

Team 6 # "The Final Project"



대한상공회의소
서울기술교육센터

FPGA Based RTL Design

UART & WATCH & SENSOR

NAME
고 종 완
김 태 헌
이 은 성
임 재 홍



AI 시스템 반도체 설계 2기

CONTENTS



CHAPTER 01

Top Module

- 개요
- Top Module Block Diagram
- Design Overview

CHAPTER 02

초음파 센서

- 구성
- 상세 설계
- 검증
- 고찰
- 시연 영상

CHAPTER 03

온습도 센서

- 구성
- 상세 설계
- 검증
- 고찰
- 시연 영상

CHAPTER 04

개발 회고

- 통합 시연 영상
- 프로젝트 요약
- 역할 분담
- 개발 후기



Top Module

CHAPTER 01

- 개요
- Top Module Block Diagram
- Design Overview

개요



OBJECTIVE



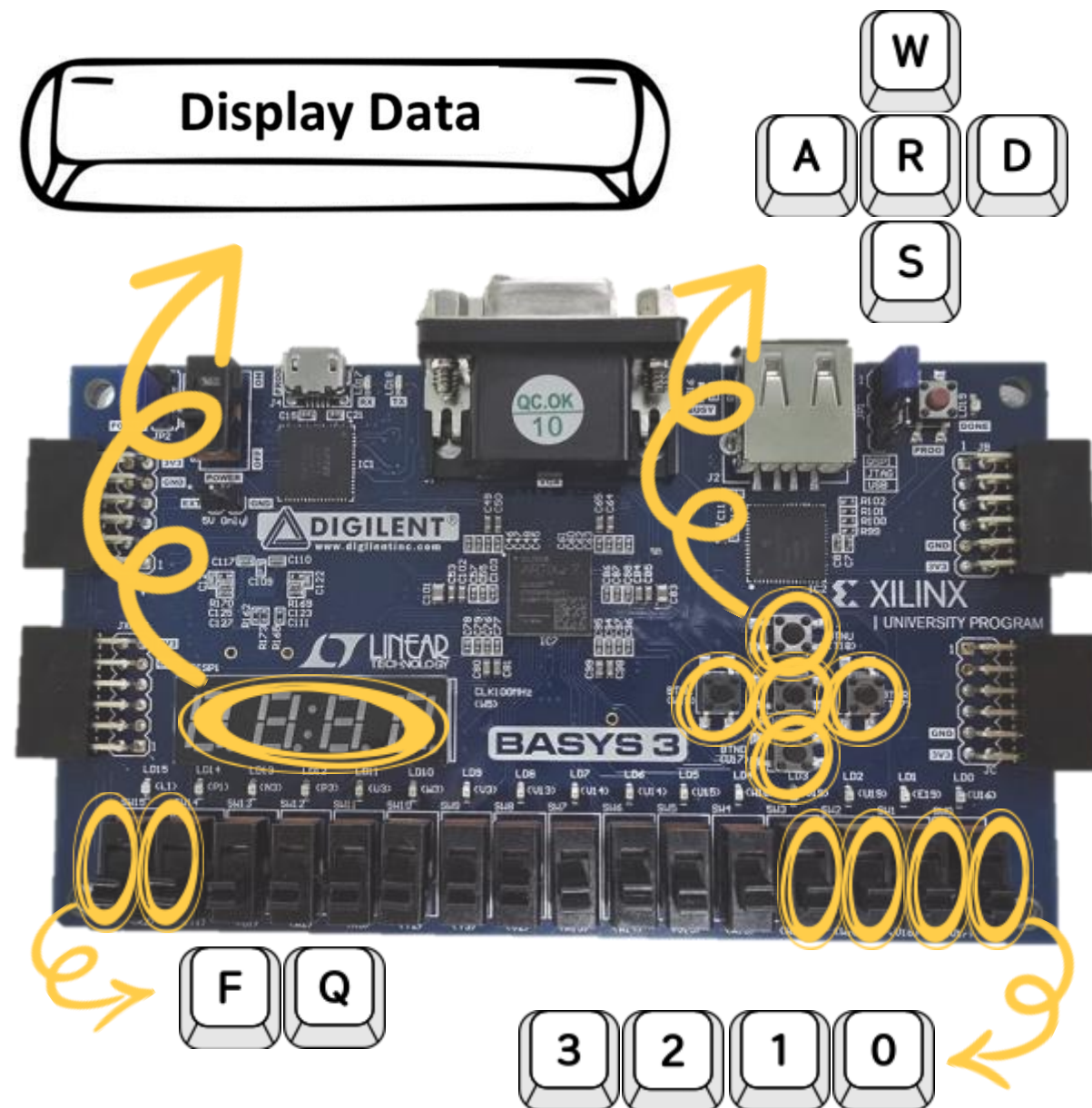
전체 모듈 통합 시스템 설계

EDA TOOL

VIVADO[®]
HLx Editions

EVALUATION BOARD

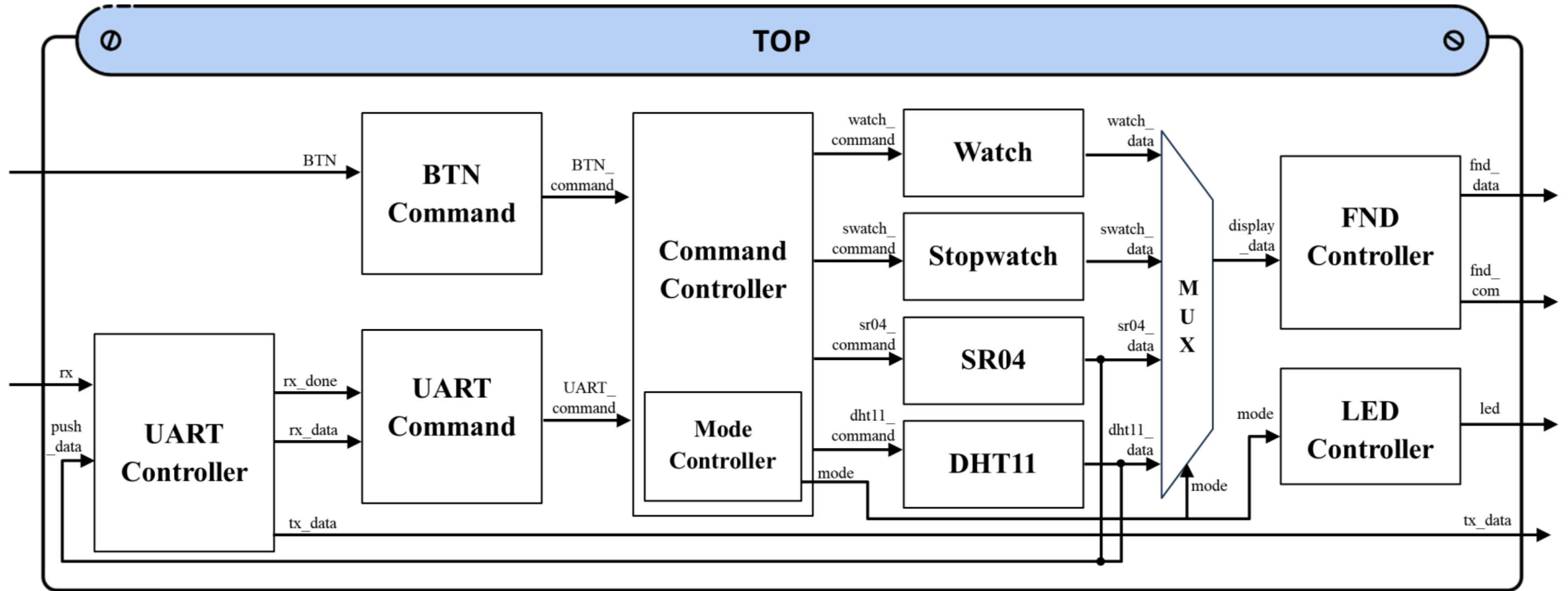
Display Data



SIGNAL

UART	Board	Operation
0	SW0	watch
1	SW1	stopwatch
2	SW2	sr04
3	SW3	dht11
Q	SW14	(watch) set mode, run mode
F	SW15	(watch & stopwatch) display switching
R	btnC	reset
A	btnL	각 control logic
D	btnR	각 control logic
W	btnU	각 control logic
S	btnD	각 control logic
T	-	(watch) TIME:HH:MM:SS:MSEC

TOP Module Block Diagram



UART 통신

신호 처리

명령 생성

기능 제어

출력 제어

Design Overview



START

mode 0



Watch

- btnL : Shift Left
- btnR : Shift Right
- btnU : Set Increase
- btnD : Set Decrease

mode 1



Stopwatch

- btnL : Clear
- btnR : Run/Stop

mode 2



SR04

- btnU : Single update
- btnR : Single / Real
Mode Switching

mode 3



DHT11

- btnU : Single update
- btnR : Single / Real
Mode Switching



초음파 센서

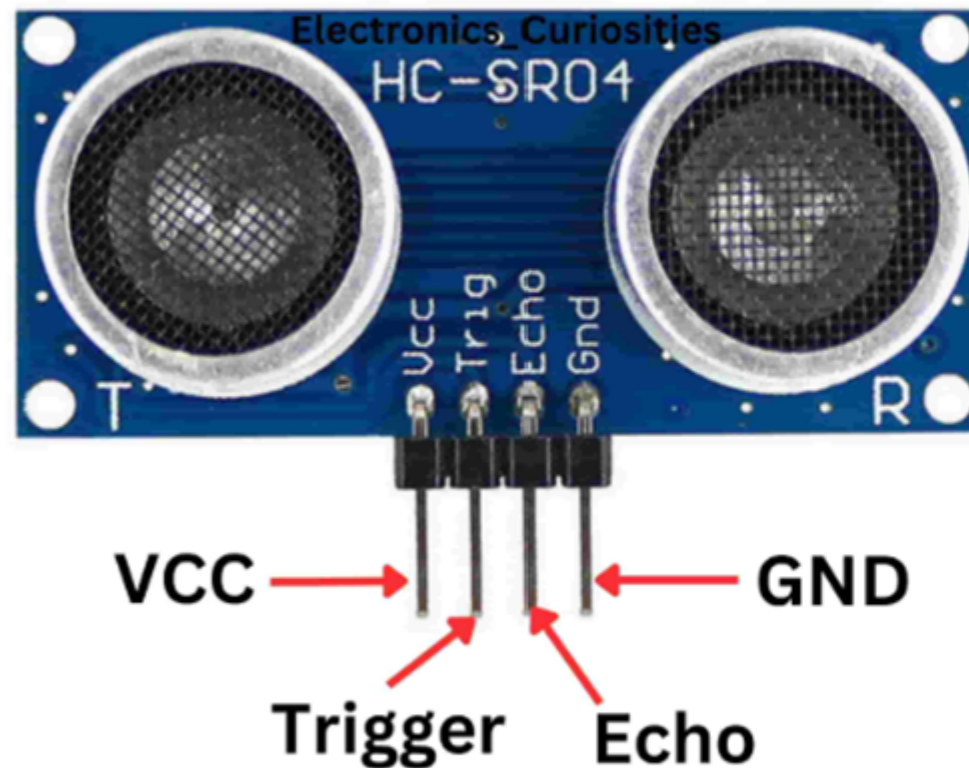
CHAPTER 02

- 구성
- 상세 설계
- 검증
- 고찰
- 시연 영상

구성 - HC_SR04 Sensor (1)



① Ultrasonic Ranging Module HC - SR04



1 HC-SR04 초음파 센서

HC-SR04는 초음파를 이용해 비접촉식으로 거리를 측정하는 센서
2cm에서 400cm(4m)까지 측정 가능하며, 오차는 약 3mm

2 HC-SR04 초음파 센서 사양

최대 동작 전압: 5V, 동작 전류: 15mA, 동작 주파수: 40kHz

측정 각도: 15도, 크기: 45 x 20 x 15mm

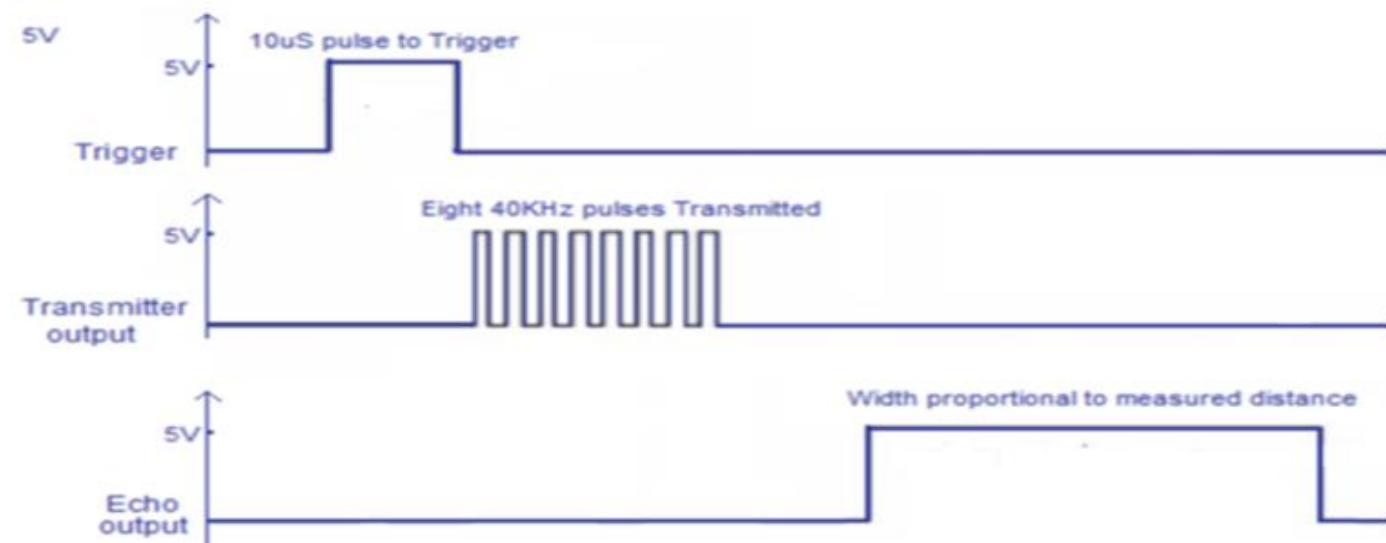
구성 - HC_SR04 Sensor (2)



①

TIMING DIAGRAM

②



$$\text{거리(cm)} = \frac{\text{Echo 펄스 시간}(\mu\text{s})}{58}$$

3

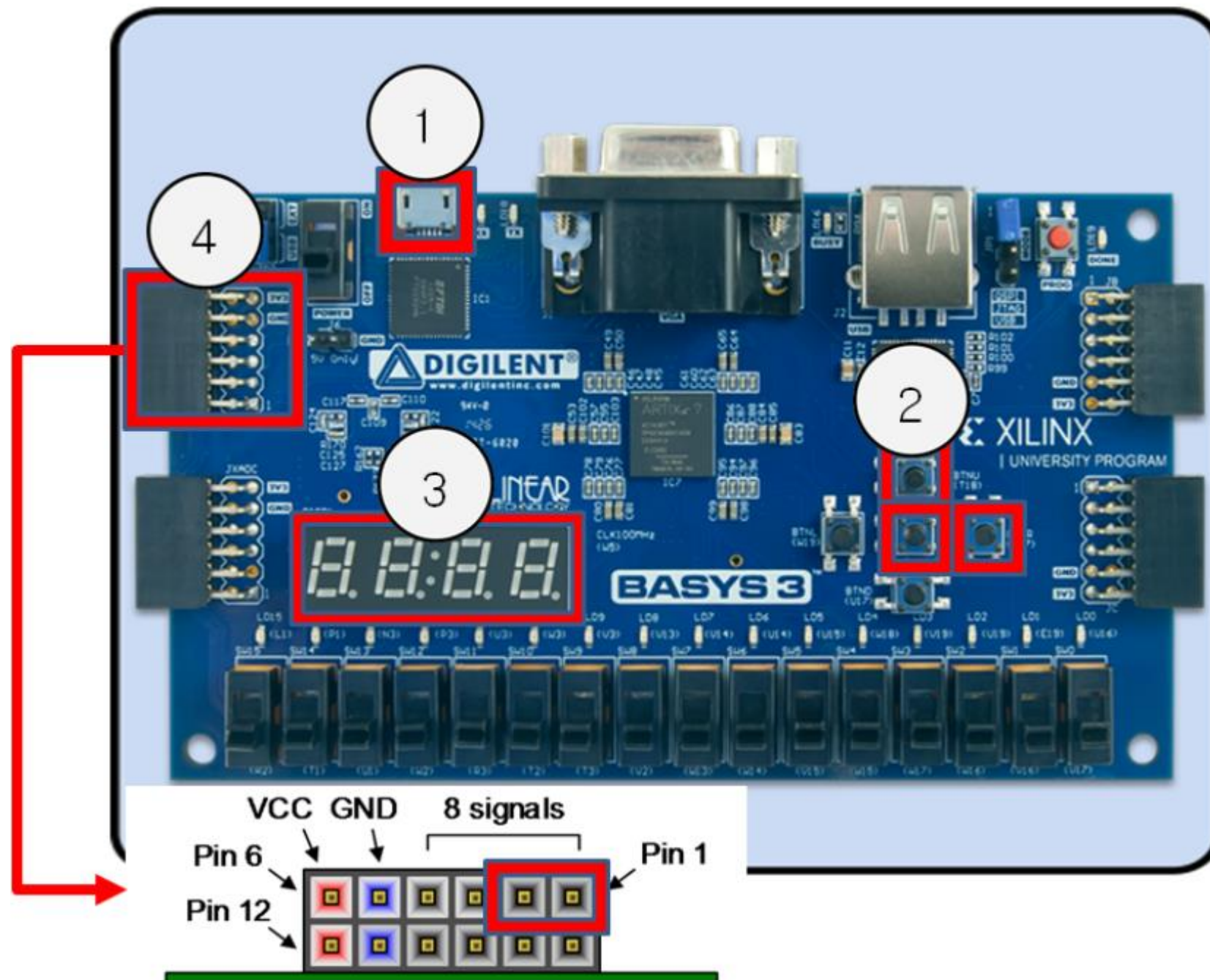
HC-SR04 초음파 센서 작동 원리

트리거(Trigger) 핀에 (10µs) 이상의 신호를 주면, 센서가 8번의 40kHz 초음파를 발사
초음파가 물체에 반사되어 돌아오면, 에코(Echo) 핀에서 신호가 출력
신호가 들어온 시간을 측정해 거리를 계산

구성 - Basys 3 Board I/O



Basys3 Board



1 UART (PC 송수신)

2 BTNU (수동 거리 측정 모드) 버튼을 누를 때마다 거리를 측정

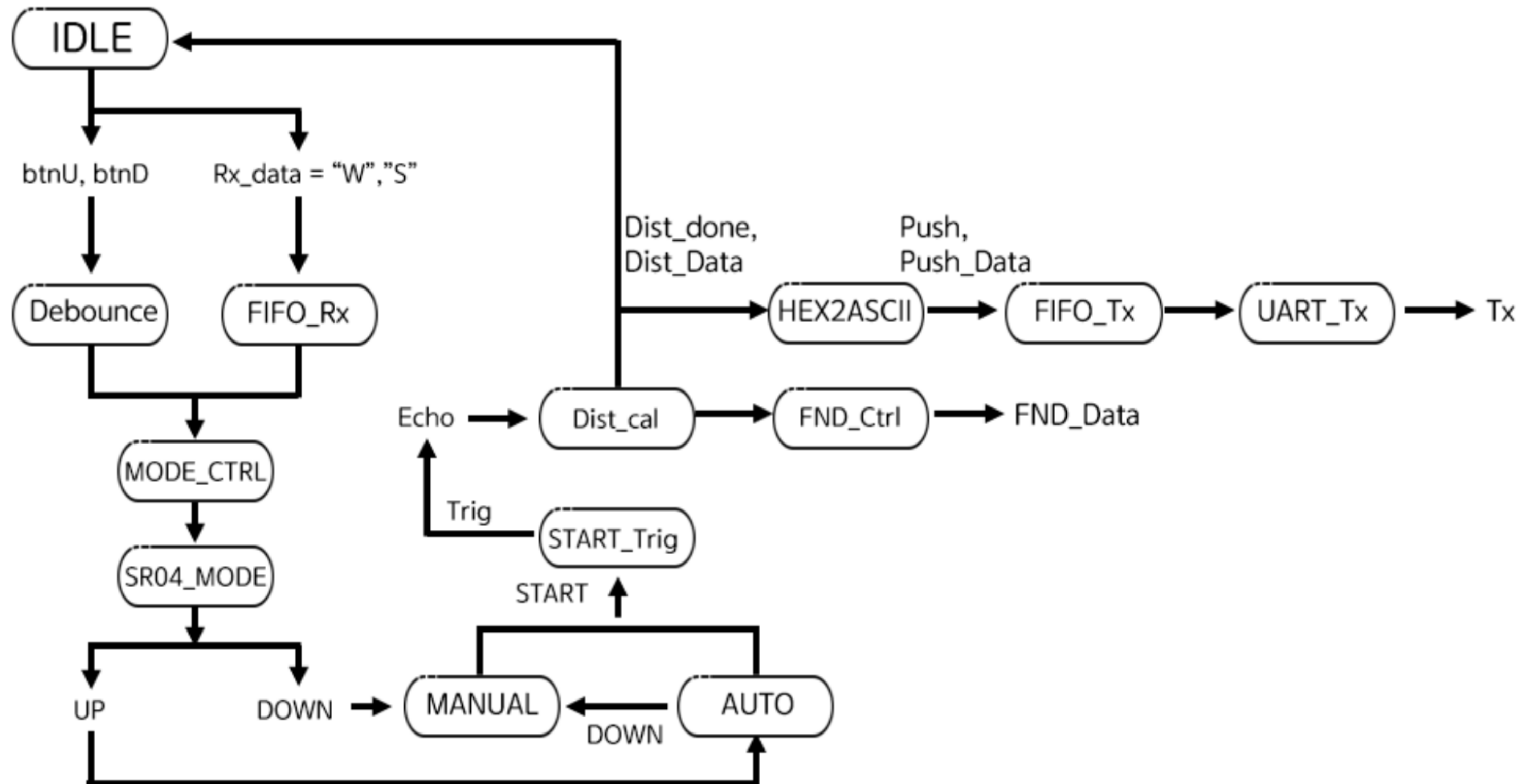
BTND (자동 거리 측정 모드) 버튼을 누르면 1초마다 자동 거리 측정

BTNC (리셋 버튼) - 시스템 초기화

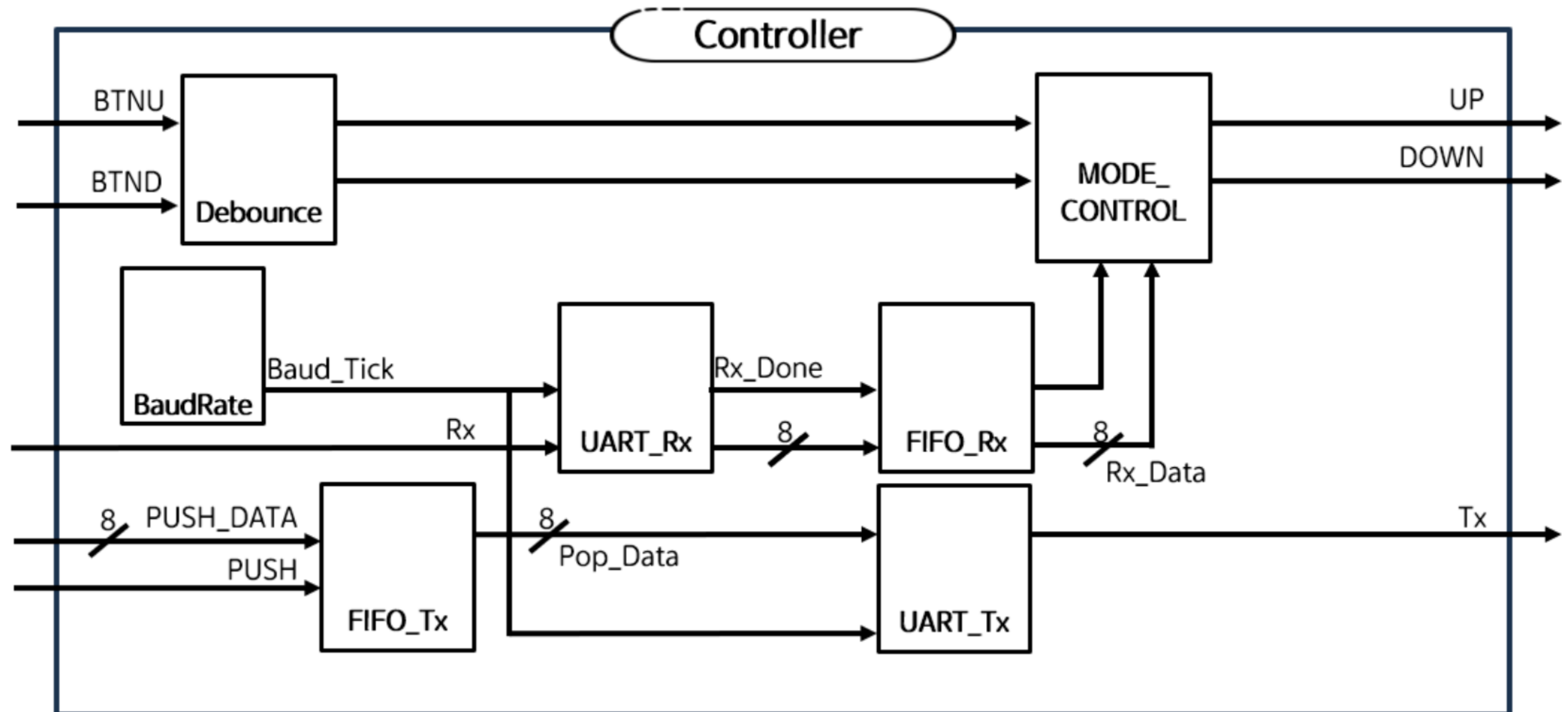
3 FND (측정 거리 표시)

4 초음파 거리 센서
(3.3V, GND, trig(JA1), echo(JA2))

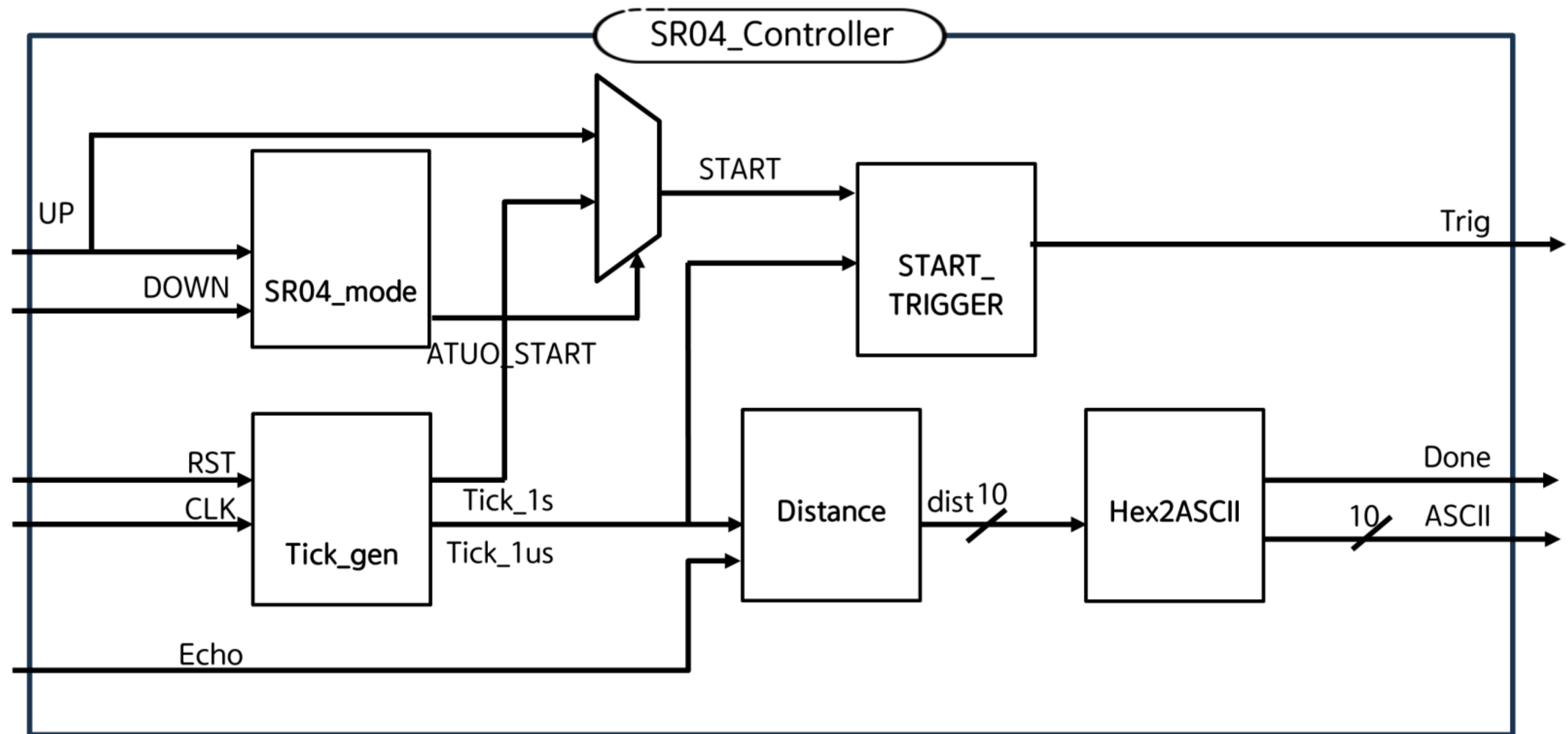
구성 - 동작 진행방식



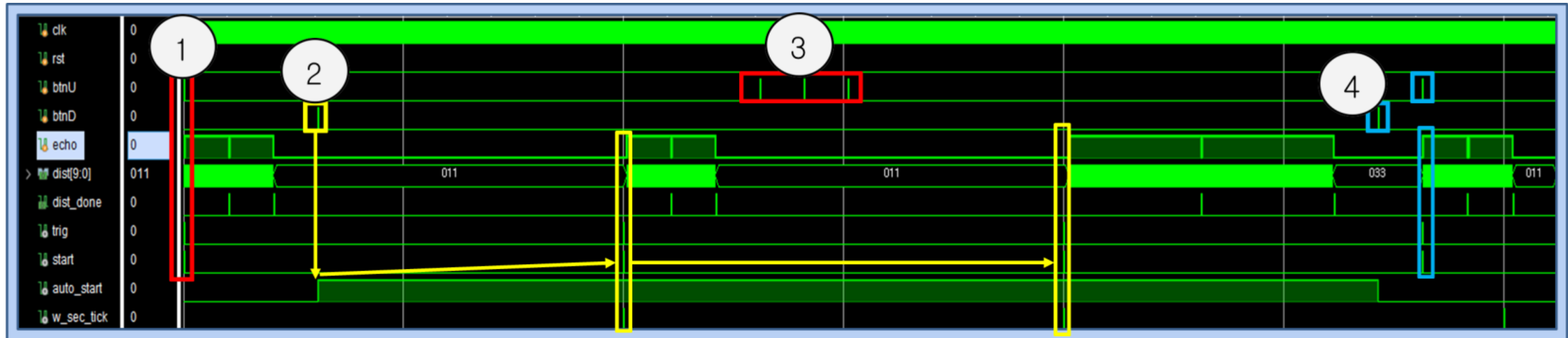
상세 설계 - Controller Block Diagram



상세 설계 - SR04_Controller Block Diagram



검증 - Simulation Result (1)



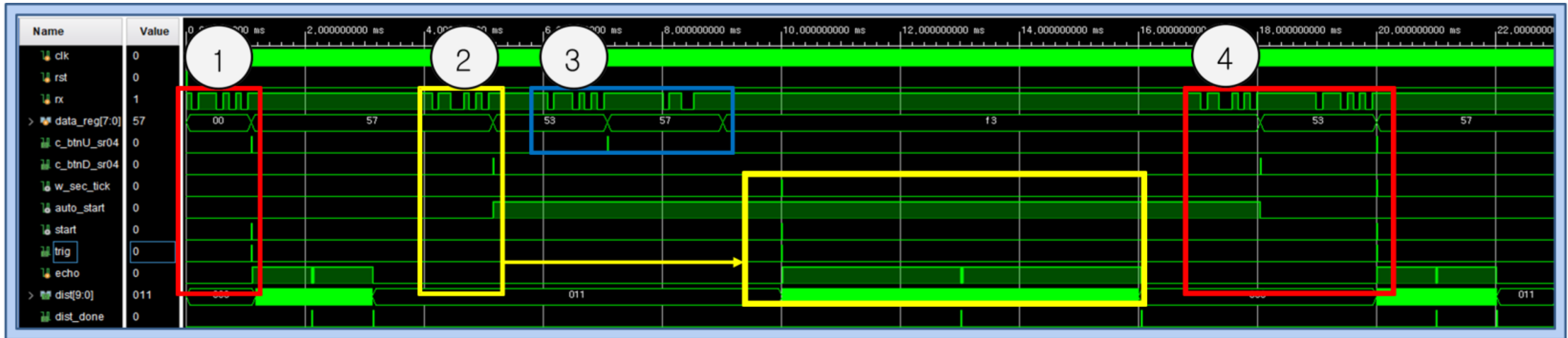
1 수동 모드: btnU high -> start high -> trig high -> echo high => 거리 측정 후 -> done high

2 자동 모드: btnD High-> Auto_start High -> start = w_sec_tick

3 자동 모드 일 때: 수동모드 버튼을 입력해도 start신호 발생하지 않음

4 수동 모드 전환 : btnD high -> auto_start low -> btnU high -> start high

검증 - Simulation Result (2)



1 수동 모드: “W”(8'h57) -> start high -> trig high -> echo high => 거리 측정 후 -> done high

2 자동 모드: “D” (8'h53)-> Auto_start High -> start = w_sec_tick

3 자동 모드 일 때: 수동모드 “W”(8'h57), 어떤 값을 입력해도 start신호 발생하지 않음

4 수동 모드 전환 : btnD high -> auto_start low -> btnU high -> start high

고찰 - Trouble Shooting (1)



문제 인지

ComPortMaster에 송신한 거리
데이터의 마지막 자리가 계속
출력이 되는 문제 발견

원인 파악

Simulation 검증 결과
send_data를 초기화 해주지
않아 끝 값이 계속 그 값으로
유지되는 것을 확인

문제 해결

send_data를 IDLE상태에서
초기화 후 해결

고찰 - Trouble Shooting (2)



문제 인지

Auto 모드에서 btnU 신호가 들어와도 거리 측정이 중단되지 않고, 수동 측정 신호와 자동 측정 신호가 동시에 활성화되는 문제가 있음을 파악.

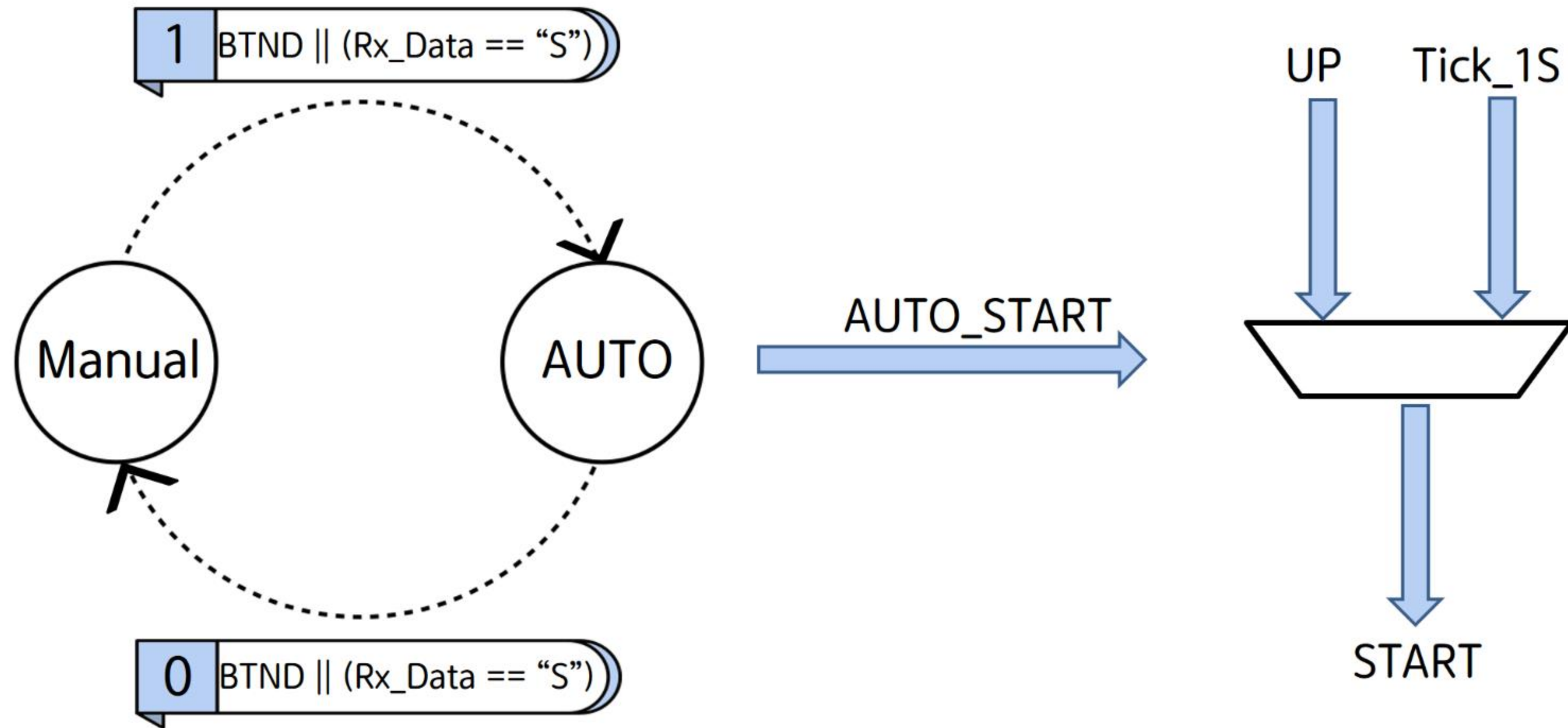
원인 파악

Simulation 검증 결과 설계 로직이 잘못되었다는 것을 인지하고, FSM 와 각 버튼 입력에 따른 동작을 하나씩 따라가며 분석.

문제 해결

Auto 상태일 때 1을 내보내는 신호를 생성하여, sel 신호로 동작하게 로직을 수정.

고찰 - Module (SR04_MODE)



고찰 - 목표 달성 및 결론



1

기존 목표

SR04 거리 측정 센서
데이터를 FND 출력 및
UART로 출력하는 시스템
설계 및 검증하기.

2

추가 기능

PC입력으로 부터의
시스템 제어, 1초마다
자동 측정 기능 추가 구현

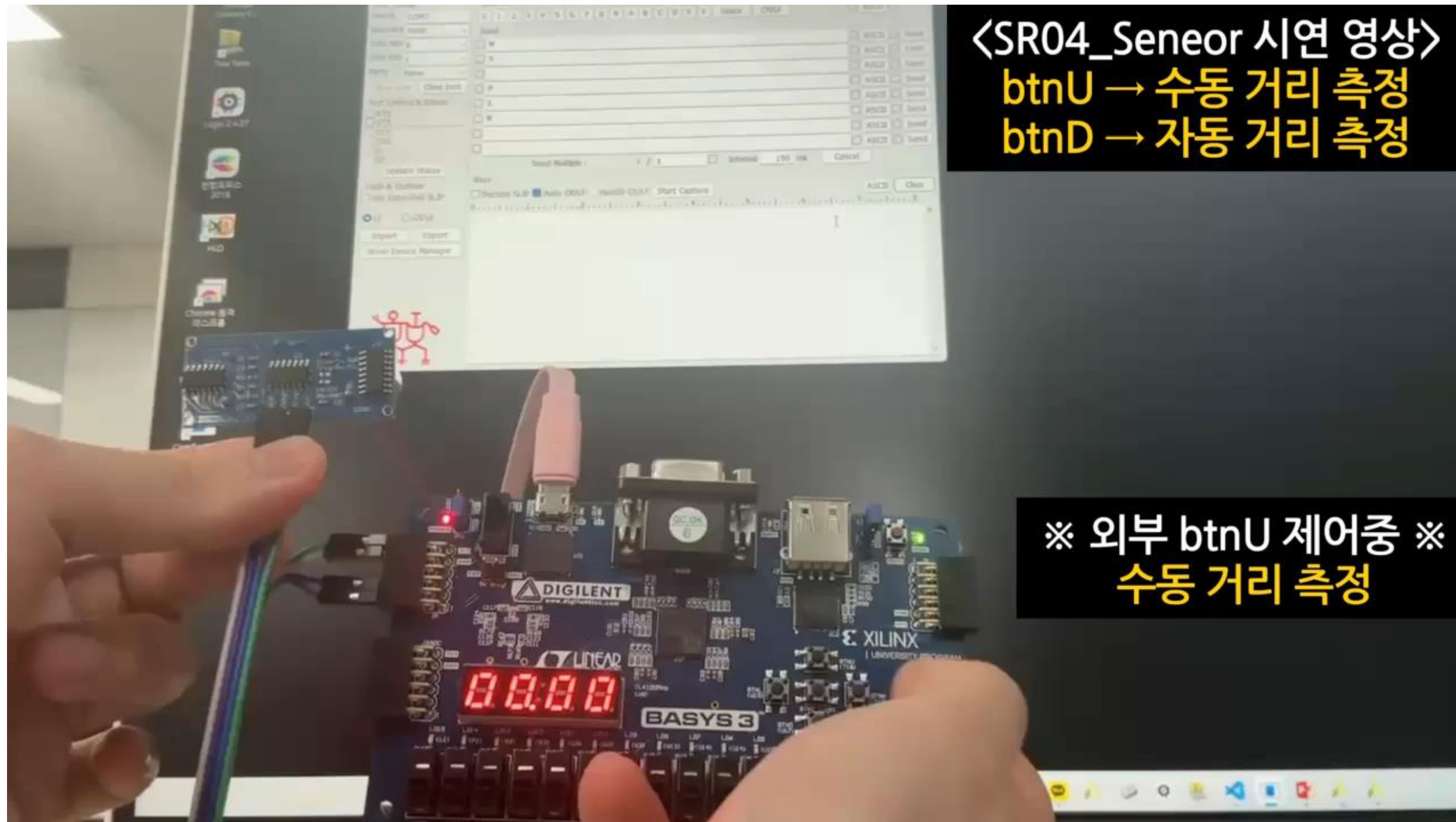
3

개선할 점

사용자 인터페이스(UI)
개선 및 오동작 방지를
위한 예외 처리 로직 보강

Conclusion : SR04 초음파 센서를 이용해 FND와 UART로 거리 데이터를 출력하는 시스템을 구현
PC입력을 통한 시스템 제어, 자동 측정 모드 추가 기능도 정상적으로 동작

시연 영상



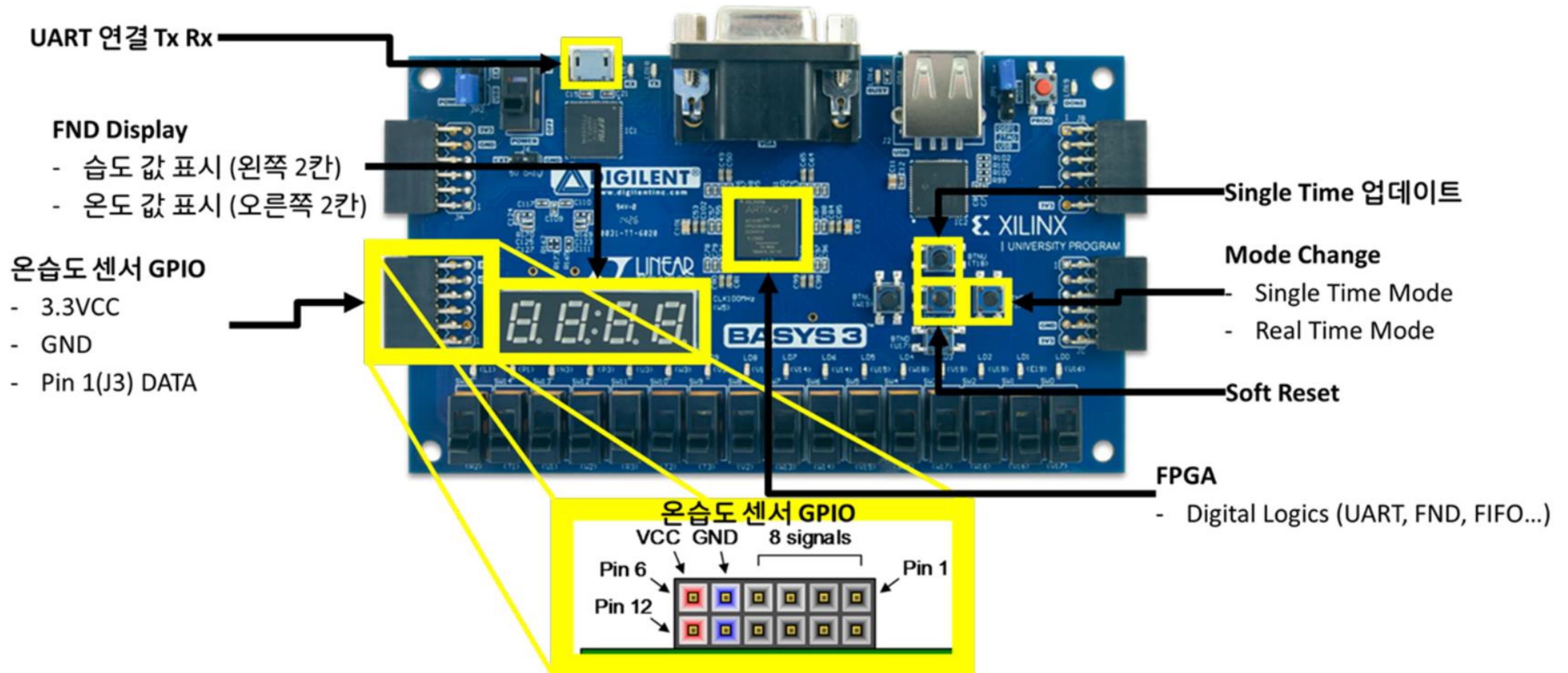


온습도 센서

CHAPTER 03

- 구성
- 상세 설계
- 검증
- 고찰
- 시연 영상

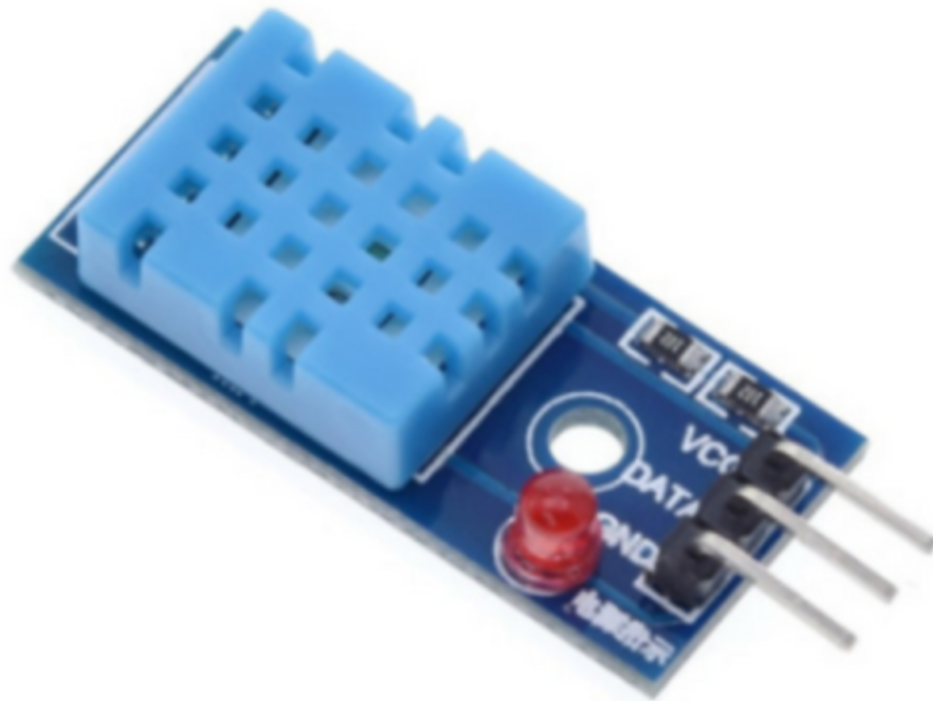
구성 - Basys 3 Board I/O



구성 - DHT11 Sensor



DHT11 Sensor



1 DHT11 Sensor 개요

이 센서는 저항식 습도 센서와 **NTC** 온도 센서로 구성되어 있으며, 독자적인 단일 버스 프로토콜을 통해 **40비트** 데이터를 송수신합니다.

2 주요 Specification

온도 측정 범위 : 0 ~ 50°C ($\pm 2^\circ\text{C}$ 정확도)

습도 측정 범위: 20 ~ 90% RH ($\pm 5\%$ 정확도)

출력 데이터 형식: 40bit (8bit x 5: 습도, 온도, checksum)

동작 전압: 3.3V ~ 5.5V

통신 방식: 단일 데이터 라인 (Single-Wire Serial)

응답 속도: 약 10초 주기

센서 구조: 저항식 습도센서 + NTC (Negative Temperature Coefficient) Thermistor

*(섭씨 25도씨 가정)

구성 - UART Commands



Commands

Send

☒	R
☐	T
☐	M
☐	
☐	
☐	
☐	
☐	
☐	

Send Multiple : /

TRH:72% T:30C

TRH:72% T:30C

TRH:72% T:30C

TRH:72% T:30C

1 R (RESET Command)

R을 보내면 각 모듈을 리셋 할 수 있습니다.
Center Button과 동일한 기능입니다.

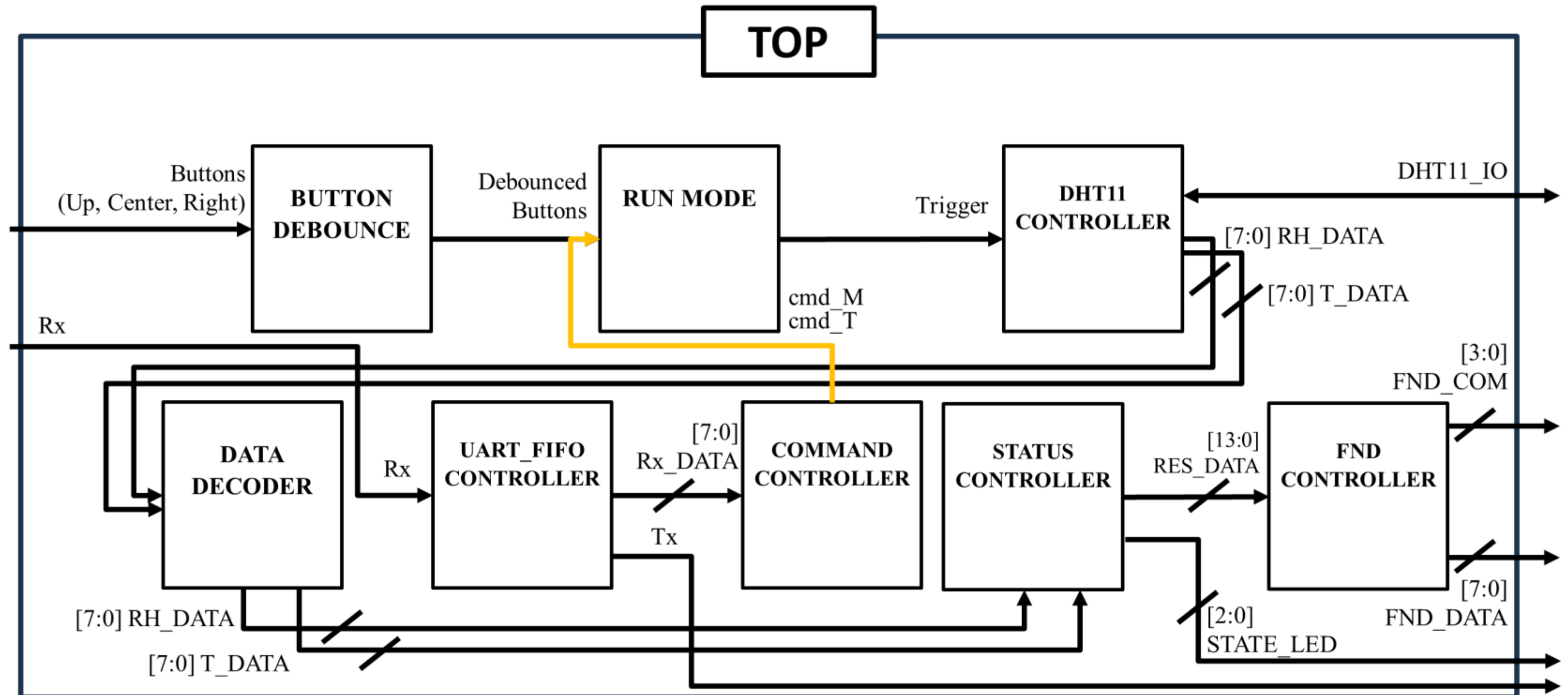
2 T (Track Command)

T를 보내면 Single Time update를 하여 온도 및 습도 값을 업데이트 할 수 있습니다.
Up Button과 동일한 기능입니다.

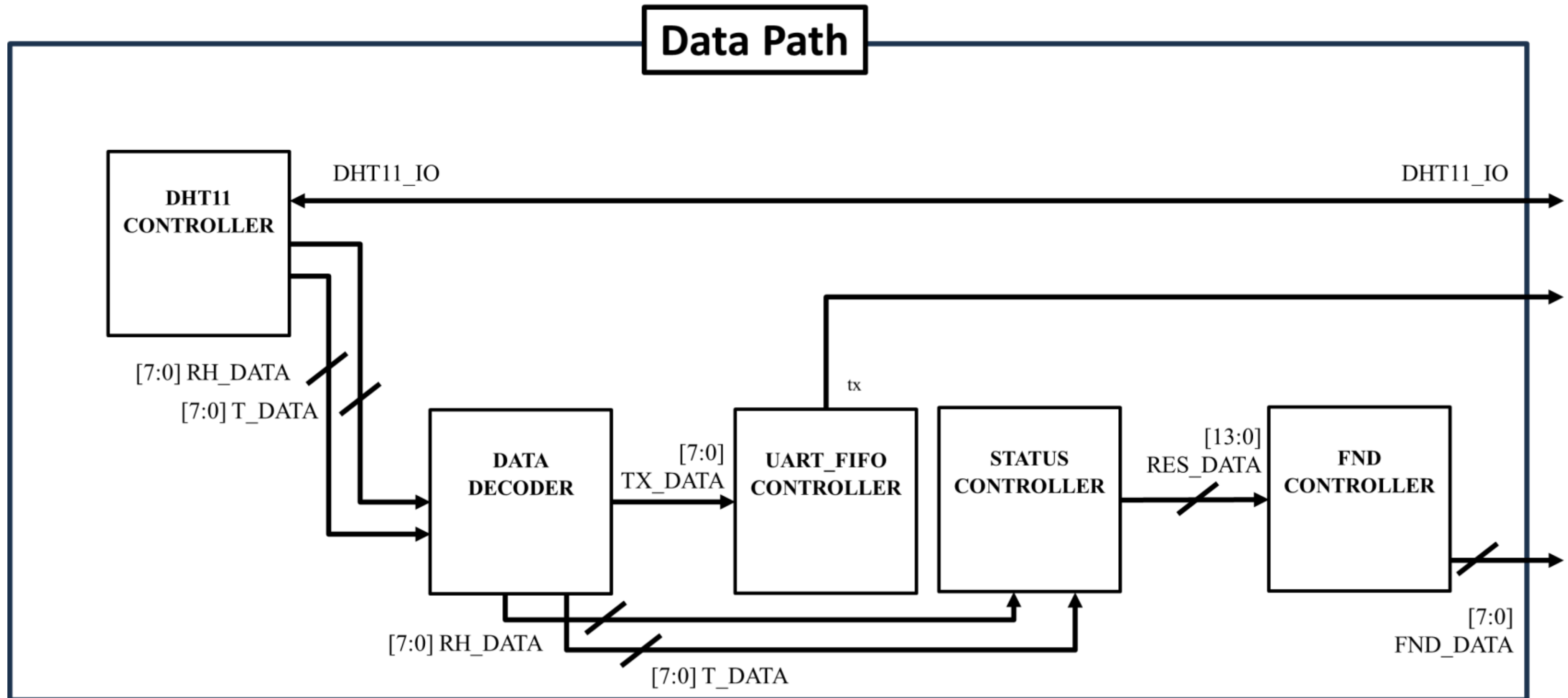
3 M (Mode Command)

Single Time과 Real Time 모드 간의 모드 변경을 가능하게 합니다.
Right Button과 동일한 기능입니다.

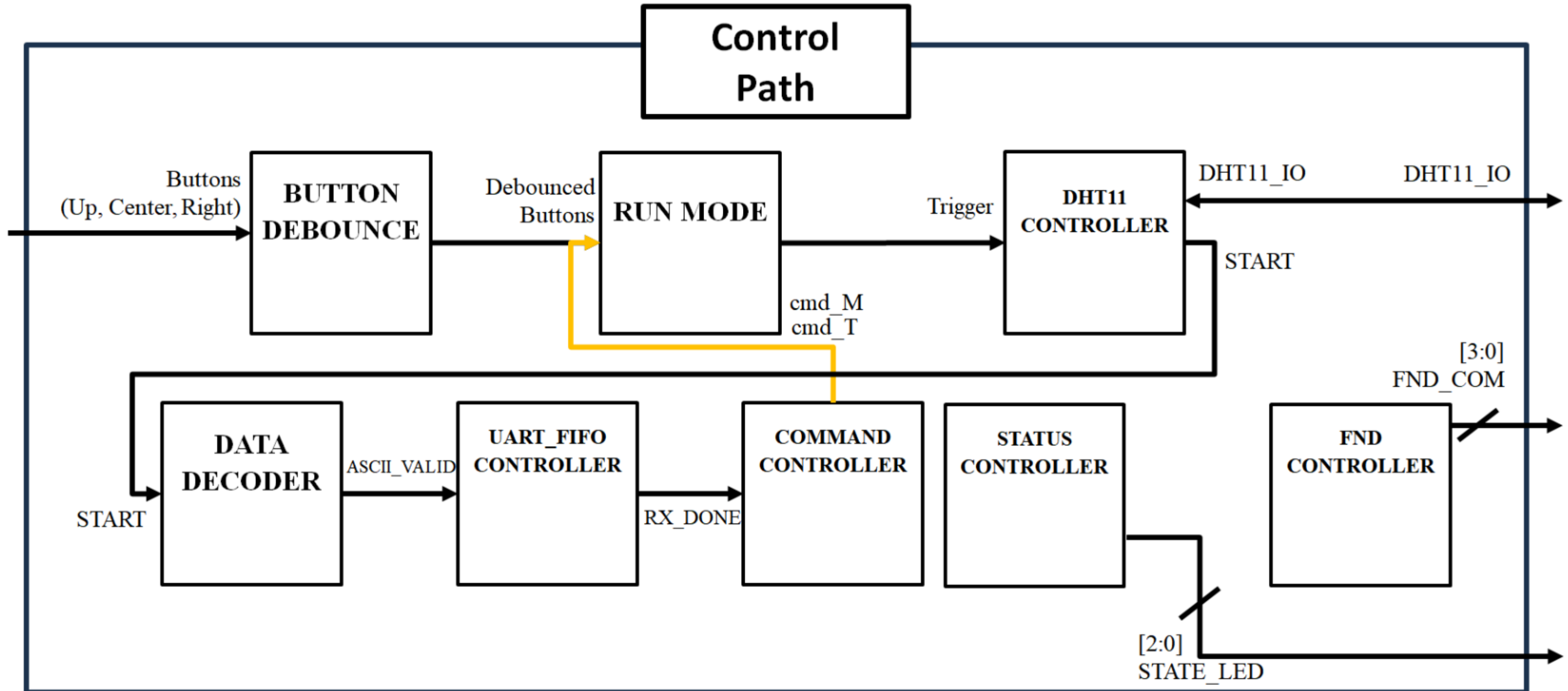
상세 설계 - TOP Module Block Diagram



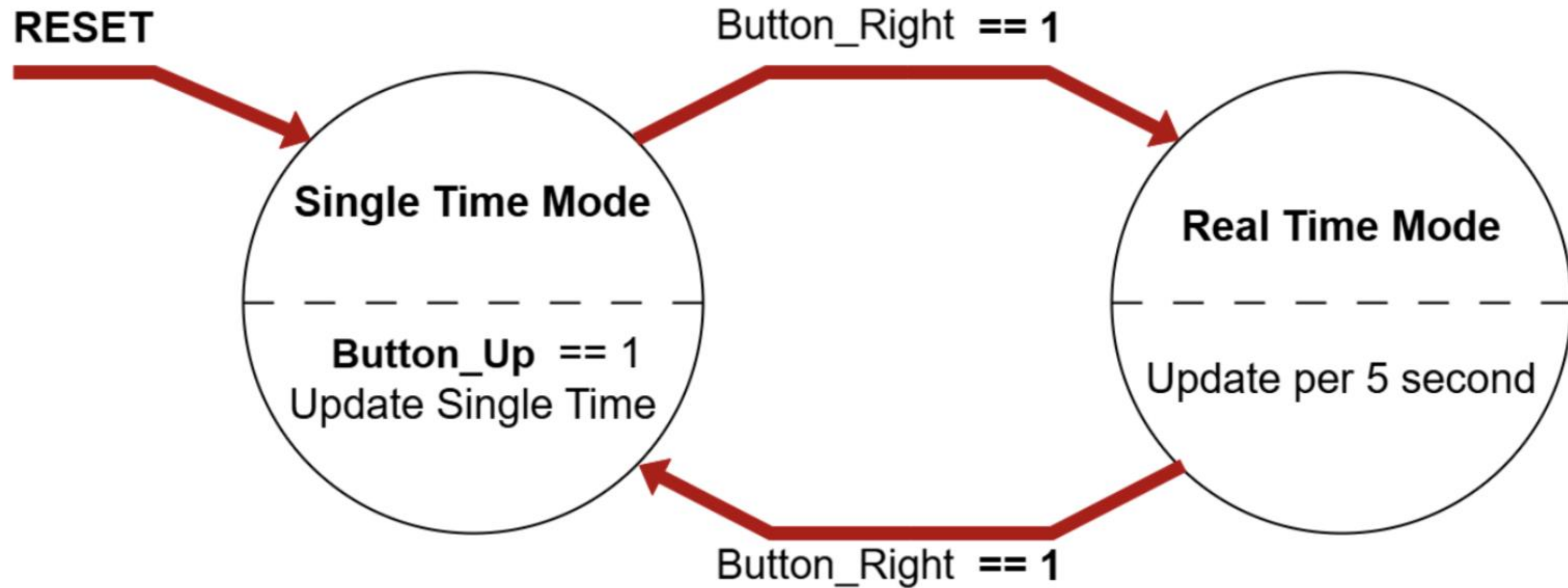
상세 설계 - Data Path Block Diagram



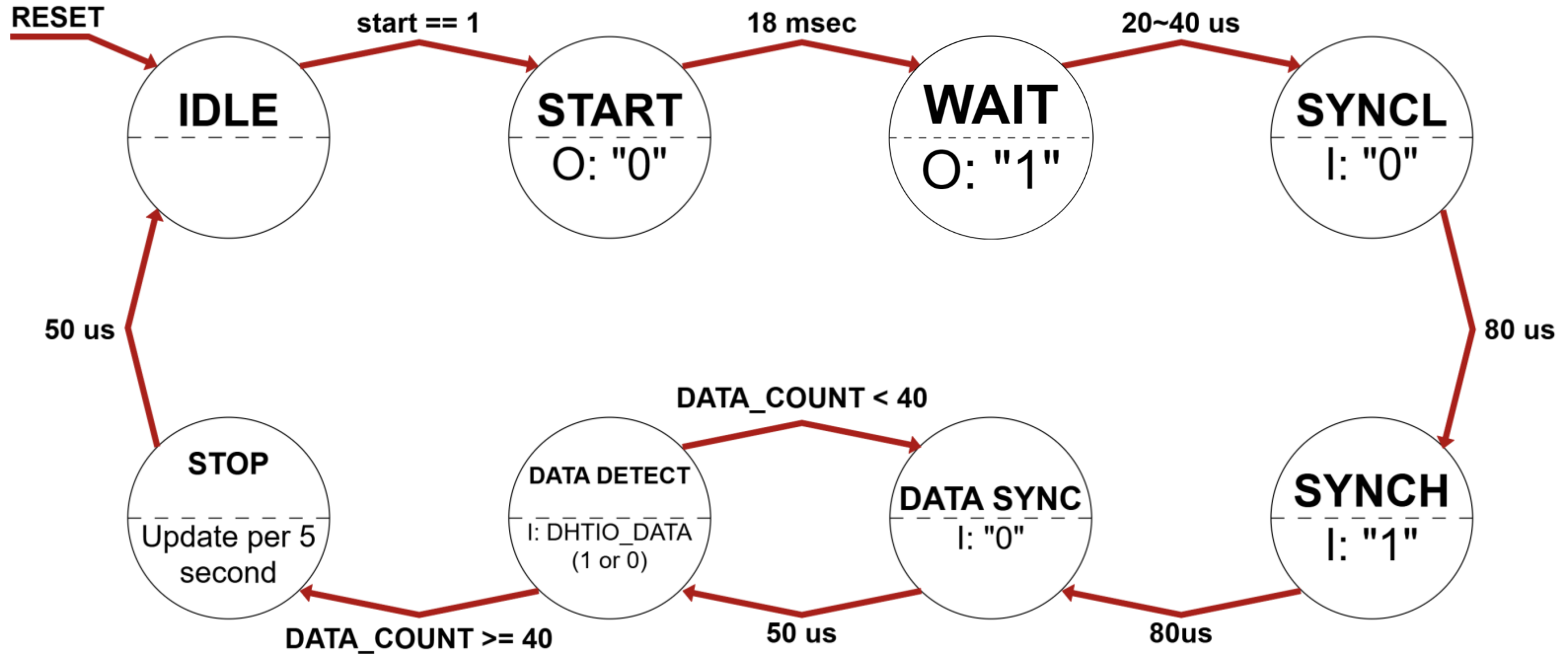
상세 설계 - Control Path Block Diagram



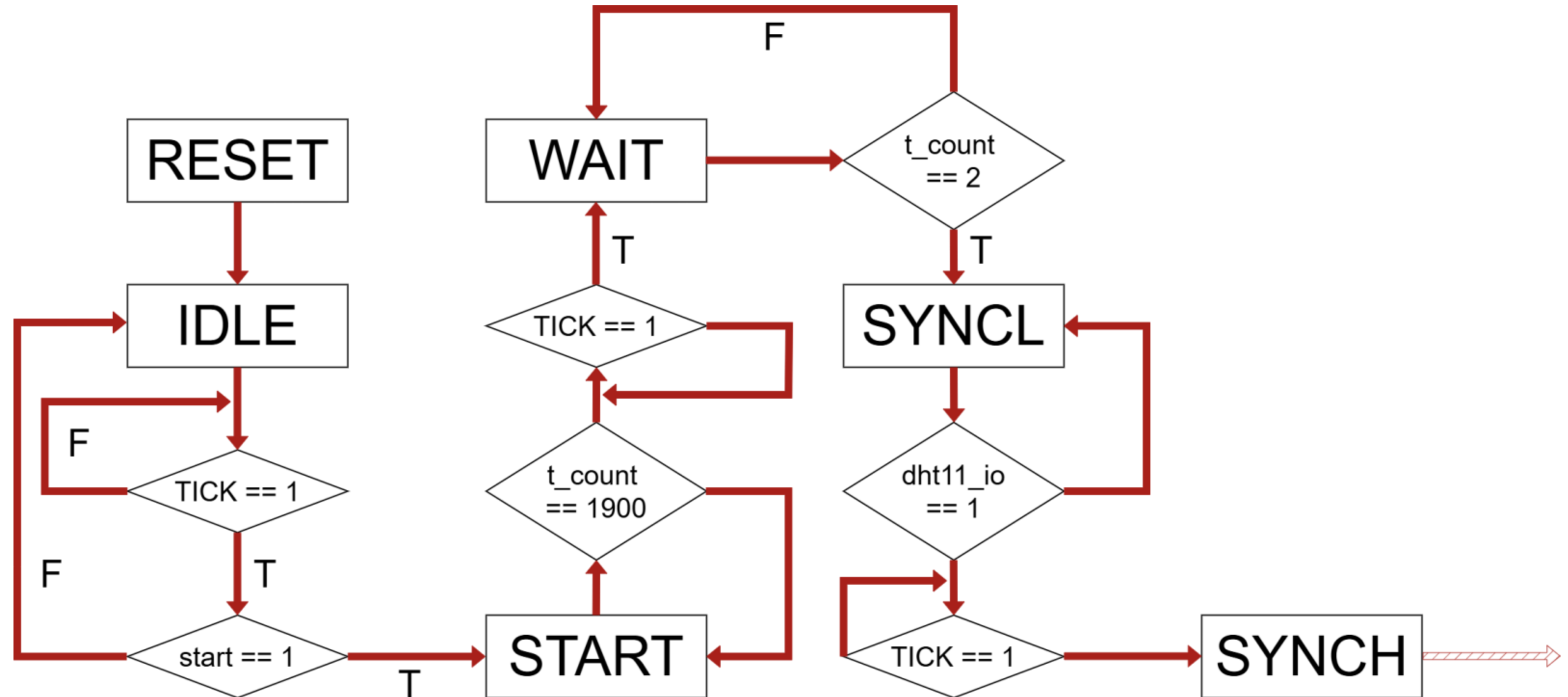
상세 설계 - Mode 전환 FSM



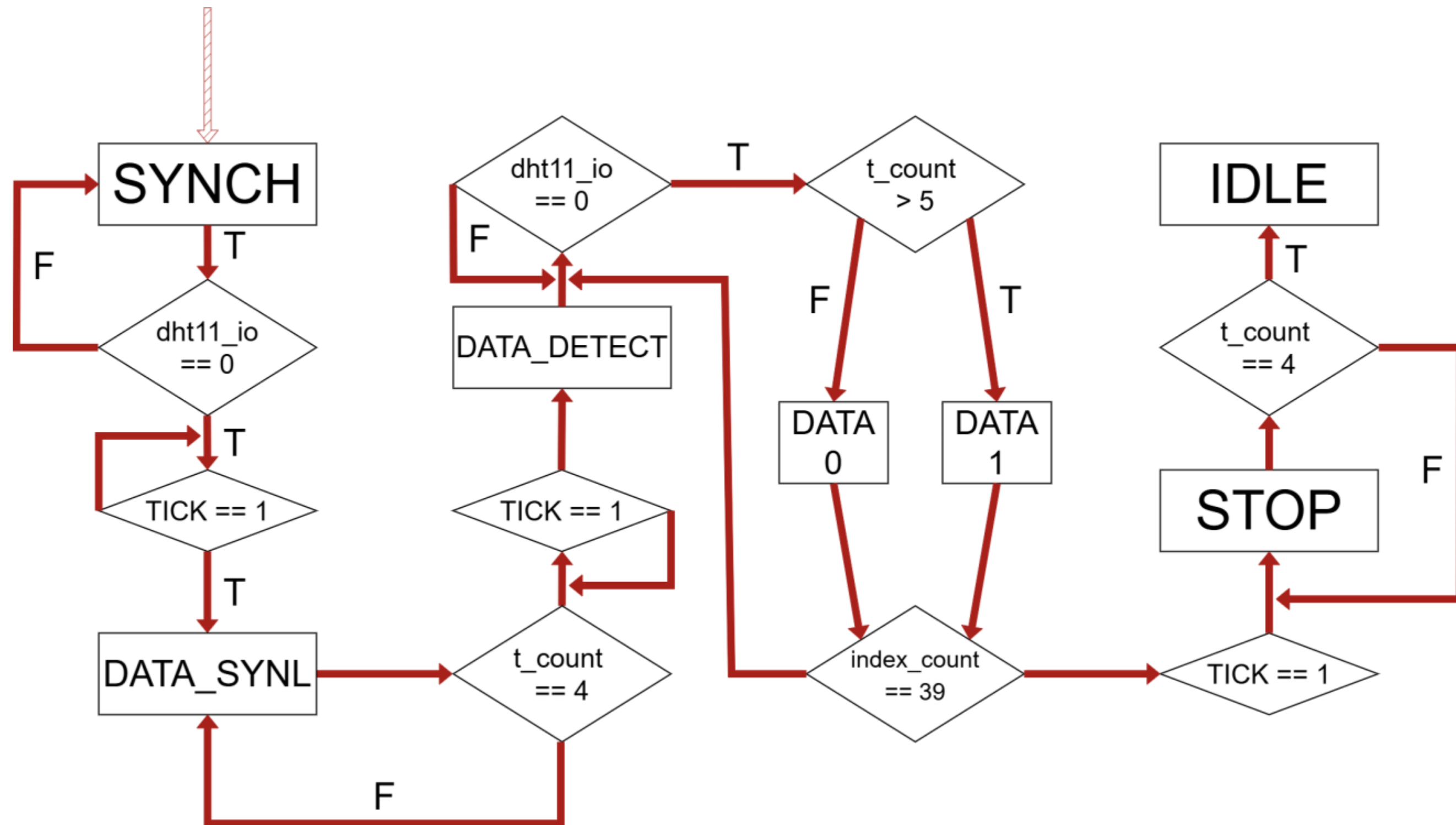
상세 설계 - DHT11 Sensor Controller FSM



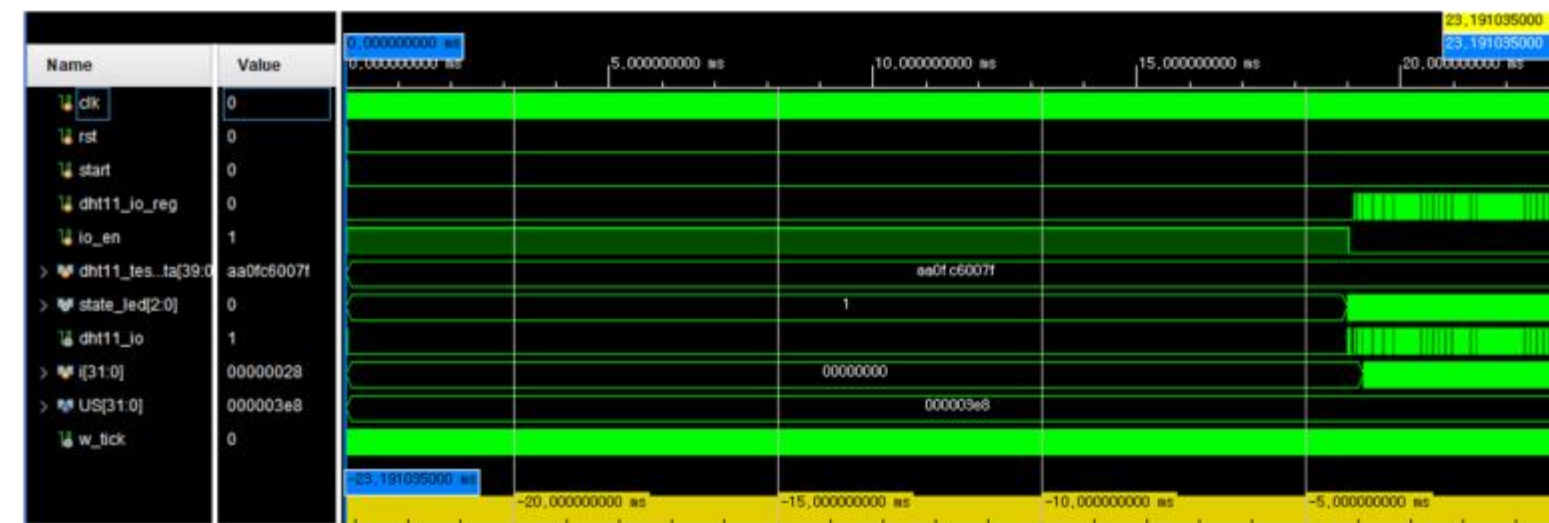
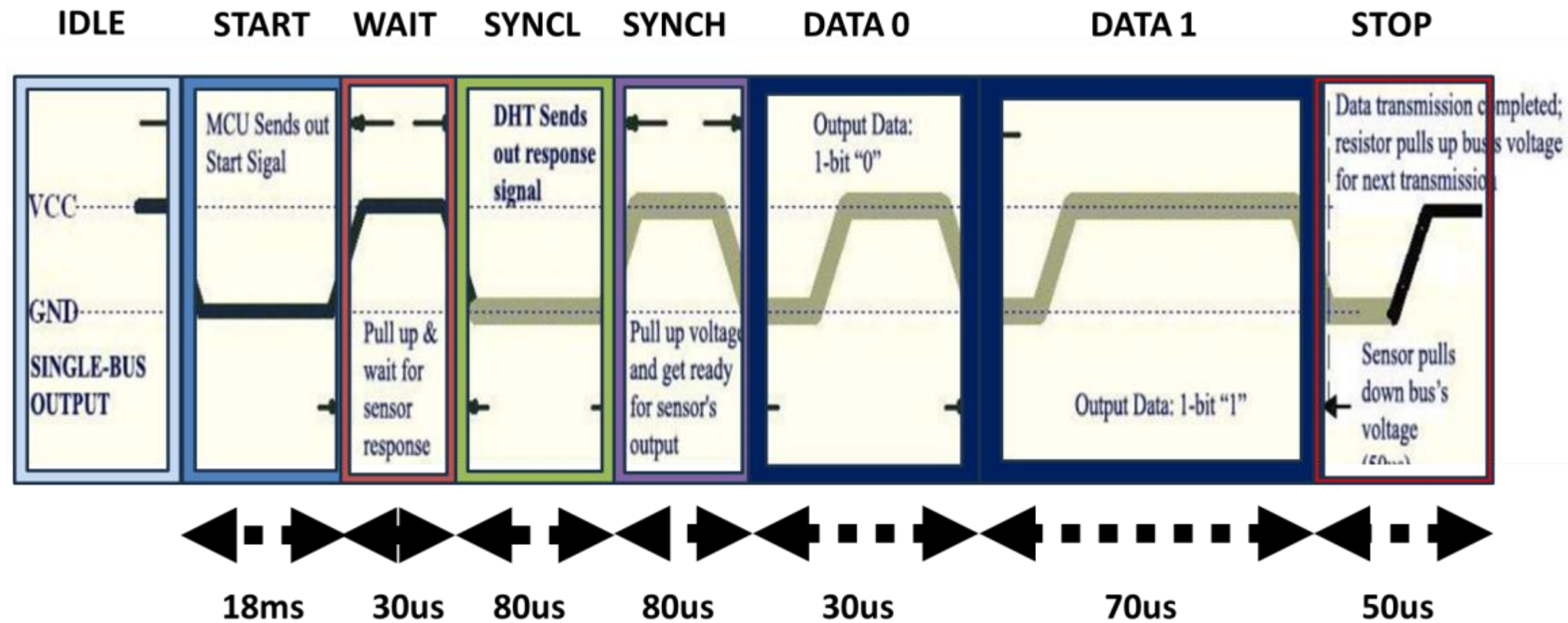
상세 설계 - DHT11 Sensor Controller ASM (1)

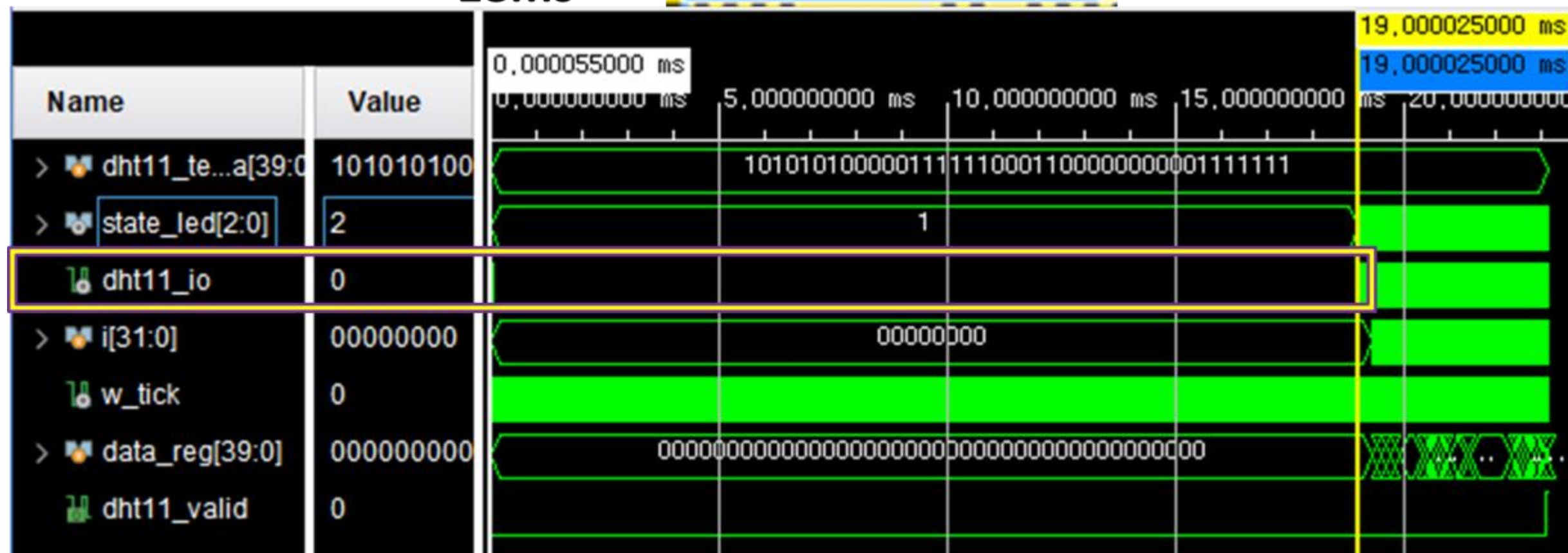


상세 설계 - DHT11 Sensor Controller ASM (2)

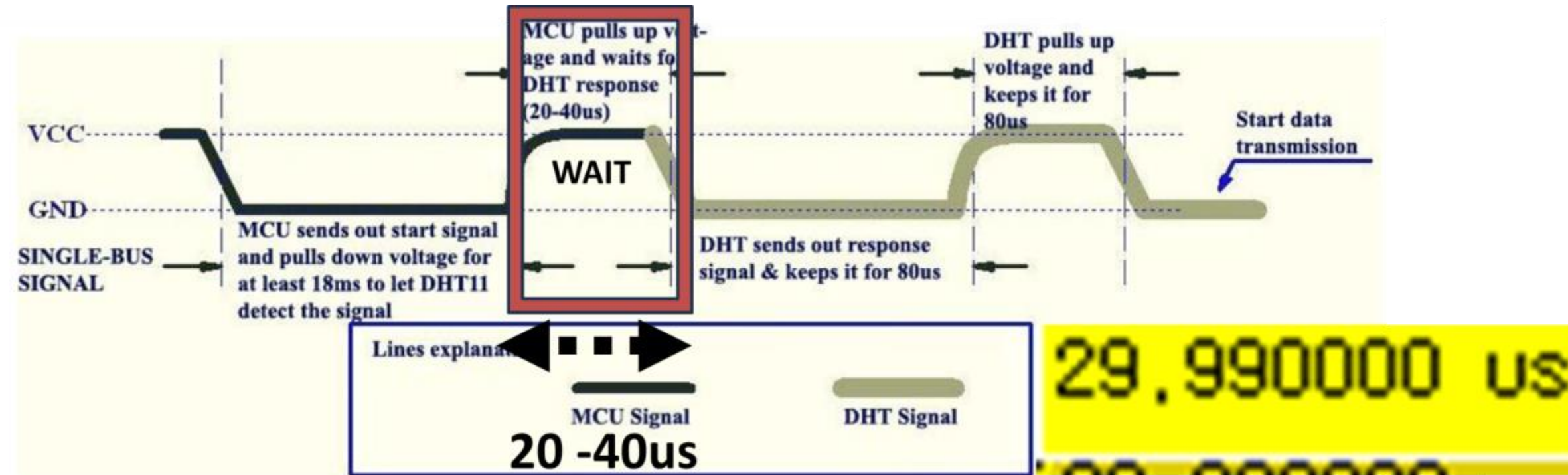


검증 - Timing 비교



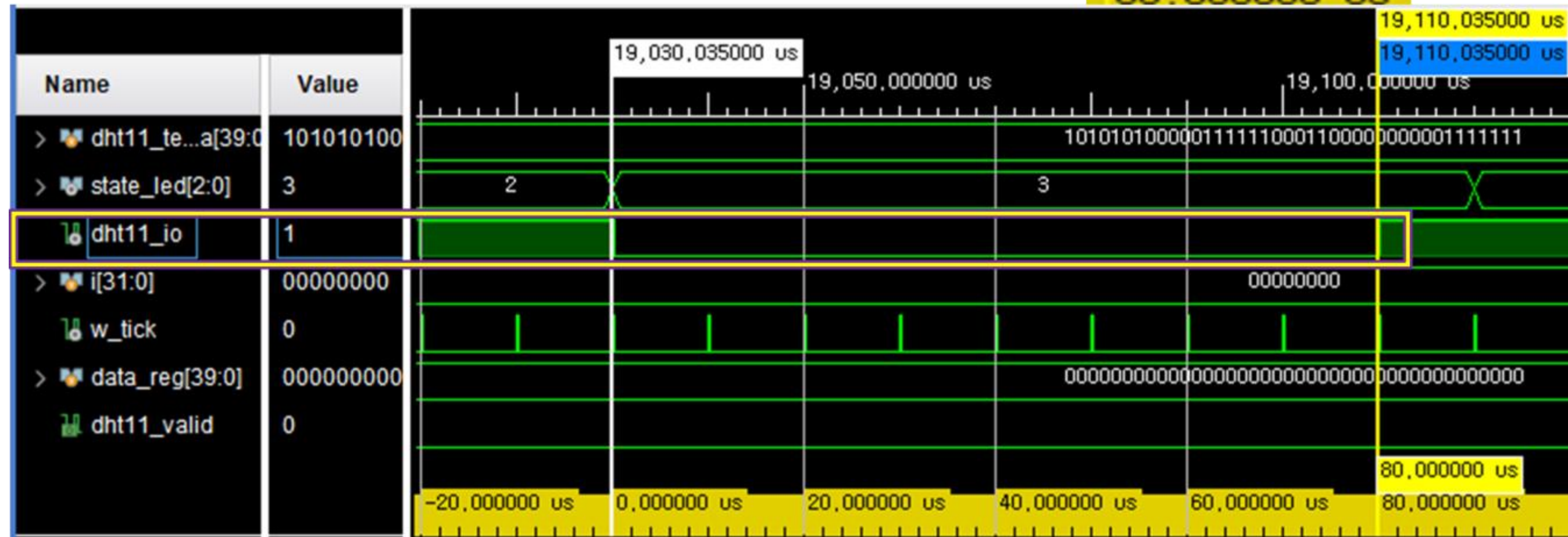
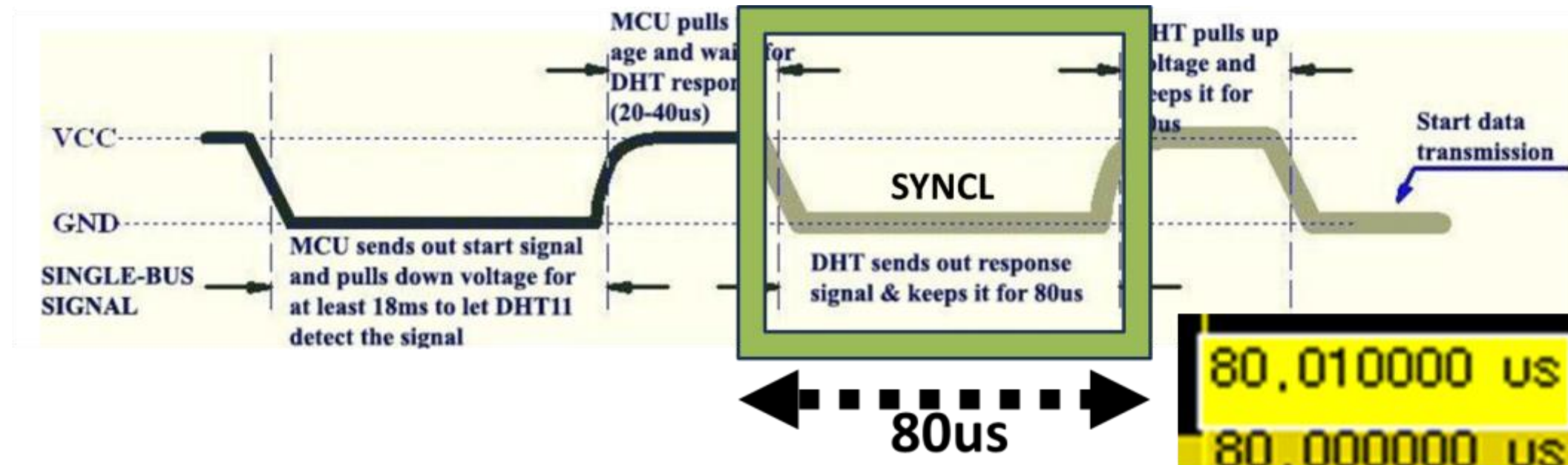


검증 - WAIT Timing 비교

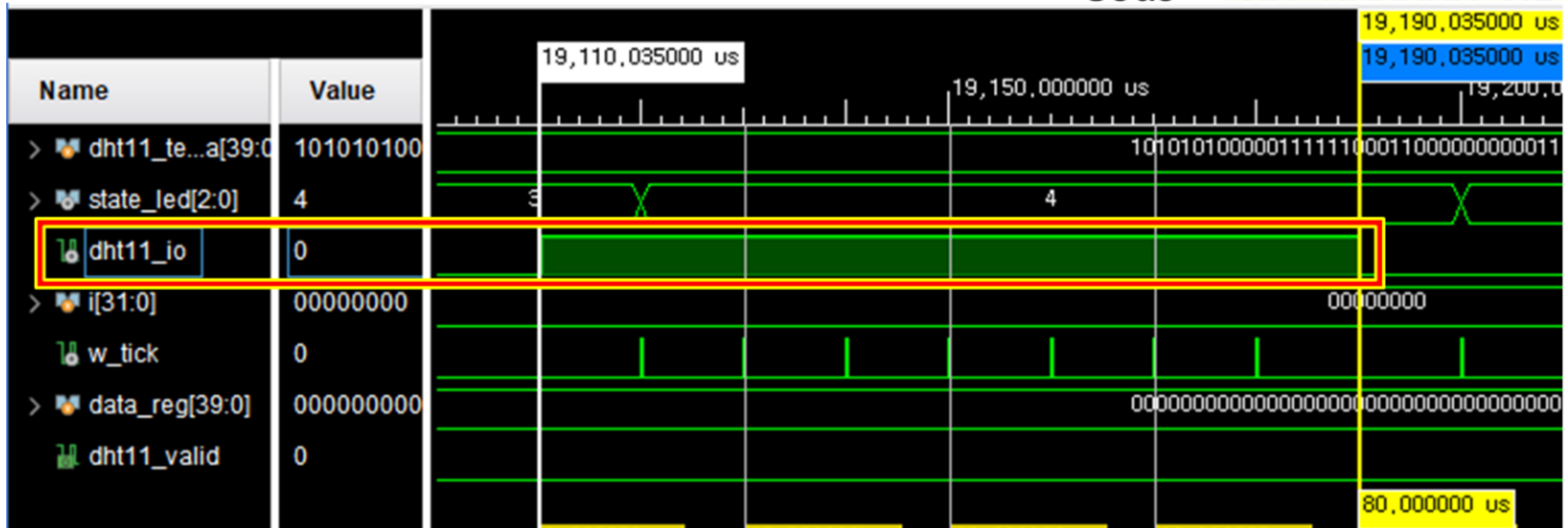
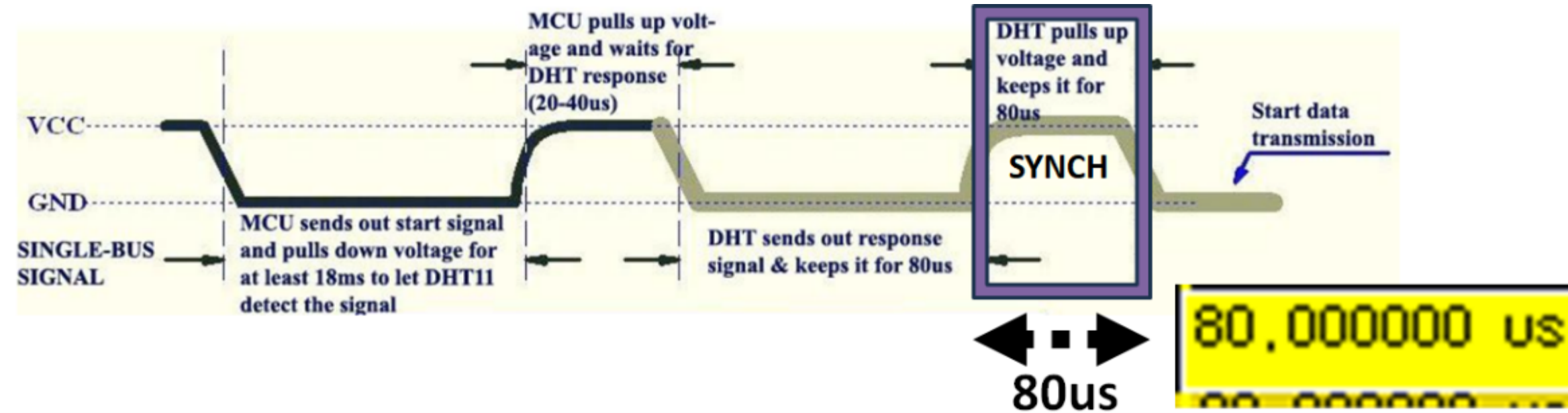


Name	Value	19,000,025000 us	19,030,035000 us
> dht11_te...a[39:0]	101010100		1010101000001111100
> state_led[2:0]	3	1	2
dht11_io	0		
> i[31:0]	00000000		0000
w_tick	0		
> data_reg[39:0]	000000000		00000000000000000000
dht11_valid	0		

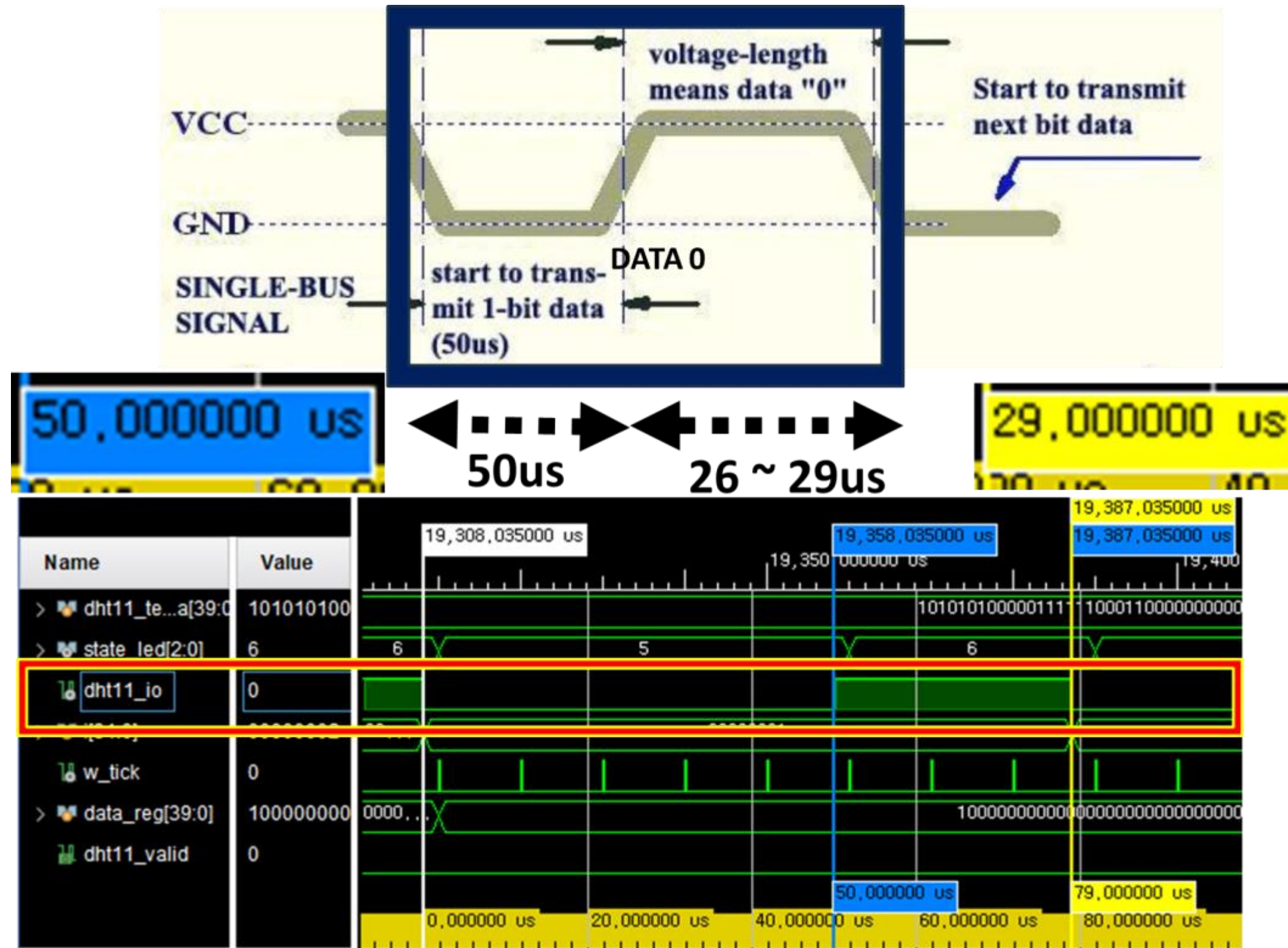
검증 - SYNCL Timing 비교



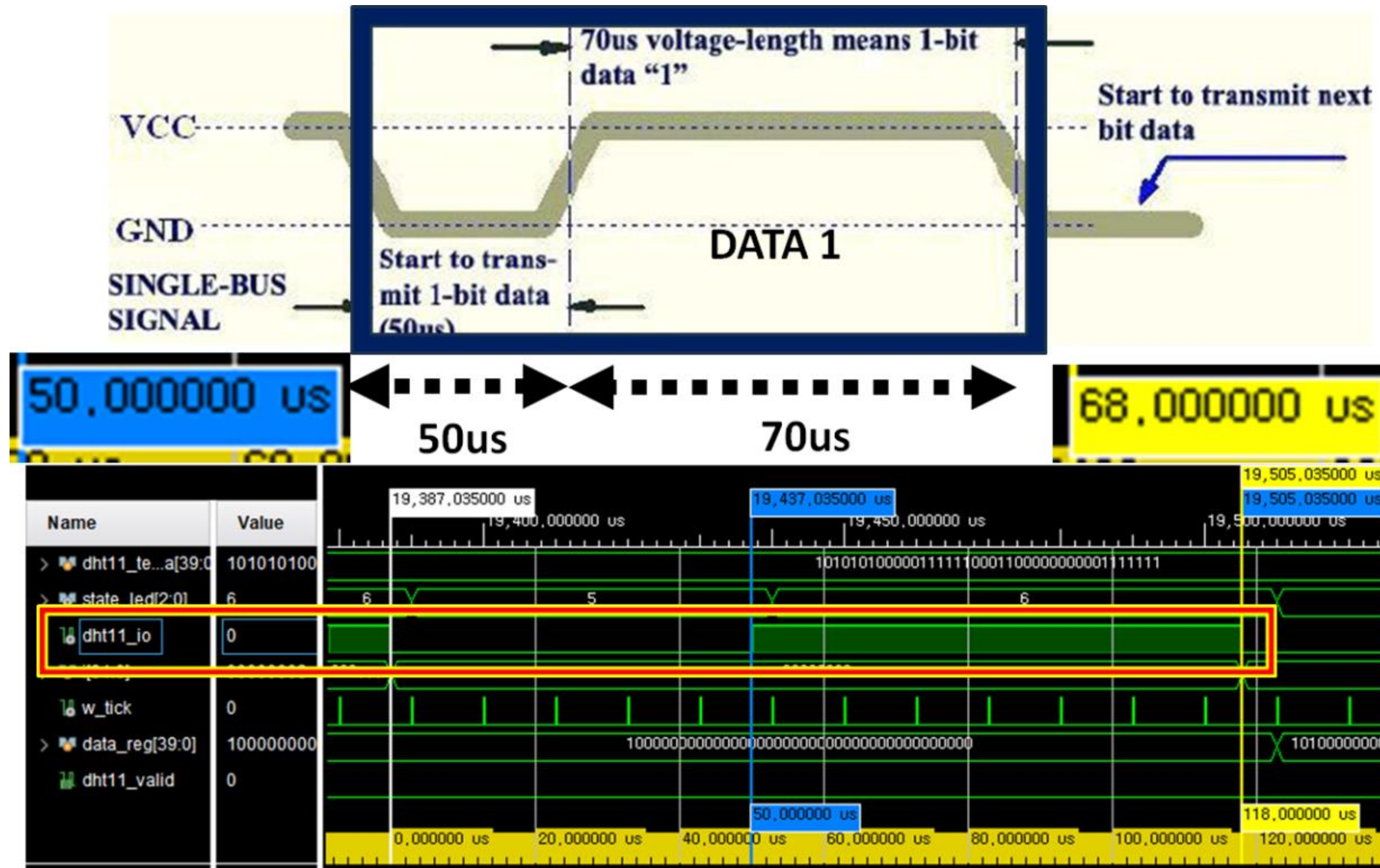
검증 - SYNCH Timing 비교



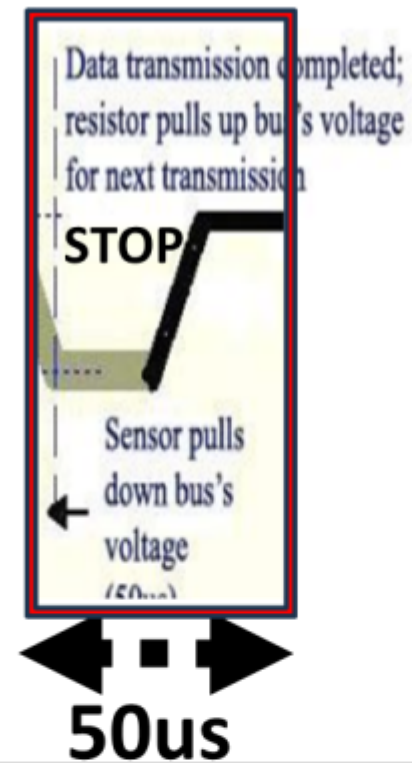
검증 - DATA 0 Timing 비교



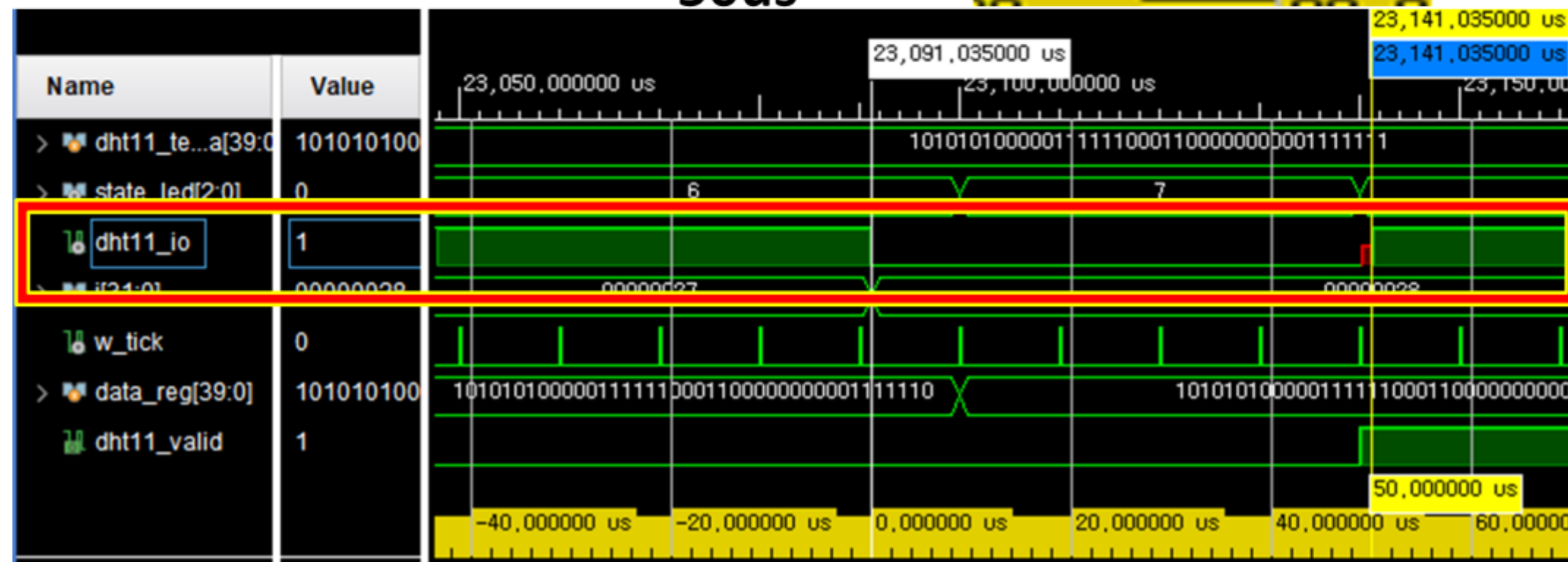
검증 - DATA 1 Timing 비교

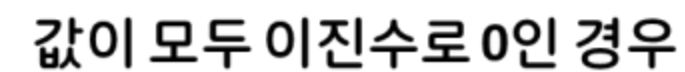
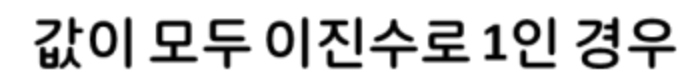


검증 - STOP Timing 비교

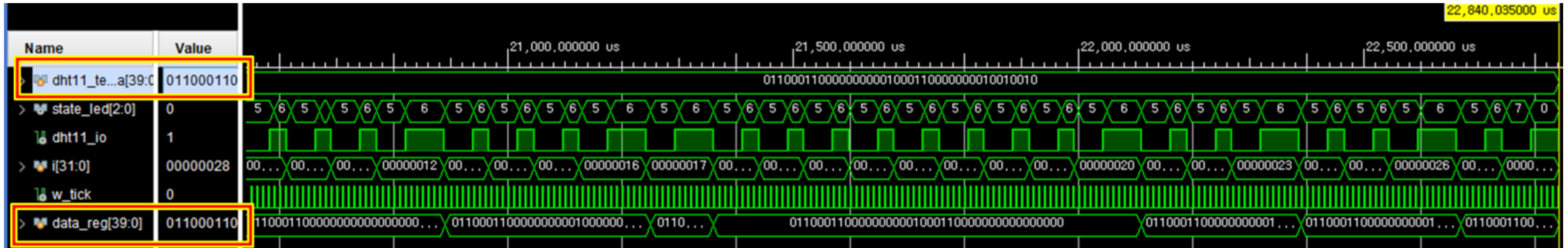


50,000000 us

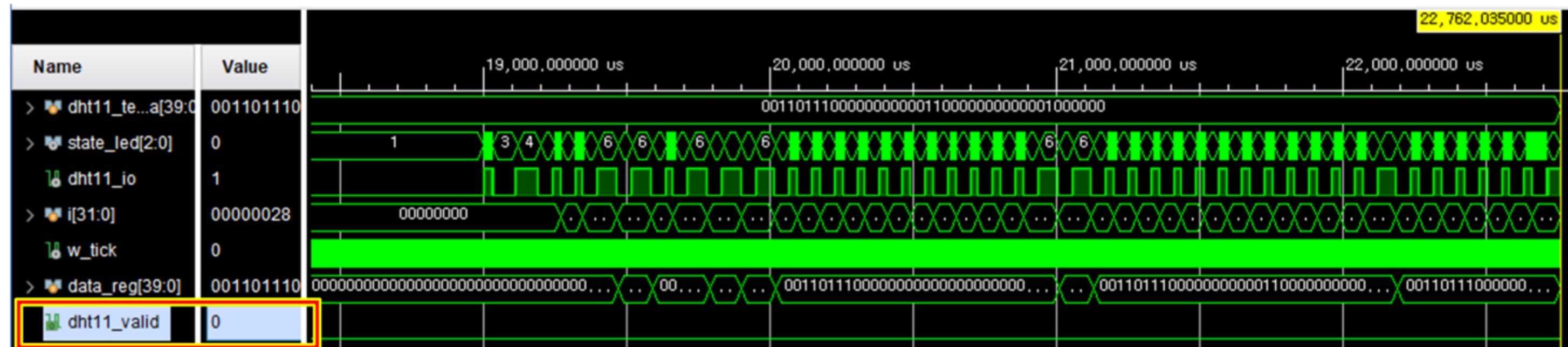




검증 - Timing 비교 (2)

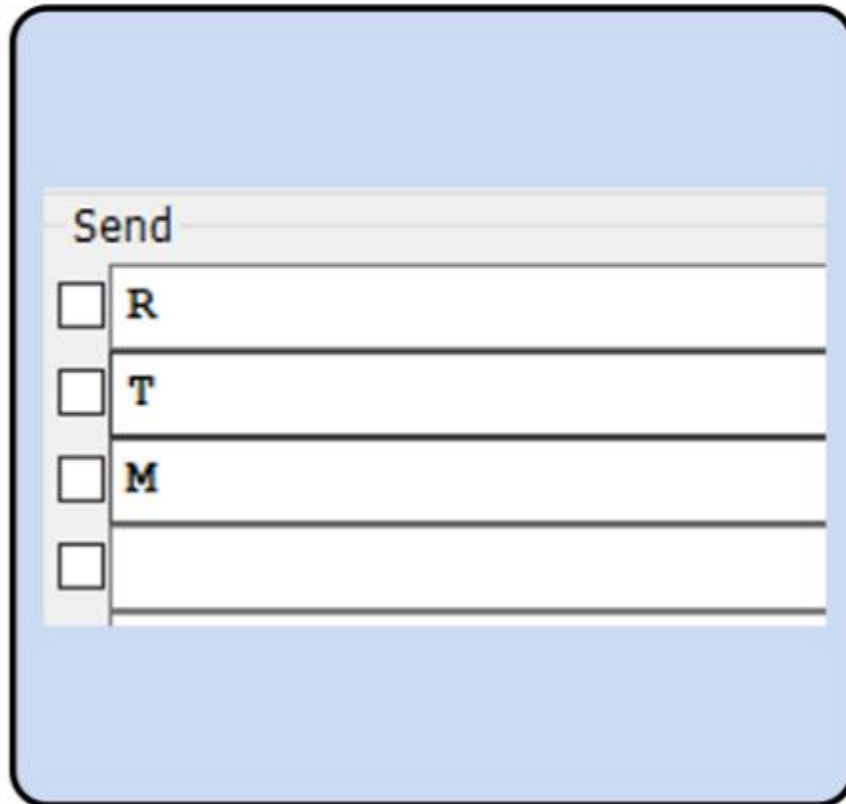


스펙의 경계값 99% 습도



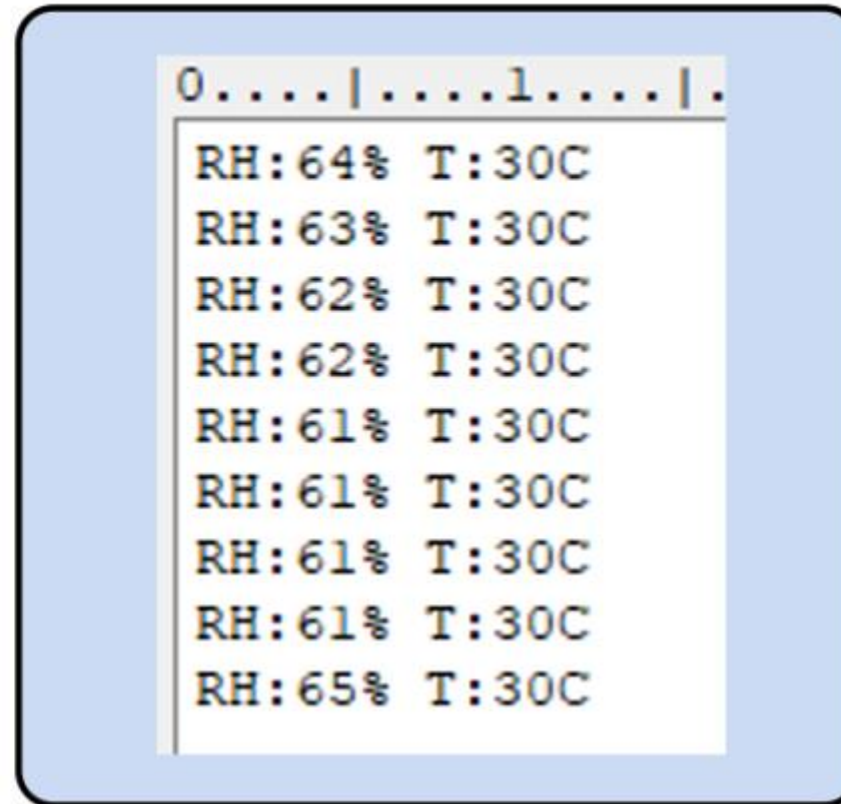
Checksum이 불일치 하는 경우

고찰 - 기존 온도 및 습도 센서 추가기능



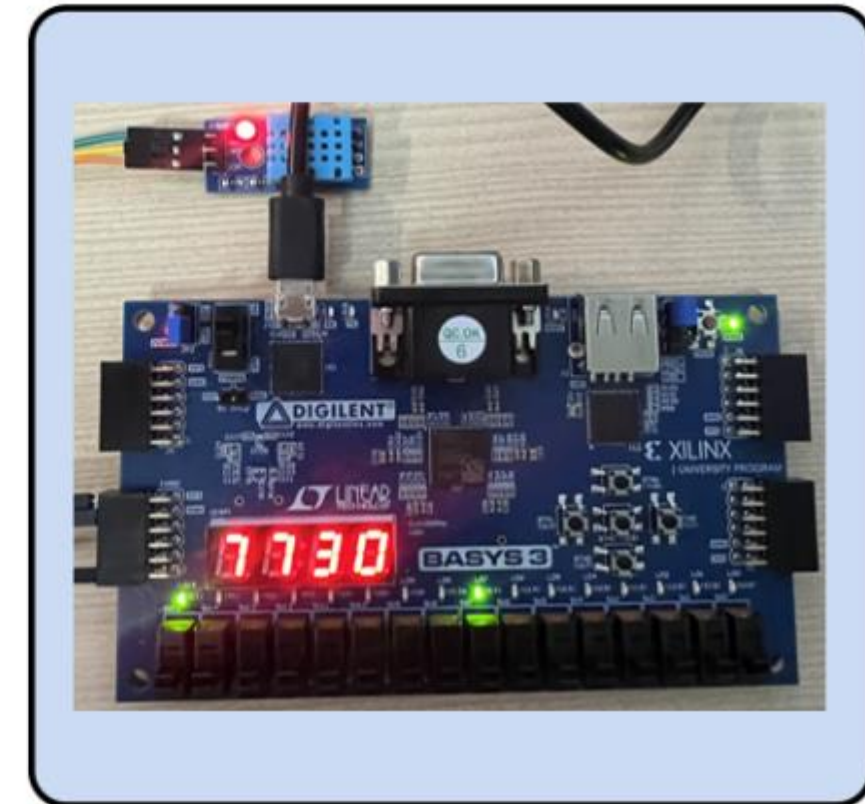
명령어 기능

5초마다 자동 측정되는 Auto 모드를 기존 버튼 측정 방식에 추가하였습니다. R, M, T와 같은 명령어를 추가하여 기존의 버튼으로 동작하는 것을 UART 제어도 가능하게 하였습니다.



Real Time 측정 기능

5초마다 자동 측정되는 Auto 모드를 기존 버튼 측정 방식에 추가하였습니다.



온도 및 습도 안전도 알림 기능

5초마다 자동 측정되는 Auto 모드를 기존 버튼 측정 방식에 추가하였습니다. 온도와 습도를 기준으로 30/32/34도 및 60/70/80% 이상일 경우, 3단계로 나누어 상태값을 출력하도록 설계하였습니다.

고찰 - Trouble shooting, Debugging



문제점

받은 센서는 정상 동작했지만, 팀원의 센서로 교체 후에는 마지막 state 7에서 멈추며 동작 정지.

원인 파악 및 해결 방안

STOP 단계에서 $50\mu\text{s}$ Low Level Signal이 필요한데, $10\mu\text{s}$ 가 더 출력되어 STALL 발생 타이밍 수정 후 정상 동작 확인

.....>



KEYWORD



해결방안 및 주의점

$50\mu\text{s}$ 타이밍을 정확히 맞추어 문제를 해결하였습니다. 추가적으로 조원들과 센서를 교환해 테스트하는 과정에서 모든 조원이 동작하지 않는 센서를 발견하였습니다. $50\mu\text{s}$ 타이밍 보정과 Edge Detection 개선에도 불구하고 센서가 동작하지 않아, 불량 센서였음을 확인했습니다.

고찰 - 목표 달성도 및 추후 개발 가능성



1

기존 목표

DHT11 센서의 온·습도 데이터를 수집하고 UART로 출력하는 시스템 설계 및 검증 완료.

2

추가 기능

UART 명령어 기반 제어, 5초 자동 측정, 온습도 안전도 표시 기능을 추가 구현하였습니다.

3

추후 개발 1

UART나 DIP 스위치를 통해 사용자 맞춤 경고 기준 구성 개발.

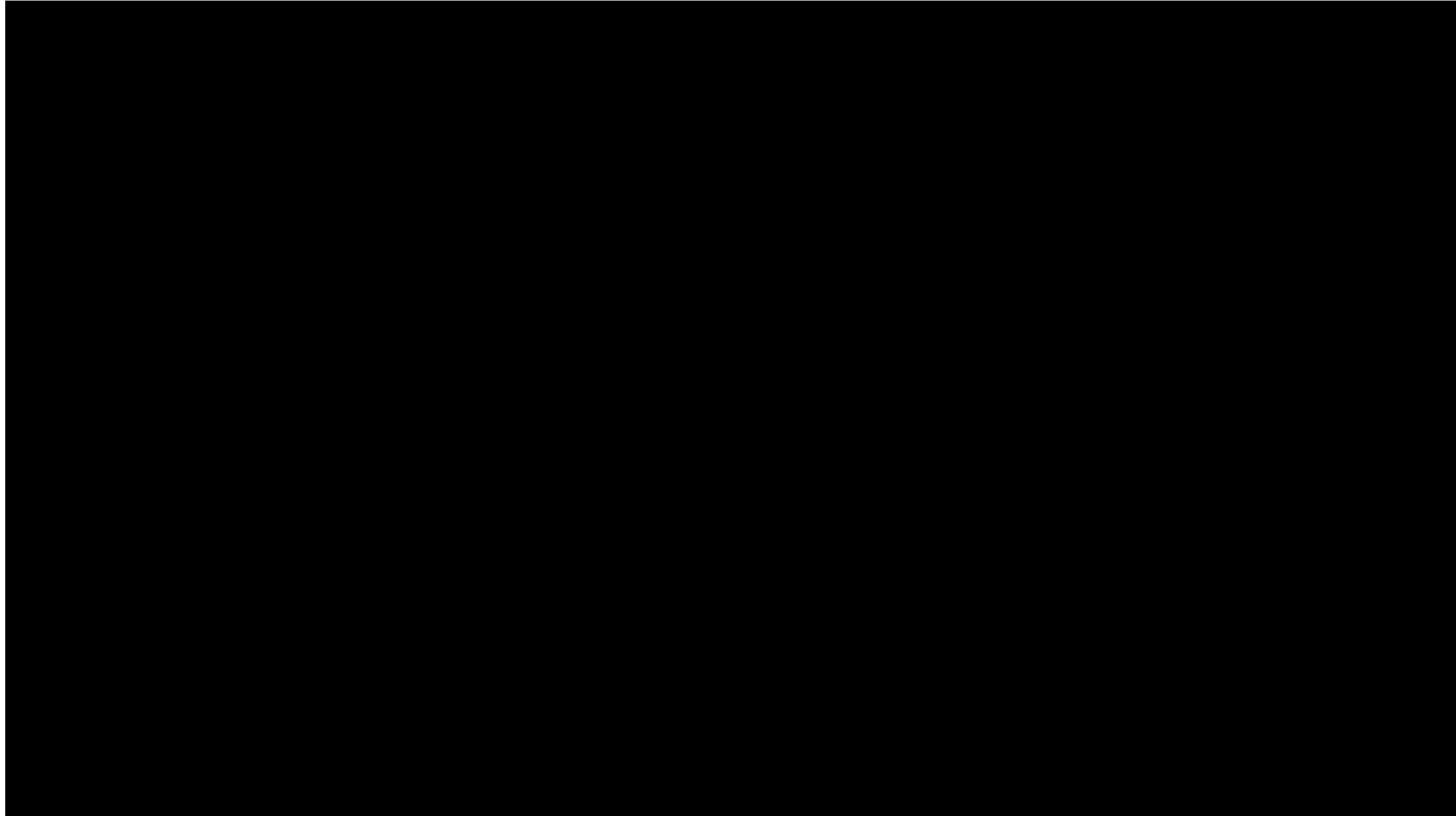
4

추후 개발 2

자동 측정 주기를 사용자가 직접 설정할 수 있는 맞춤형 Real Time 모드 기능 개발.

기존의 개발 목표 달성 및 추가적으로 3개 기능 추가하여 시뮬레이션 및 FPGA를 활용한 실측 검증까지 완료

시연 영상



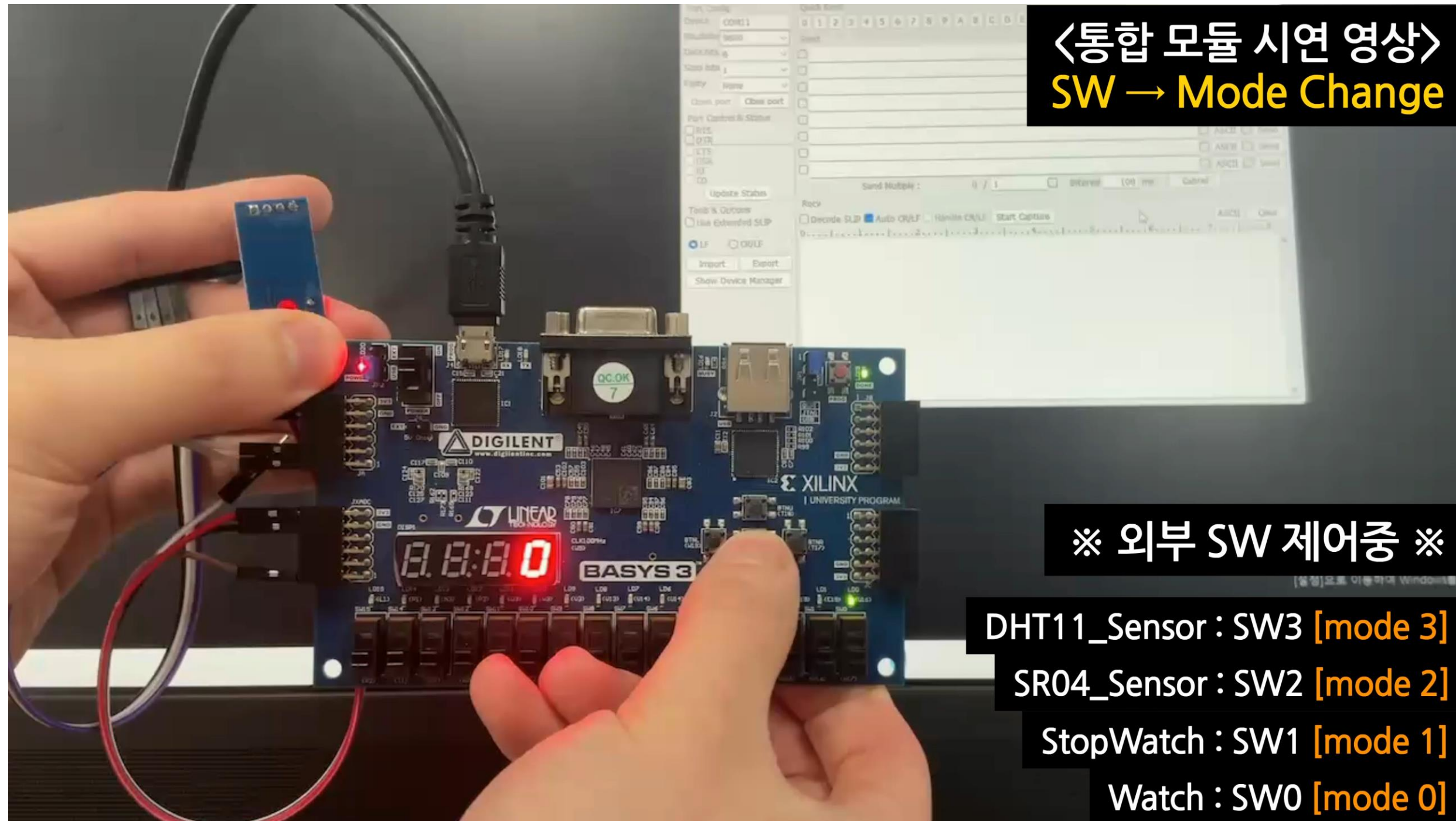


개발 회고

CHAPTER 04

- 통합 시연 영상
- 프로젝트 요약
- 역할 분담
- 개발 후기

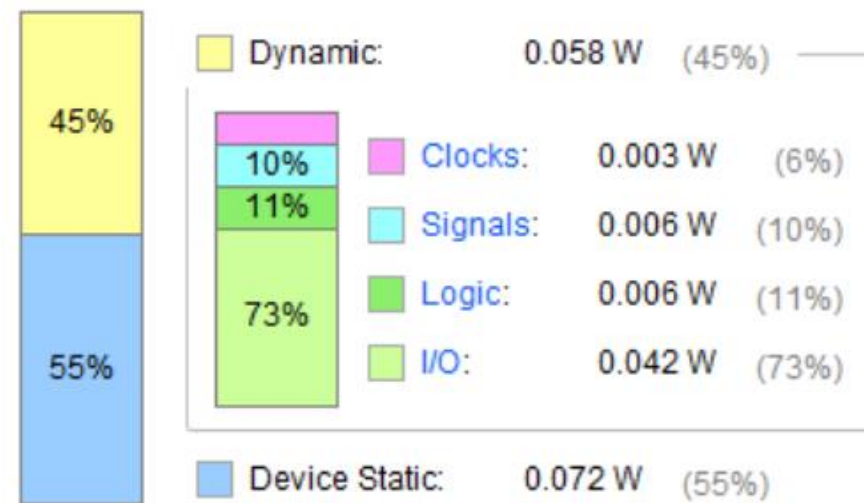
통합 시연 영상



프로젝트 요약



On-Chip Power



STA

Item	Value
Worst Setup Time	2.139 ns
Worst Hold Time	0.137 ns

리소스 사용 현황

Resource	Available	Utilization	Utilization (%)
LUT	20800	867	4.17 %
LUTRAM	9600	16	0.17 %
FF	41600	527	1.27 %
BUFG	32	1	3.13 %

온도 및 열 분석

Item	Value
Total On-Chip Power	0.13 W
Junction Temperature	섭씨 25.6 도
Thermal Margin	섭씨 59.4 도 (11.8W)

역할 분담



고종완 - Team leader

- DHT11 Sensor Design
- DHT11 Documentation
- Adding features to Watch Module



김태현

- UART & Watch Design
- Module Signal Integration
- TOP Module Design

이은성

- SR04 Sensor Design
- SR04 Documentation



임재홍

- Watch & Stopwatch Design
- document integration

개발 후기



고종완

FSM 기반 개발을 통해 단계별 로직을 명확히 나눌 수 있어 검증이 쉬워지고 실수가 줄어든다는 점을 체감했습니다. UART 외의 다른 통신 프로토콜도 타이밍 설계만 맞추면 구현이 가능하다는 점에서 자신감을 얻었습니다.



이은성

데이터시트에는 센서의 동작 원리와 타이밍, 입출력 조건 등이 명확히 정리되어 있어 다른 센서 모듈 구현에도 자신감을 얻었습니다. 또한, 시뮬레이션을 통해 오류를 직접 찾아 수정하며 검증 능력도 향상되었다고 느꼈습니다.



김태현

Top Module을 설계하며 모듈화의 중요성을 다시 한 번 느낄 수 있었습니다. 모듈화를 통해 명령 제어를 일원화하니 기능 추가 및 수정에 유리함이 있었습니다. 앞으로도 시스템 설계 시 모듈화를 적극적으로 활용해야겠다는 생각을 갖게 되었습니다.



임재홍

Bitstream만으로는 정상 동작하는 것처럼 보여도, 예상하지 못한 입력에서 오동작이 발생하는 경험을 했습니다. 이후 단순한 기능이라도 시뮬레이션하는 습관을 기르게 되었고, 설계 초기보다 더욱 신중하게 회로를 검토하는 태도를 갖게 되었습니다.



대한상공회의소
서울기술교육센터



THANK YOU!

Q&A

[Team 6] 고종완 | 김태현 | 이은성 | 임재홍