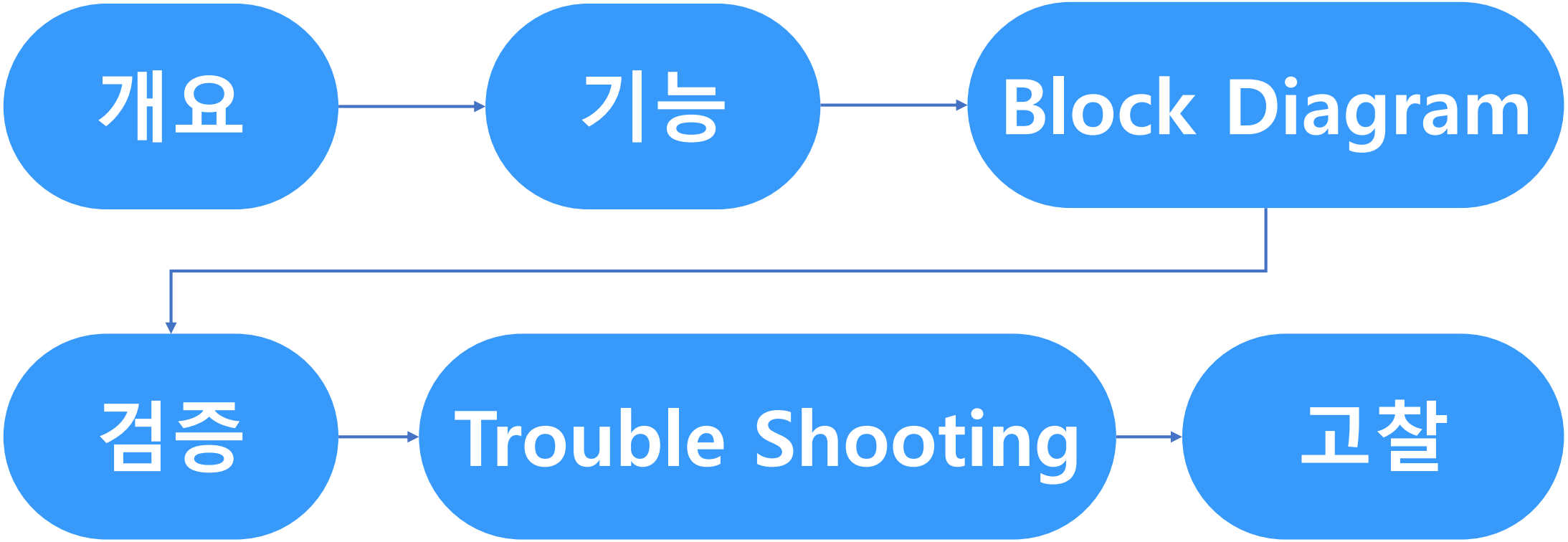
UART + FIFO System Verilog검증

Harman 2기 2조 조민준 허현강

목차

대한상공회의소 서울기술교육센터



개요

대한상공회의소 서울기술교육센터

- **프로젝트 목적**

10000진 카운터 설계와 SystemVerilog를 이용한 UART_TOP의 설계 검증

- **사용 언어**

Verilog + SystemVerilog를 이용해 설계
SystemVerilog를 기반으로 검증

- **검증 목표**

256개의 데이터값에 대한 정상 동작 확인
비정상 조건에서의 동작 확인

- **주요 시나리오**

- 랜덤 데이터 전송 시나리오
(0~ 255 범위의 데이터를 256개 전송)

- 예외 상황 시나리오
(잘못된 스타트 / 스톱 비트, 비트오류 삽입)

개요

베릴로그 확장 언어 (디자인 + 검증 모두 지원)

객체지향 프로그래밍(OOP)

randomization 기능으로 유연한 테스트

인터페이스, 모듈 간 연결 구조 간소화

Assert 기능 지원

기능

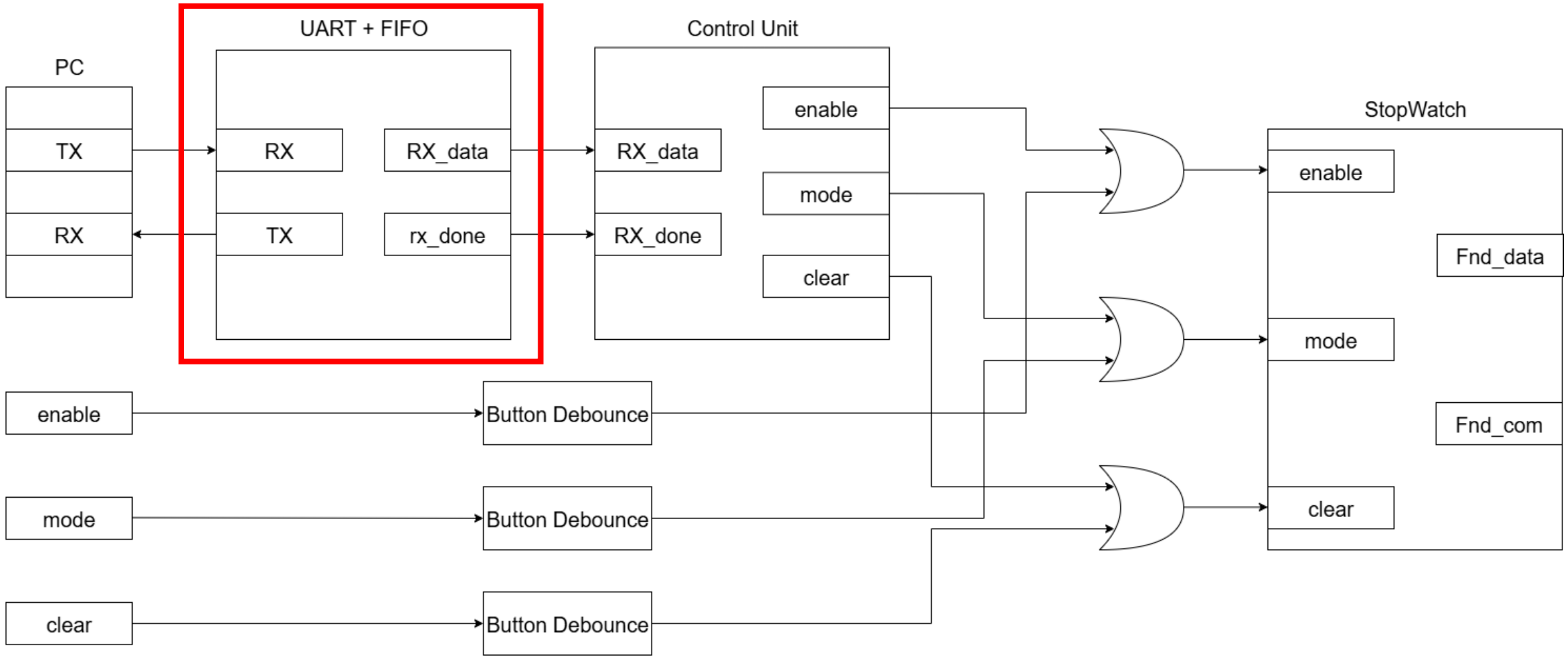
- 7-segment를 이용한 숫자 표시

- 버튼을 통해 STOPWATCH의 Count up / down(mode), stop / run (enable), clear 기능 구현

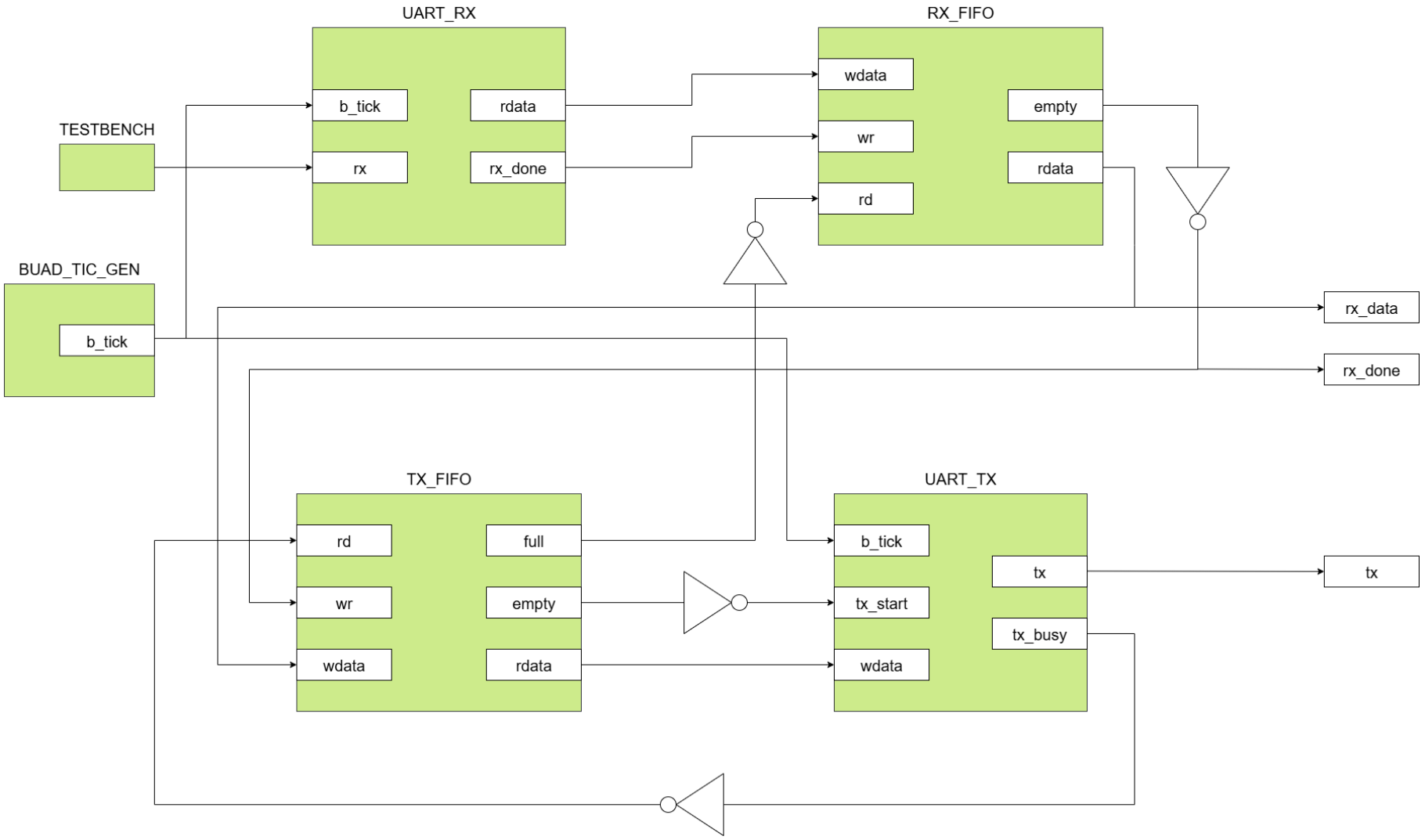
- UART를 통한 STOPWATCH의 기능 제어

- UART의 LOOP-BACK 구조를 이용한 검증

Block Diagram



UART Block diagram



검증 – 시나리오1

Generator : 테스트용 데이터를 만들어서 보냄

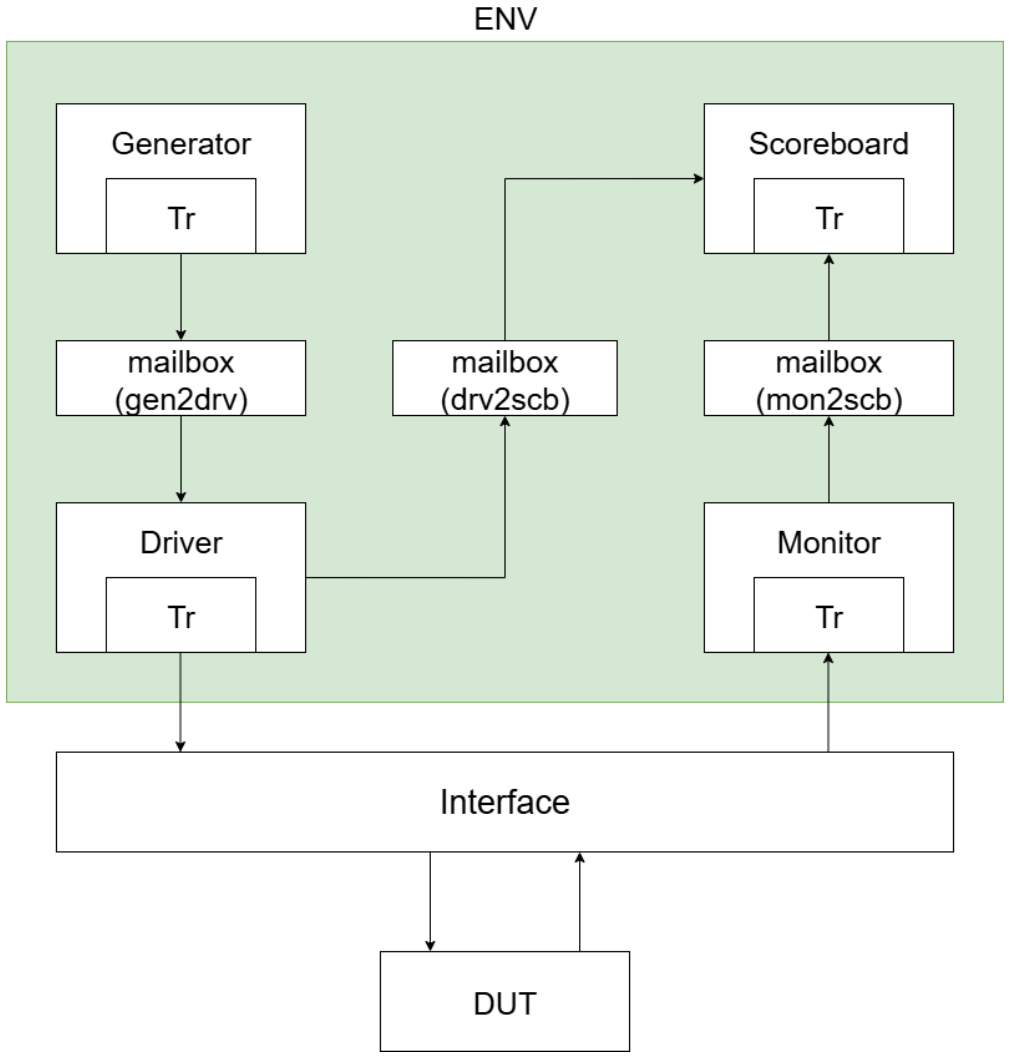
Mailbox: 데이터를 전달하는 통로

Driver : 생성된 데이터를 인터페이스로 전송

Interface : DUT와 테스트벤치 간 신호 연결

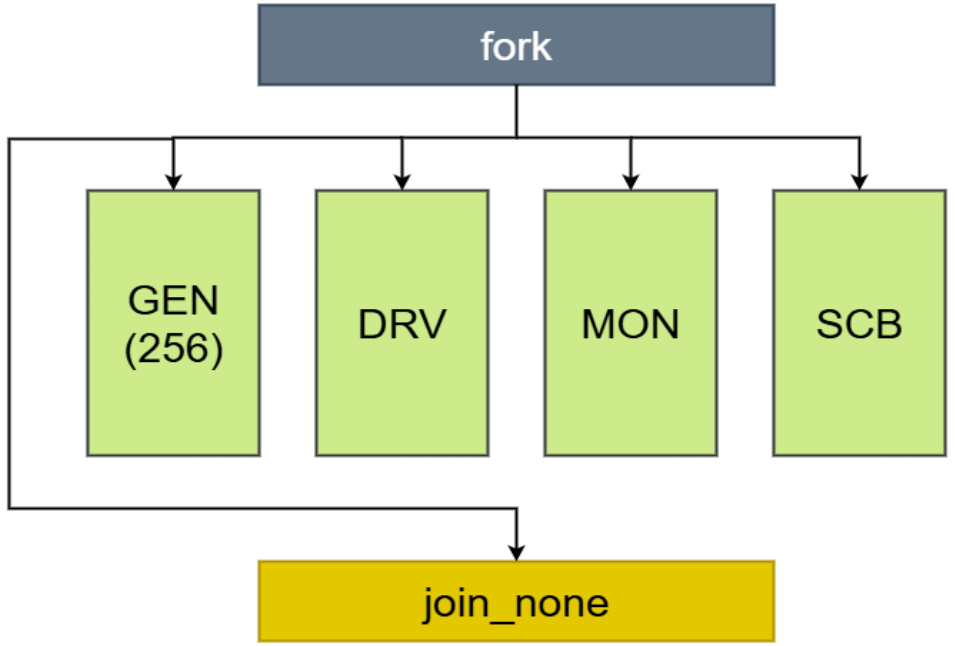
Monitor : 하드웨어에서 보낸 데이터 수집후 전송

Scoreboard : Driver 에서 보낸 데이터와 interface 에서 받은 데이터를 비교



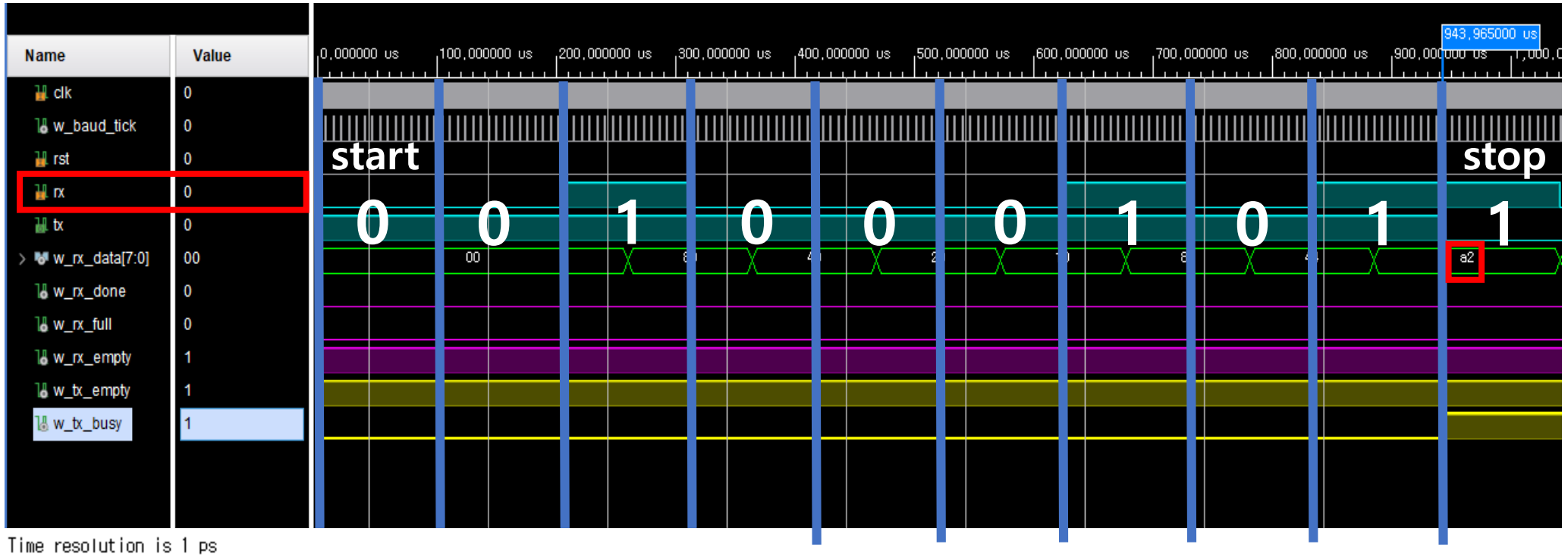
검증 – 시나리오1

1. Driver reset (10ns)
2. 2^8개의 Random data 생성
3. Random data를 mailbox에 담아 DRV로 전송
4. 1byte씩 Interface에 전송
5. MON에서 interface의 출력을 받아 SCB에 전송
6. SCB에서 값을 비교해 pass/fail 결정



검증 – Timing/log 결과

대한상공회의소 서울기술교육센터

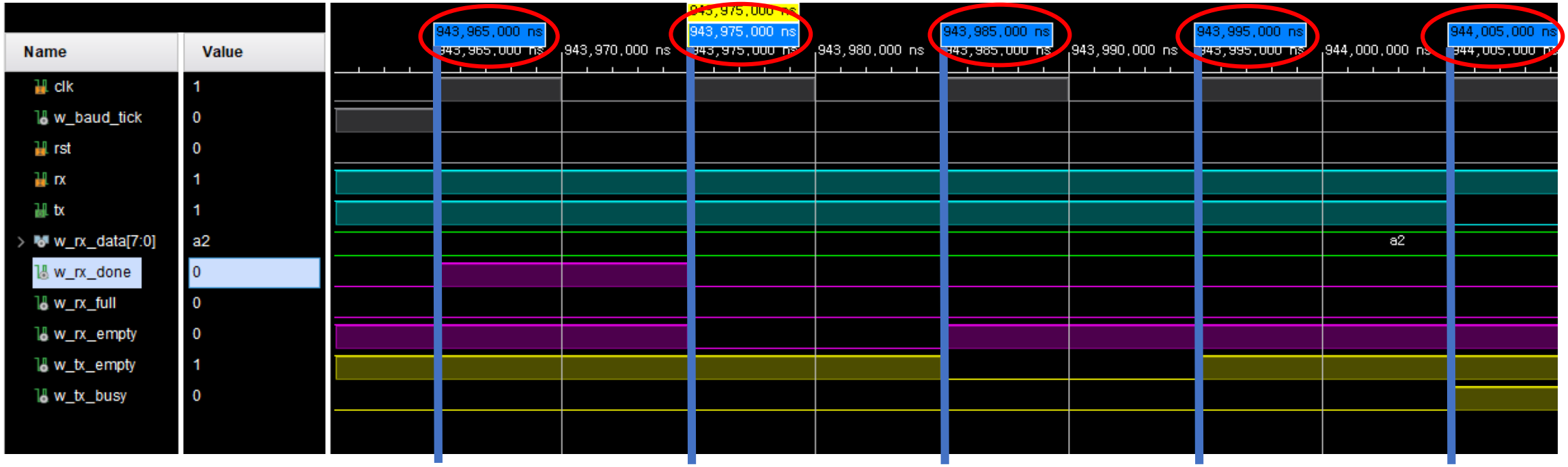



```
Time resolution is 1 ps
[          10 ns] : [  GEN] : Generating 256 shuffled transactions (0x00 to 0xFF)...
[          10 ns] : [GEN] rand_wdata = 0xa2, rdata = 0xxx
[          10 ns] : [GEN] rand_wdata = 0x31, rdata = 0xxx
[          10 ns] : [GEN] rand_wdata = 0x4f, rdata = 0xxx
[          10 ns] : [GEN] rand_wdata = 0xf8, rdata = 0xxx

[     1041610 ns] : [DRV] rand_wdata = 0xa2, rdata = 0xxx
[     1933525 ns] : [MON] rand_wdata = 0xxx, rdata = 0xa2
```

rand_wdata = 0xa2(1010_0010)

검증 – Timing/log 결과

대한상공회의소 서울기술교육센터

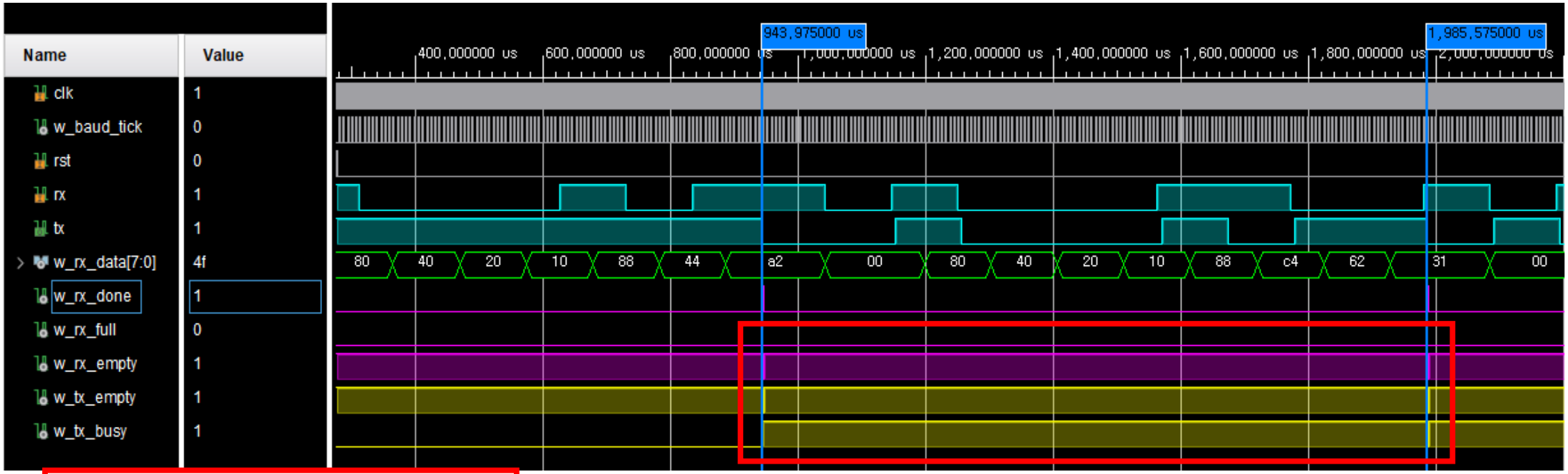


RX_done = 1 RX_empty = 0 RX_empty = 1 TX_empty = 1 TX_busy = 1

TX_empty = 0

검증 – Timing/log 결과

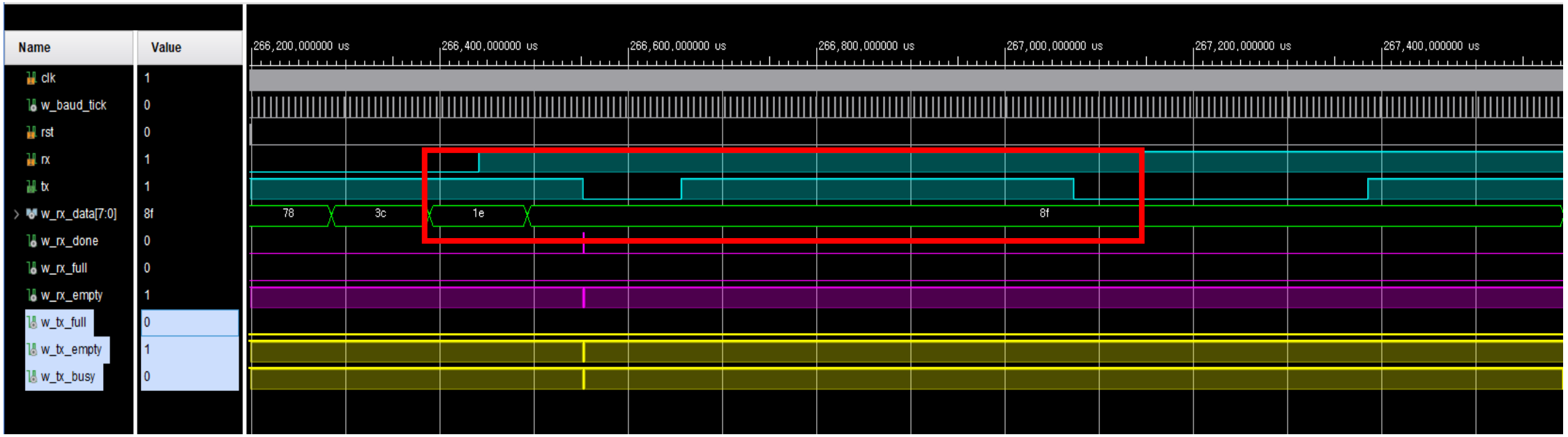
대한상공회의소 서울기술교육센터




```
[     1933525 ns] : [MON] rand_wdata = 0xxx, rdata = 0xa2
[     1985605 ns] : ==============  SCORE BOARD  ===========
[     1985605 ns] : [SCB] PASS: Sent 0xa2, Received 0xa2
[     1985605 ns] : ========================================

[     2083210 ns] : [DRV] rand_wdata = 0x31, rdata = 0xxx
[     2975125 ns] : [MON] rand_wdata = 0xxx, rdata = 0x31
[     3027205 ns] : ==============  SCORE BOARD  ===========
[     3027205 ns] : [SCB] PASS: Sent 0x31, Received 0x31
[     3027205 ns] : ========================================
```

검증 – Timing/log 결과

대한상공회의소 서울기술교육센터

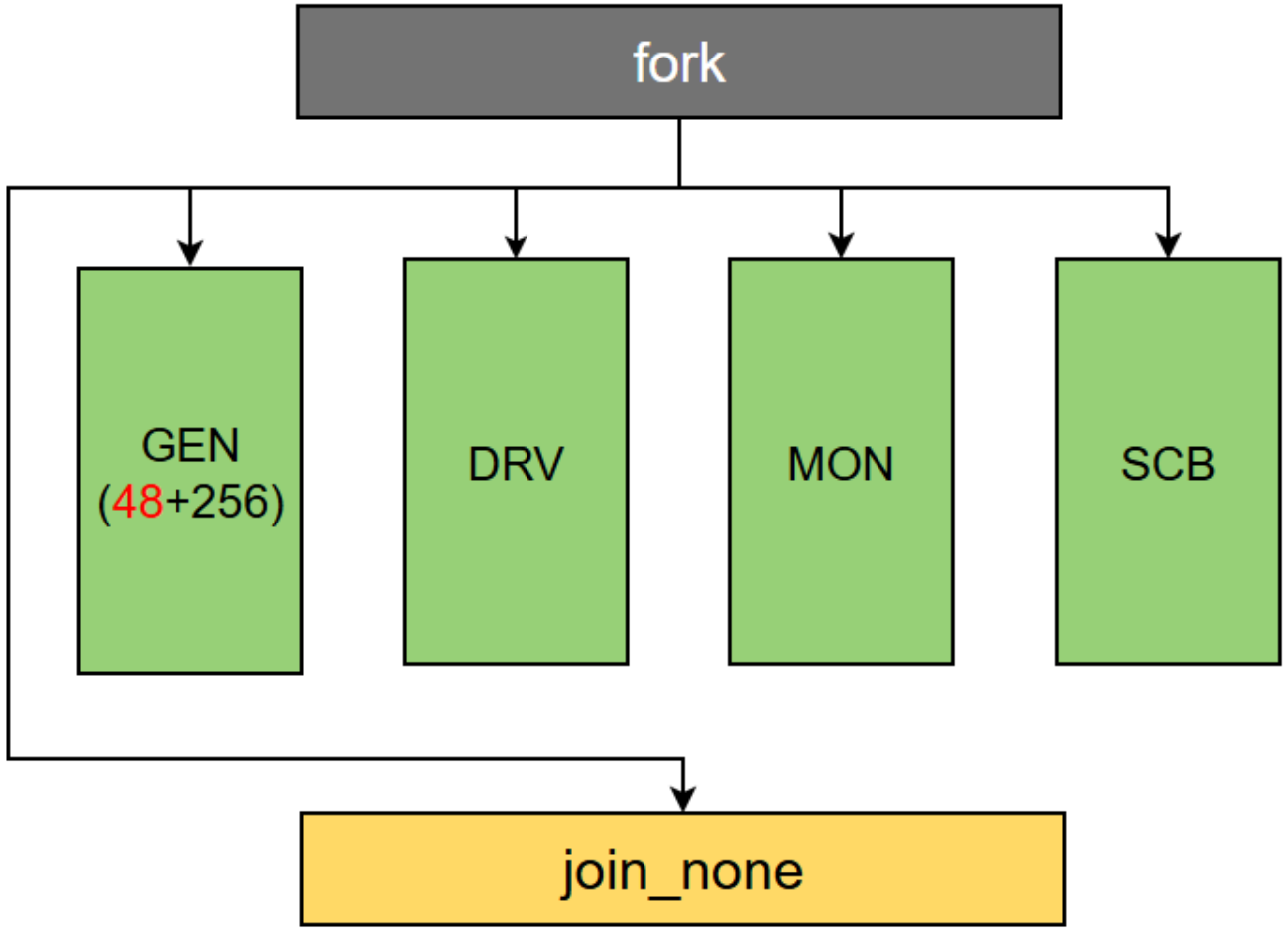



```
[          10 ns] : [GEN] rand_wdata = 0x32, rdata = 0xxx
[          10 ns] : [GEN] rand_wdata = 0x56, rdata = 0xxx
[          10 ns] : [GEN] rand_wdata = 0xed, rdata = 0xxx
[          10 ns] : [GEN] rand_wdata = 0xfb, rdata = 0xxx
[          10 ns] : [GEN] rand_wdata = 0x8f, rdata = 0xxx
[     1041610 ns] : [DRV] rand_wdata = 0xa2, rdata = 0xxx

[   266649610 ns] : [DRV] rand_wdata = 0x8f, rdata = 0xxx
[   267541525 ns] : [MON] rand_wdata = 0xxx, rdata = 0x8f
[   267593605 ns] : ==============  SCORE BOARD  ===========
[   267593605 ns] : [SCB] PASS: Sent 0x8f, Received 0x8f
[   267593605 ns] : ========================================
```

```
============= TEST SUMMARY =============
========================================
  Total Transactions : 256
  Passed             : 256
  Failed             : 0
----------------------------------------
  TEST RESULT: ** PASSED **
========================================
```

검증 목적
※ 비정상/여

1. Driver r
강건성 data
2. start/st
3. pass cou



까지 wait

Transaction Class

```
class transaction;
    logic [7:0] rand_wdata;
    logic [7:0] rdata;
    // Start bit와 Stop bit를 제어하기 위한 변수 추가
    // 기본값은 정상적인 UART 통신 값으로 설정
    logic start_bit = 1'b0;
    logic stop_bit = 1'b1;

    // 이 트랜잭션이 강건성 테스트용인지 표시하는 플래그
    bit is_robustness_vector = 0; // 기본값은 0 (정상 트랜잭션)
```

Start/Stop bit 제어와 noise data 구분을 위한 flag 추가

Generate Class task

```
task generate_robustness_vectors();
    logic four_states[4] = '{1'b0, 1'b1, 1'bx, 1'bz};
    logic [7:0] data_patterns[3];

    data_patterns[0] = 8'hxx; // 모두 X
    data_patterns[1] = 8'hzz; // 모두 Z
    data_patterns[2] = 8'b10xz_01zx; // 섞인 경우
```

```
// 중첩 루프를 사용하여 48개 모든 조합 생성
foreach (four_states[i]) begin      // Start bit 루프 (4가지)
    foreach (four_states[j]) begin  // Stop bit 루프 (4가지)
        foreach (data_patterns[k]) begin // Data 패턴 루프 (3가지)
            tr = new();
            tr.is_robustness_vector = 1;

            tr.start_bit = four_states[i];
            tr.stop_bit  = four_states[j];
            tr.rand_wdata = data_patterns[k];

            total_cnt++;
            // start bit가 0 , stop bit가 1일 때만 기대응답 개수 증가
            if (tr.start_bit === 1'b0 && tr.stop_bit == 1'b1) begin
                env.expected_responses++;
                gen2drv_mbox.put(tr);
                tr.display("GEN");
```

Start	Random data 8bit	Stop
4	3	4

총 48개 강건성 data

검증 – Timing/log 결과

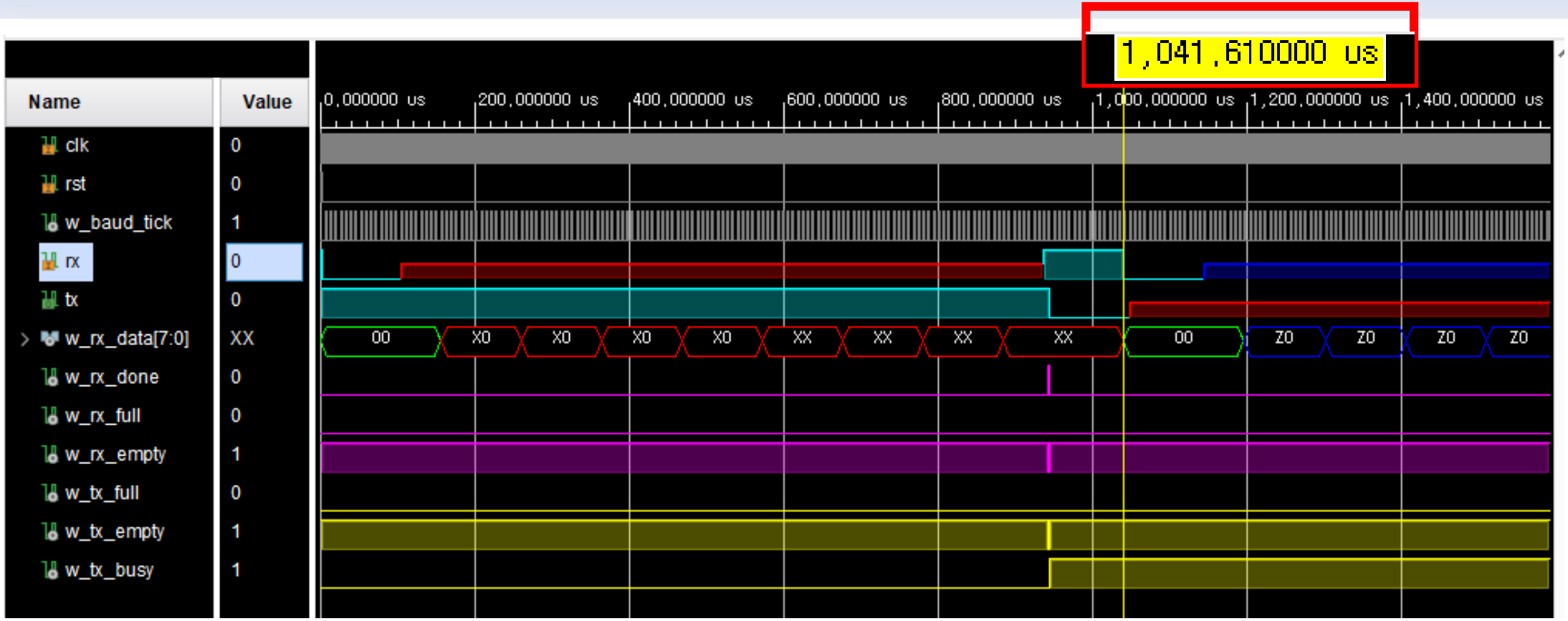
대한상공회의소 서울기술교육센터

콘솔

```
[           0 ns] : [ENV] : Starting Test...
[          10 ns] : [GEN] : Generating 4x4x3=48 robustness vectors...
[          10 ns] : [GEN] : data = 0xxx, start=0, stop=1
[          10 ns] : [GEN] : data = 0xzz, start=0, stop=1
[          10 ns] : [GEN] : data = 0xXX, start=0, stop=1
[          10 ns] : [GEN] : Generating 304 shuffled transactions (0x00 to 0xFF)...
[          10 ns] : [GEN] : data = 0x19, start=0, stop=1
[          10 ns] : [GEN] : data = 0xc6, start=0, stop=1
[          10 ns] : [GEN] : data = 0x9c, start=0, stop=1
[          10 ns] : [GEN] : data = 0xb1, start=0, stop=1
[          10 ns] : [GEN] : data = 0x00, start=0, stop=1
```

Driver 리셋 후 강건성 data와 정상 data 한번에 생성

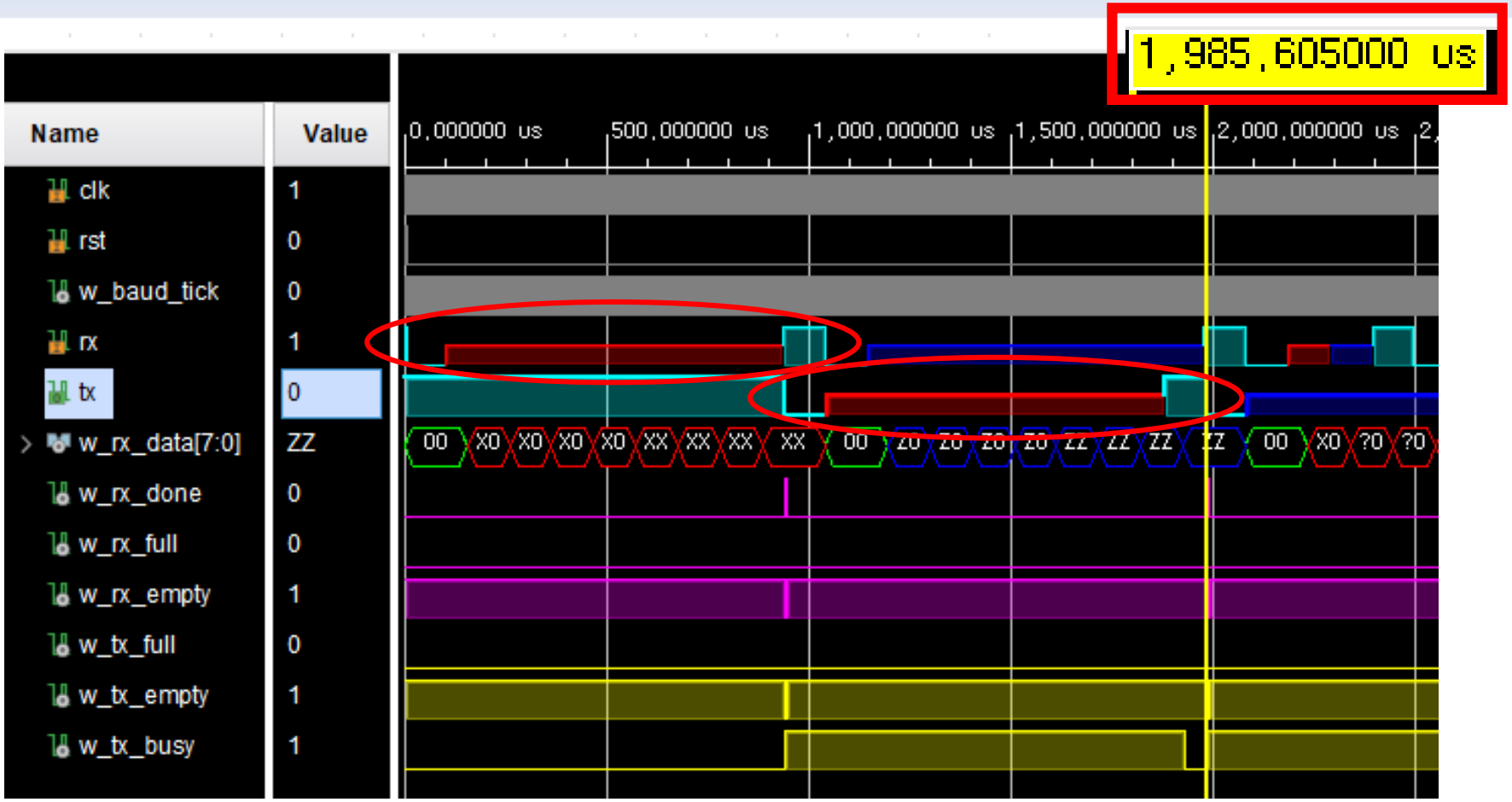
검증 – Timing/log 결과



생성은 한번에 했지만,
Drive는 UART Protocol에 맞게 순차적으로 진행

검증 – Timing/log 결과

대한상공회의소 서울기술교육센터



첫 번째 data 수신 시점 -> 1985605 ns

검증 – Timing/log 결과

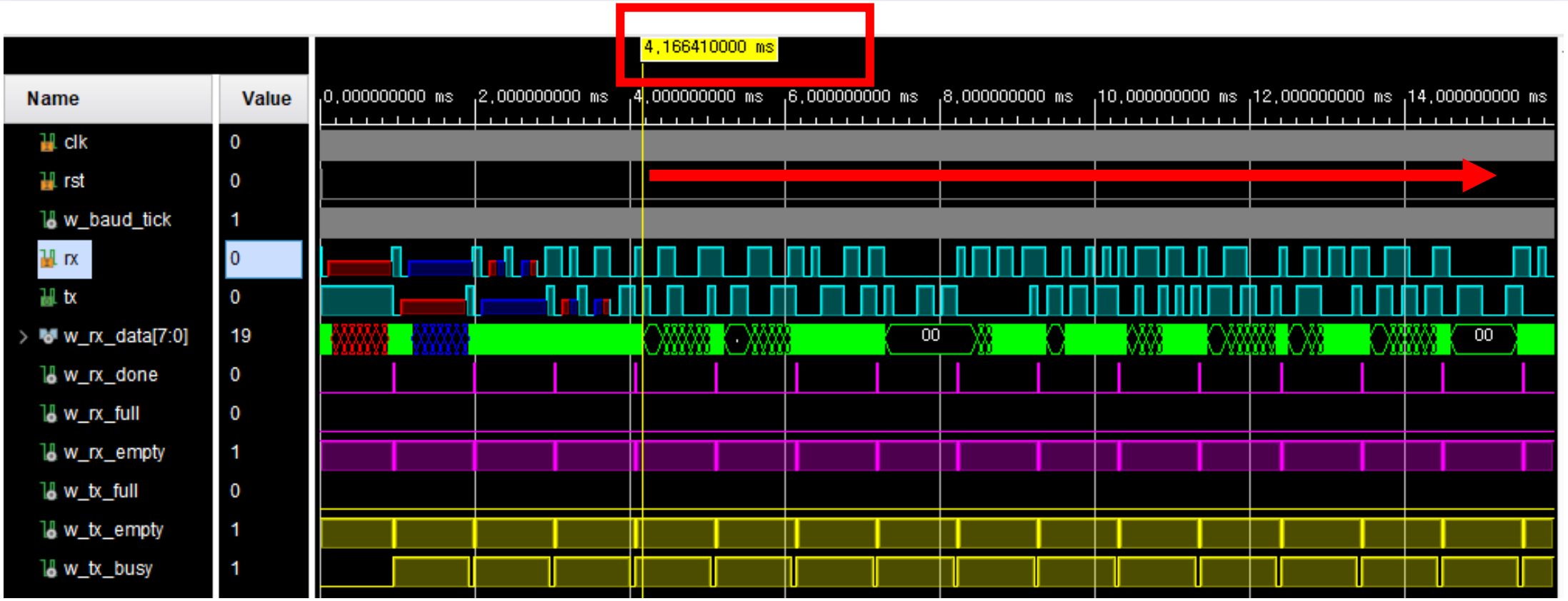
콘솔

```
[     1041610 ns] : [DRV] : Sent data = 0xxx, start=0, stop=1
[     1985605 ns] : [MON] : Received data = 0xxx
[     1985605 ns] : ==============  SCORE BOARD   ===========
[     1985605 ns] : [SCB] FAIL: Sent 0xxx, Received 0xxx
[     1985605 ns] : ========================================

[     2083210 ns] : [DRV] : Sent data = 0xzz, start=0, stop=1
[     3027205 ns] : [MON] : Received data = 0xzz
[     3027205 ns] : ==============  SCORE BOARD   ===========
[     3027205 ns] : [SCB] FAIL: Sent 0xzz, Received 0xzz
[     3027205 ns] : ========================================

[     3124810 ns] : [DRV] : Sent data = 0xXX, start=0, stop=1
[     4068805 ns] : [MON] : Received data = 0xXX
[     4068805 ns] : ==============  SCORE BOARD   ===========
[     4068805 ns] : [SCB] FAIL: Sent 0xXX, Received 0xXX
[     4068805 ns] : ========================================
```

Wire 특성에 의해
Start, Stop 잘 인가 되었다면
x,z가 그대로 기록됨을 확인

검증 – Timing/log 결과

대한상공회의소
서울기술교육센터



강건성 data 이후 정상 data 검증

검증 – Timing/log 결과

콘솔

```
[   269774410 ns] : [DRV] : Sent data = 0x2b, start=0, stop=1
[   270718405 ns] : [MON] : Received data = 0x2b
[   270718405 ns] : ==============  SCORE BOARD   ===========
[   270718405 ns] : [SCB] PASS: Sent 0x2b, Received 0x2b
[   270718405 ns] : ========================================
```

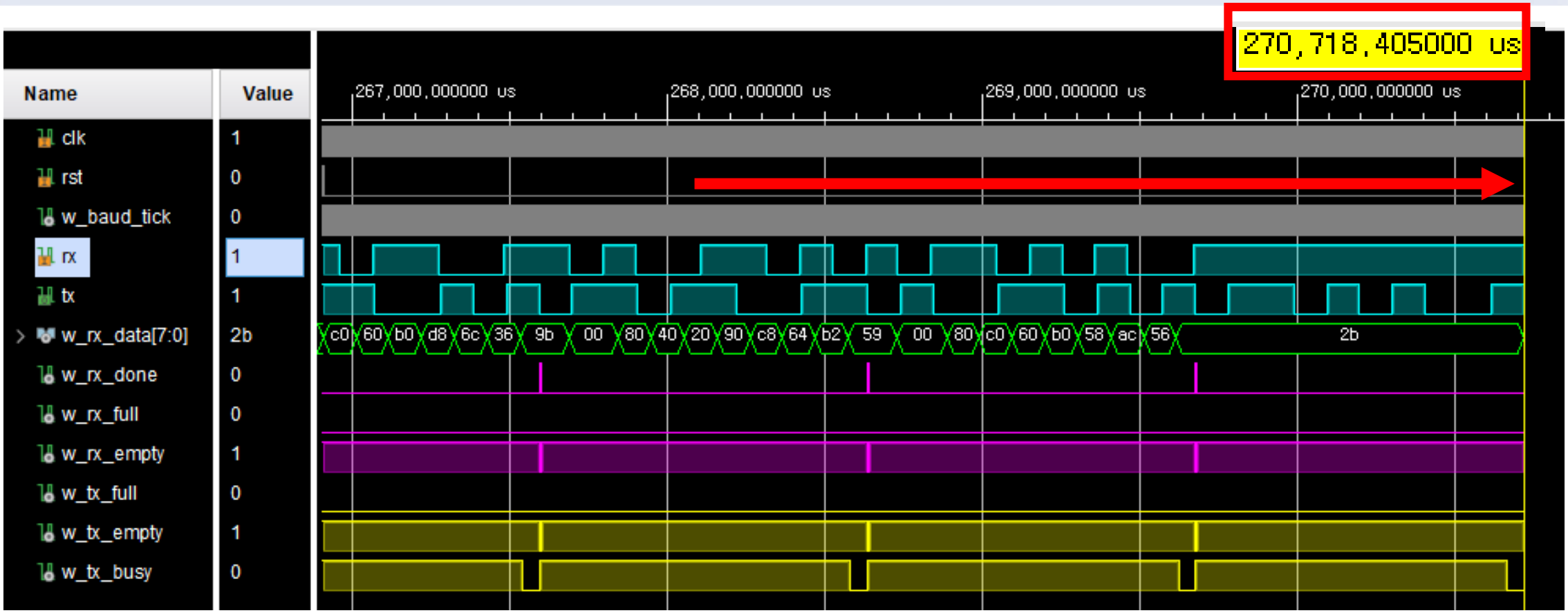
```
========================================
============ TEST SUMMARY ============
========================================
  Total Transactions : 304
  Expected Responses : 259
  Passed             : 256
  Failed             : 3
----------------------------------------
  TEST RESULT: ** FAILED **
========================================
========== COVERAGE REPORT ===========
========================================
  Data Value Coverage: 100.00 % (256 / 256)
========================================
```

모든 transaction 진행 후 Test Summary 출력

0~255의 모든 경우를 cover 했는가?

검증 – Timing/log 결과

대한상공회의소
서울기술교육센터



Loopback 시나리오 종료시점과
Log상의 시간이 일치함을 확인

추가 검증 – data 개수 감소

```
========================================
============ TEST SUMMARY ============
========================================
  Total Transactions : 294
  Expected Responses : 249
  Passed             : 246
  Failed             : 3
----------------------------------------
  TEST RESULT: ** FAILED **
========================================
========== COVERAGE REPORT ===========
========================================
  Data Value Coverage: 96.09 % (246 / 256)
  Uncovered values : 0xf6, 0xf7, 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff,
========================================
```

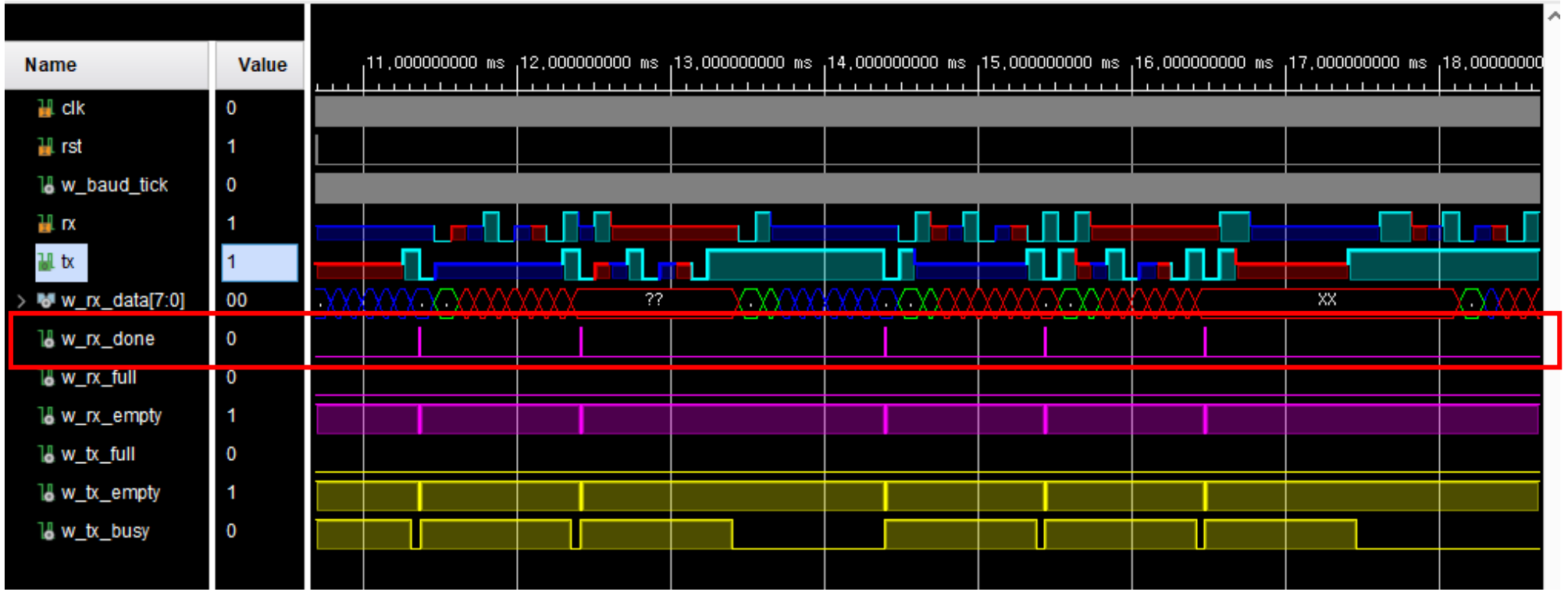
제외한 10개의 data에 따라 Coverage report 출력

추가 검증 – Protocol 위반의 경우

```
//start bit가 0 , stop bit가 1일 때만 기대응답 개수 증가
// if (tr.start_bit === 1'b0 && tr.stop_bit == 1'b1) begin
//     env.expected_responses++;
//     gen2drv_mbox.put(tr);
//     tr.display("GEN");
// end

// 모든 경우를 mailbox로 전달
env.expected_responses++;
gen2drv_mbox.put(tr);
tr.display("GEN");
```

모든 transaction을 mailbox로 전달

추가 검증 – Protocol 위반의 경우



기존 Multibyte 전송 간격 망가짐

추가 검증 – Protocol 위반의 경우

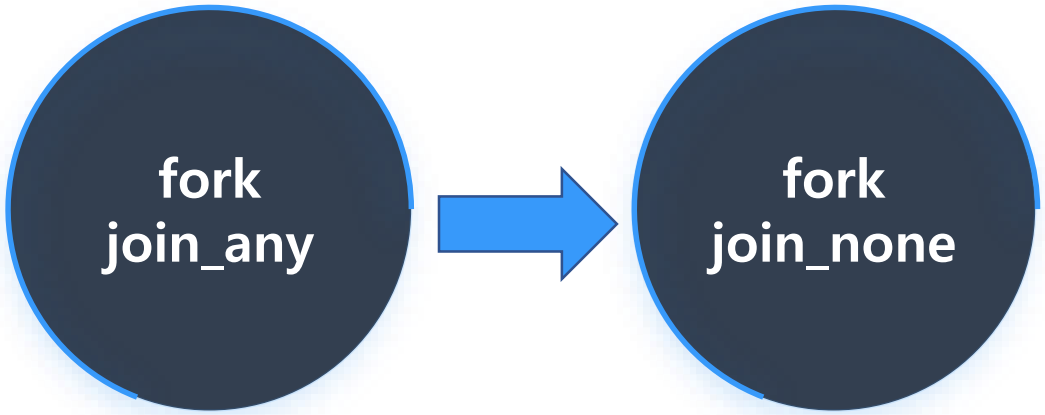
```
[    248844805 ns] : [MON] : Received data = 0x3e
[    248942410 ns] : [DRV] : Sent data = 0x1e, start=0, stop=1
[    248942410 ns] : ==============  SCORE BOARD   ===========
Error: [    248942410 ns] : [SCB]: Expected 0x84, But got 0xea
Time: 248942410 ns  Iteration: 0  Process: /$unit_fifo_sv/scorel
[    248942410 ns] : ========================================

[    249886405 ns] : [MON] : Received data = 0x1e
[    249984010 ns] : [DRV] : Sent data = 0x84, start=0, stop=1
[    249984010 ns] : ==============  SCORE BOARD   ===========
Error: [    249984010 ns] : [SCB]: Expected 0x1b, But got 0x6e
Time: 249984010 ns  Iteration: 0  Process: /$unit_fifo_sv/scorel
[    249984010 ns] : ========================================

[    250000010 ns] : [ENV] : Test finished by TIMEOUT.
```

```
========================================
========== COVERAGE REPORT ===========
========================================
Data Value Coverage: 75.39 % (193 / 256)
Uncovered values : 0x06, 0x09, 0x0a, 0x0e, 0x1d, 0x1f, 0x21, 0x23, 0x2a,
========================================
```

Start/Stop noise에 따라 Process 순서 꼬임 확인
-> 설계한 모듈이 잘못된 Protocol에 대응하지 못함을 알 수 있다.

TroubleShooting

1.



기존 gen 이벤트 제어

gen후 wait을 통해 시간 흐르도록 제어

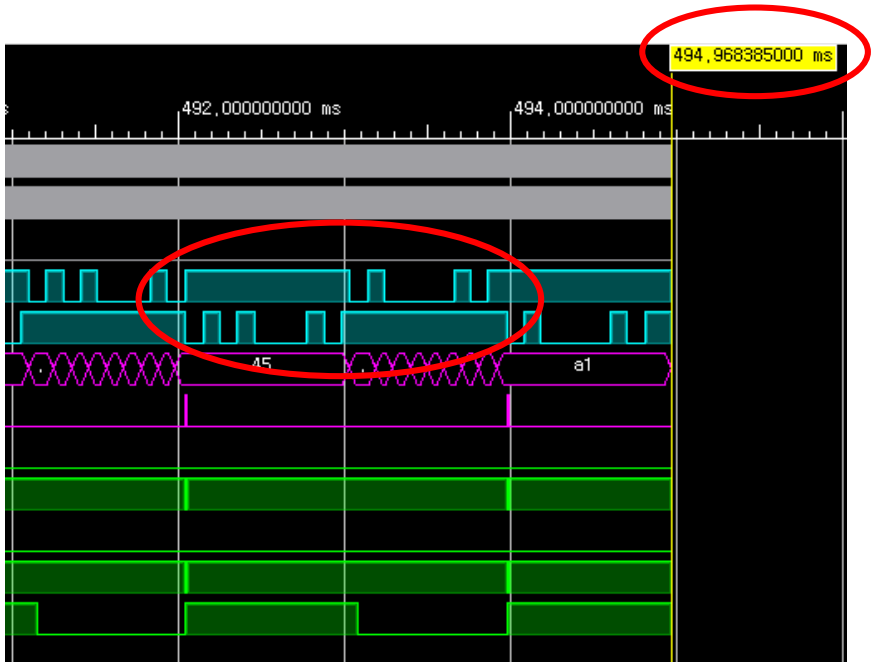
2.

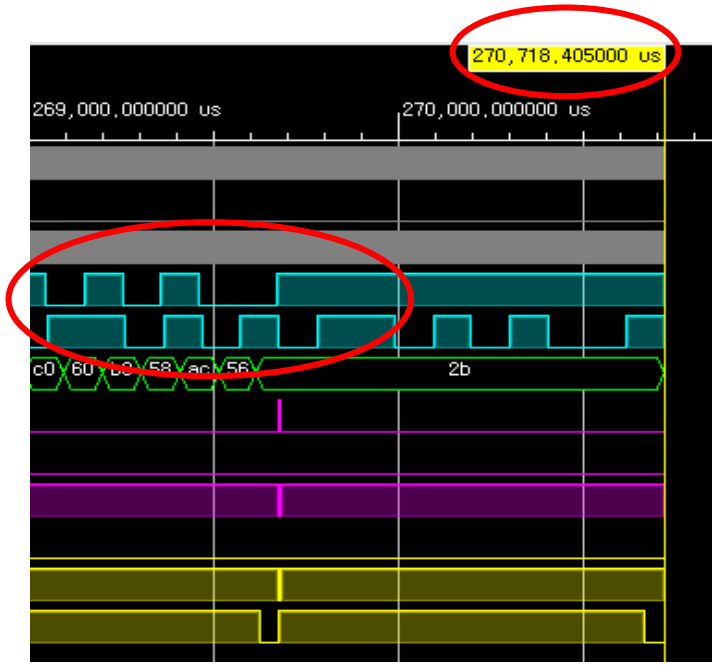
```
task run();
    forever begin
        gen2drv_mbox.get(tr);
        drv2scb_mbox.put(tr);

        uart_if.rx = tr.start_bit;
        #(BIT_PERIOD);

        for (int i = 0; i < 8; i++) begin
            uart_if.rx = tr.rand_wdata[i];
            #(BIT_PERIOD);
        end

        uart_if.rx = tr.stop_bit;
        #(BIT_PERIOD);

        tr.display("DRV");
    end
endtask
endclass  //driver
```

display 출력 시점에 따른 Timing 비교

고찰

대한상공회의소
서울기술교육센터





최초 시나리오에서 시나리오1로 변경 후,
시뮬레이션 Time이 약 46% 줄어듬을 알 수 있다.

고찰


```
// 0~255 모든 값을 섞어서 생성하는 태스크
task generate_shuffled_vectors();
    int data_q[$];
    for (int i = 0; i < 256; i++) begin
        data_q.push_back(i);
    end

    // 직접 shuffle
    // for (int i = data_q.size() - 1; i > 0; i--) begin
    //     int j = $urandom_range(i, 0);
    //     int temp = data_q[i];
    //     data_q[i] = data_q[j];
    //     data_q[j] = temp;
    // end

    data_q.shuffle();
```

Simulation (1 error)

sim_1 (1 error)

[XSIM 43-3980] File "D:/Working/harman_8/2025_0915_uart_fifo/2025_0915_uart_fifo.srcs/sim_1/new/tb_uart_fifo.sv" Line 107 : The SystemVerilog feature "Array methods sort/rsort/reverse/shuffle on queue/dynamic-array" is not supported yet for simulation.

메소드 함수를 통해 더 간략하게 할 수 있으면 좋을 것이다.

고찰


```
class generator;

    transaction tr;
    mailbox #(transaction) gen2drv_mbox;

    int total_cnt = 0;

    [VRFC 10-2989] 'environment' is not declared xvlog(VRFC 10-2989)
    View Problem (Alt+F8)    No quick fixes available
    function new(mailbox#(transaction) gen2drv_mbox, environment env);
        this.gen2drv_mbox    = gen2drv_mbox;
        this.env             = env;
    endfunction  //new()
```

```
typedef class environment;
```

```
class environment;
```

typedef 를 통해 generator가 environment를 바라볼 수 있도록 해야함

질의응답

감사합니다