Experiment 10:

Symbol Synchronization

포항공과대학교 전자전기공학과

Communication and Intelligent Systems Lab. (CISL)

Outline

1. Review
2. Symbol Synchronization
3. Today’s Experiment
4. About Report

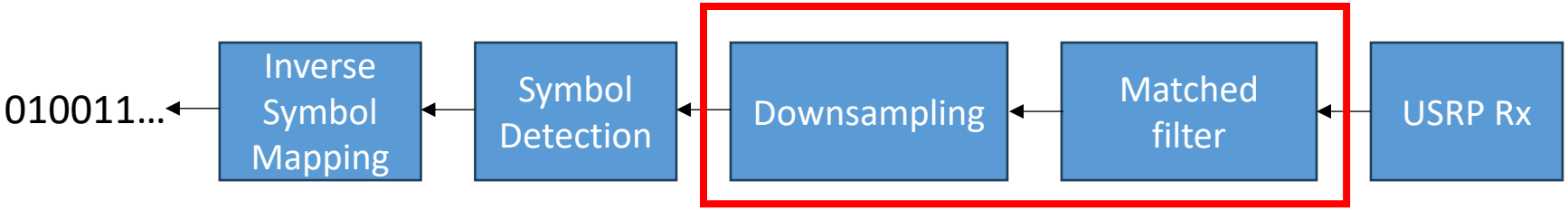
Outline

1. Review
2. Symbol Synchronization
3. Today's Experiment
4. About Report

Review

- What we did in last time.

< Transmitter >



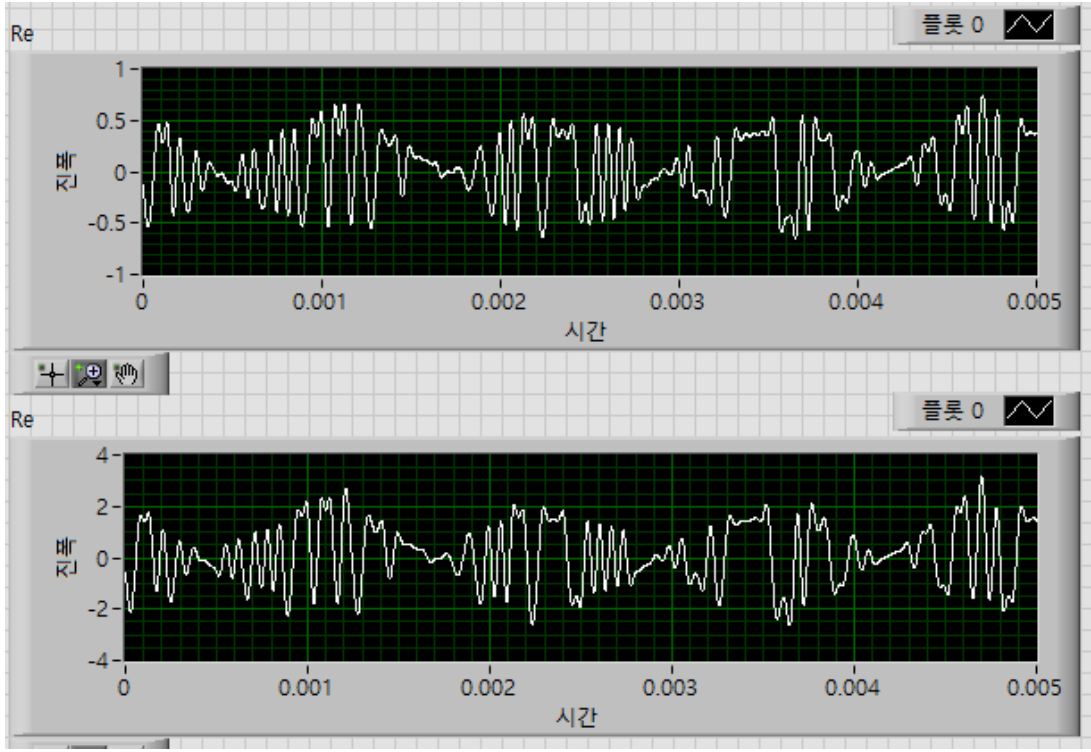
< Receiver >

Review

- Matched filtering 전/후 (time domain)

Matched Filtering 전

Matched Filtering 후

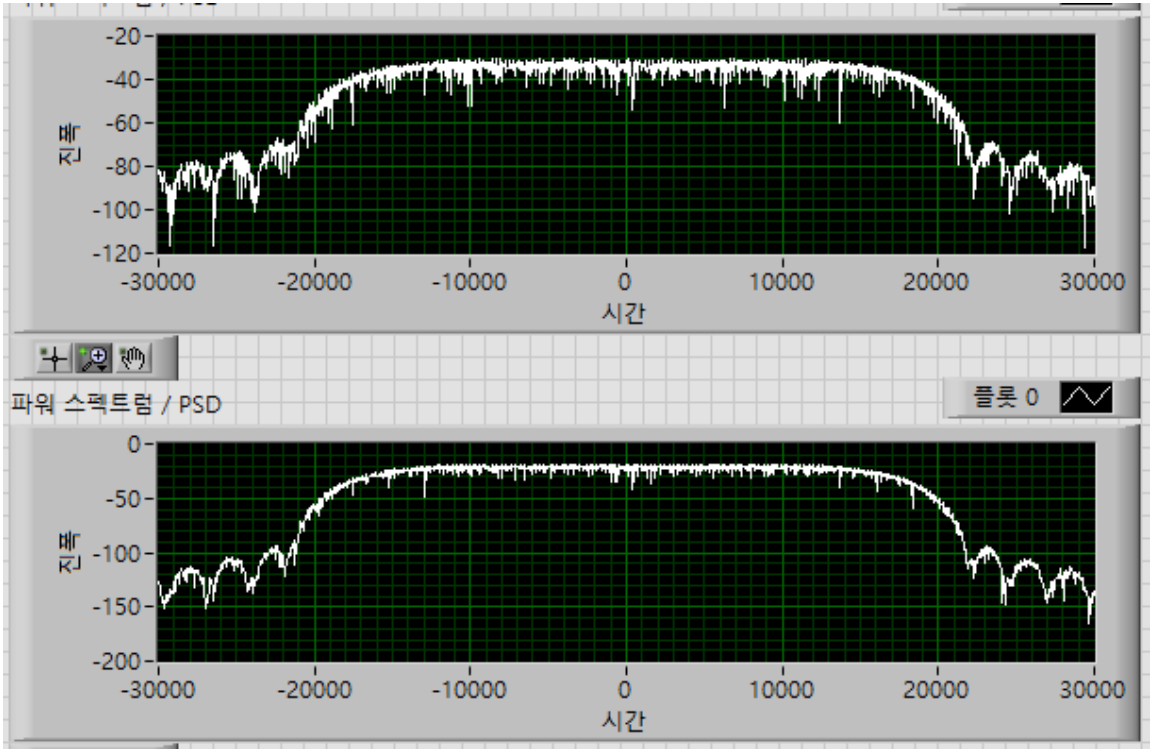


Review

- Matched filtering 전/후 (frequency domain)

Matched Filtering 전

Matched Filtering 후

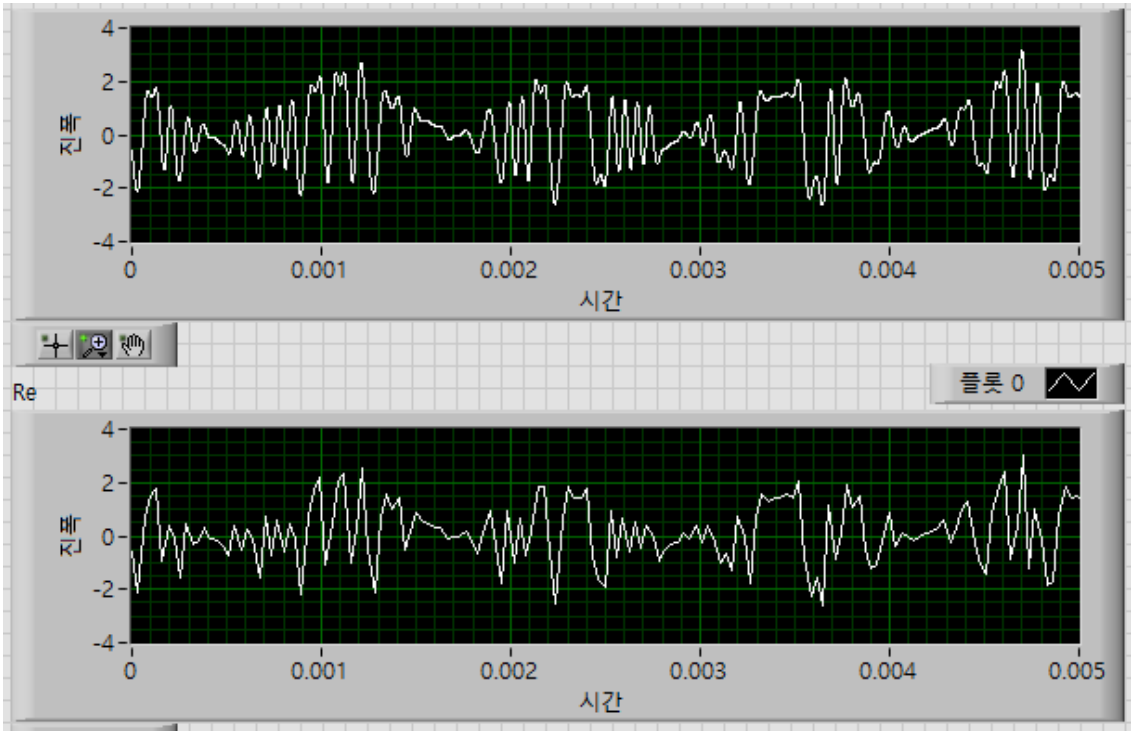


Review

- Downsampling 전/후 (time domain)

Downsampling 전

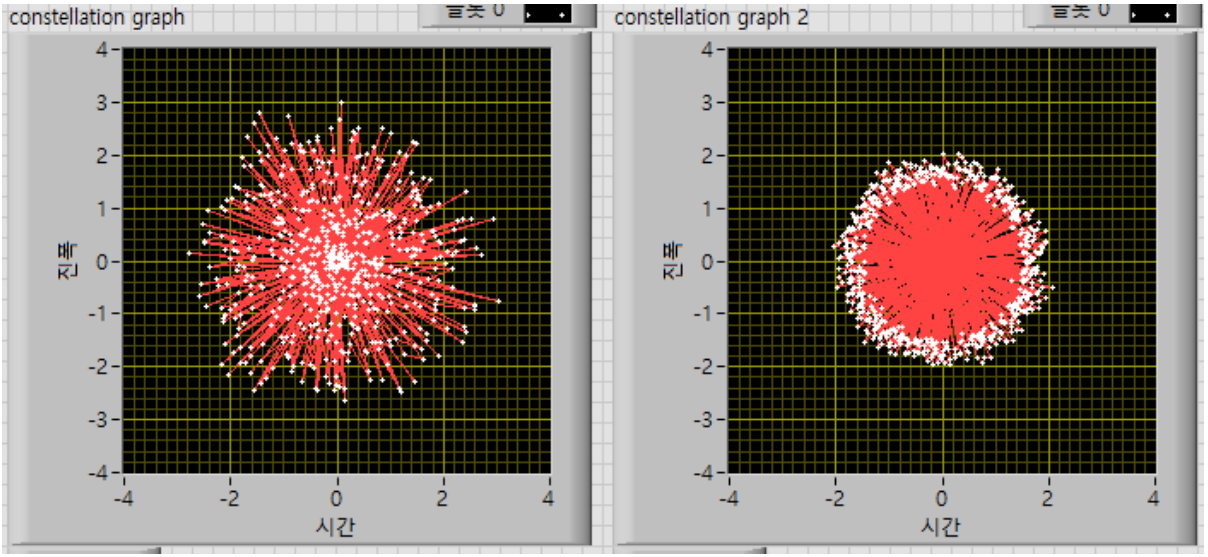
Downsampling 후



Review

- Downsampling 후 (complex plane)

운이 좋다면…



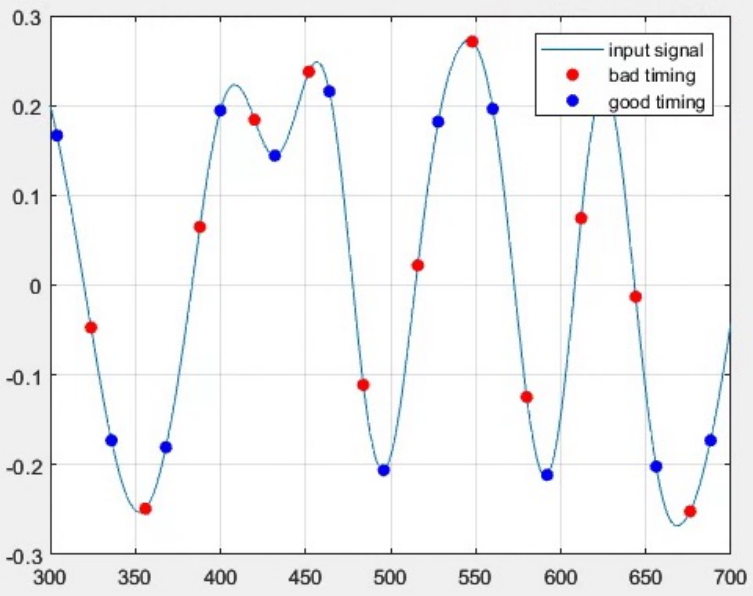
- 이 상태에서는 Symbol Detection 과정이 이루어지기 어렵다.
  - Symbol timing, frequency offset에 의한 phase rotation, etc…

Outline

1. Review
2. Symbol Synchronization
3. Today's Experiment
4. About Report

Symbol Synchronization

- Symbol synchronization
  - ➢ 수신단에서의 delay 발생으로 인한 timing error
  - ➢ Incorrect point에서의 downsampling 시 inter-symbol interference (ISI) 발생
    → 통신 성능 저하



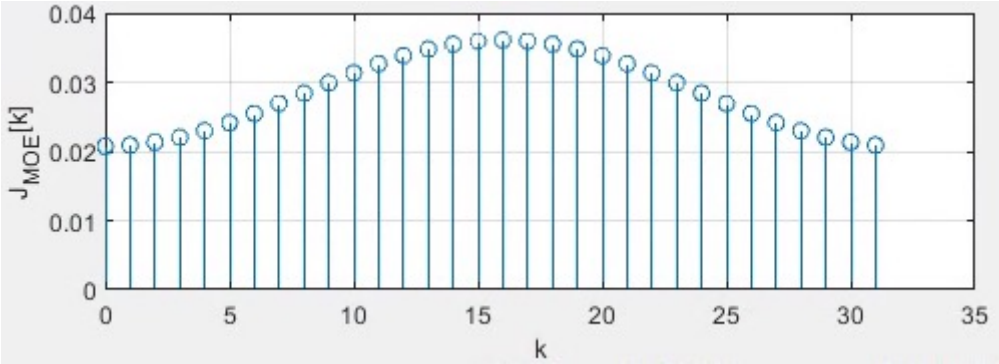
  - ➢ Downsampling 전에 ISI 최소화를 위한 offset를 찾을 필요가 있음.

Symbol Synchronization

- Symbol synchronization : Maximum Output Energy (MOE) Criterion
  - Received signal : $\tilde{y}[n]$
  - Samples per symbol : M_{rx}
  - Sample offset이 k일 때 ($k = 0, 1, \cdots, M_{rx}$)
    output energy after downsampling (factor : M_{rx})

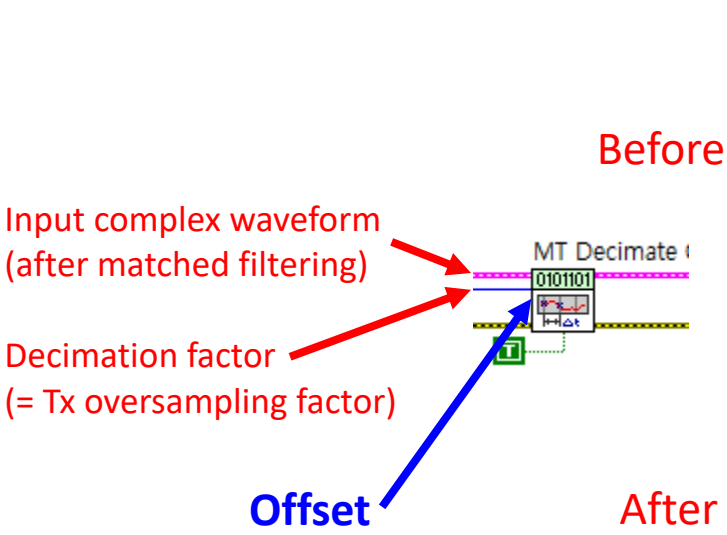
$$J_{\mathrm{MOE,d}}[k] = \mathbb{E}\left[\left|\tilde{y}[nM_{\mathrm{rx}} + k]\right|^2\right]$$

  - 모든 k에 대해 $J_{MOE,d}[k]$를 계산 후,
    $J_{MOE,d}[k]$ 가 가장 클 때의 k를 offset으로 하여 downsampling 수행.



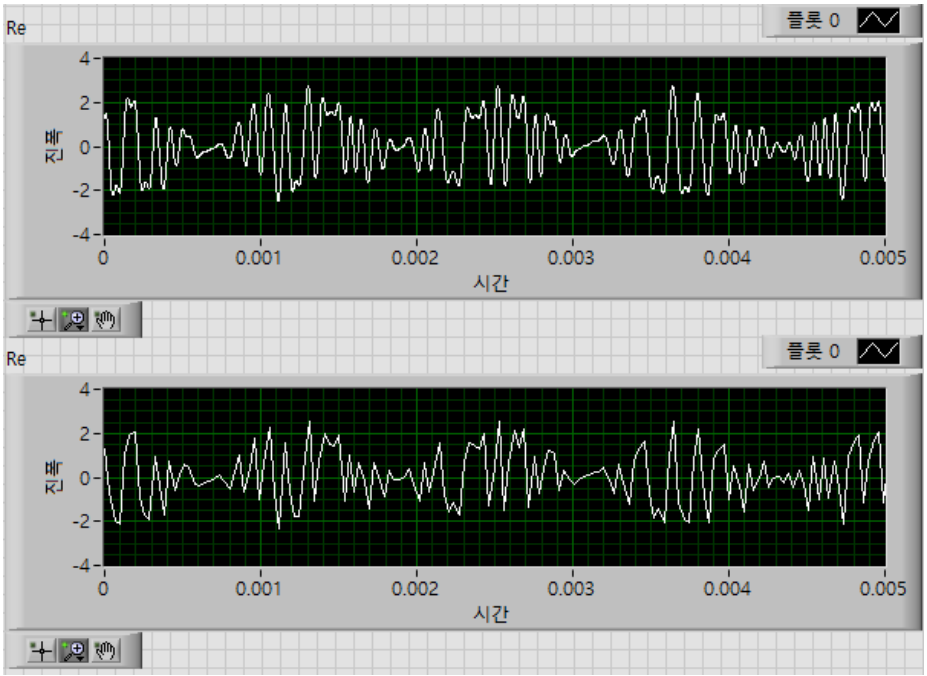
Symbol Synchronization

- Decimation : downsampling
  - Tx에서 M배 oversampling한 신호를 전송하면
  - Rx에서는 M배 downsampling하여 symbol detection 수행.
  - Modulation → Digital → Utilities → Decimate Cmplx Wfm



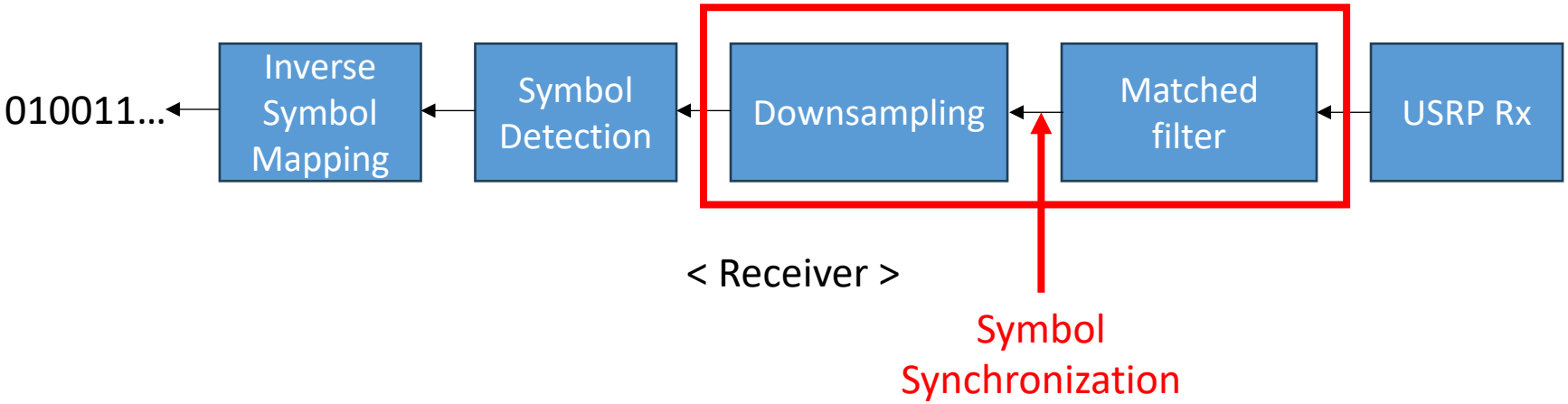
Before

After



Symbol Synchronization

- Symbol synchronization은
  Matched filtering과 Downsampling 사이에서 수행.



Outline

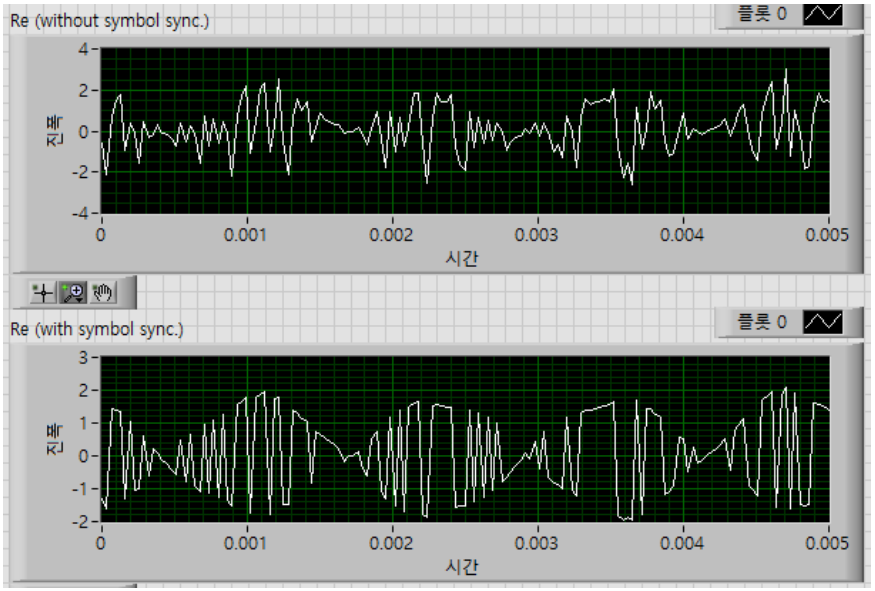
1. Review
2. Symbol Synchronization
3. Today’s Experiment
4. About Report

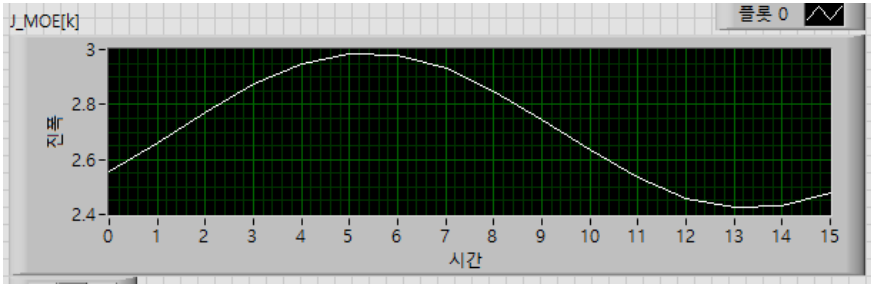
Today's Experiment

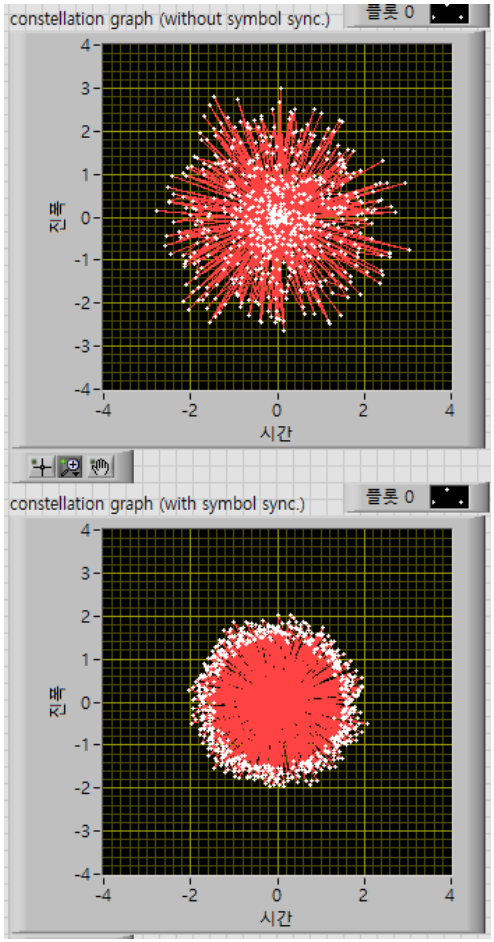
- MOE Criterion 코드 작성하여 symbol synchronization 수행.
  - 송신 신호 정보
    - IQ rate : 500,000 samples/s [500 kS/s]
    - Filter : Root Raised Cosine
    - Roll-off factor : 0.35
    - Oversampling factor : 16
    - Center frequency : 2.3765 GHz
  - Symbol sync. 수행 후 downsampling 결과를 그래프로 출력
    - Symbol sync. 수행하지 않고 downsampling한 결과 [1]: time domain(Re, Im), constellation
    - Symbol sync. 수행 후 downsampling한 결과 [2]: time domain(Re, Im), constellation
    - $J_{MOE,d}[k]$ 그래프[3]
  - IQ rate를 1,000,000 samples/s, samples per symbol을 32로 하여 Symbol sync.
    - Symbol sync. 수행하지 않고 downsampling한 결과[4] : time domain(Re, Im), constellation
    - Symbol sync. 수행 후 downsampling한 결과[5] : time domain(Re, Im), constellation
    - $J_{MOE,d}[k]$ 그래프[6]

Today's Experiment

- 결과 예시 (IQ rate : 500,000 samples/s, samples per symbol : 16)

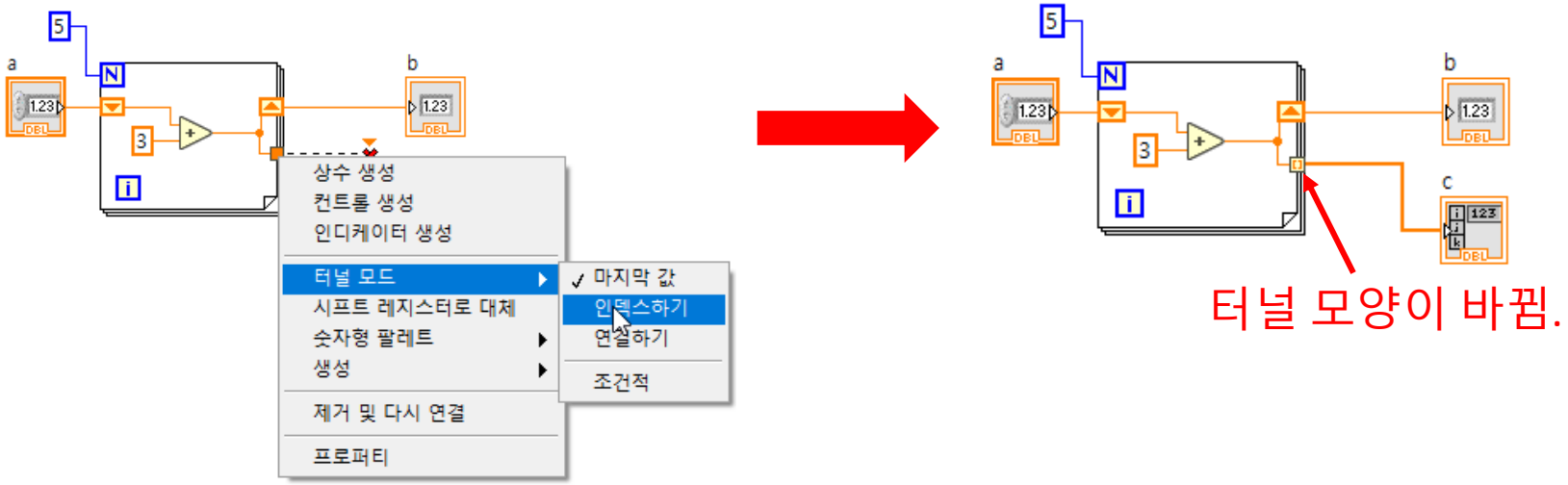






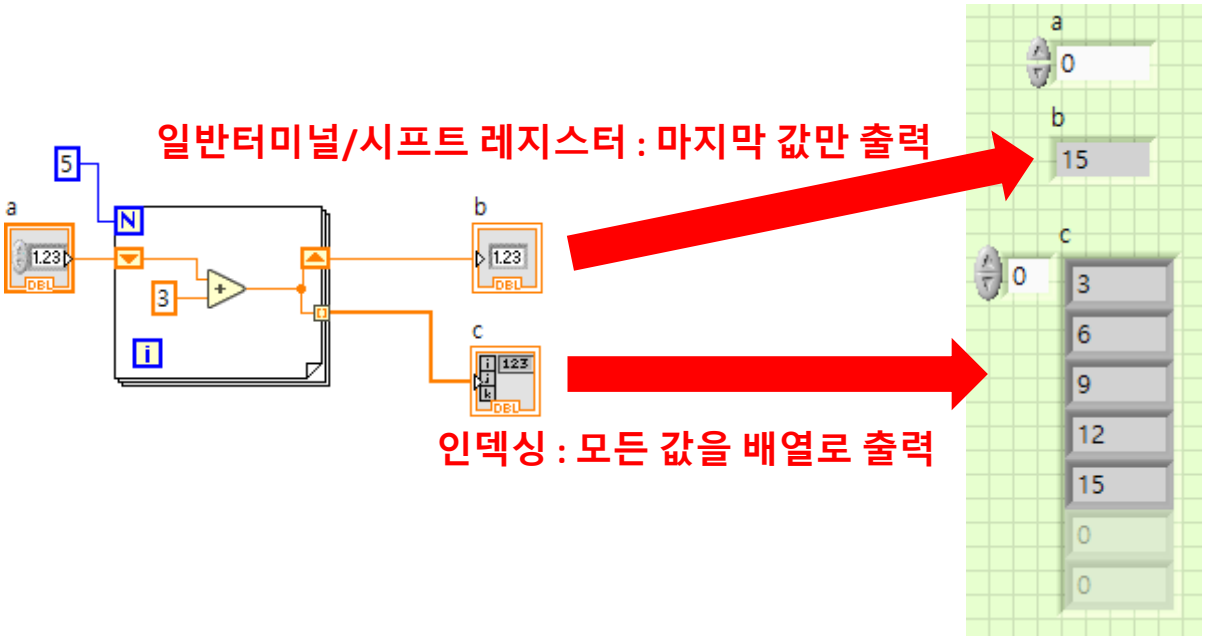
Today's Experiment

- 루프 indexing : 매 반복마다 터널의 값을 순차적으로 배열로 저장
  - ➢ While, for 루프에서 사용 가능.



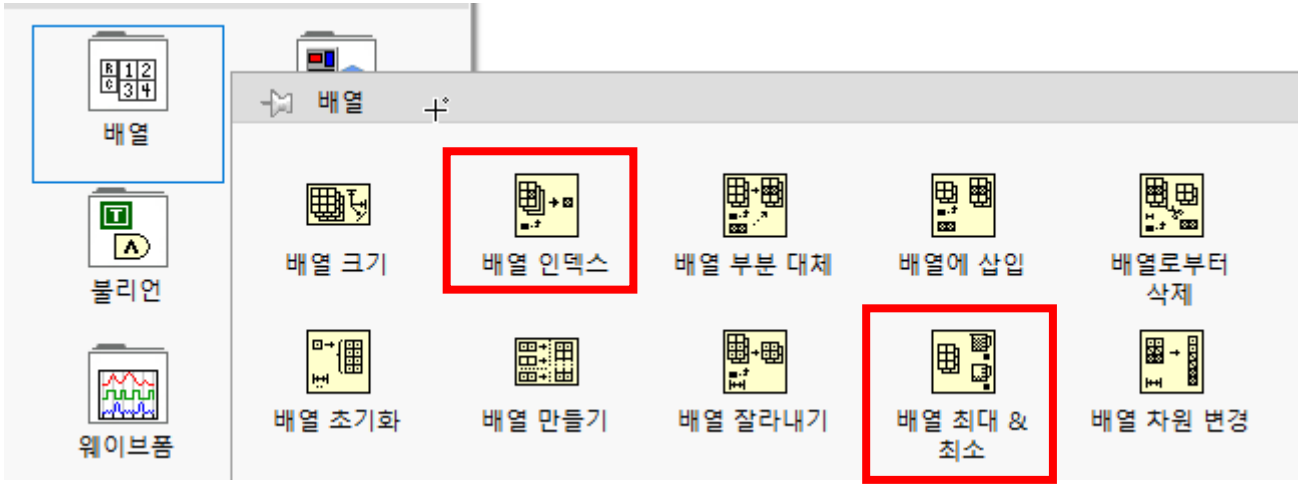
Today's Experiment

- For 루프 구성요소
  - 인덱싱 : 루프의 매 반복마다 터널의 값을 순차적으로 배열로 저장
  - While 루프에서도 사용 가능.



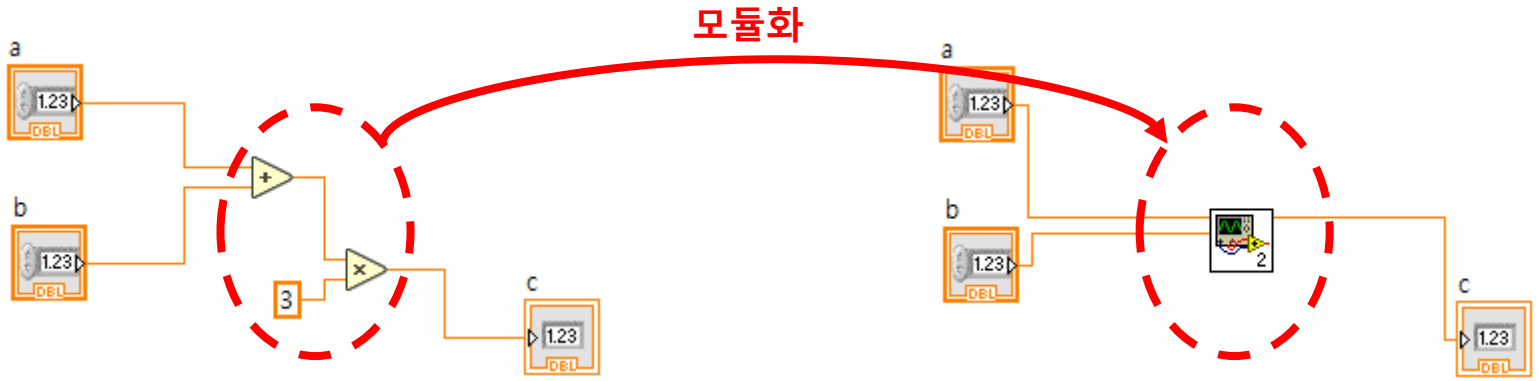
Today's Experiment

- 배열 인덱스
  - 입력 : 배열, 인덱스
  - 출력 : 인덱스 위치의 배열 원소
- 배열 최대 & 최소
  - 입력 : 배열
  - 출력 : 배열 내 최댓값, 최대 인덱스, 최솟값, 최소 인덱스



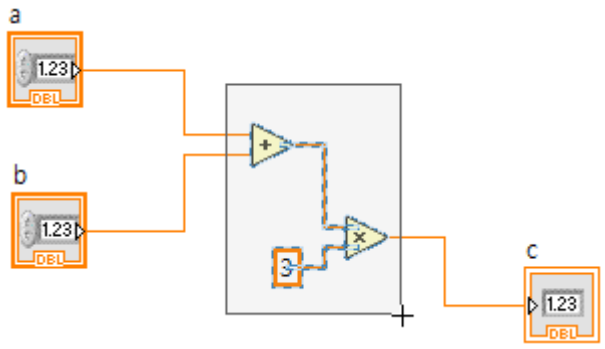
Today's Experiment

- subVI : 자주 사용하는 부분의 코드를 다른 VI나 프로젝트에서 재사용하기 위해 모듈화하여 저장한 것.
  - C/python/matlab에서 "사용자가 직접 만든" 함수와 유사.
  - 하나의 subVI에 하나의 기능
- 모듈화 : 코드의 특정 부분을 subVI로 만드는 것.
  - 코드의 특정 부분만을 모듈화
  - 코드 전체를 subVI로 저장

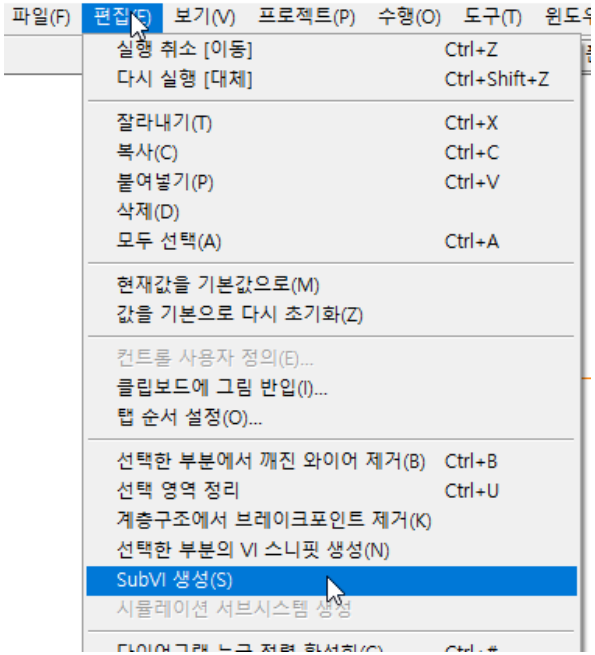


Today's Experiment

- subVI 만드는 방법 – 코드의 특정 부분만을 모듈화
  - 모듈화하고자 하는 부분을 Drag & Drop으로 선택
  - "편집 → SubVI 생성" 선택.



1. Drag & Drop



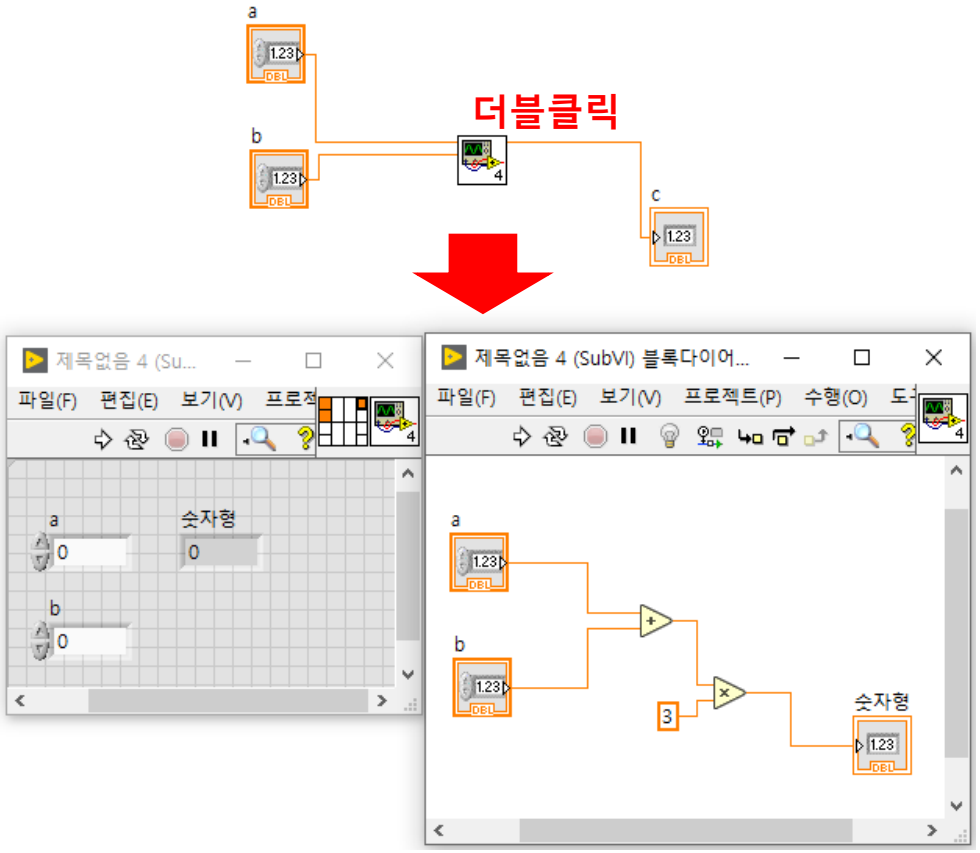
2. "SubVI 생성" 선택

Today's Experiment

- subVI 만드는 방법 – 코드의 특정 부분만을 모듈화
  - 모듈화하고자 하는 부분을 Drag & Drop으로 선택
  - "편집 → SubVI 생성" 선택.

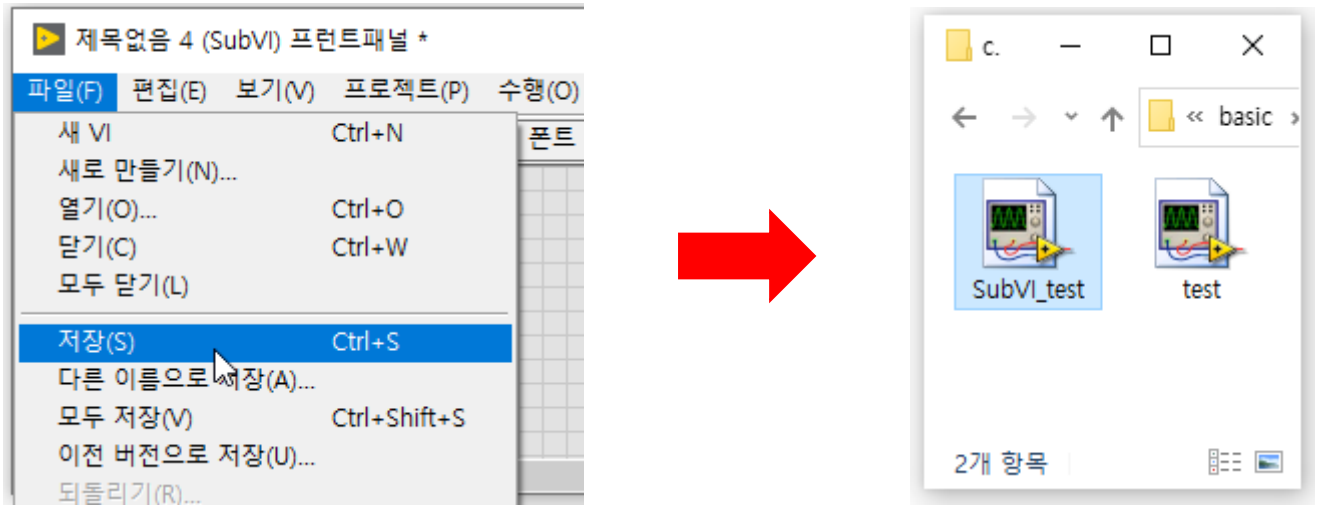
Today's Experiment

- subVI 만드는 방법 – 코드의 특정 부분만을 모듈화
  - ➢ subVI 더블클릭 시 해당 subVI의 프런트패널/블록다이어그램 확인 가능



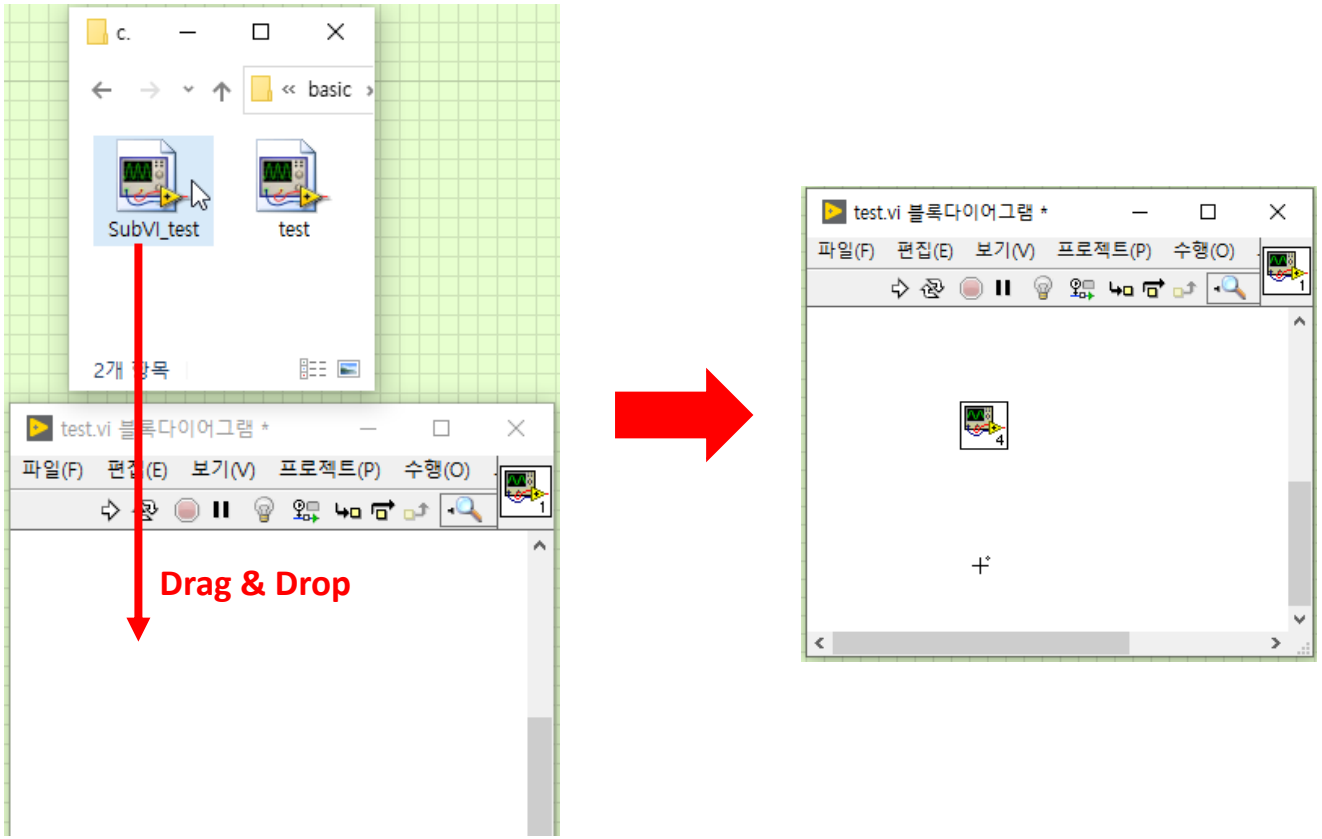
Today's Experiment

- subVI 만드는 방법 – 코드의 특정 부분만을 모듈화
  - subVI의 프런트패널/블록다이어그램에서 "파일 → 저장" 선택
  - 저장할 위치, subVI 이름 지정하여 저장



Today's Experiment

- 저장한 subVI 사용하기
  - ➢ subVI 파일을 "블록다이어그램" 상에 Drag & Drop



Outline

1. Review
2. Symbol Synchronization
3. Today’s Experiment
4. About Report

About Report

- Due: 2024. 11. 25. (월) 오후 04:59
- Must be included :
  - Symbol synchronization 수행 결과
    - 전체 코드(block diagram), 프런트 패널[1]
      - 반드시 Symbol synchronization 수행 부분을 모듈화
      - 만들어진 subvi의 block diagram도 첨부.
  - Case 1 : IQ rate 500,000, samples per symbol 16
    - Symbol sync. 수행하지 않고 downsampling : time domain(Re, Im), constellation [2]
    - Symbol sync. 수행 후 downsampling : time domain(Re, Im), constellation [3]
    - $J_{MOE,d}[k]$ 그래프 [4]
  - Case 2 : IQ rate 1,000,000, samples per symbol 32
    - Case 1과 동일. [5]~[7]
  - Discussion
    - symbol sync. 유무에 따른 통신 performance 차이에 samples per symbol이 미치는 영향?
- jhchoi0226@postech.ac.kr 로 제출