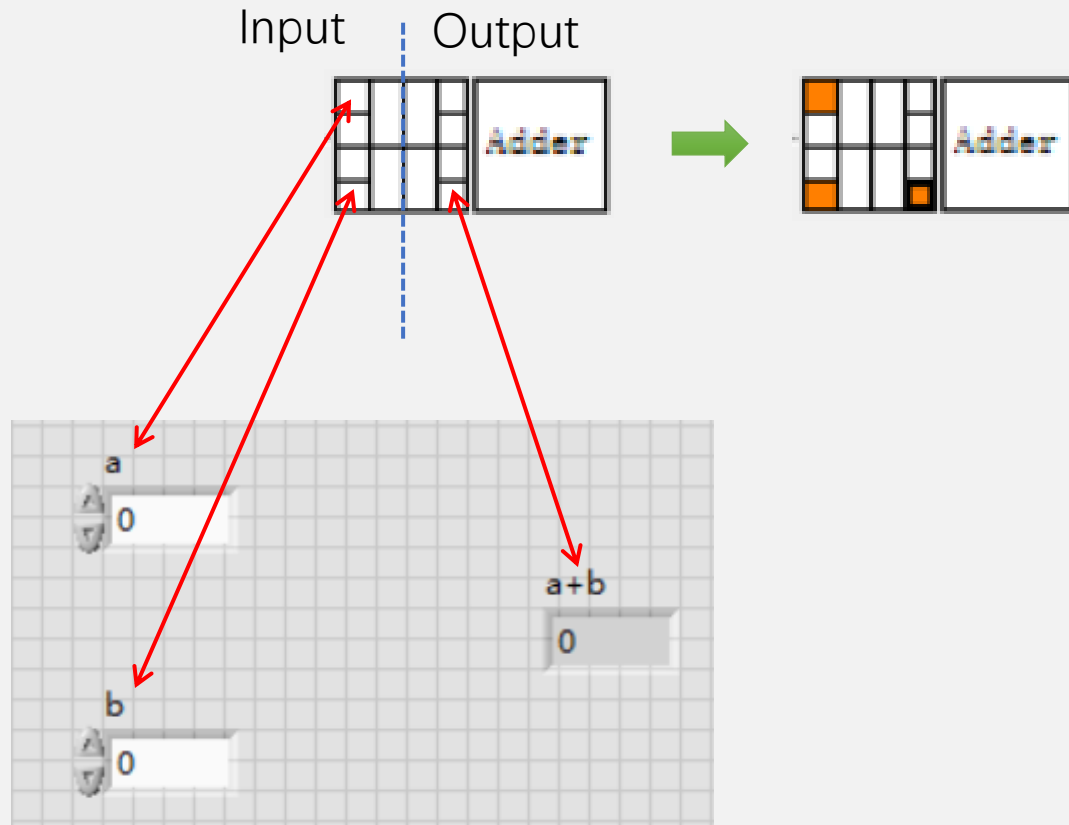




Sub VI

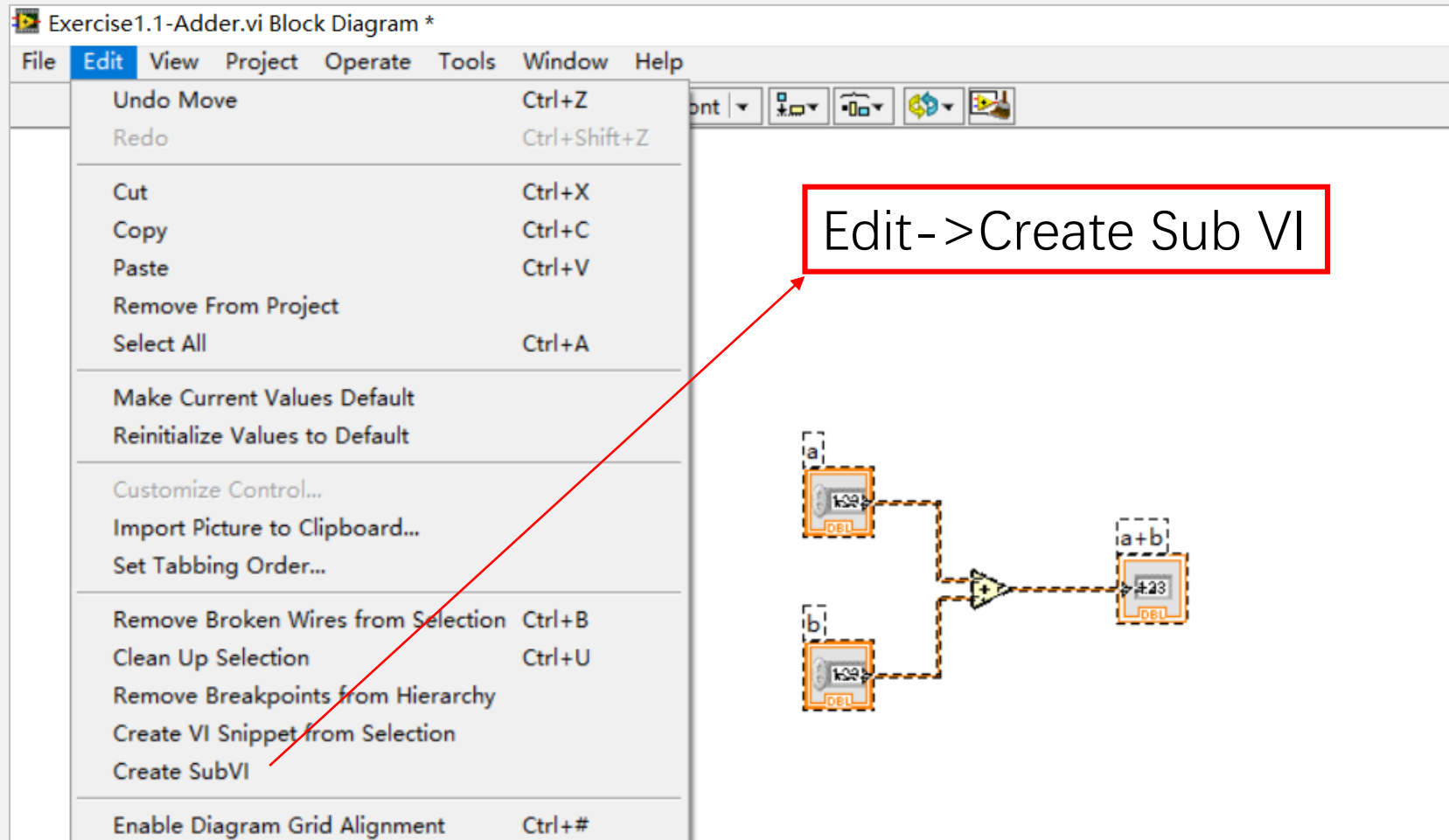


# Create & Call Sub VI



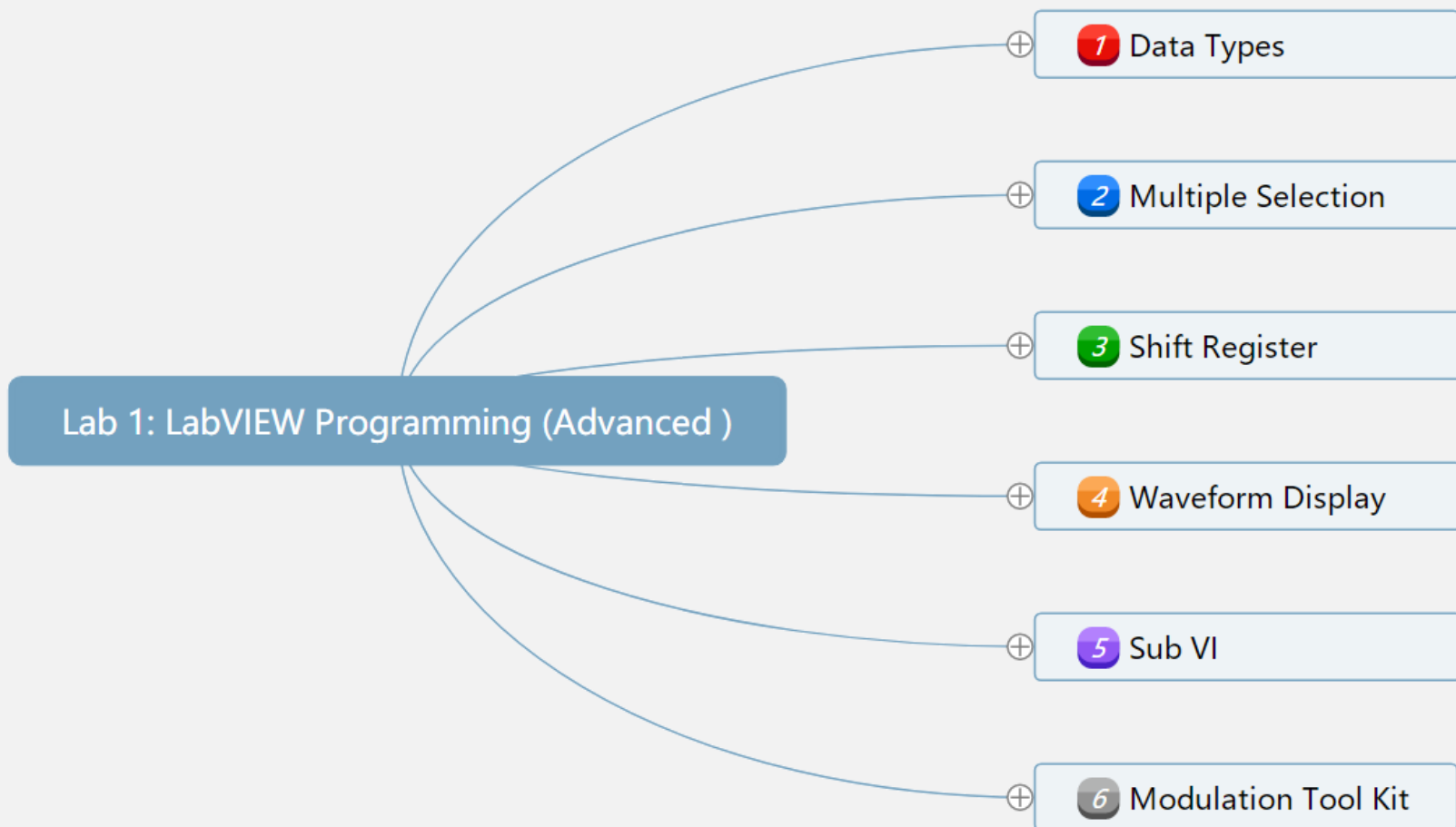
- Measurement I/O ▶
  - Instrument I/O ▶
  - Mathematics ▶
  - Signal Processing ▶
  - Data Communication ▶
  - Connectivity ▶
  - Express ▶
  - Addons ▶
  - Select a VI...
  - FPGA Interface ▶
  - RF Communications ▶
- ⏏

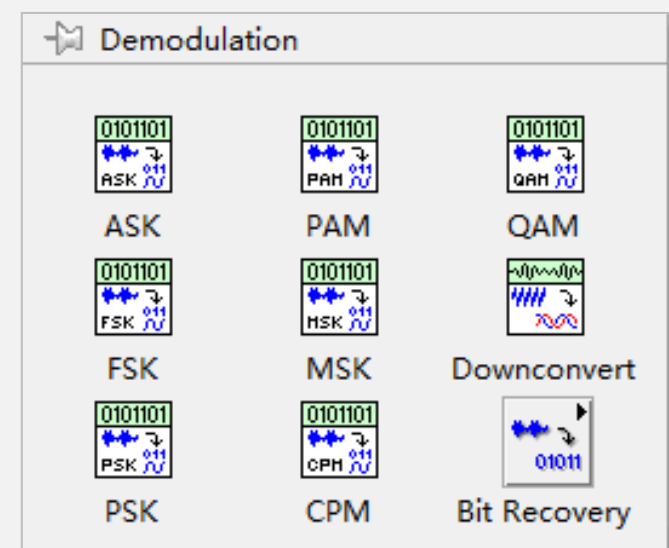
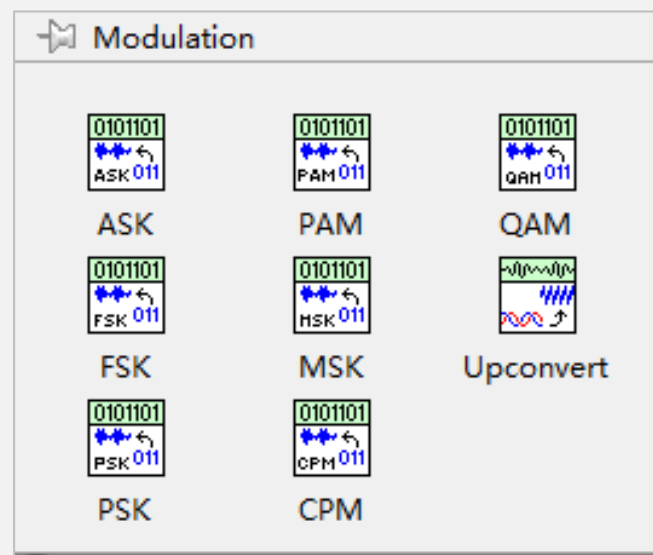
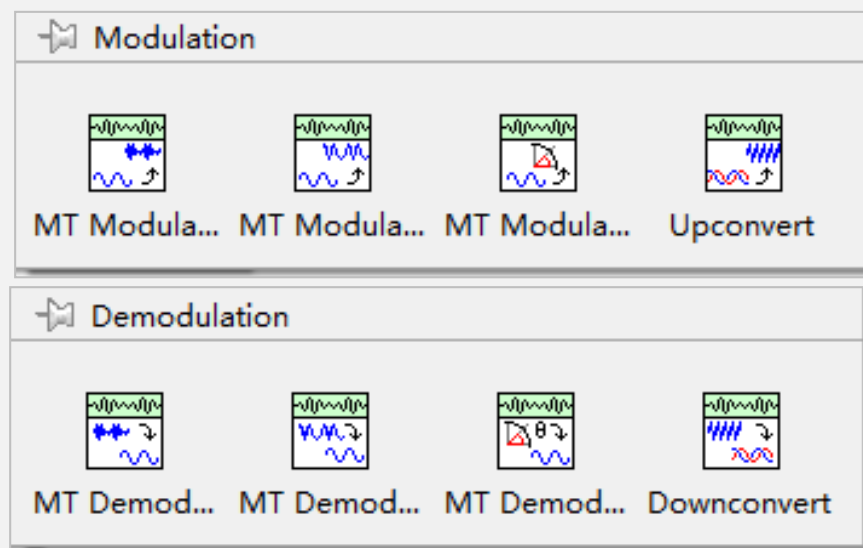
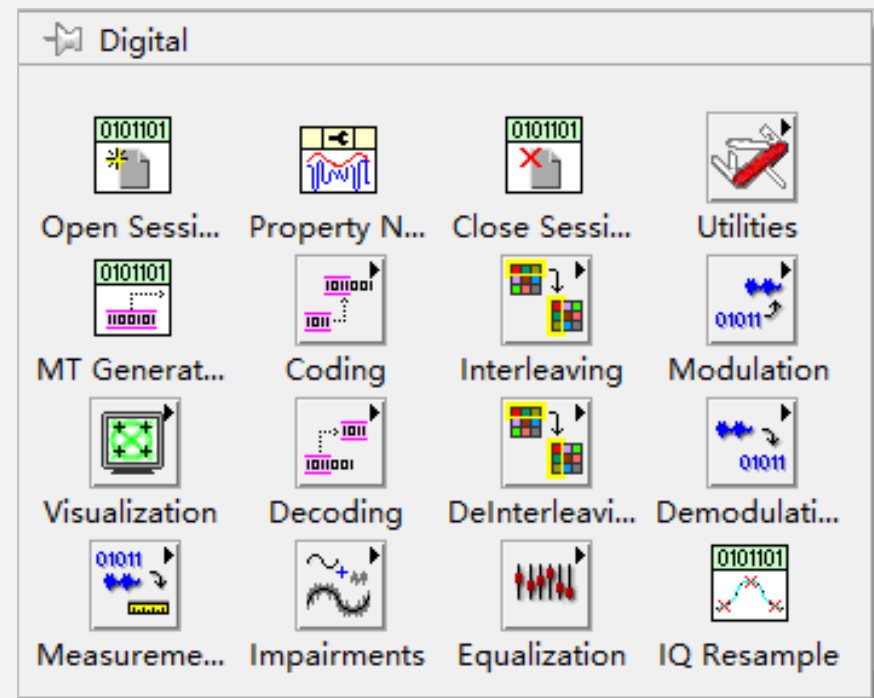
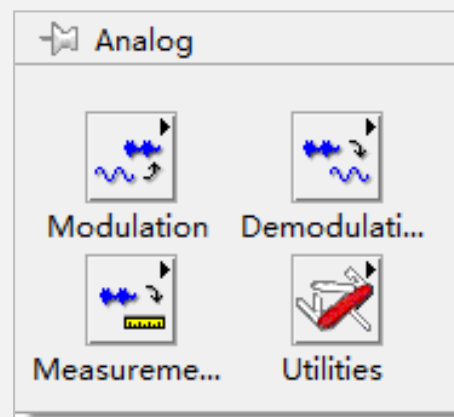
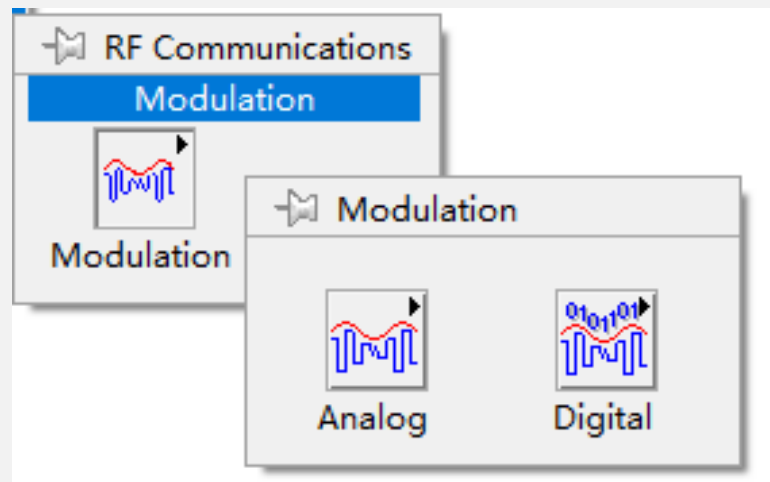
# Create & Call Sub VI



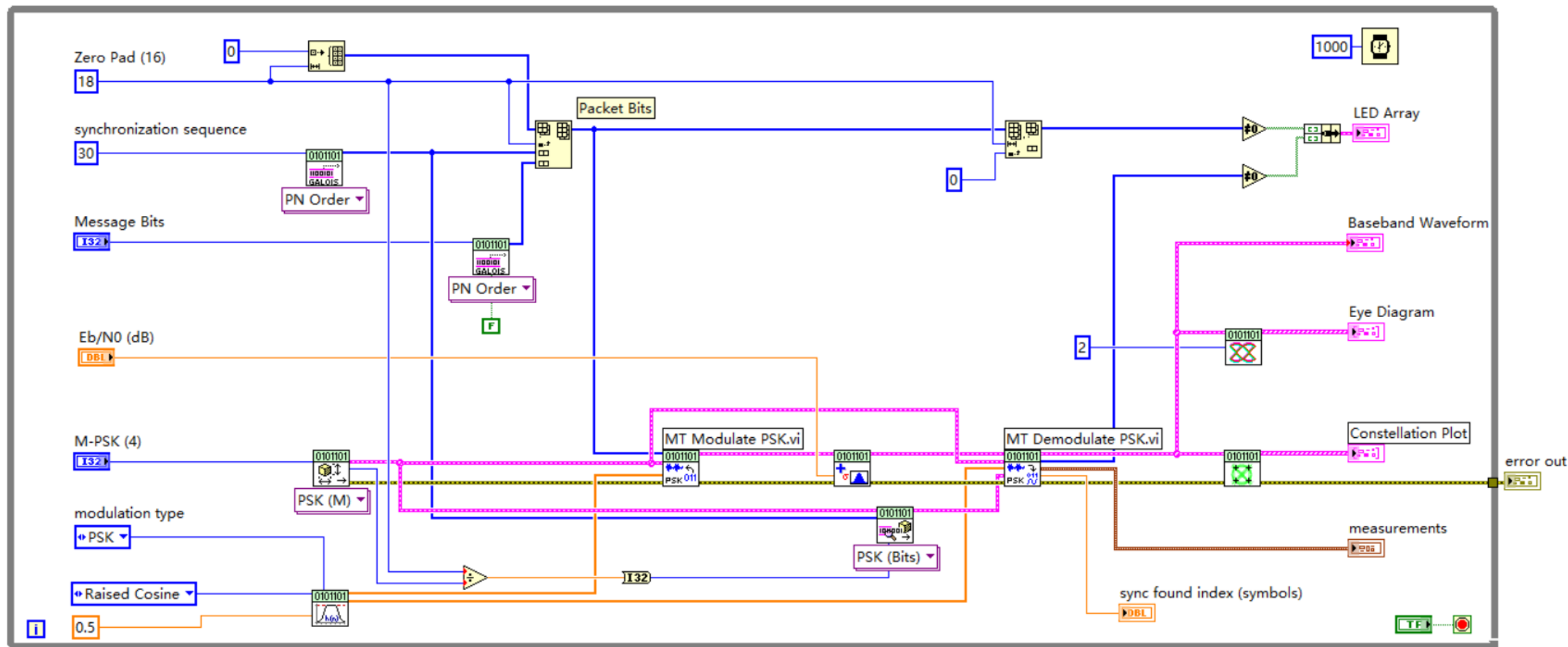


# I have an ability to understand:





Tab Control





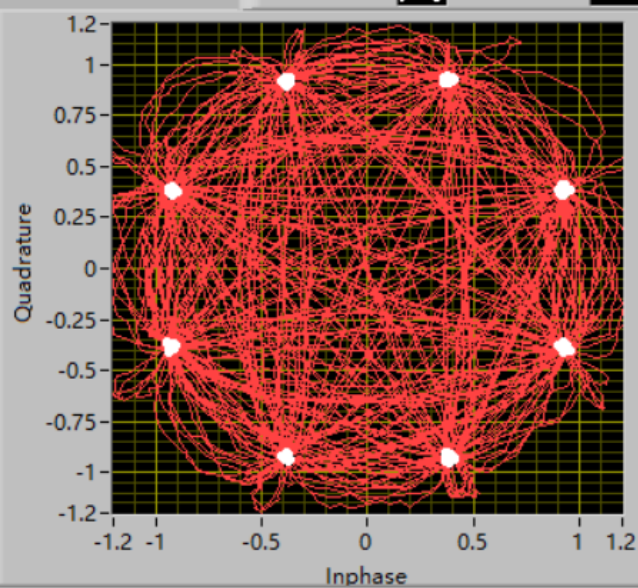
## 8-PSK Simulation

Constellation

Eye Diagram

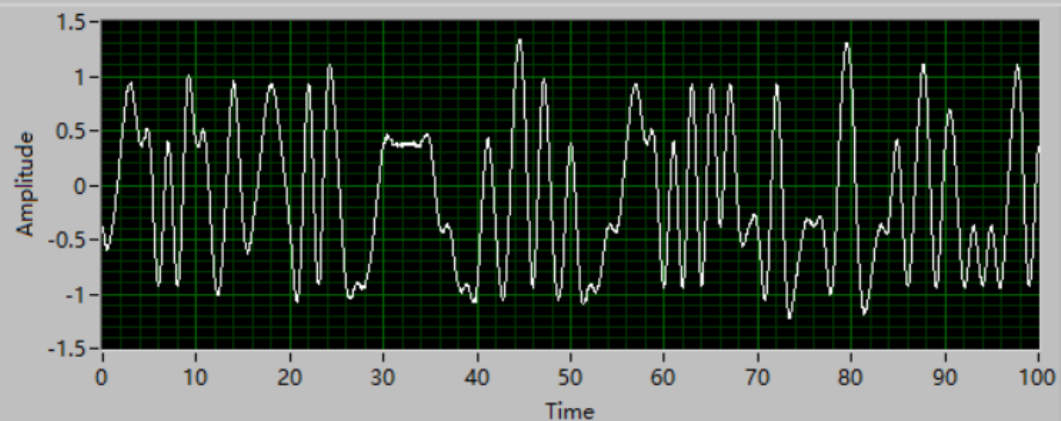
Constellation Plot

Constellation Transitions



Baseband Waveform

Plot 0



M-PSK (4)

Message Bits

8

1000

Eb/N0 (dB)

-10 0 20 40 50

Measurements

frequency offset (Hz)

-0.00

frequency drift (Hz)

-0.00

phase offset

0.04

Input Bitstream

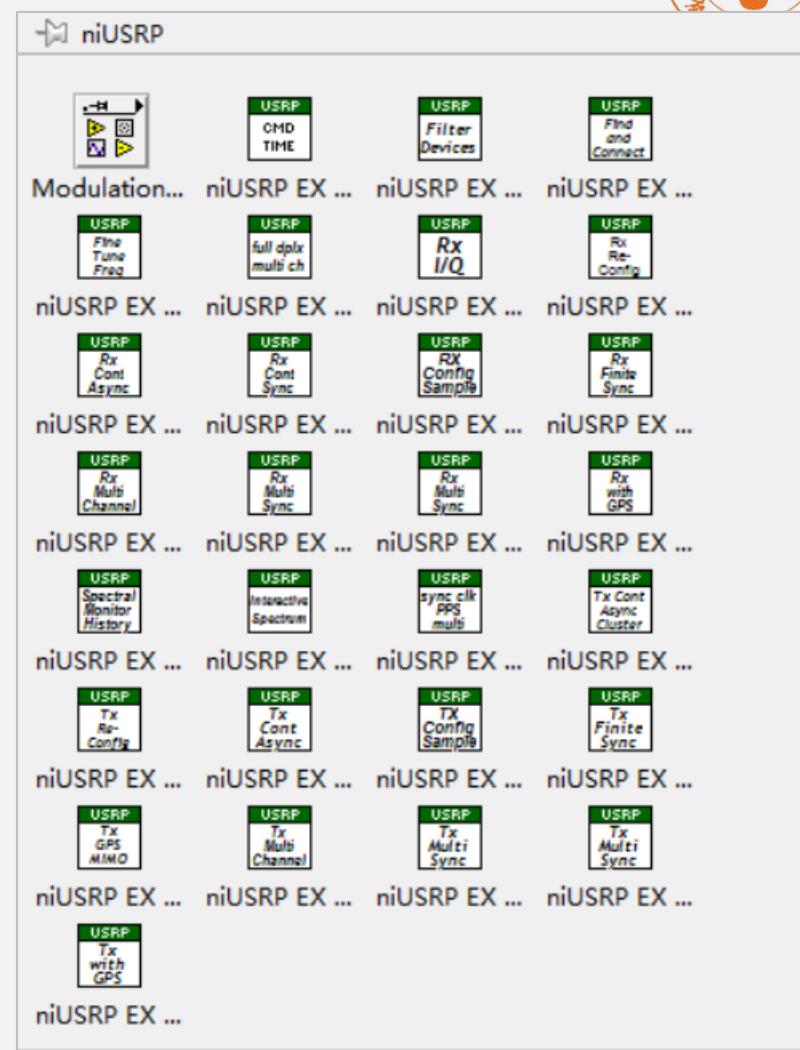
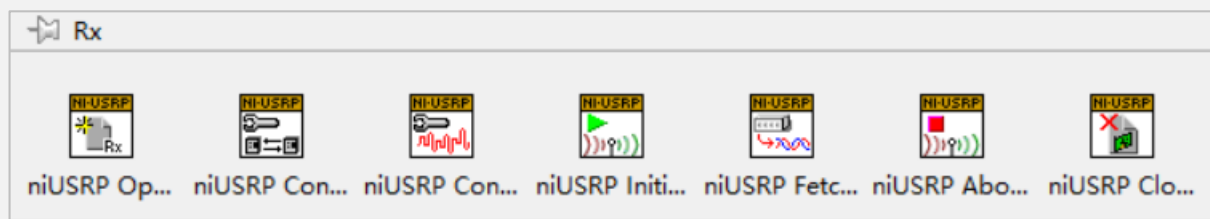
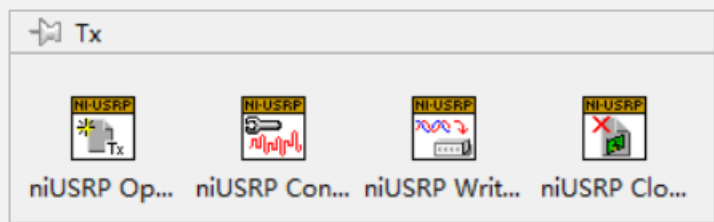
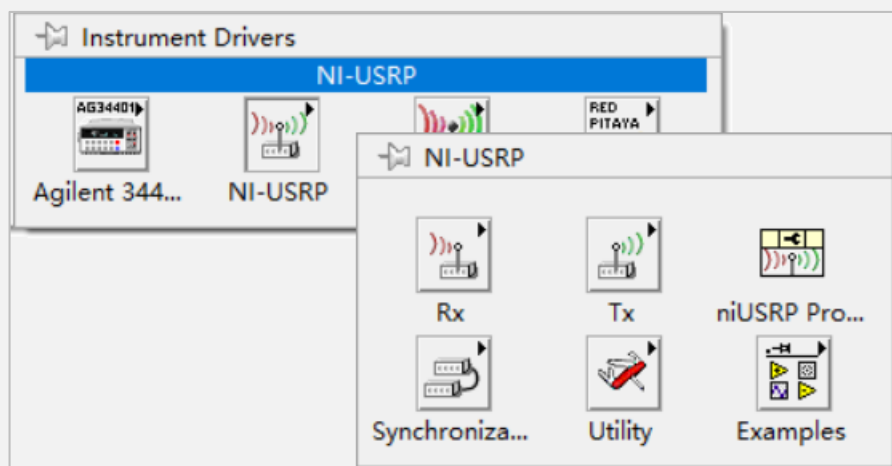
101



Output Bitstream

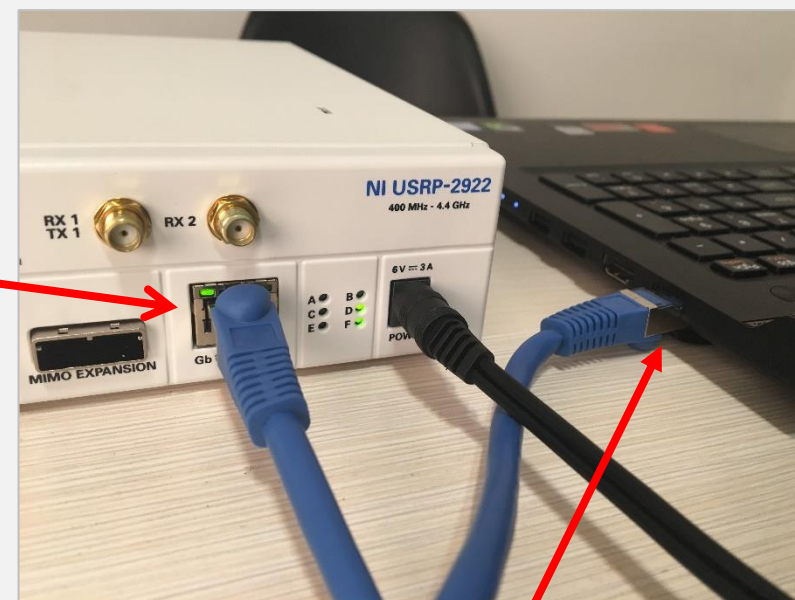
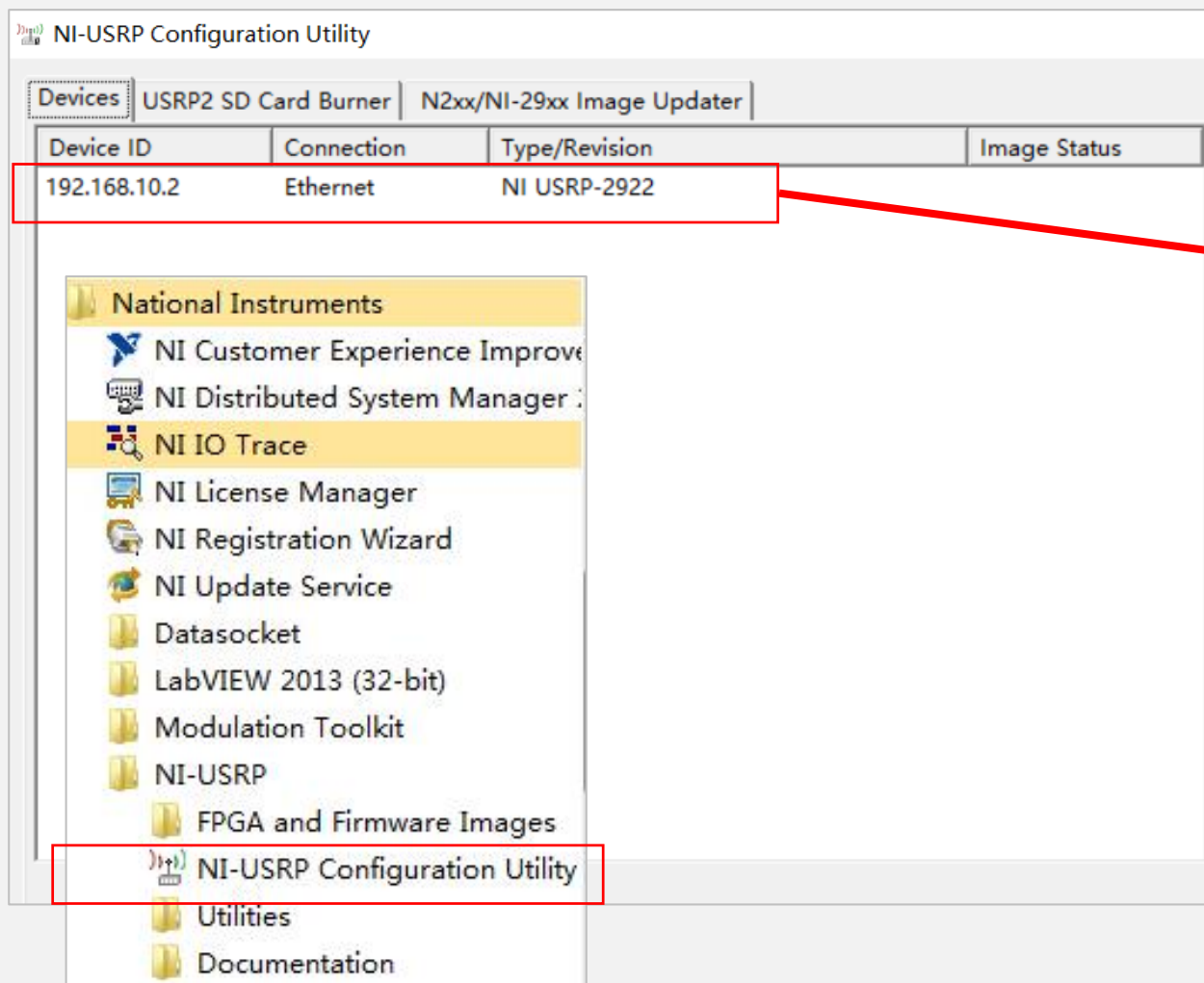
101







# USRP Experiment



Host computer's IP:  
**192.168.10.1**

# Most-used USRP functions



Configure

Read/Write

Close

USRP Transmitter

niUSRP Open Rx Session.vi



niUSRP Configure Signal.vi



niUSRP Initiate.vi



niUSRP Fetch Rx Data (poly).vi



CDB Cluster ▼

niUSRP Abort.vi



niUSRP Close Session.vi



USRP Receiver

niUSRP Open Tx Session.vi



niUSRP Configure Signal.vi



niUSRP Write Tx Data (poly).vi

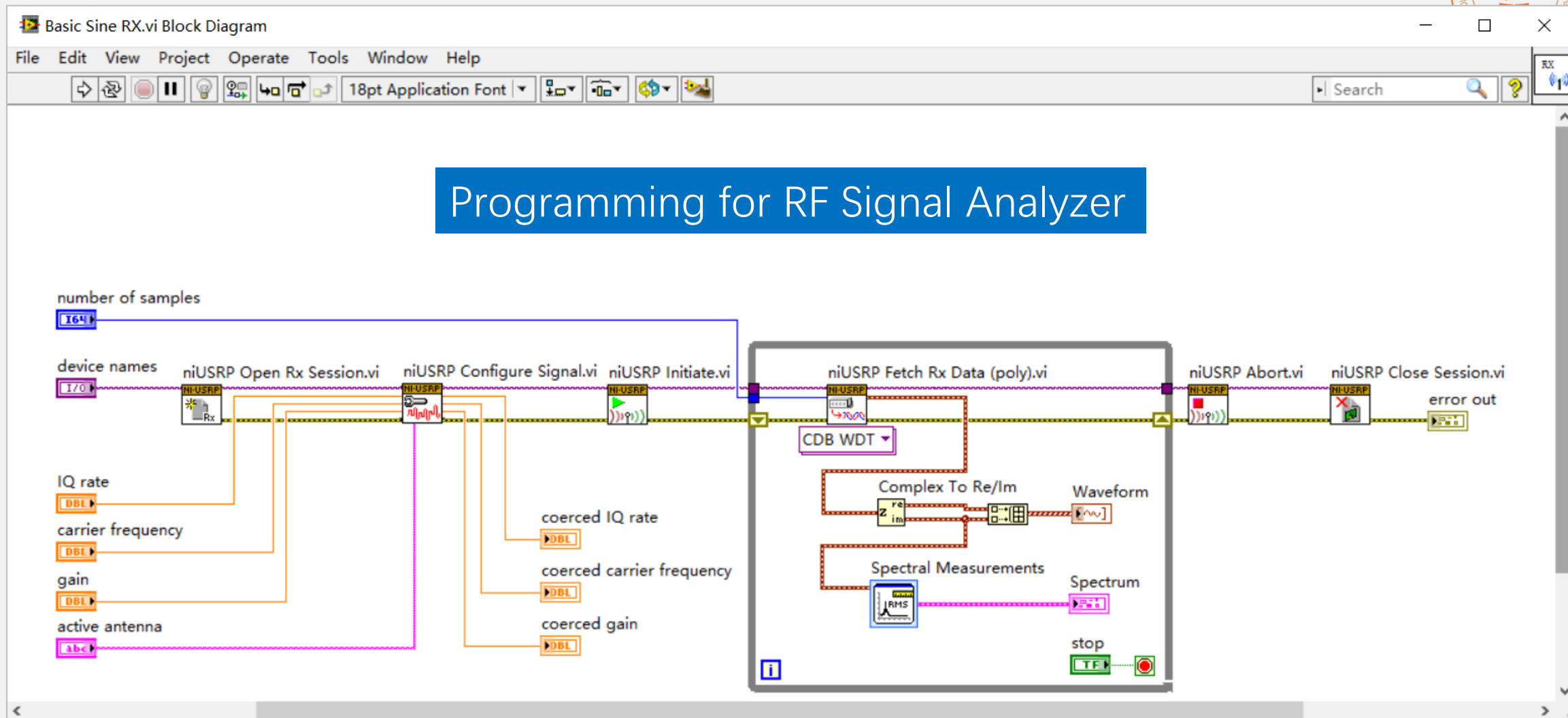


CDB Cluster ▼

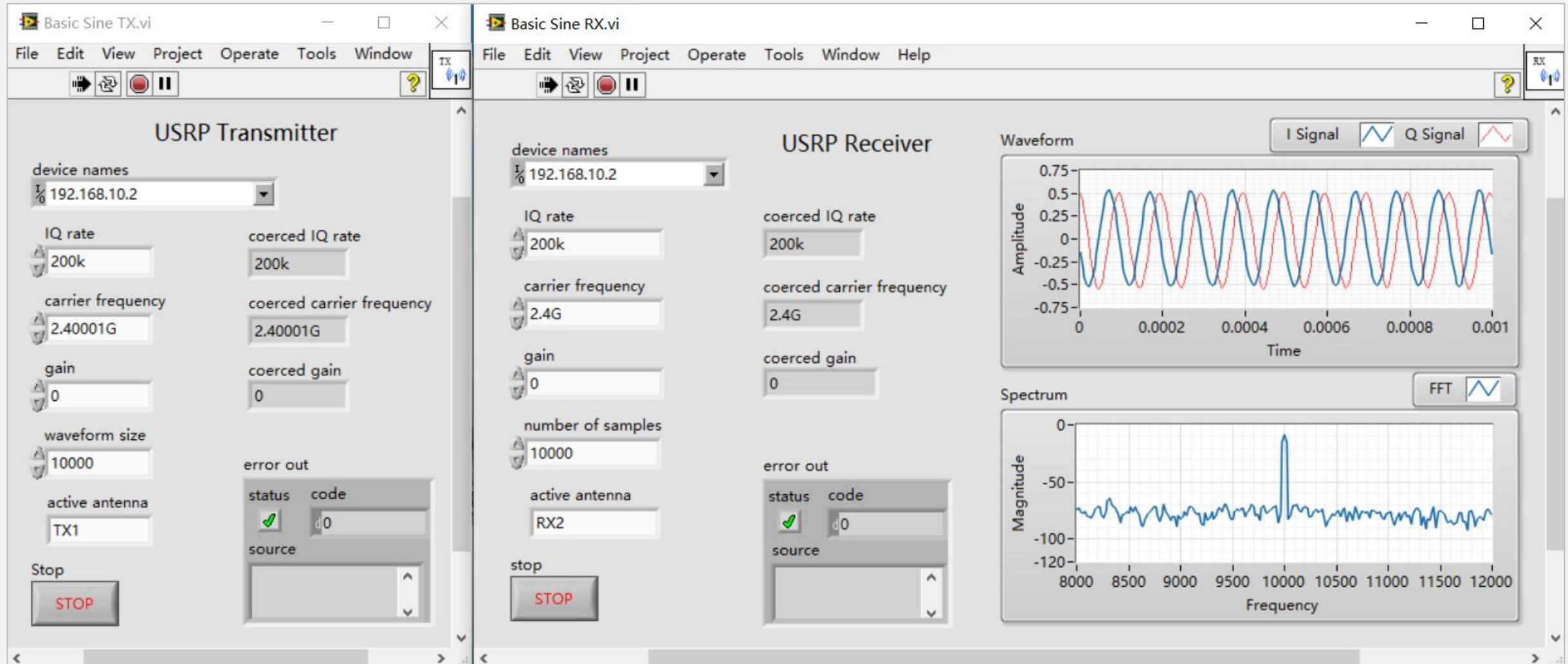
niUSRP Close Session.vi



# Programming for RF Signal Analyzer

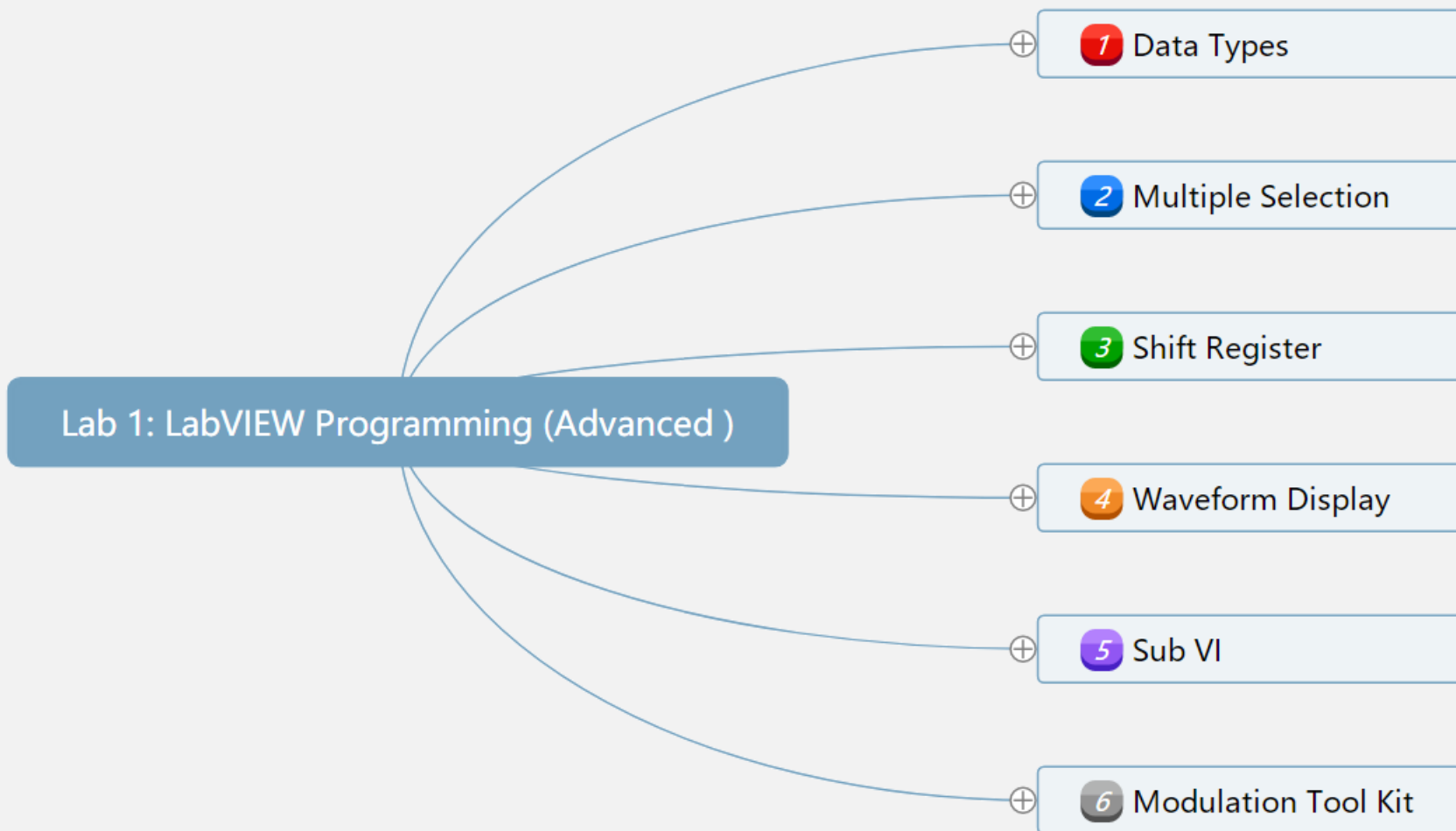


# RF Signal Analyzer





# I have an ability to understand:





- Question ?





【通信新说】



腾讯课堂