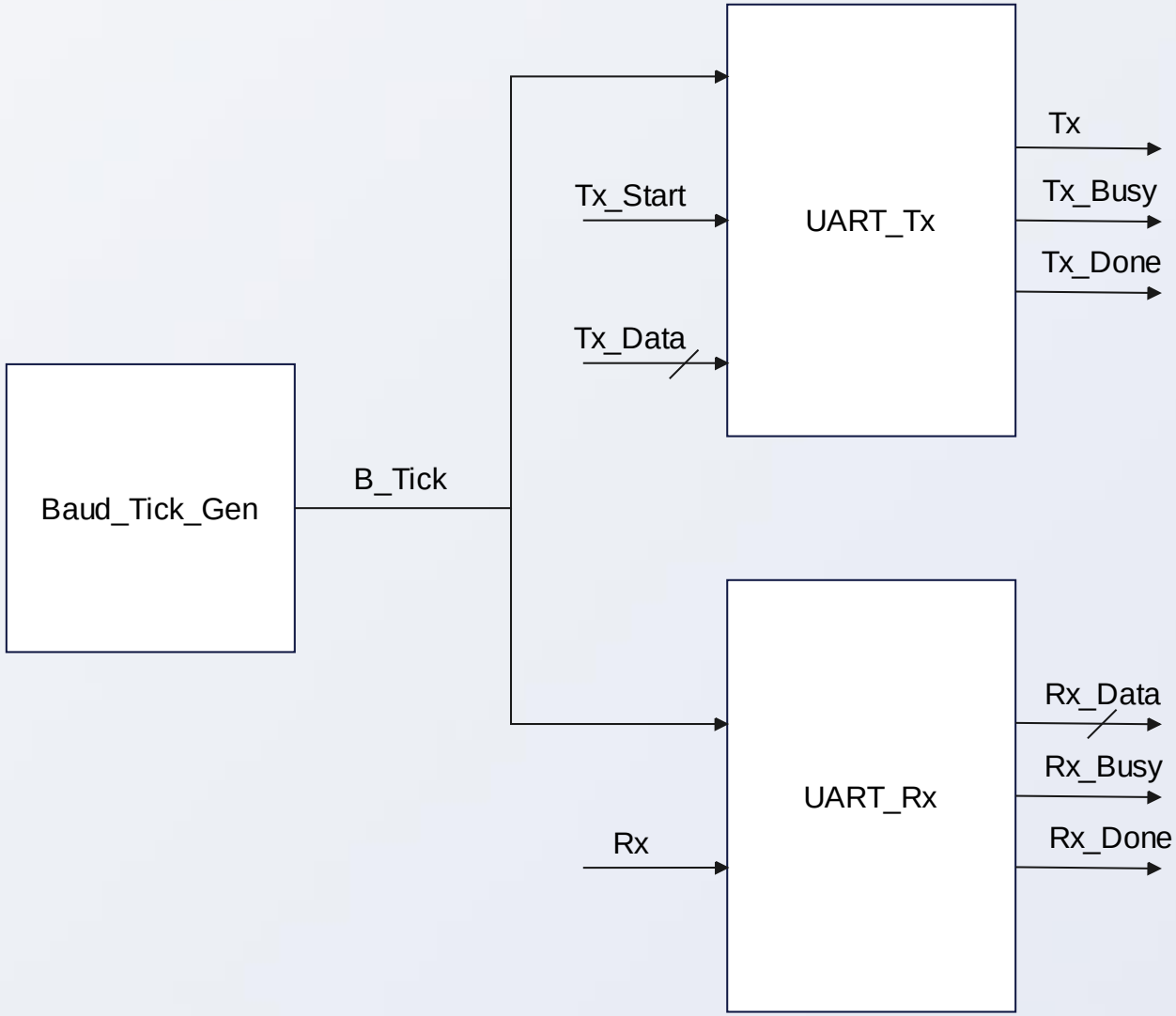
UART-FIFO 설계 및 검증

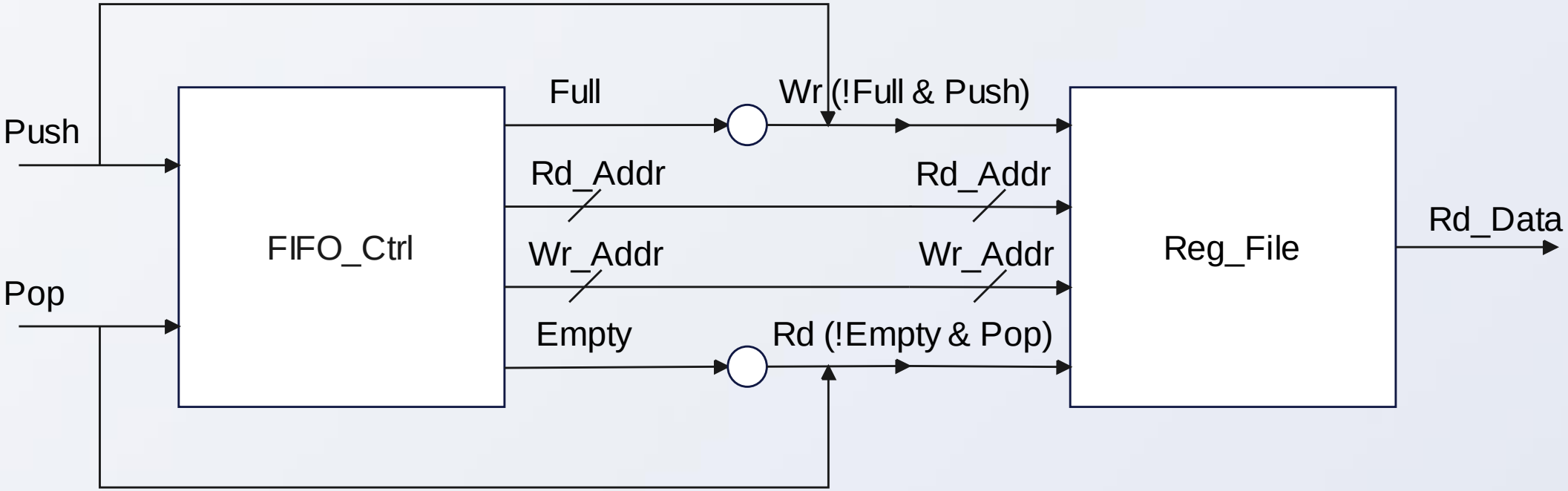
2025.09.15

UART_FIFO

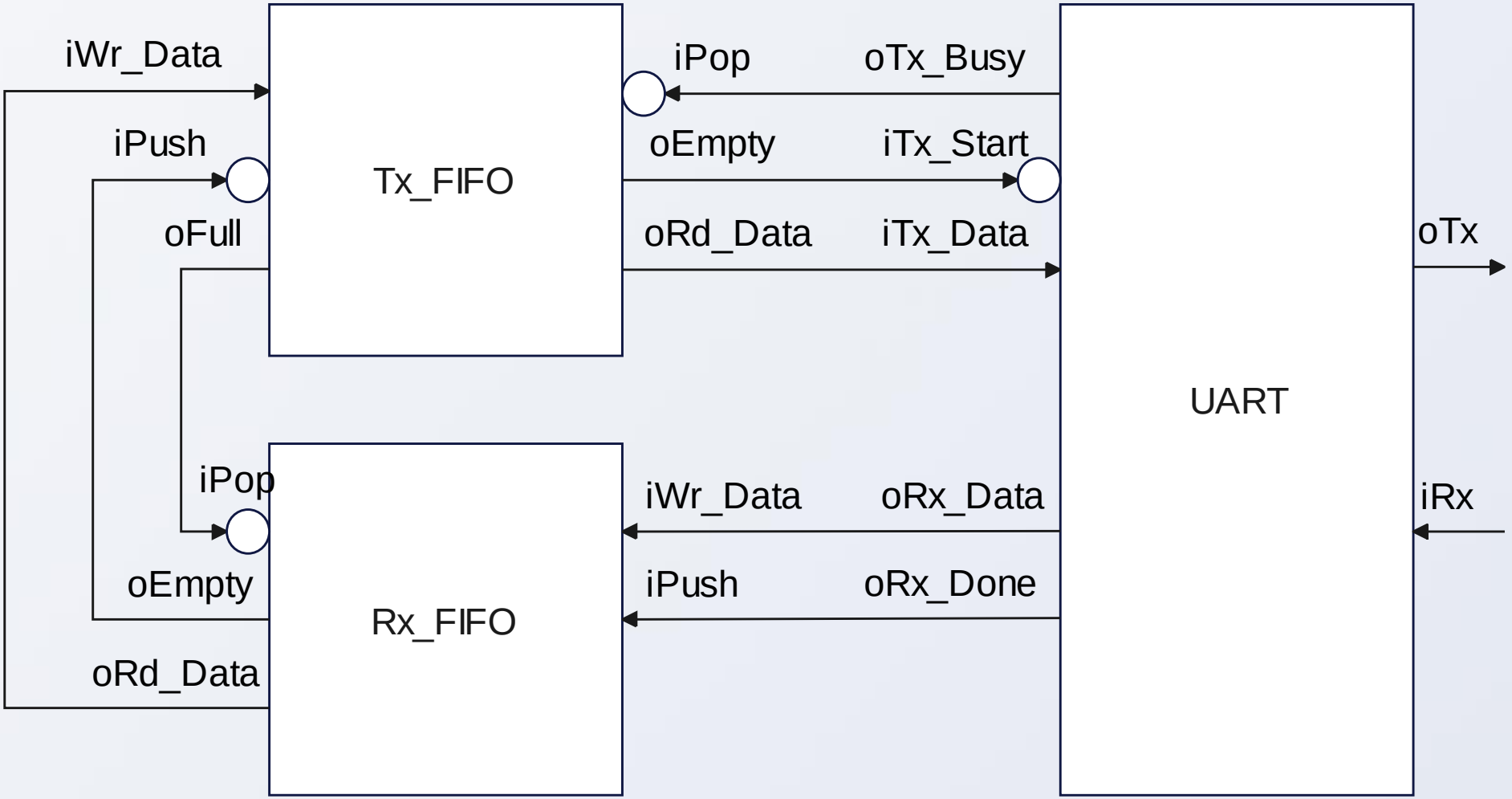
UART 구조


Baud_Tick_Gen
B_Tick
Tx_Start
Tx_Data
UART_Tx
Tx
Tx_Busy
Tx_Done
Rx
UART_Rx
Rx_Data
Rx_Busy
Rx_Done

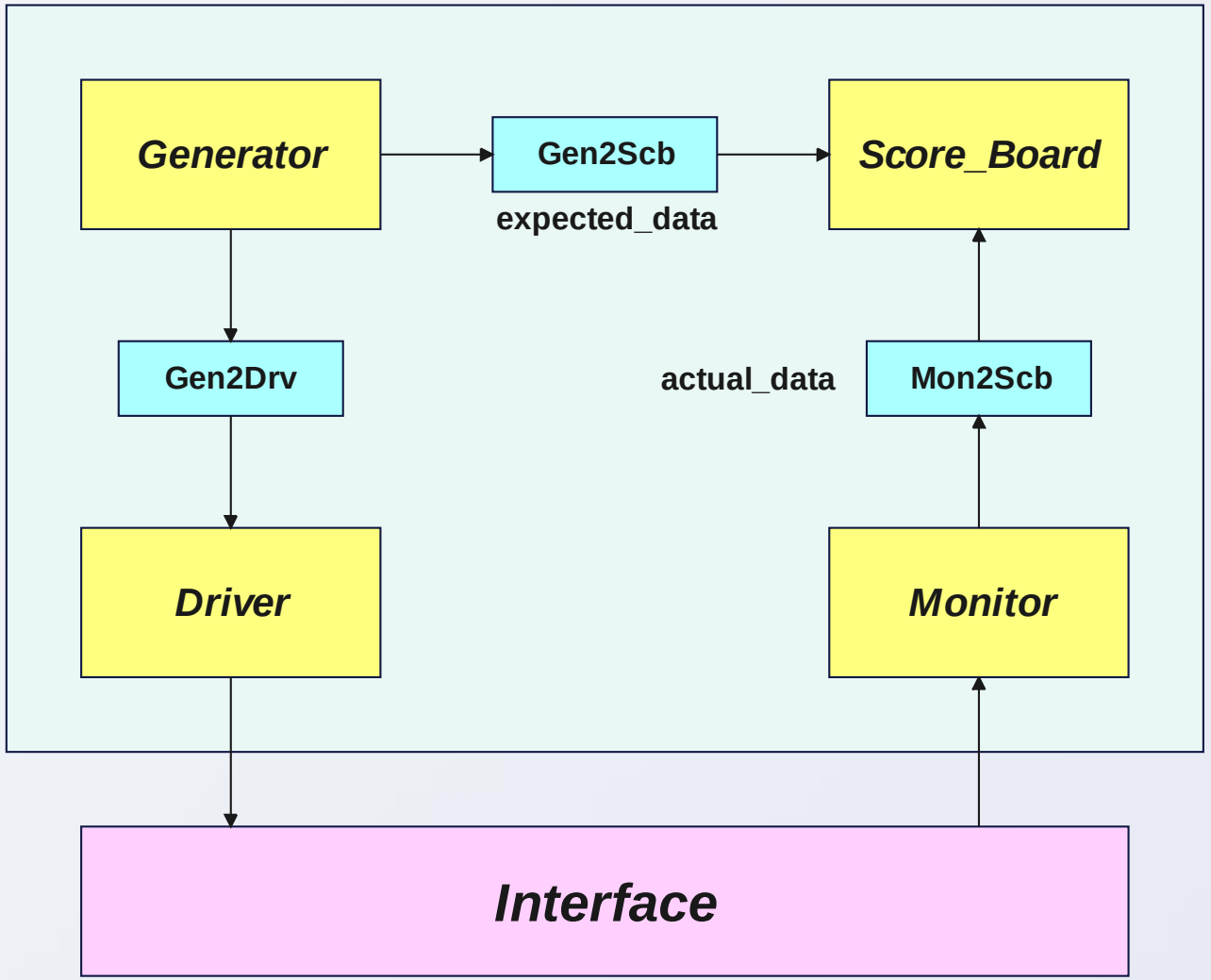
FIFO 구조


Push
Pop
FIFO_Ctrl
Full
Empty
Wr (!Full & Push)
Rd (!Empty & Pop)
Rd_Addr
Wr_Addr
Rd_Addr
Wr_Addr
Reg_File
Rd_Data

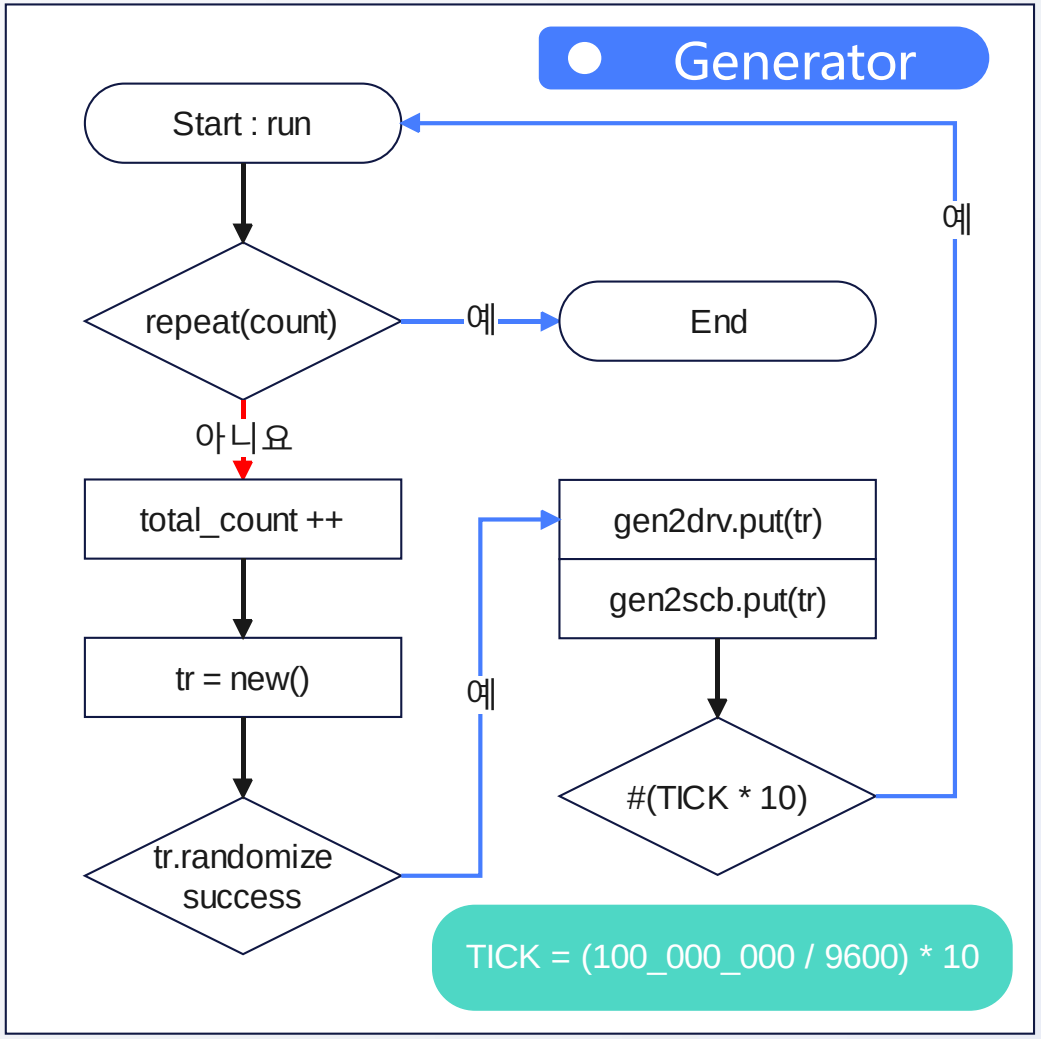
UART_FIFO 구조

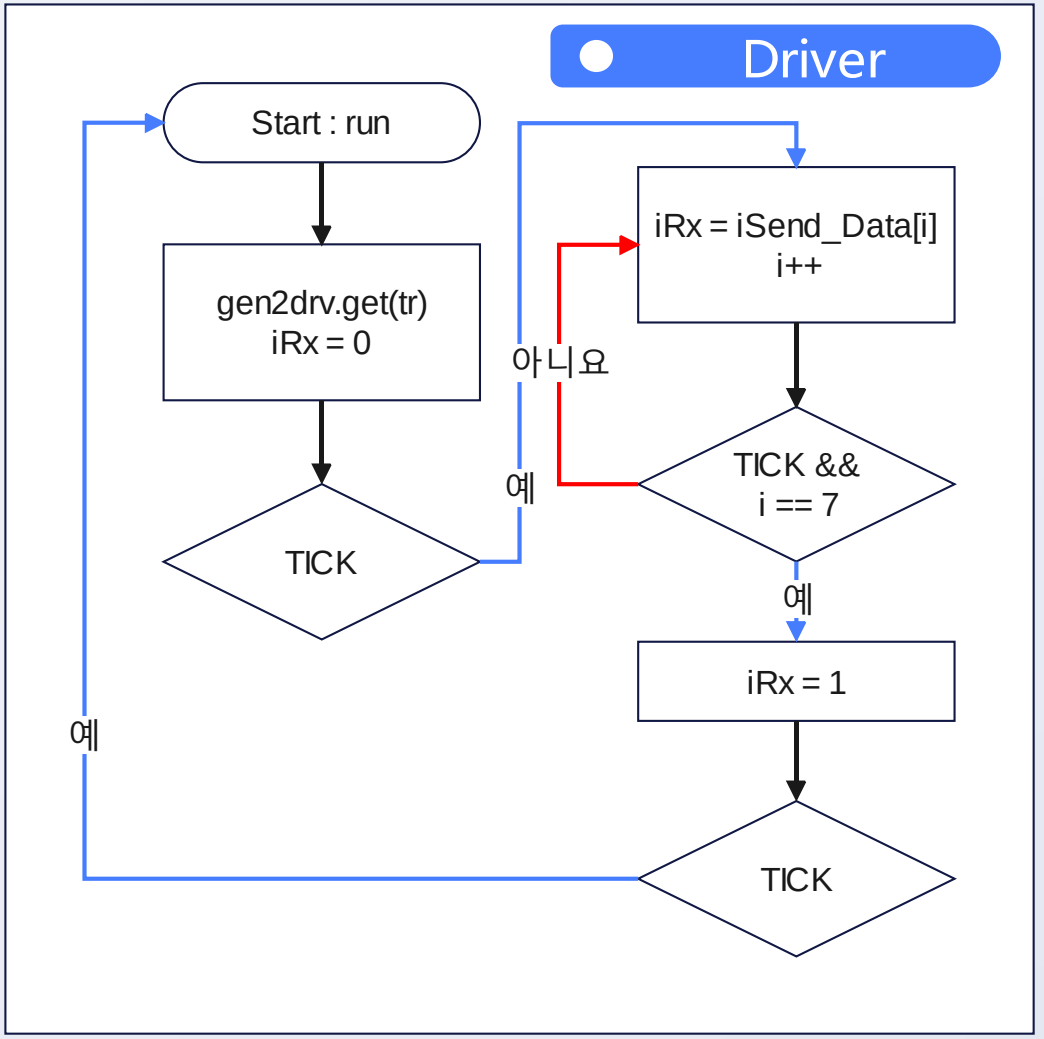

iWr_Data
iPush
oFull
Tx_FIFO
iPop
oTx_Busy
oEmpty
iTx_Start
oRd_Data
iTx_Data
oTx
UART
iPop
oEmpty
Rx_FIFO
oRd_Data
iWr_Data
oRx_Data
iPush
oRx_Done
iRx

Testbench 구조

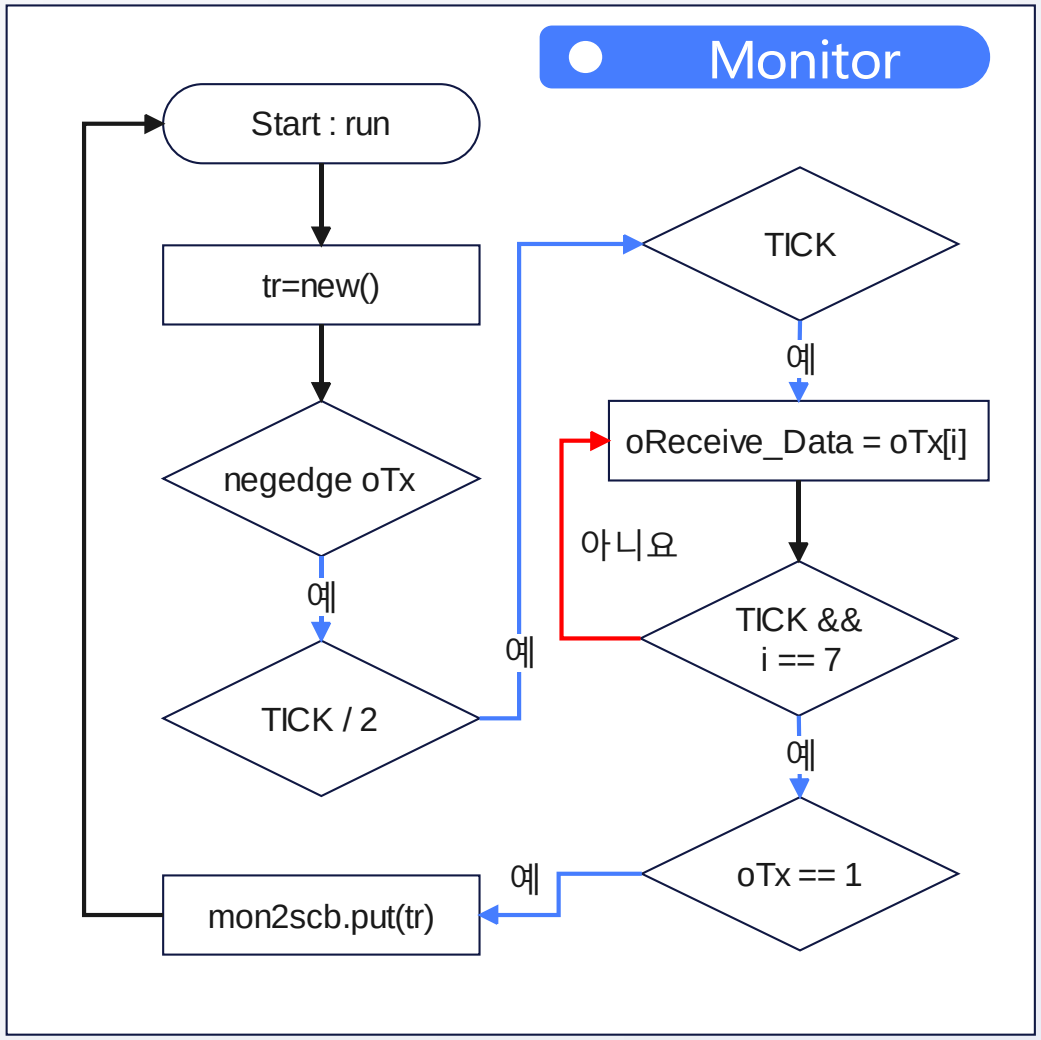


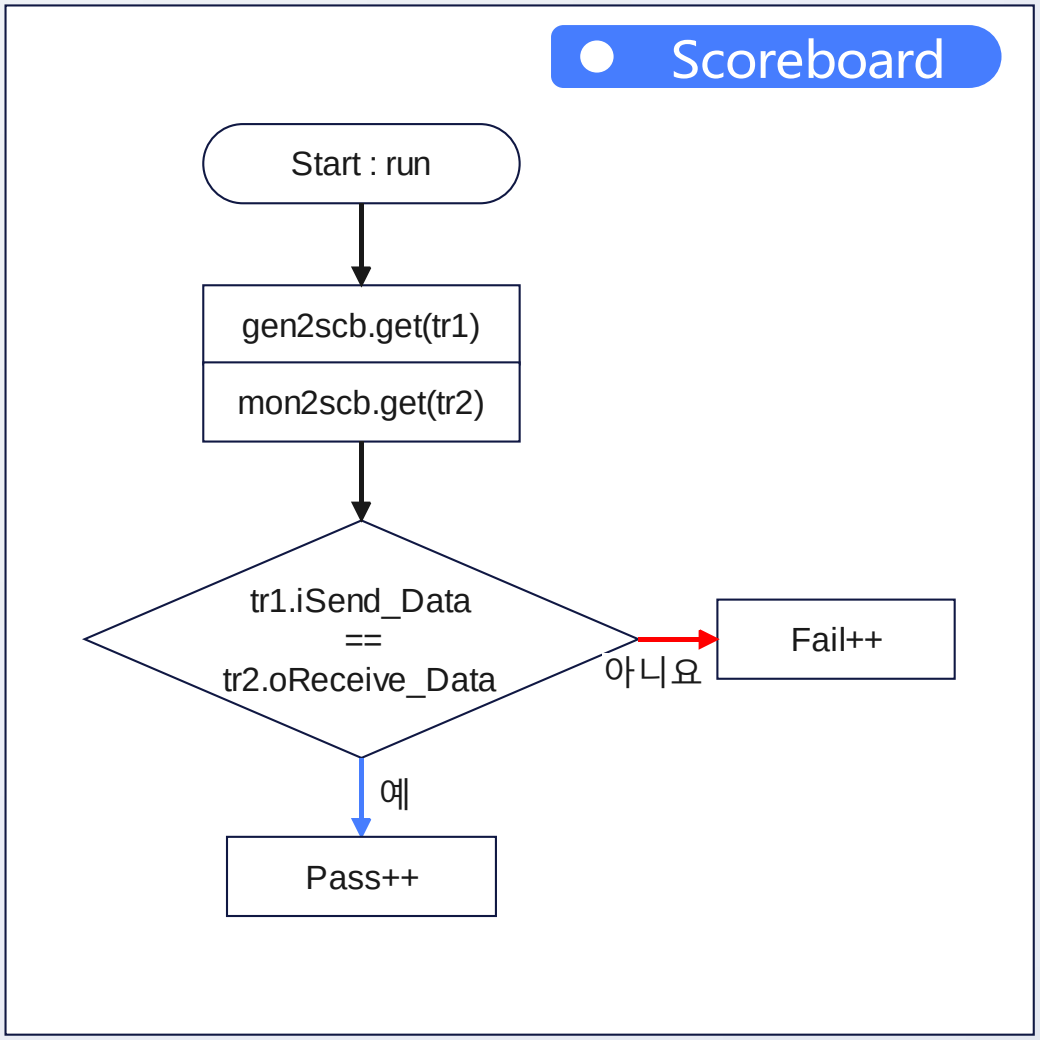
Testbench ASM (Gen / Drv)



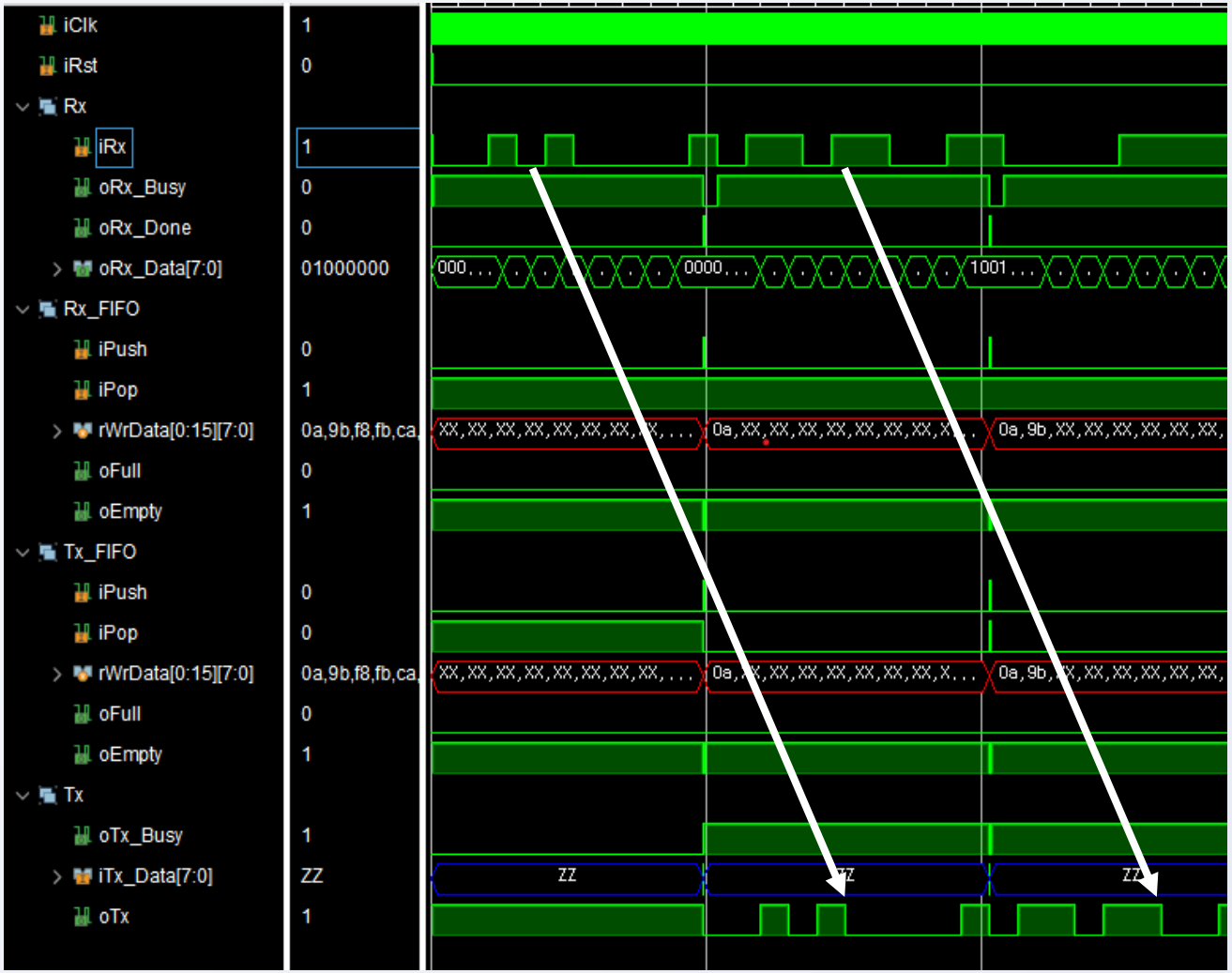


Testbench ASM (Mon / Scb)





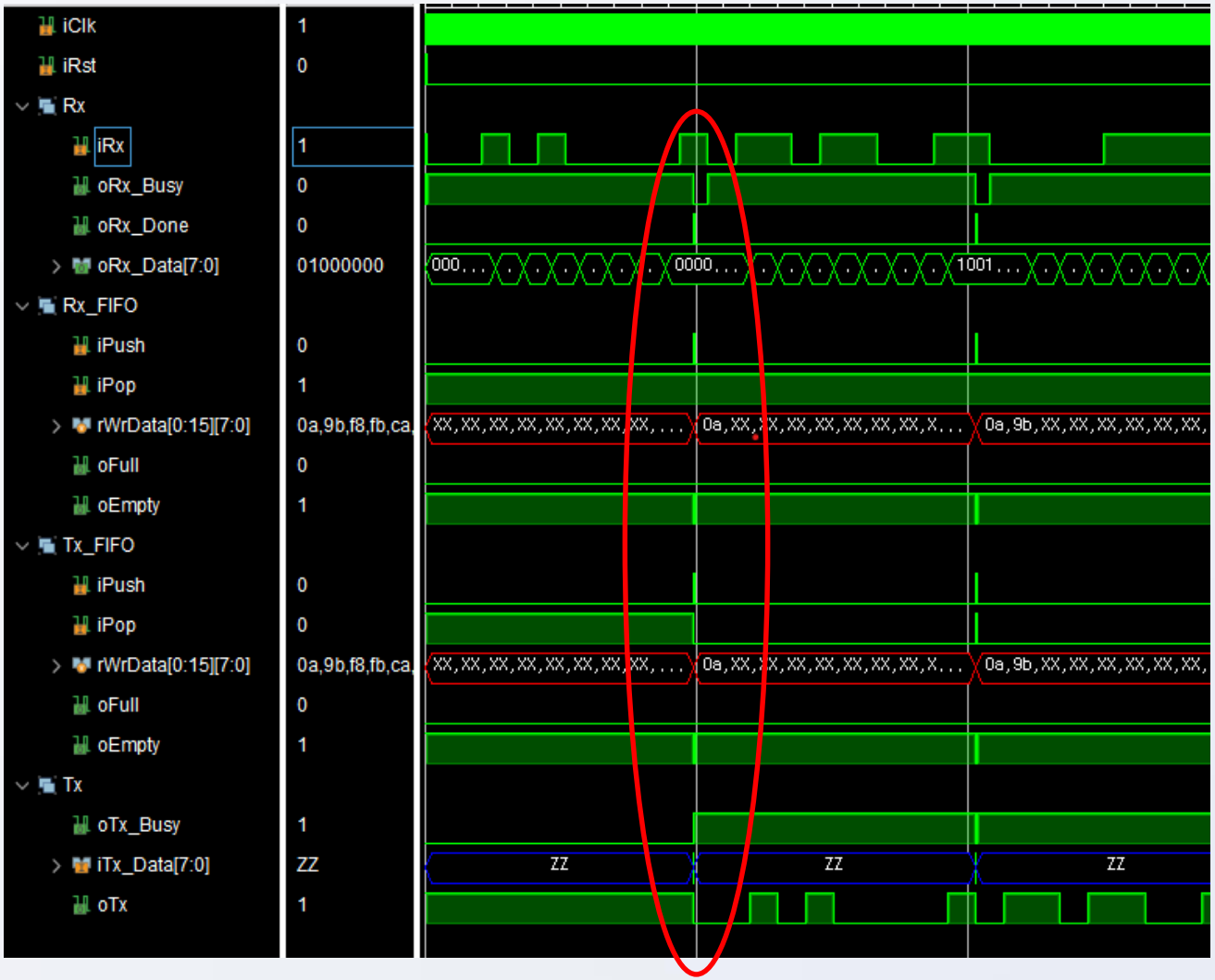
Testbench Result




```
        10416015000 : [GEN] : iSend_Data =  64, oReceive_Data =   0
        10416015000 : [DRV] : iSend_Data =  64, oReceive_Data =   0
Send Done!!
        11353485000 : [MON] : iSend_Data =   0, oReceive_Data =  64
        11353485000 : [SCB] : iSend_Data =   0, oReceive_Data =  64
Data Receive Success!!
Send_Data =  64, Receive_Data =  64
======================
Count =          10
======================
```

```
==================================
========== Test Report ==========
==================================
==        Total Test :   10      ==
==         Pass Test :   10      ==
==         Fail Test :    0      ==
==================================
==        Testbench Finish      ==
==================================
```

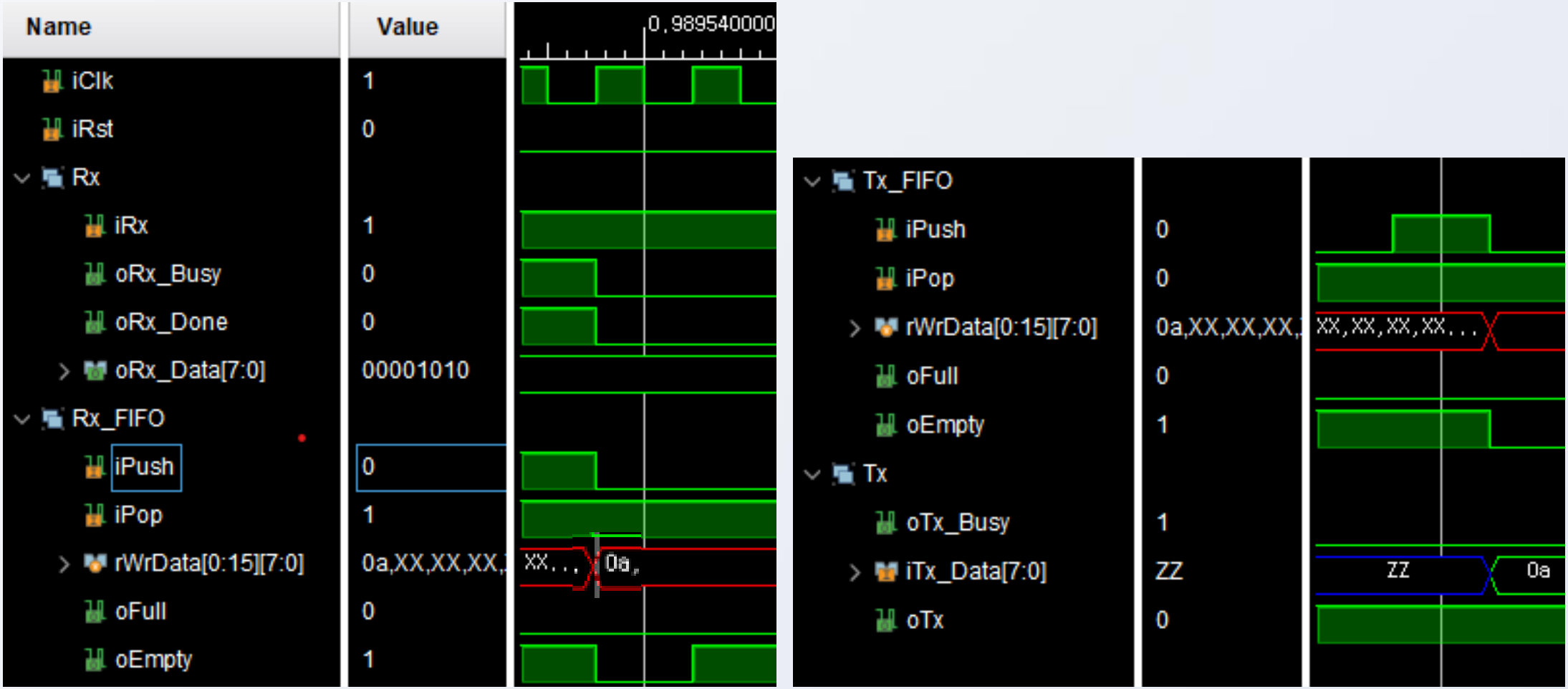
Testbench Result




```
        10416015000 : [GEN] : iSend_Data =  64, oReceive_Data =   0
        10416015000 : [DRV] : iSend_Data =  64, oReceive_Data =   0
Send Done!!
        11353485000 : [MON] : iSend_Data =   0, oReceive_Data =  64
        11353485000 : [SCB] : iSend_Data =   0, oReceive_Data =  64
Data Receive Success!!
Send_Data =  64, Receive_Data =  64
======================
Count =          10
======================
```

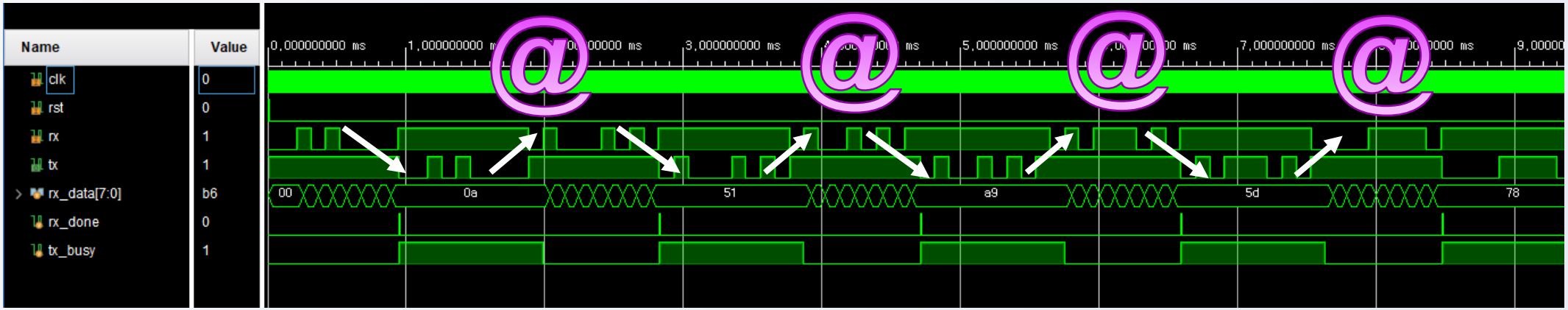
```
==================================
========== Test Report ==========
==================================
==        Total Test :   10      ==
==         Pass Test :   10      ==
==         Fail Test :    0      ==
==================================
==        Testbench Finish      ==
==================================
```

Testbench Rx_Tx Connect



결 론

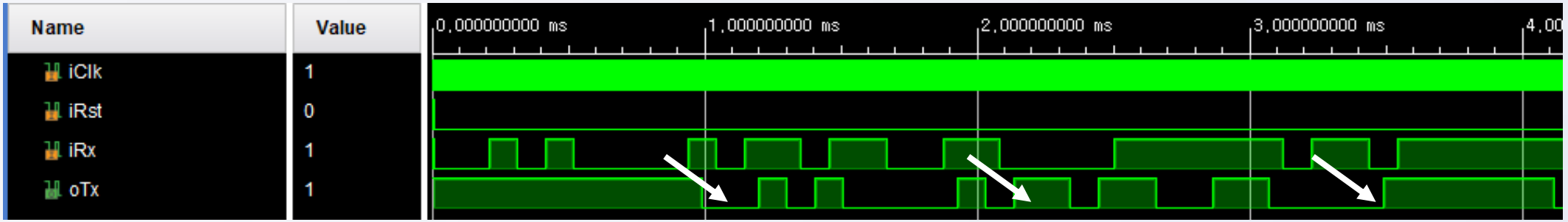
Trouble Shooting 1



Generator ➡ Driver ➡ Moniter ➡ Scoreboard ➡ Gen….

Evevt 제어 시 모든 출력이 끝나야 다시 입력을 받는 상황이 발생

Trouble Shooting 1



Event 제어 대신 Clk 제어를 통해 출력 중에도 입력을 받도록 조절 #(TICK * 10) Delay

➡ Pipeline 구조의 시뮬레이션 환경을 구성

Trouble Shooting 2

```
        10416015000 : [GEN] : iSend_Data =  64, oReceive_Data =   0
        10416015000 : [DRV] : iSend_Data =  64, oReceive_Data =   0
Send Done!!
        11353485000 : [MON] : iSend_Data =   0, oReceive_Data =  64
        11353485000 : [SCB] : iSend_Data =   0, oReceive_Data =  64
Data Receive Fail!!
Send_Data =   0, Receive_Data =  64
----------------------
Count =          19
======================
```

Send_Data랑 Receive_Data가 맞지 않는 현상 발생

➡ Gen2Scb mailbox를 생성하여
입력 데이터를 따로 저장해 비교함

```
        10416015000 : [GEN] : iSend_Data =  64, oReceive_Data =   0
        10416015000 : [DRV] : iSend_Data =  64, oReceive_Data =   0
Send Done!!
        11353485000 : [MON] : iSend_Data =   0, oReceive_Data =  64
        11353485000 : [SCB] : iSend_Data =   0, oReceive_Data =  64
Data Receive Success!!
Send_Data =  64, Receive_Data =  64
----------------------
Count =          10
======================

================================
========== Test Report ==========
================================
==       Total Test :   10      ==
==        Pass Test :   10      ==
==        Fail Test :    0      ==
================================
==       Testbench Finish      ==
================================
```