2025년 7월 29일

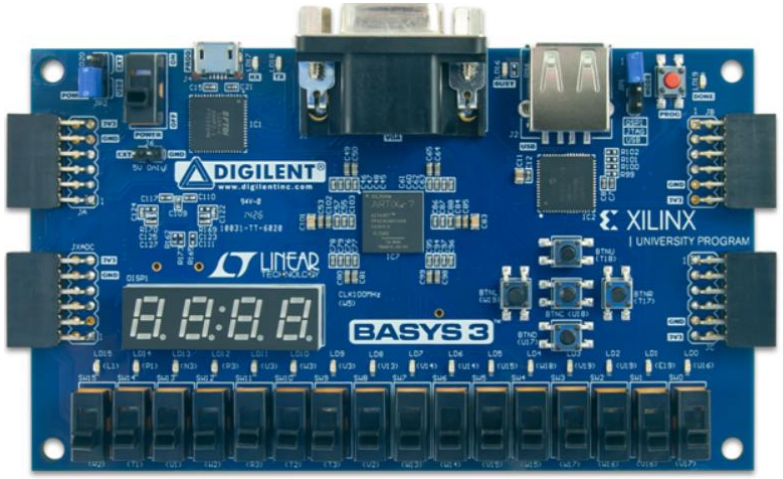
FINAL PROJECT

3조 박찬호 석경현 박주원 신상혁

HARMAN 세미콘아카데미 시스템반도체 설계 검증 엔지니어 2기

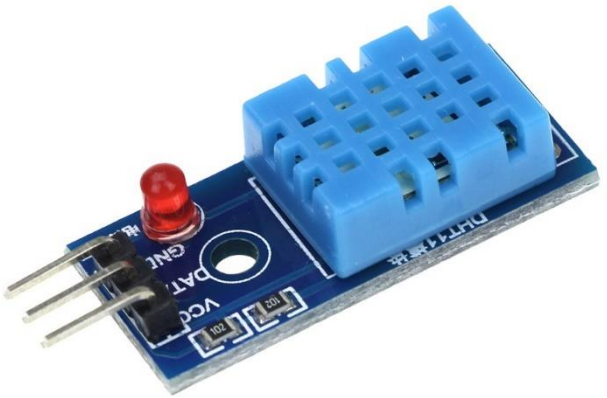
INDEX

BOM



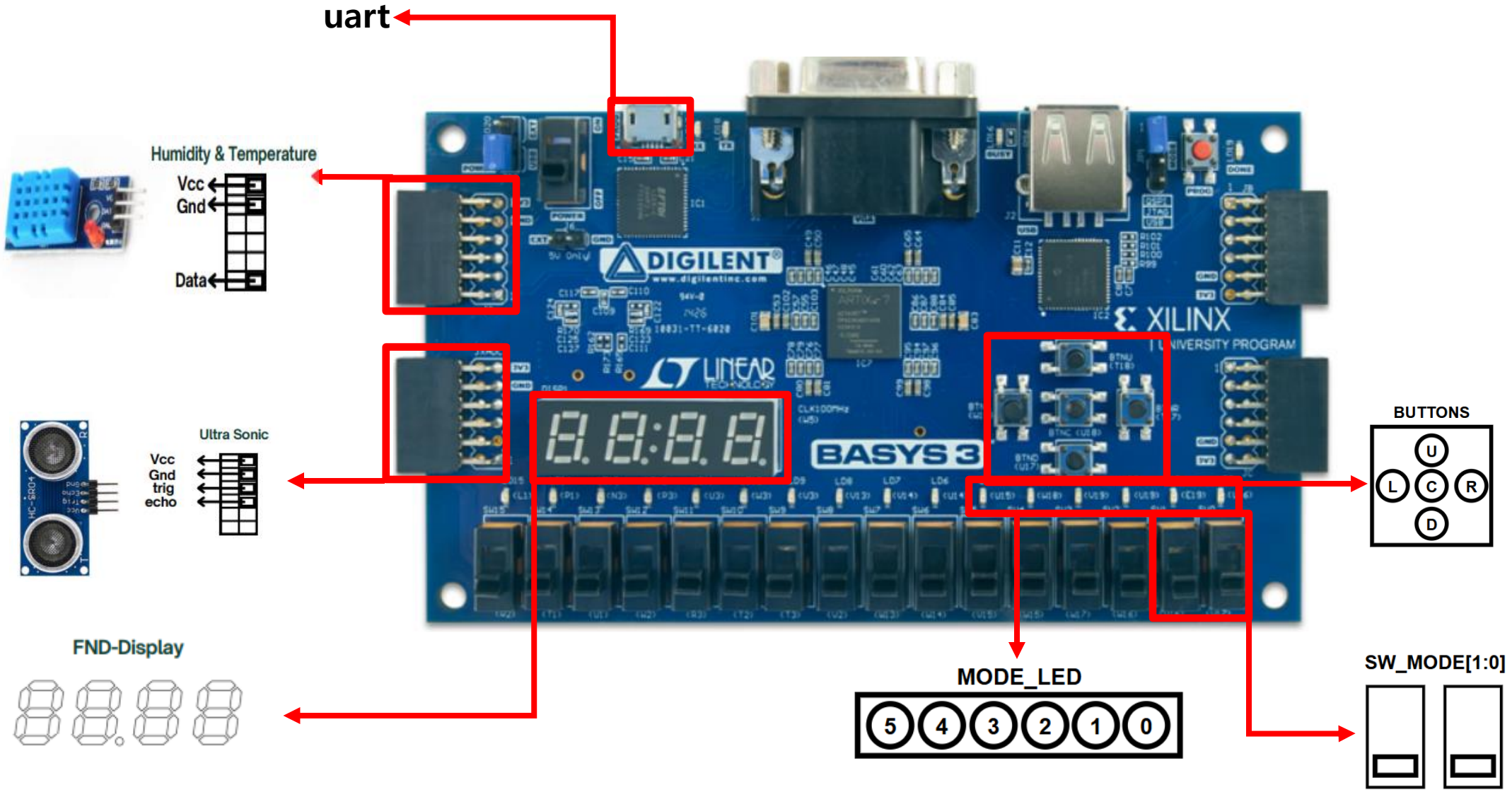
Basys3

SR04

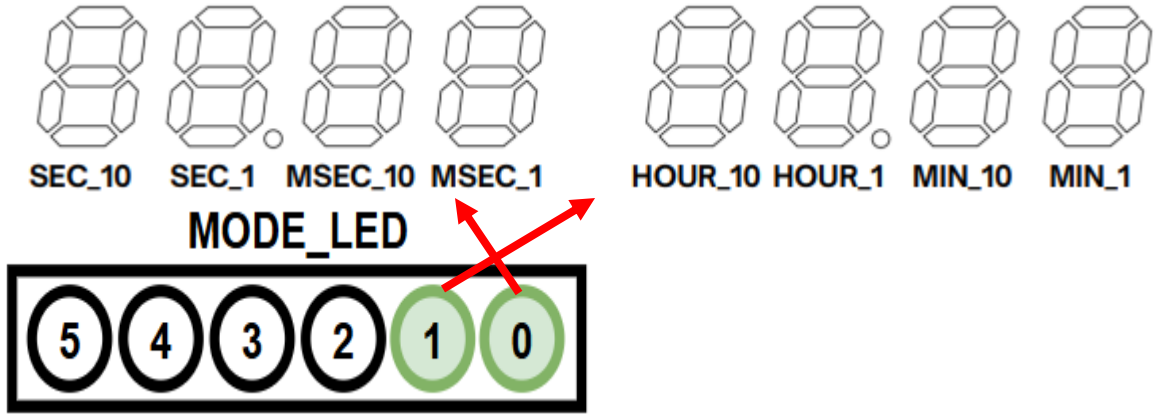


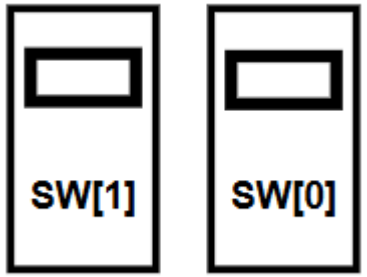
DHT11

기능 구성


uart
Humidity & Temperature
Vcc
Gnd
Data
Ultra Sonic
Vcc
Gnd
trig
echo
FND-Display
DIGILENT
XILINX
LINEAR TECHNOLOGY
BASYS 3
BUTTONS
U
L
C
R
D
MODE_LED
5
4
3
2
1
0
SW_MODE[1:0]

STOP_WATCH MODE





- SWITCH[0] =

1'b1 ->HOUR/MIN FND-Display 출력

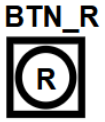
1'b0 -> SEC/MSEC FND-Display 출력

- SWITCH[1]

TOGGLE 할 때마다 모드 변경

STOPWATCH -> WATCH -> DISTANCE -> TEMP_HUMID MODE

Button 기능



Run & Stop toggle

Clear

Reset

UART 기능

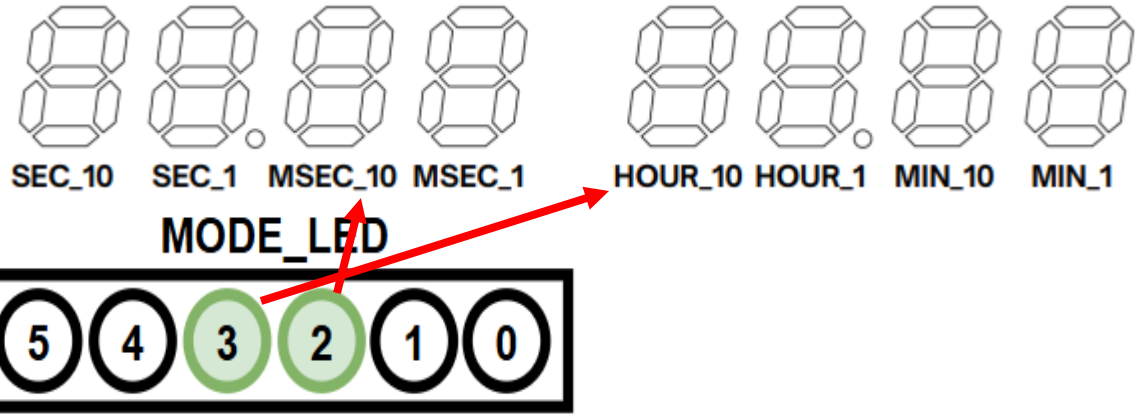
R : Run & Stop

L : Clear

M : Mode change

T : 시간 출력

WATCH MODE



- SWITCH[0] =

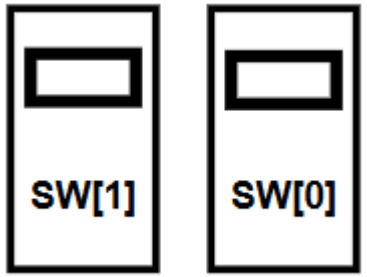
1'b1 ->HOUR/MIN FND-Display 출력

1'b0 -> SEC/MSEC FND-Display 출력

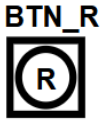
- SWITCH[1]

TOGGLE 할 때마다 모드 변경

STOPWATCH -> WATCH -> DISTANCE -> TEMP_HUMID MODE



Button 기능



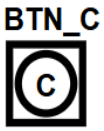
Selet_Adjust_Time

msec/sec, min/hour

Clear

Increase

Decrease



Reset

UART 기능

R : Selet_Adjust_Time

L : Clear

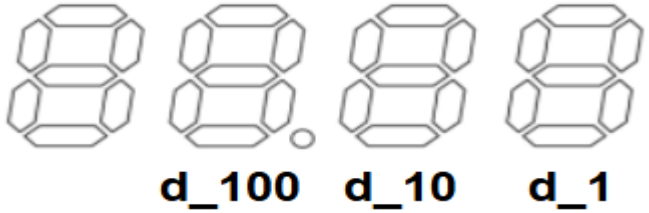
M : Mode change

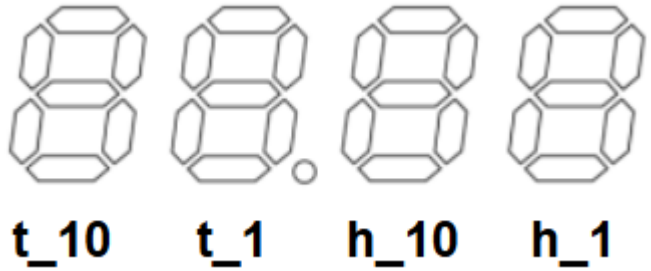
U : Increase

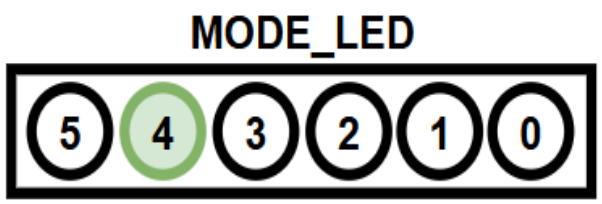
D : Decrease

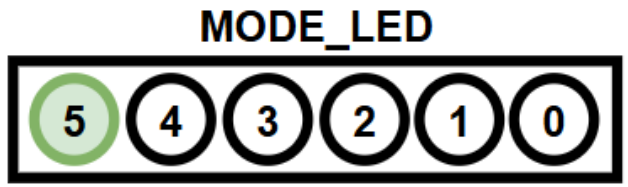
T : 시간 출력

Ultra_Sonic, Humidity & Temperature Mode









- SWITCH[1]

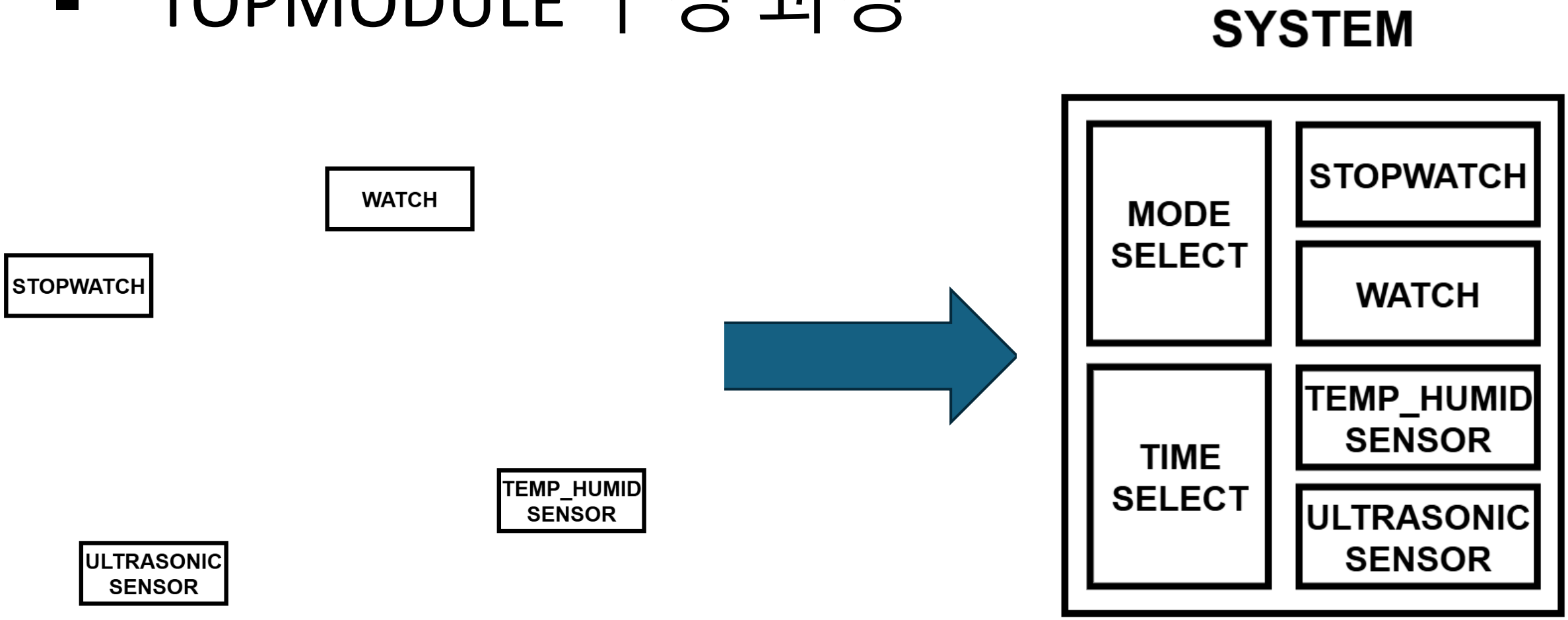
TOGGLE 할 때마다 모드 변경

STOPWATCH -> WATCH -> DISTANCE -> TEMP_HUMID MODE

Button 기능		UART 기능
BTN_R		
R	Run	T : Run & display
		M : Mode change

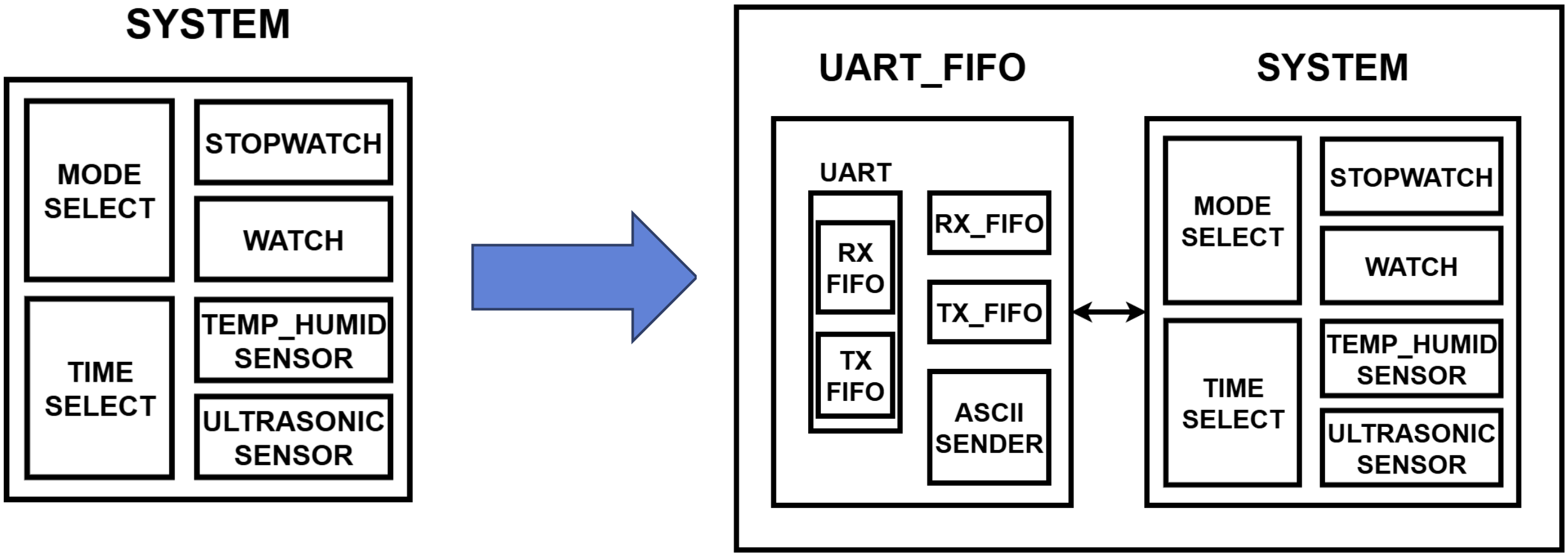
FINAL_TOP MODULE

TOPMODULE 구성 과정



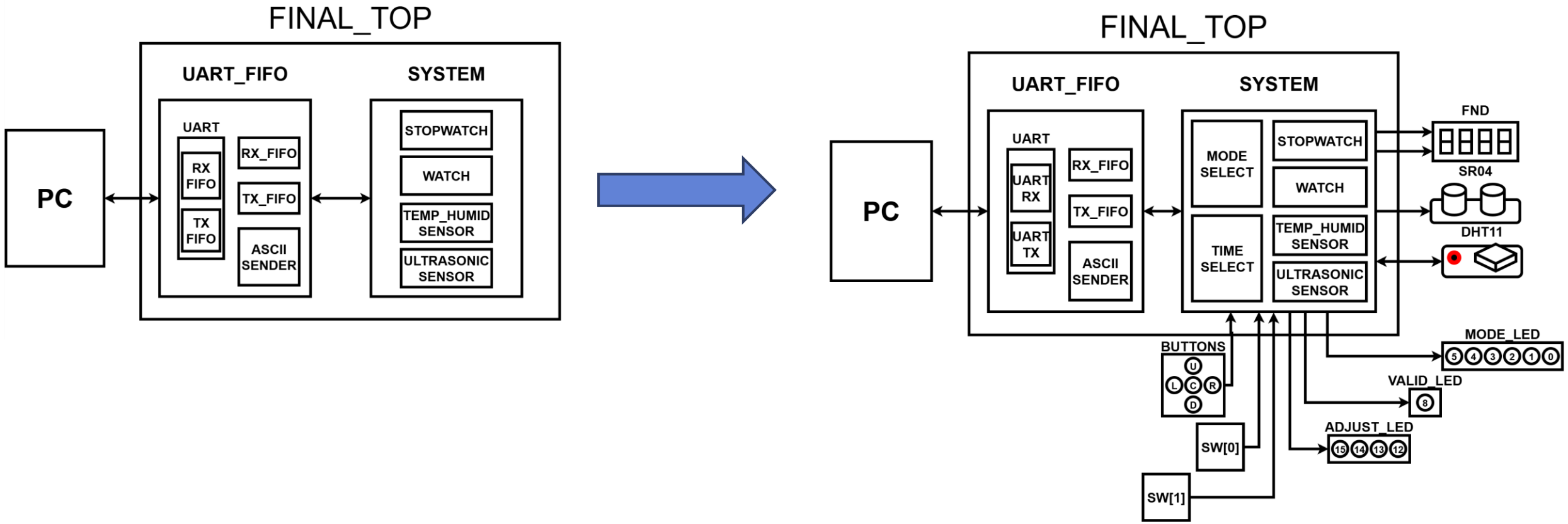
FINAL_TOP모듈을 만들기 전에 SYSTEM MODULE을 만들어서 흩어져 있는 WATCH_STOPWATCH와 SR04센서 DHT11센서를 합친다.

TOPMODULE 구성 과정



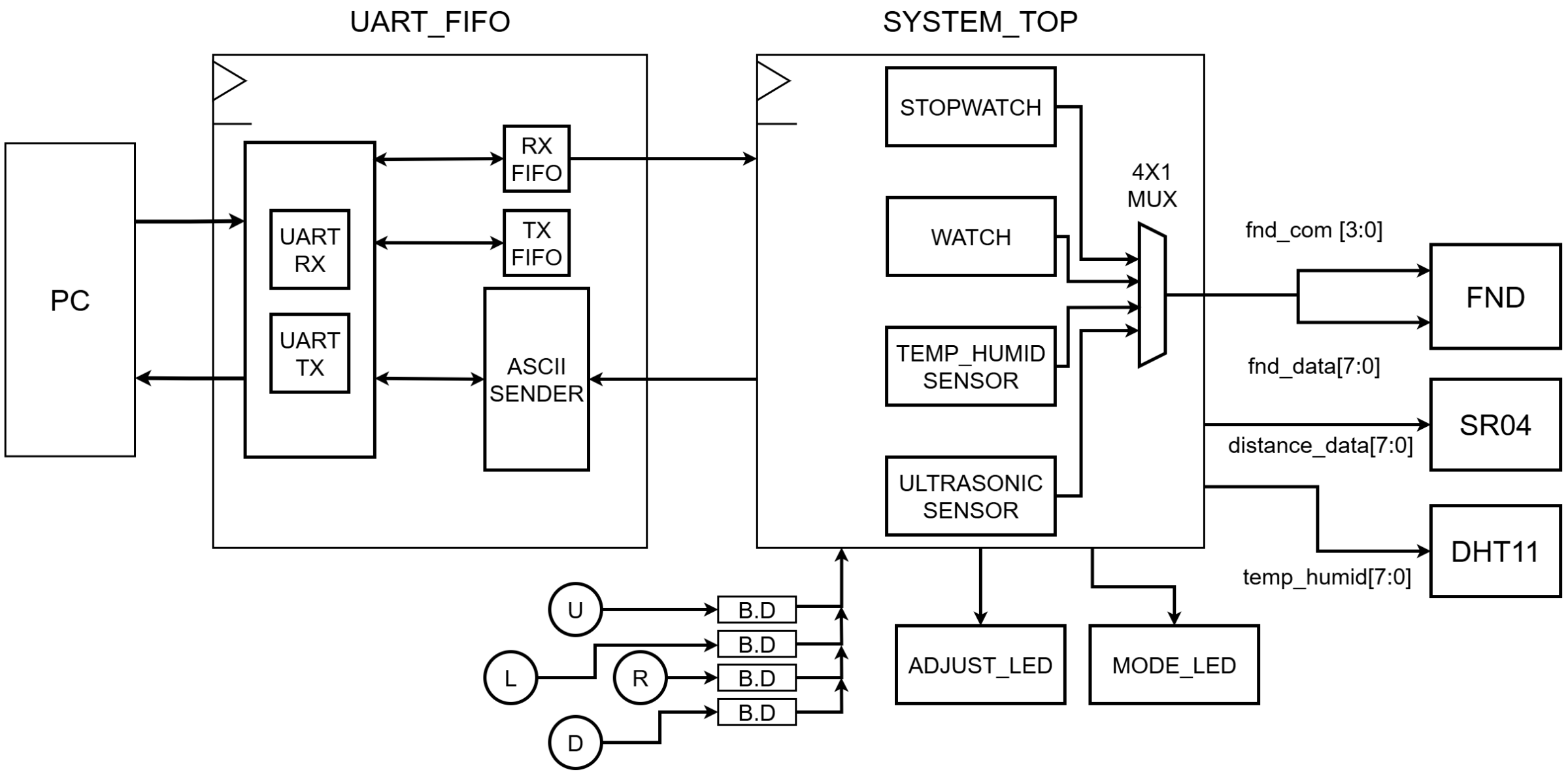
UART_FIFO와 SYSTEM_TOP을 인스턴스화해 FINAL_TOP 모듈을 만든다.

TOPMODULE 구성 과정



센서의 기능을 UART로 구현하고 LED, BUTTON, SWITCH를 재구성한다.

FINAL_TOP DATAPATH



SR04 Ultrasonic Sensor

박찬호

SR04 specification

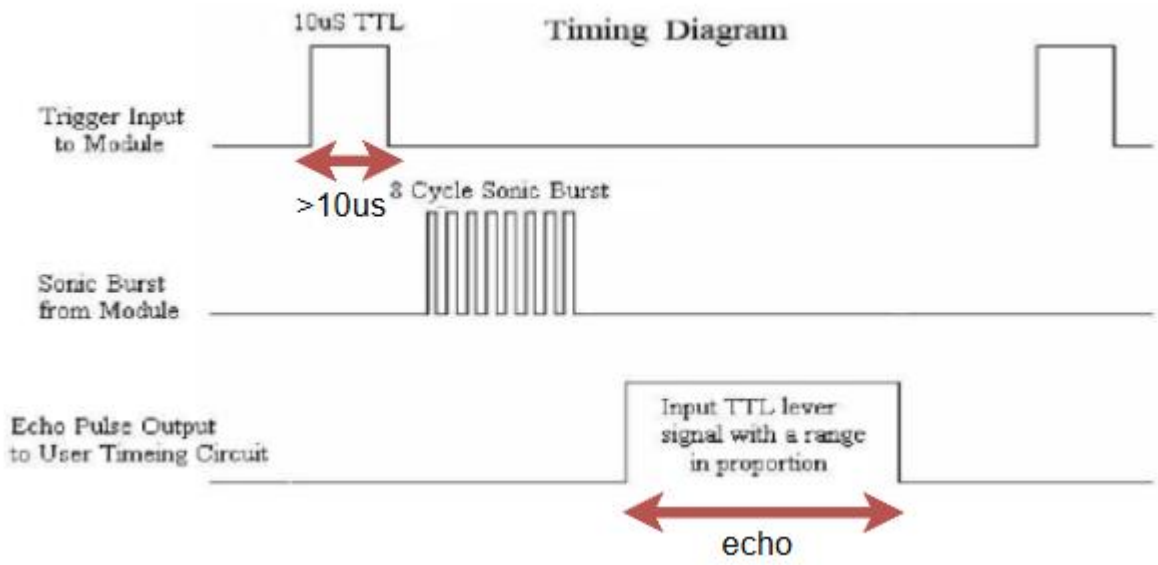
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

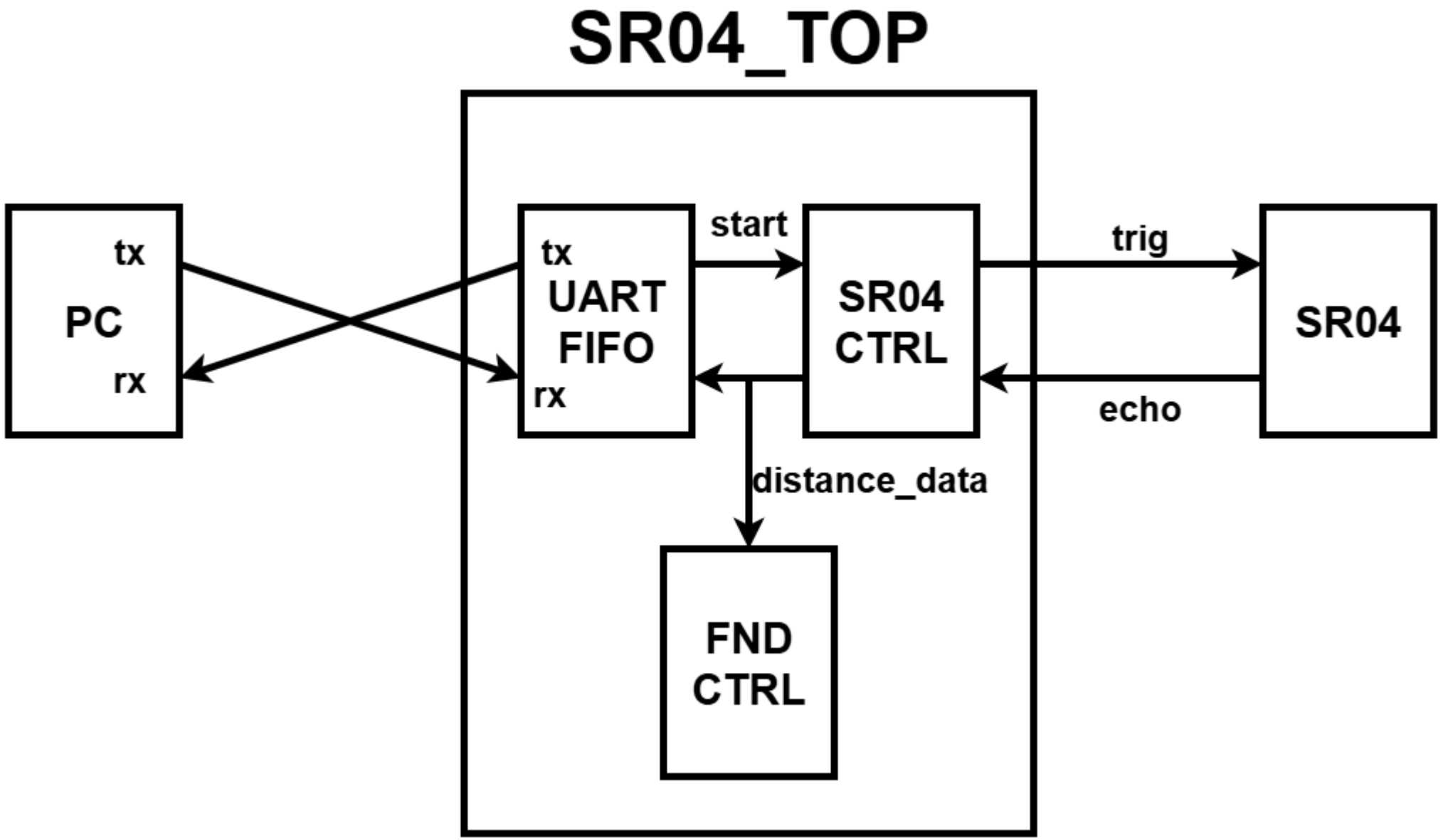
SR04 Timing Diagram

1. Trig 핀에 10μs 이상의 High 신호 입력
2. 센서 내부에서 초음파 발사
3. Echo 핀이 High 상태로 유지 시간 측정

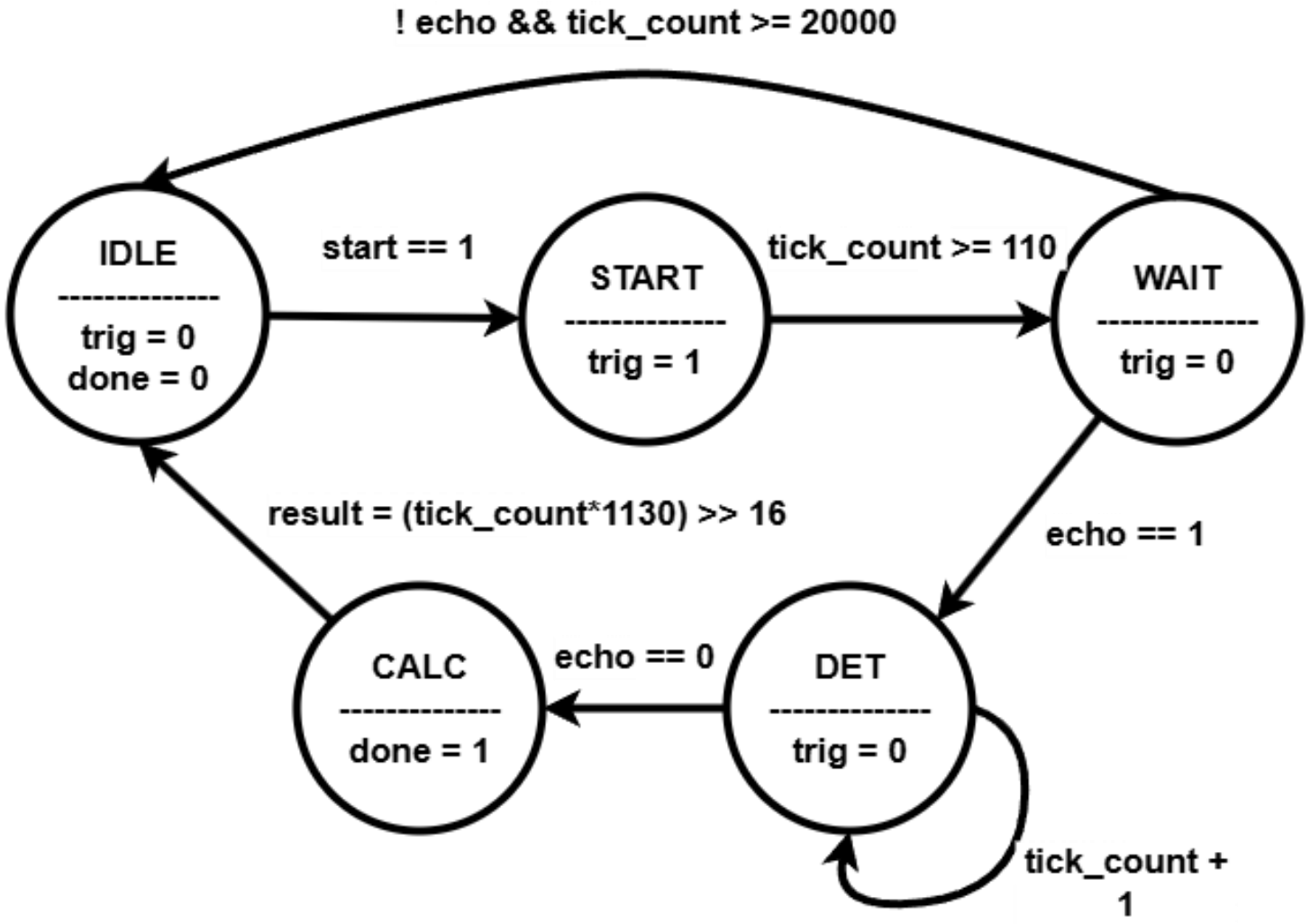
- uS / 58 = cm
- uS / 148 =inch



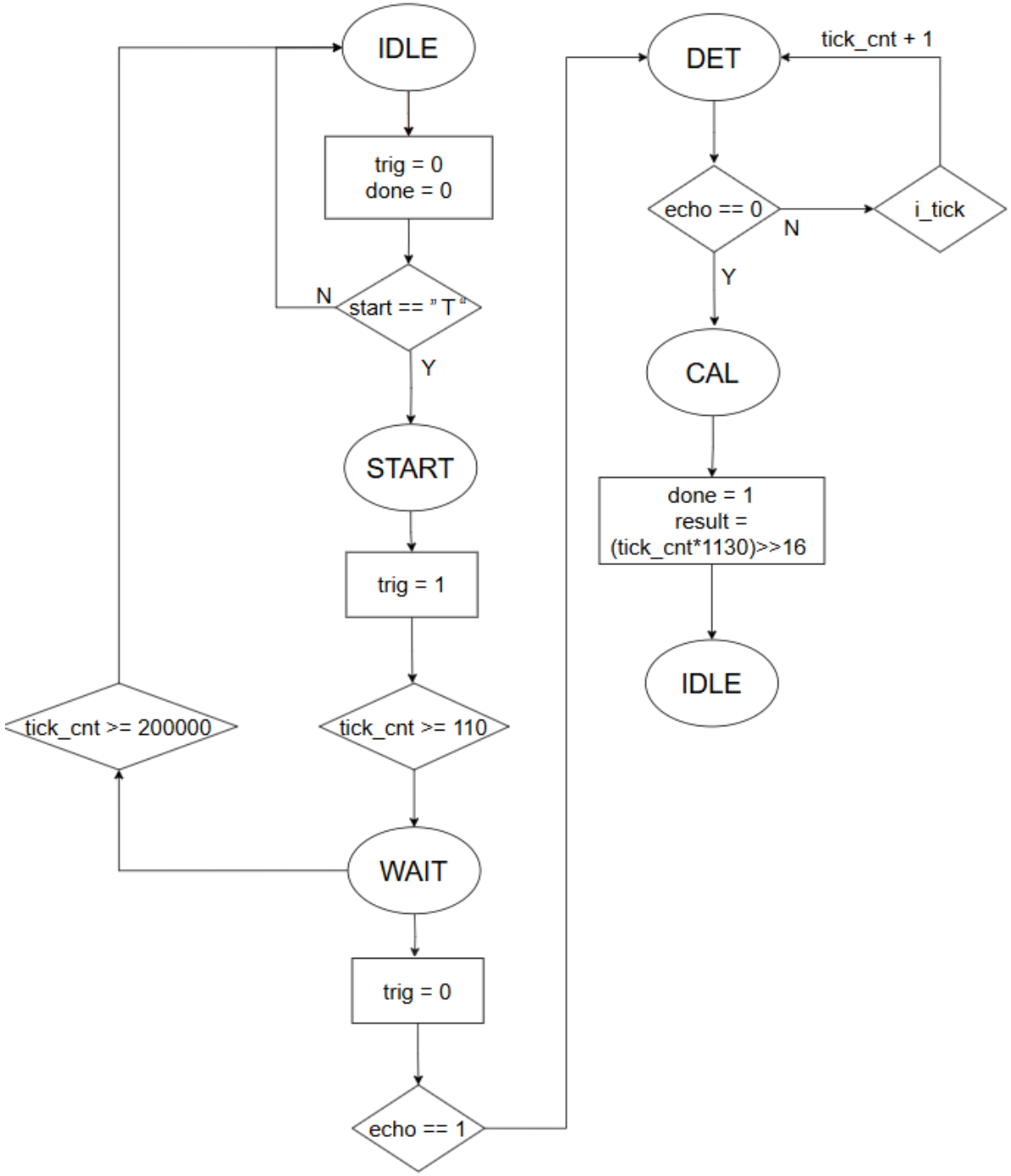
SR04 Block Diagram



SR04 FSM Chart

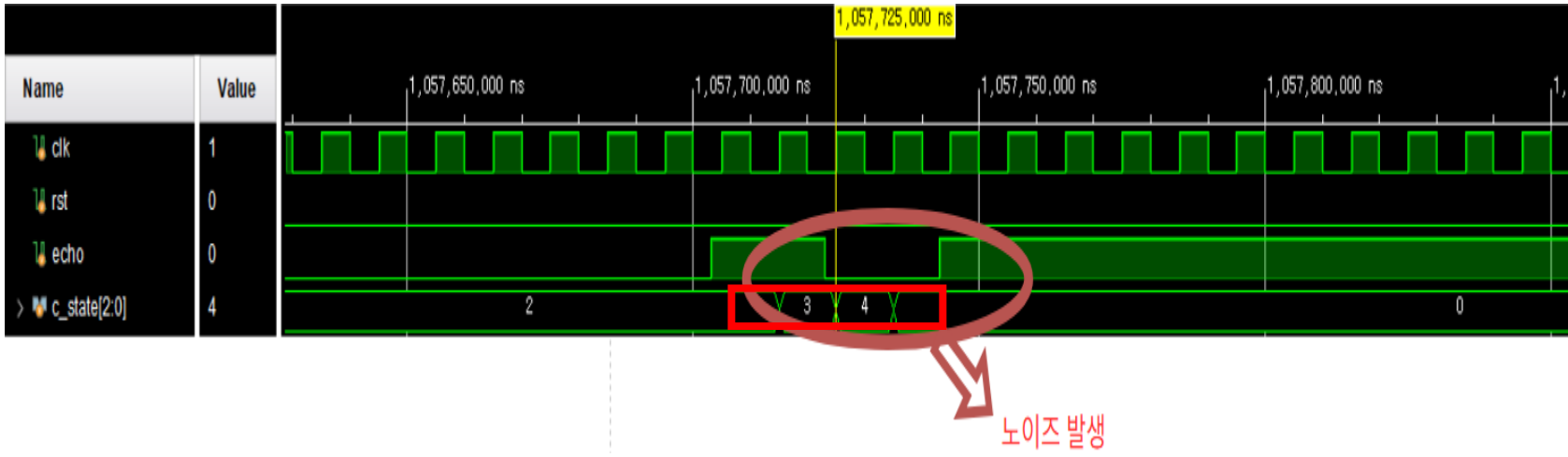


SR04 ASM Chart



Noise filtering

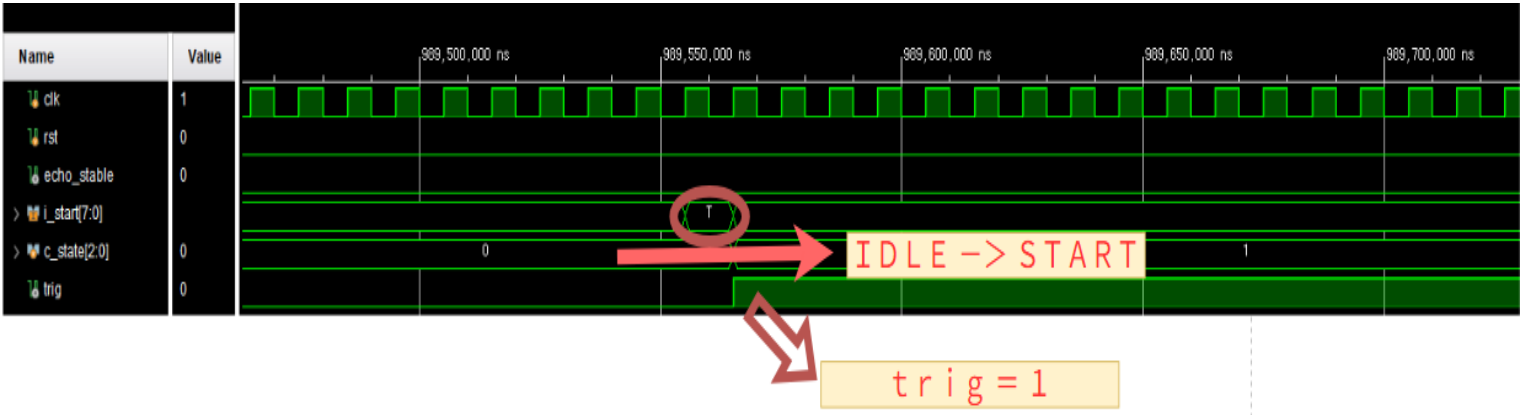
```
always @(posedge clk, posedge rst) begin
    if (rst) begin
        echo_shift <= 3'b000;
    end else begin
        echo_shift <= {echo_shift[1:0], i_echo};
    end
end
assign echo_stable = &echo_shift;
```

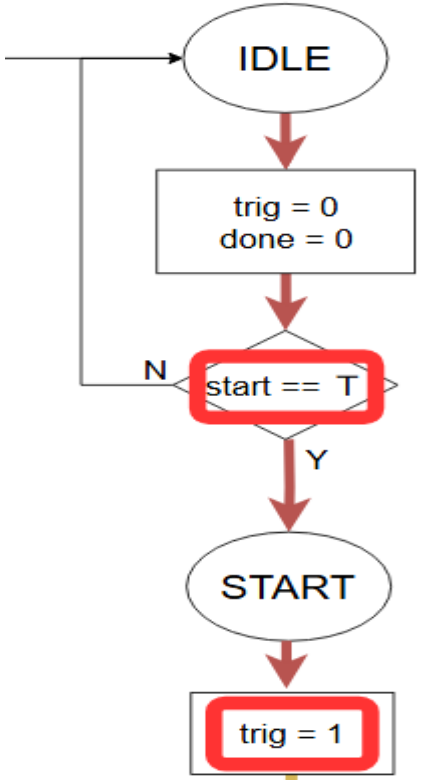


- 장점
  - 노이즈 필터링
  - 신호 안정화
- 단점
  - 응답 지연 발생

Simulation

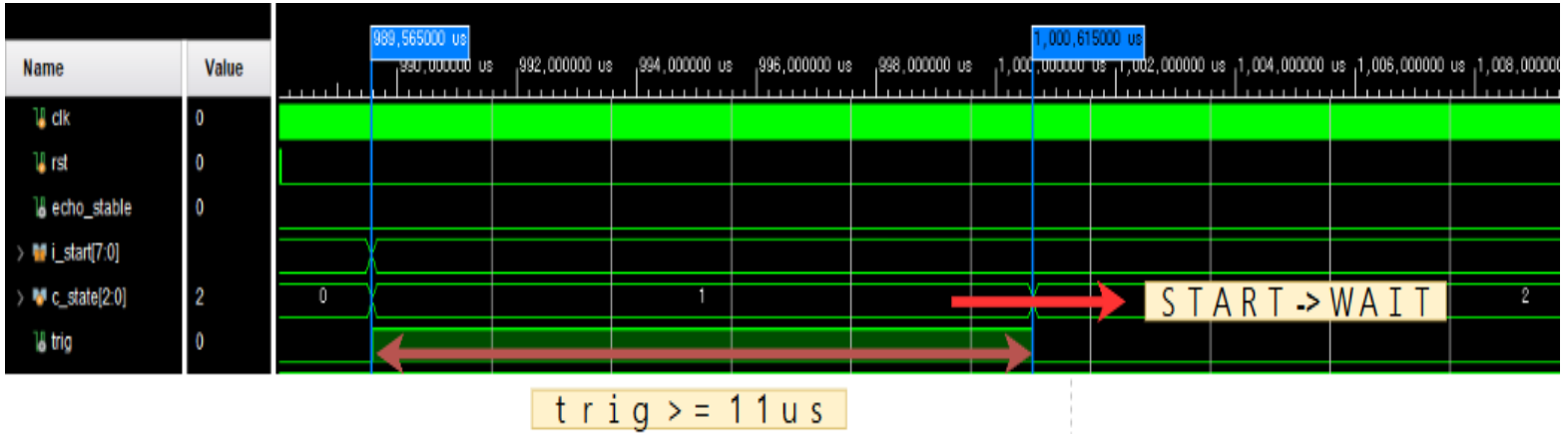
- IDLE - > START



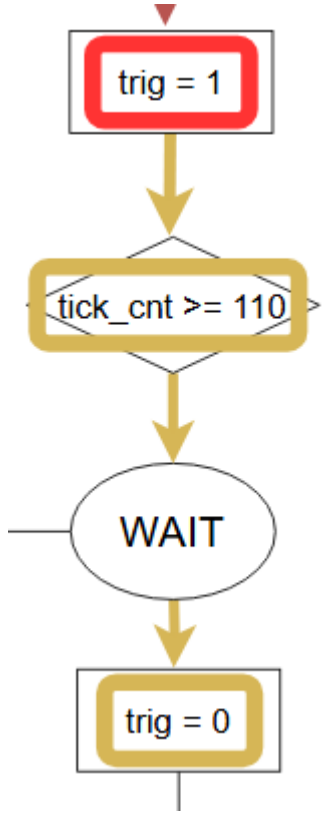


IDLE - > START

- START - > WAIT

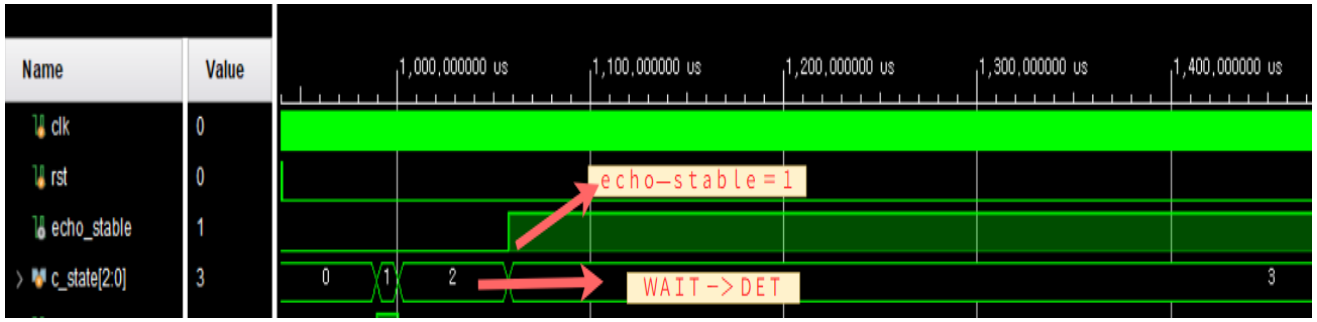


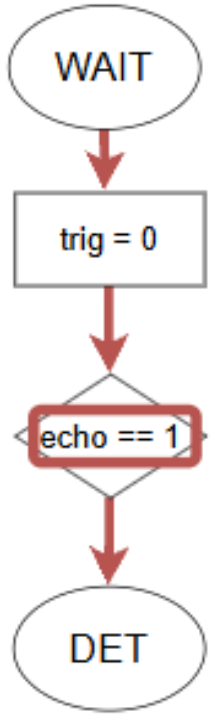
START - > WAIT



Simulation

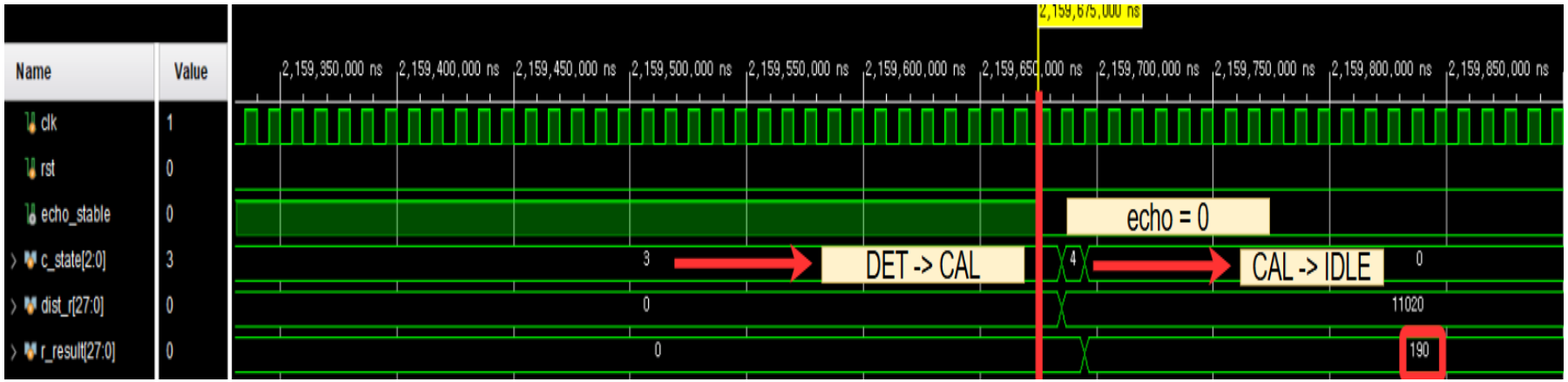
- WAIT - > DET



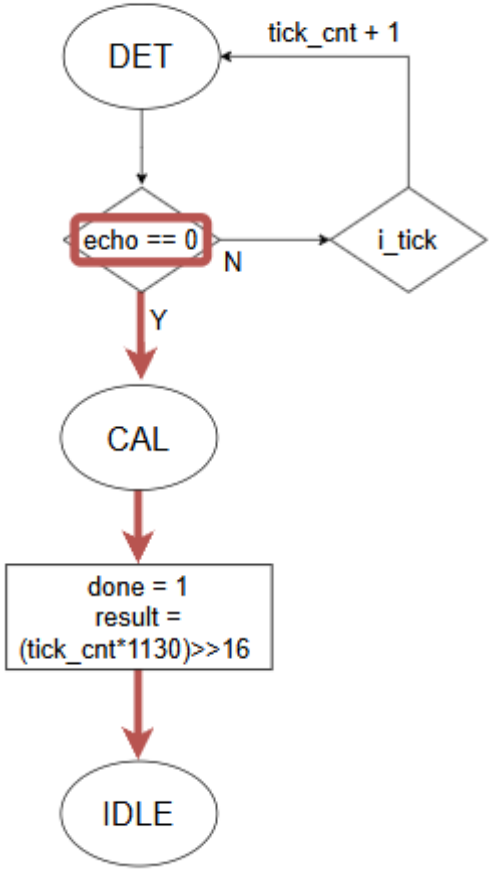


WAIT - > DET

- DET - > CAL -> IDLE



DET - > CAL -> IDLE



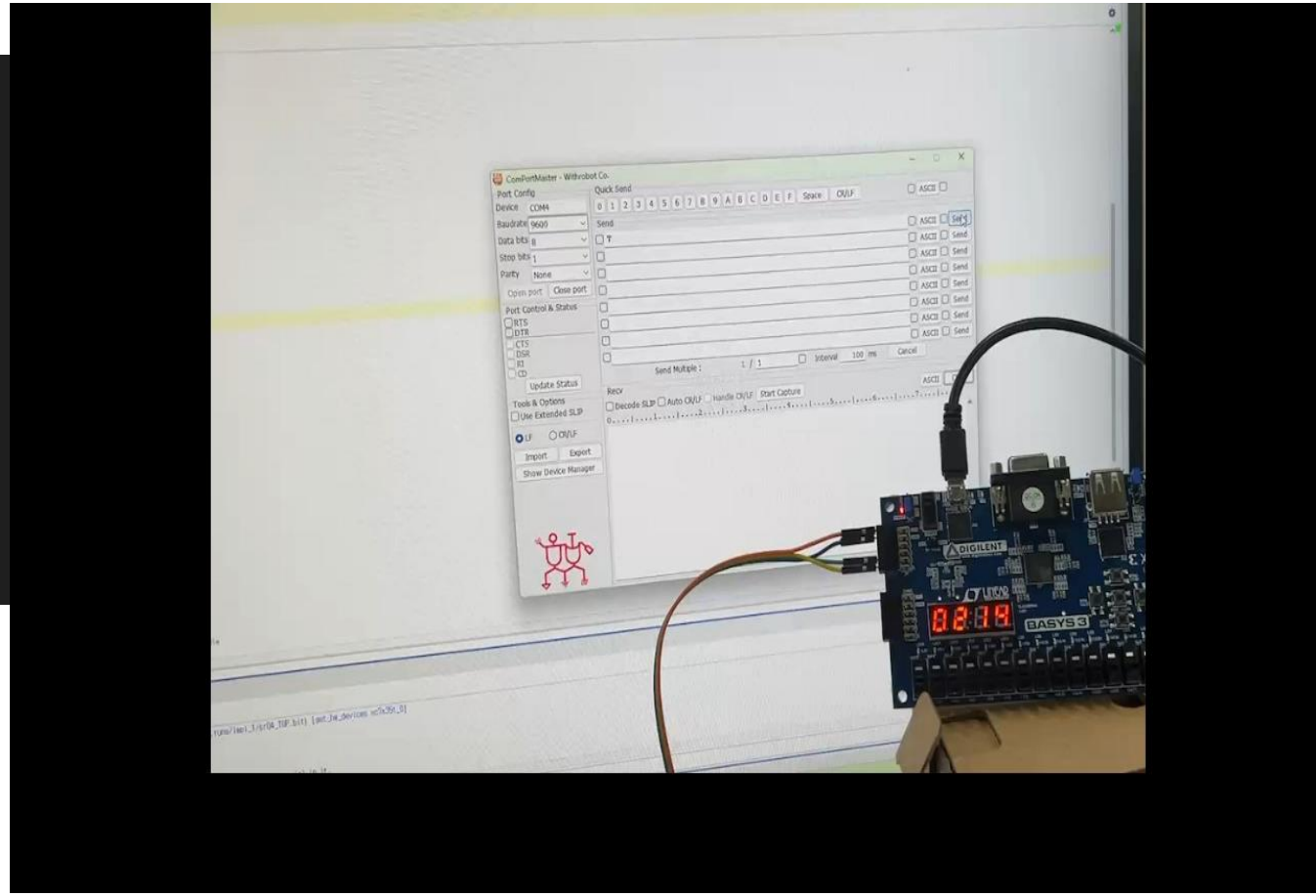
Simulation & 동작 영상

```
for (i = 0; i < 10; i = i + 1) begin
    random_data = $urandom_range(2, 200);
    send_data = random_data;
    send_uart(send_data);
    #15_000
    echo = 1;
    #(58*1000*random_data);
    echo = 0;
    wait(o_done==1);
    #100;
    if (o_dist == random_data*10) begin
        $display("Pass : random_dist = %0d, measured_dist = %0d", random_data*10, o_dist);
    end  else begin
        $display("Fail : random_dist = %0d, measured_dist = %0d", random_data*10, o_dist);
    end
    #1000;
end
    $stop;
```

2~200 정수 생성

소수점까지 출력되기 때문에 *10

```
Pass : random_dist = 340, measured_dist = 340
Pass : random_dist = 970, measured_dist = 970
Pass : random_dist = 350, measured_dist = 350
Pass : random_dist = 910, measured_dist = 910
Pass : random_dist = 1310, measured_dist = 1310
Pass : random_dist = 1170, measured_dist = 1170
Pass : random_dist = 1510, measured_dist = 1510
Pass : random_dist = 1170, measured_dist = 1170
Pass : random_dist = 1890, measured_dist = 1890
Pass : random_dist = 1840, measured_dist = 1840
```

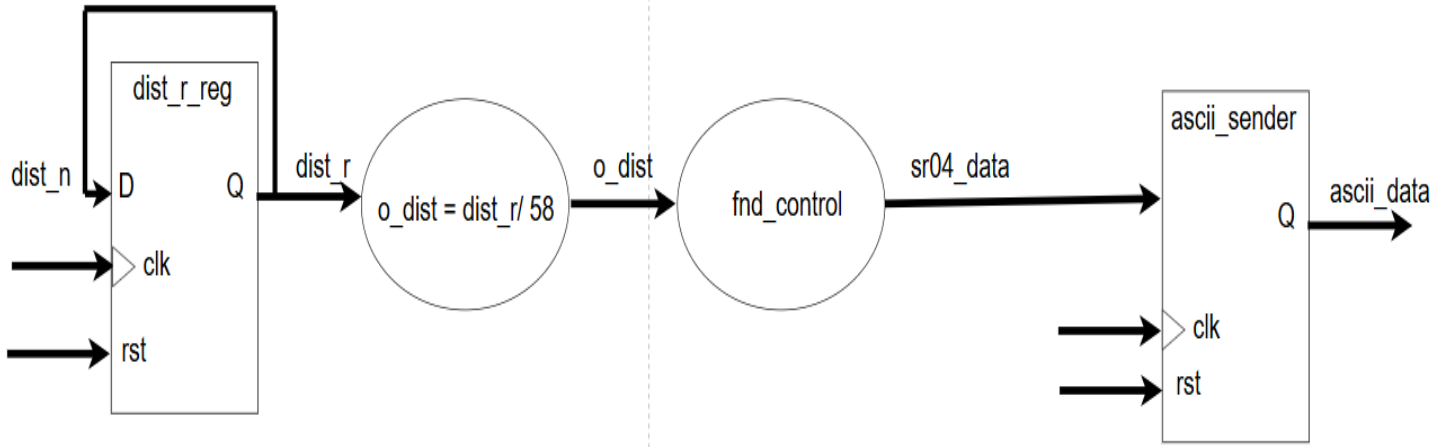


Troubleshooting

Total Negative Slack (TNS): -163.862 ns

```
        DETECT: begin
            if (~echo_stable) begin
                n_state = CAL;
                dist_n  = tick_cnt_r;
            end else begin
                if ( i_tick) begin
                    tick_cnt_n = tick_cnt_r + 1;
                end
            end
        end
        CAL: begin
            n_done  = 1;
            n_state = IDLE;
        end
    endcase
end

assign o_dist = dist_r / 58;
```



Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-10.500 ns	Worst Hold Slack (WHS):	0.084 ns	Worst Pulse Width Slack (WPWS):	2.750 ns
Total Negative Slack (TNS):	-163.862 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	16	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	383	Total Number of Endpoints:	383	Total Number of Endpoints:	223

Timing constraints are not met.

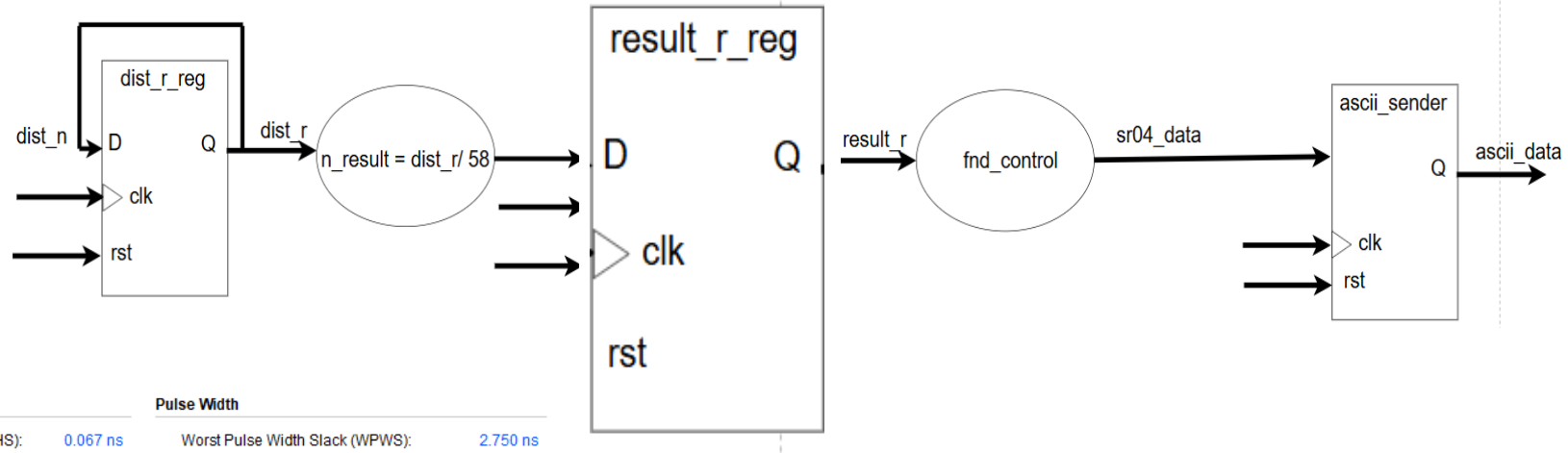
- 나눗셈 연산과 레지스터 사이의 경로가 너무 길기 때문에 setup time violation 발생

Troubleshooting

Total Negative Slack (TNS): -163.862 ns

Total Negative Slack (TNS): -49.278 ns

```
        DETECT: begin
            if (~echo_stable) begin
                n_state = CAL;
                dist_n  = tick_cnt_r;
            end else begin
                if ( i_tick) begin
                    tick_cnt_n = tick_cnt_r + 1;
                end
            end
        end
        CAL: begin
            n_done  = 1;
            n_result = (dist_r/58);
            n_state = IDLE;
        end
    endcase
end
```



Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-3.260 ns	Worst Hold Slack (WHS):	0.067 ns	Worst Pulse Width Slack (WPWS):	2.750 ns
Total Negative Slack (TNS):	-49.278 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	28	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	437	Total Number of Endpoints:	437	Total Number of Endpoints:	250

Timing constraints are not met.

- 레지스터 사이의 경로를 레지스터 하나를 추가해 slack을 늘렸지만 dist_r 레지스터와 result_r 레지스터의 나눗셈 연산에서 setup time violation 발생

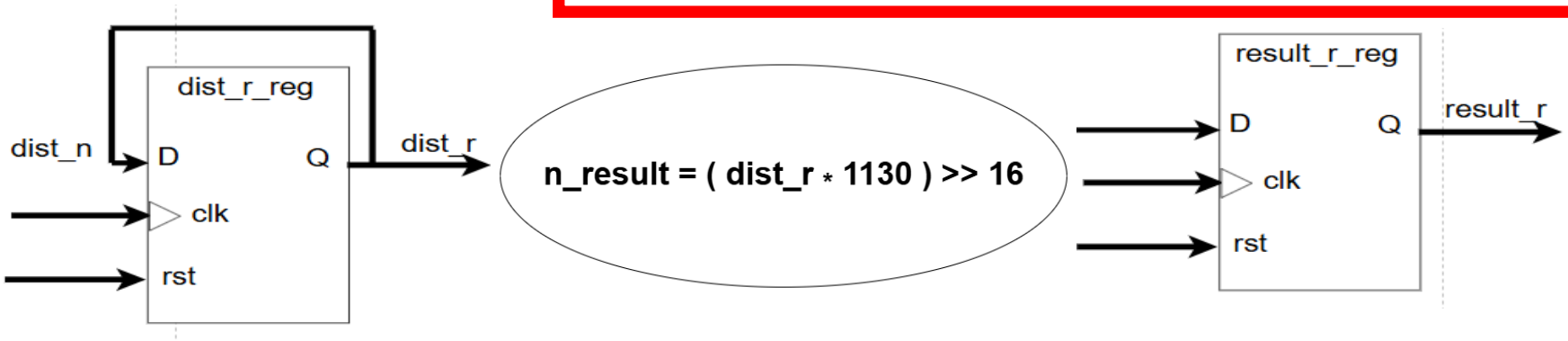
Troubleshooting

Total Negative Slack (TNS): -163.862 ns

Total Negative Slack (TNS): -49.278 ns

Total Negative Slack (TNS): 0.000 ns

```
        CAL: begin
            n_done  = 1;
            n_result = (dist_r*1130) >>16;
            //n_result = (dist_r*71) >> 12;
            n_state = IDLE;
        end
    endcase
end
```



Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.151 ns	Worst Hold Slack (WHS):	0.136 ns	Worst Pulse Width Slack (WPWS):	2.750 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	405	Total Number of Endpoints:	405	Total Number of Endpoints:	234

All user specified timing constraints are met.

- 곱하기와 right shift 연산을 통해 나눗셈을 없애고 레지스터를 추가해 레지스터와 레지스터 사이의 경로를 줄여 time violation 해결

Trade off

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT as Memory (9600)	Bonded IOB (106)	BUFGCTRL (32)
sr04_TOP	305	204	117	289	16	18	1
U_FC (fnd_controller)	62	23	34	62	0	0	0
U_SR04_CON (sr04_controller)	137	61	60	137	0	0	0
U_UF (uart_fifo)	118	120	52	102	16	0	0

```
n_result = (dist_r * 71) >> 12
```

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT as Memory (9600)	Bonded IOB (106)	BUFGCTRL (32)
sr04_TOP	320	204	116	304	16	18	1
U_FC (fnd_controller)	67	23	40	67	0	0	0
U_SR04_CON (sr04_controller)	147	61	56	147	0	0	0
U_UF (uart_fifo)	118	120	48	102	16	0	0

```
n_result = (dist_r  * 1130) >> 16
```

- n_result = (dist_r * 71) >> 12
  - 오차 : 0.537%
- n_result =(dist_r * 1130) >> 16
  - 오차 : 0.0293%
- 리소스 사용의 차이가 있지만 리소스가 부족하지 않기 때문에 리소스를 더 사용하고 오차가 적은 16bit shift 사용

DHT11 Sensor Design

석경현

Overview

DHT11 Sensor

- 디지털 온습도 센서
- 단일 핀으로 데이터 송수신

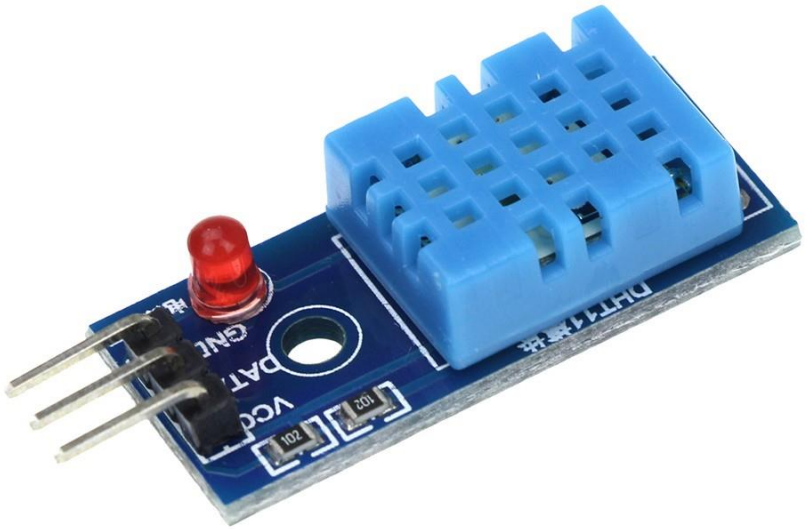
설계 목적

- 센서의 데이터를 수신하고 처리하는 로직 구현
- UART 모듈과 호환되도록 설계

Module Specification

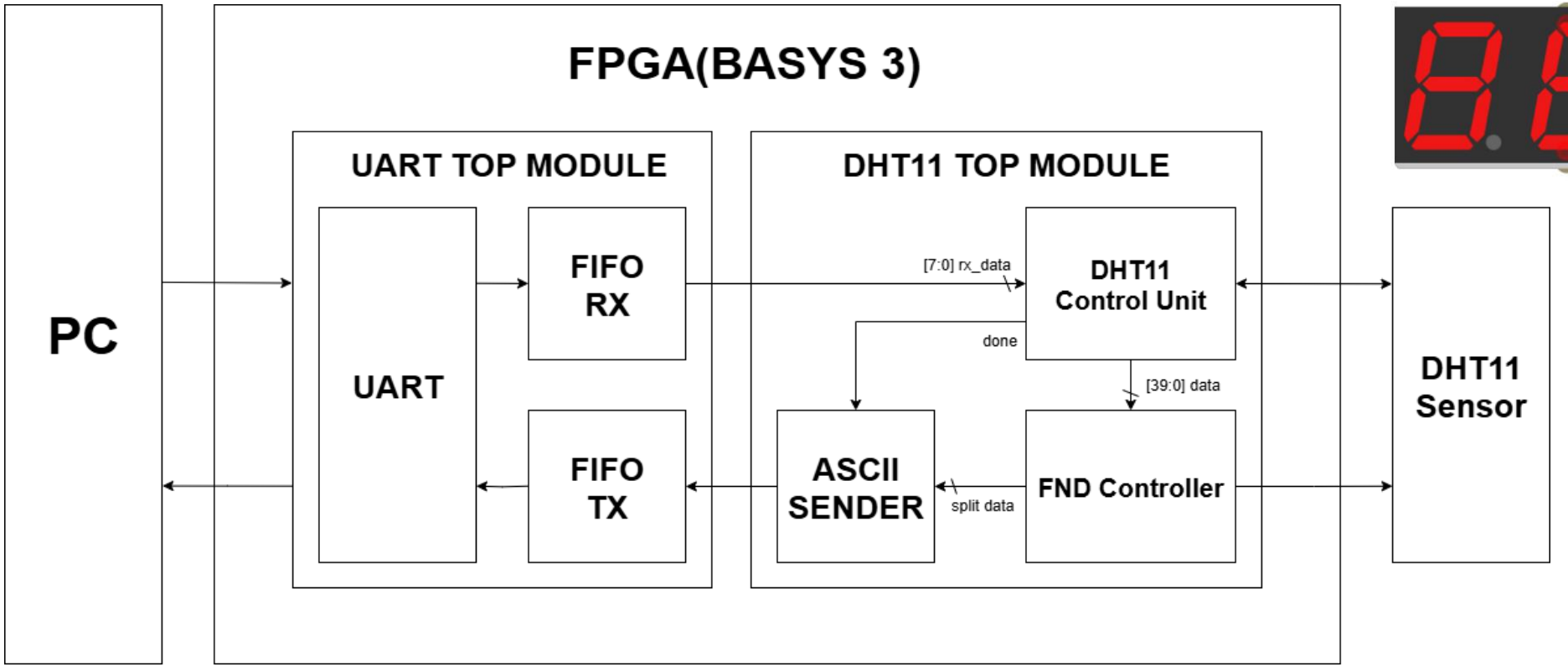
대략적인 스펙

- 공급 전압 = 3~5.5(V)
- 측정 범위 = 0~50℃ / 20~90%RH
- 정확도 = ±2℃ / ±5%RH
- 해상도 = 1
- 핀 = VCC, GND, DATA

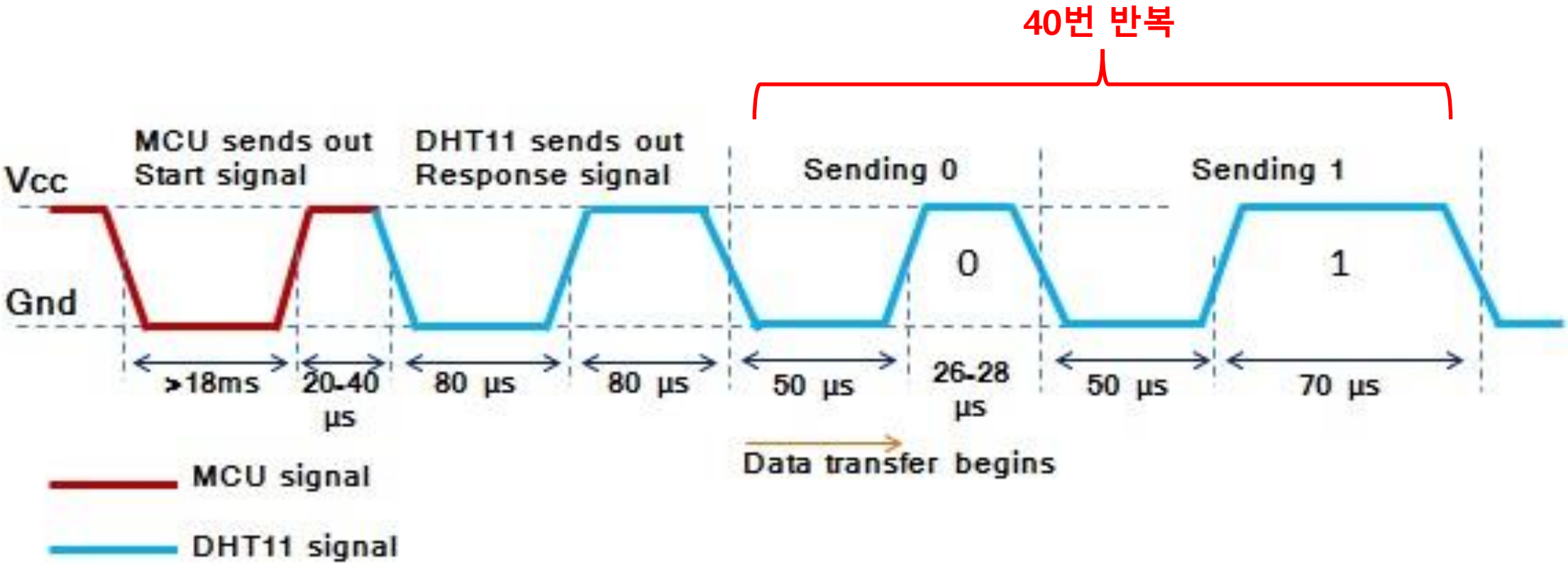


DHT11 Sensor

Block Diagram



Function



최종적으로 40비트의 데이터 값을 불러오고 저장함

Function

40-bit 데이터 형식(총 5 byte)

- 예) data = 40'b10101010_00001111_11100001_00110011_11001101

MSB부터 LSB까지

- 10101010 = 습도 정수부
- 00001111 = 습도 소수부
- 11100001 = 온도 정수부
- 00110011 = 온도 소수부
- 11001101 = Checksum 데이터

Function

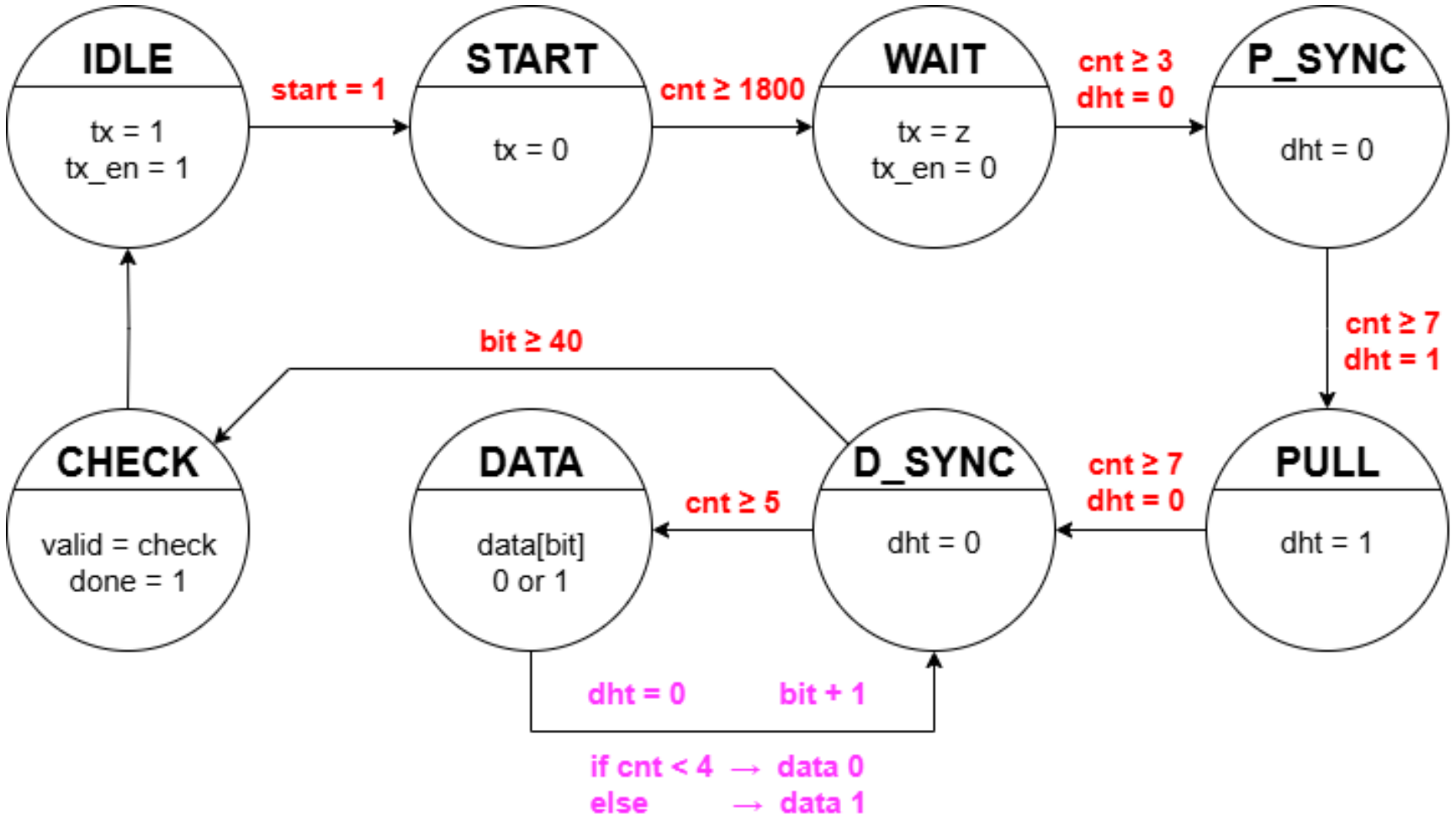
Checksum 원리

- data[39:32] + data[31:24] + data[23:16] + data[15:8] = data[7:0]
- 캐리 비트는 무시함

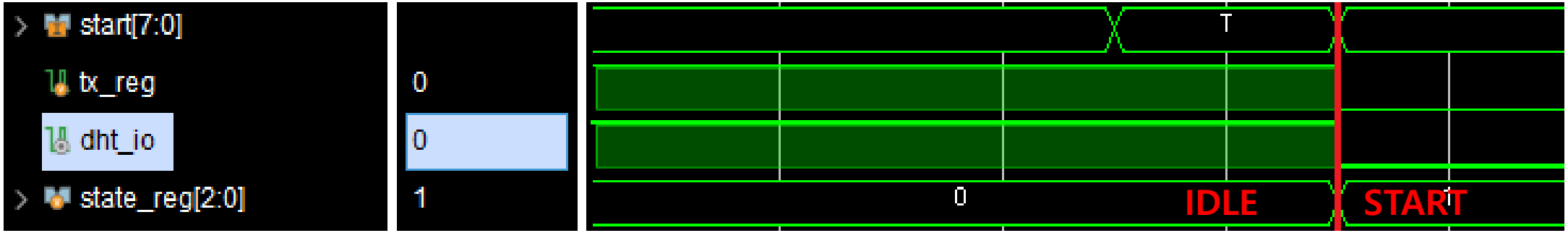
출력 로직

- Start 신호를 통해 센서 동작하고 데이터를 수신함
- 이후 Valid 신호가 참(Checksum 통과)이면 데이터를 출력하게 됨
- Done 신호를 매 동작 사이클마다 1 tick 만큼 출력함

FSM Chart


IDLE
tx = 1
tx_en = 1
start = 1
START
tx = 0
cnt ≥ 1800
WAIT
tx = z
tx_en = 0
cnt ≥ 3
dht = 0
P_SYNC
dht = 0
cnt ≥ 7
dht = 1
PULL
dht = 1
cnt ≥ 7
dht = 0
D_SYNC
dht = 0
cnt ≥ 5
DATA
data[bit]
0 or 1
bit ≥ 40
CHECK
valid = check
done = 1
dht = 0
bit + 1
if cnt < 4 → data 0
else → data 1

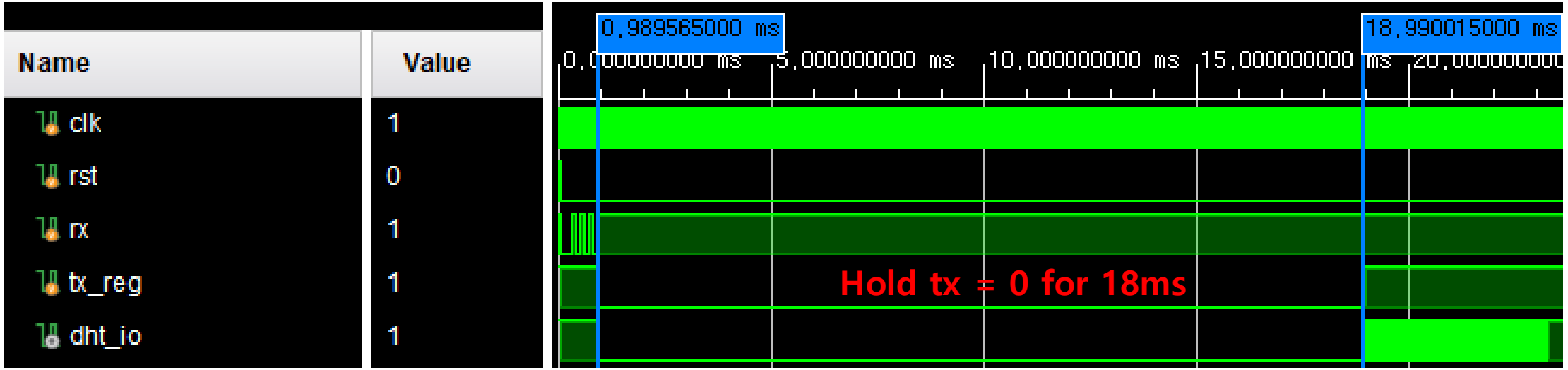
Simulation



IDLE → START if(start == “T”)

tx = 1 → 0

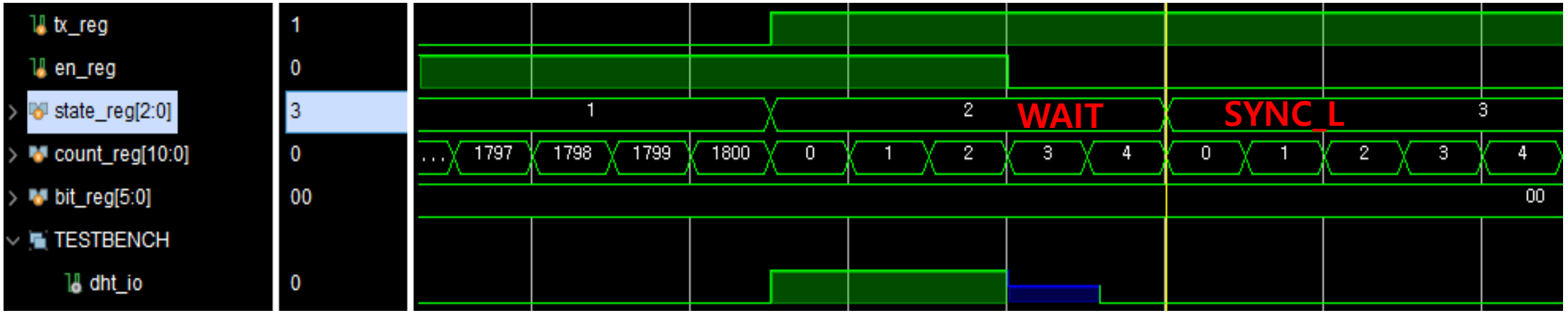
Simulation



START → WAIT if(cnt > 1800)

tx = 0 → 1

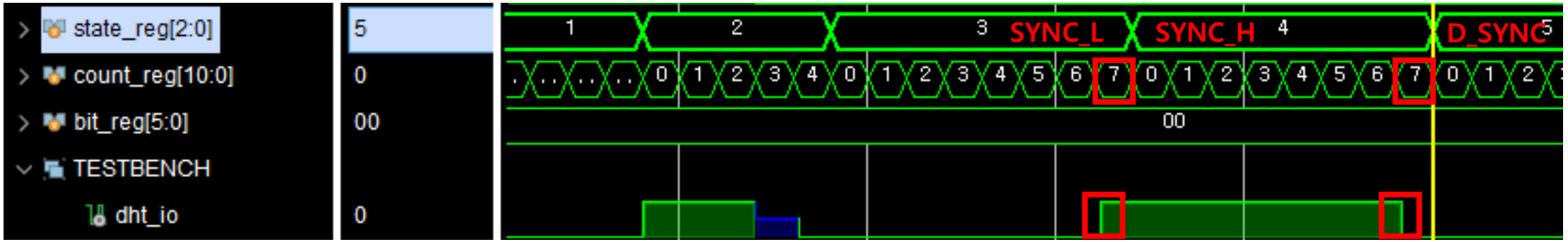
Simulation



WAIT → SYNC_L if(dht_io == 0)

tx = 1 → z / rx = z → 0

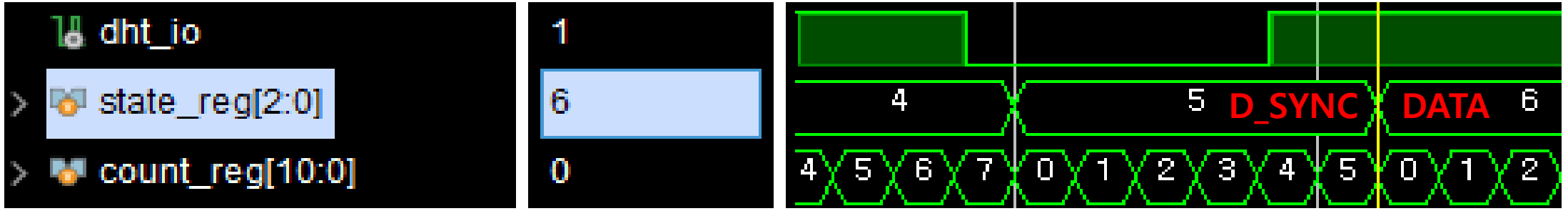
Simulation



SYNC_LOW → SYNC_HIGH
if(dht_io == 1 && cnt > 7)

SYNC_HIGH → DATA_SYNC
if(dht_io == 0 && cnt > 7)

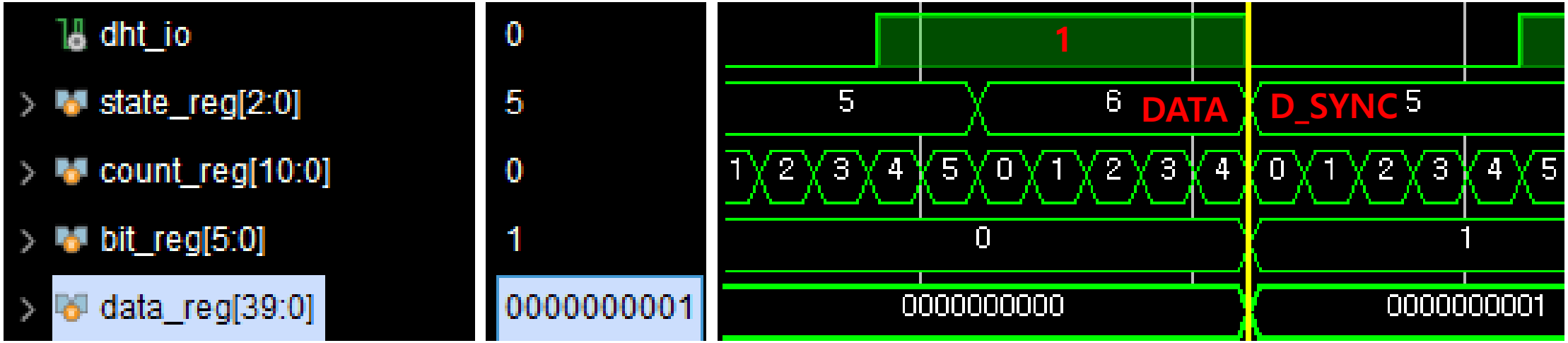
Simulation



D_SYNC → DATA

if(dht_io == 1 && cnt > 4)

Simulation

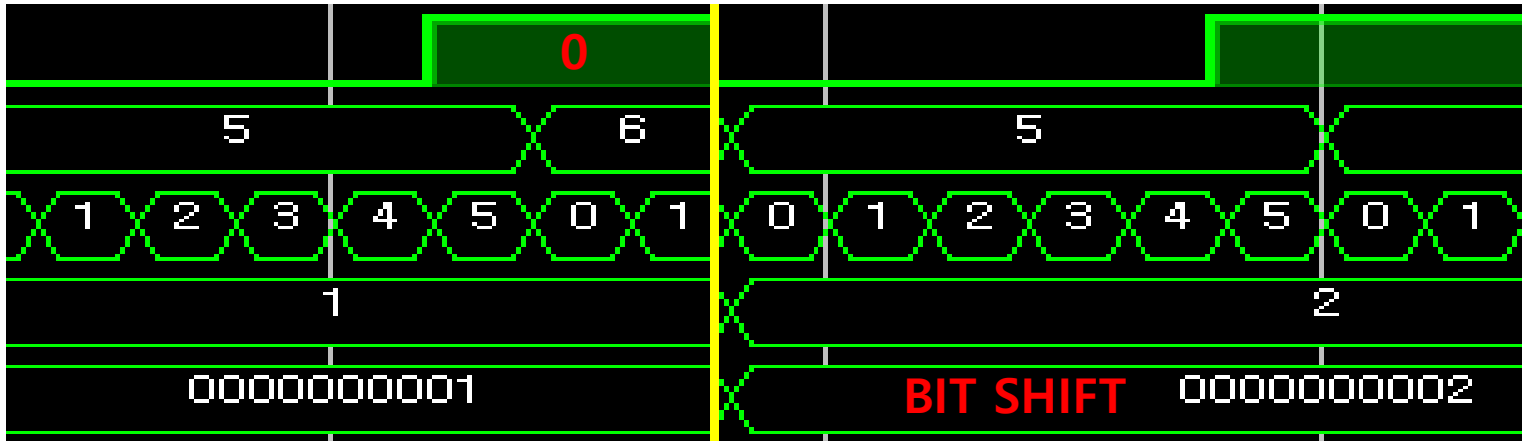


DATA → D_SYNC

if(dht_io == 0)

cnt > 4 → data = 1

Simulation

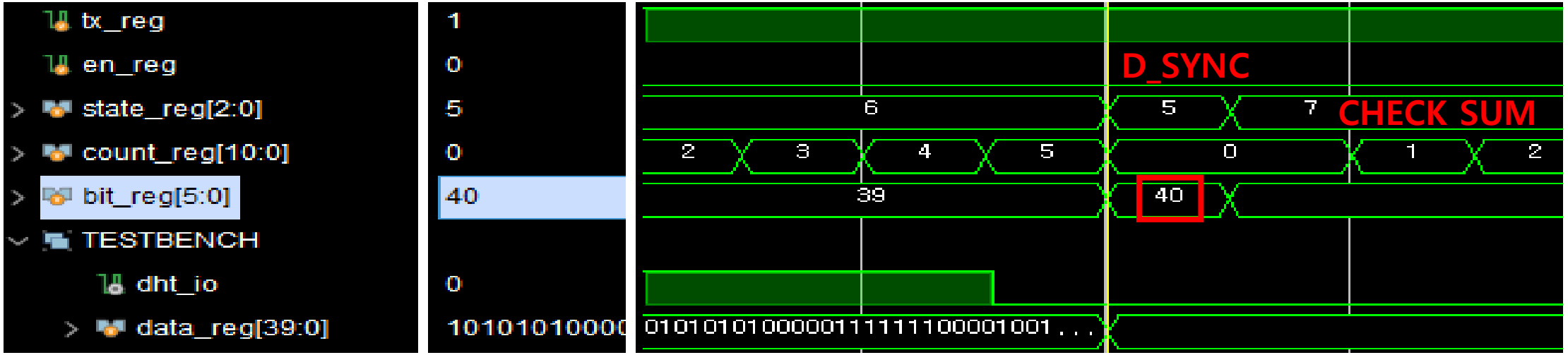


DATA → D_SYNC

if(dht_io == 0)

cnt < 4 → data = 0

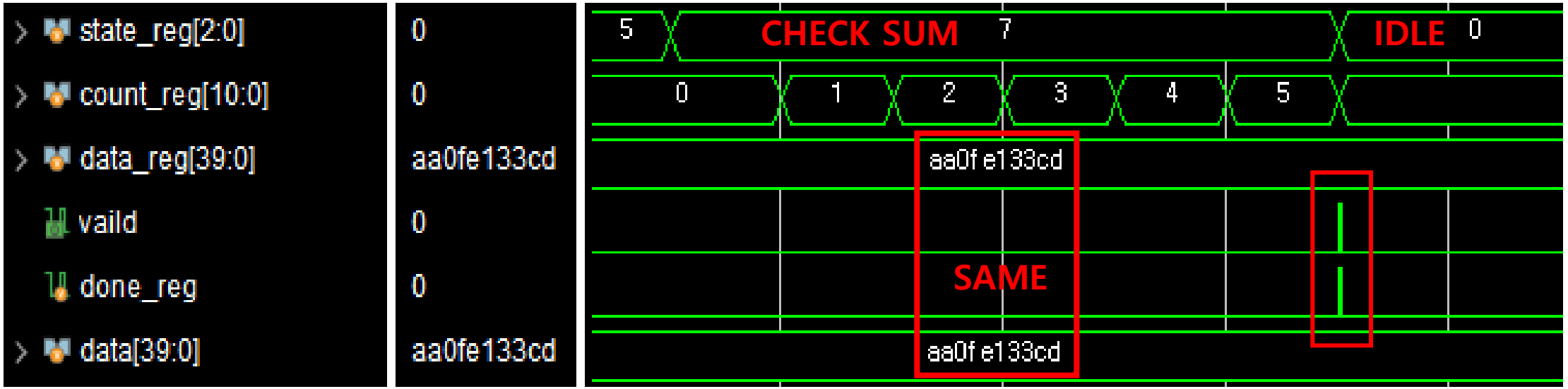
Simulation



D_SYNC → CHECK_SUM

if(bit_cnt == 40)

Simulation



CHECK_SUM → IDLE

if(cnt ≥ 5 && valid == 1)

done = 0 → 1 / tx = z → 1

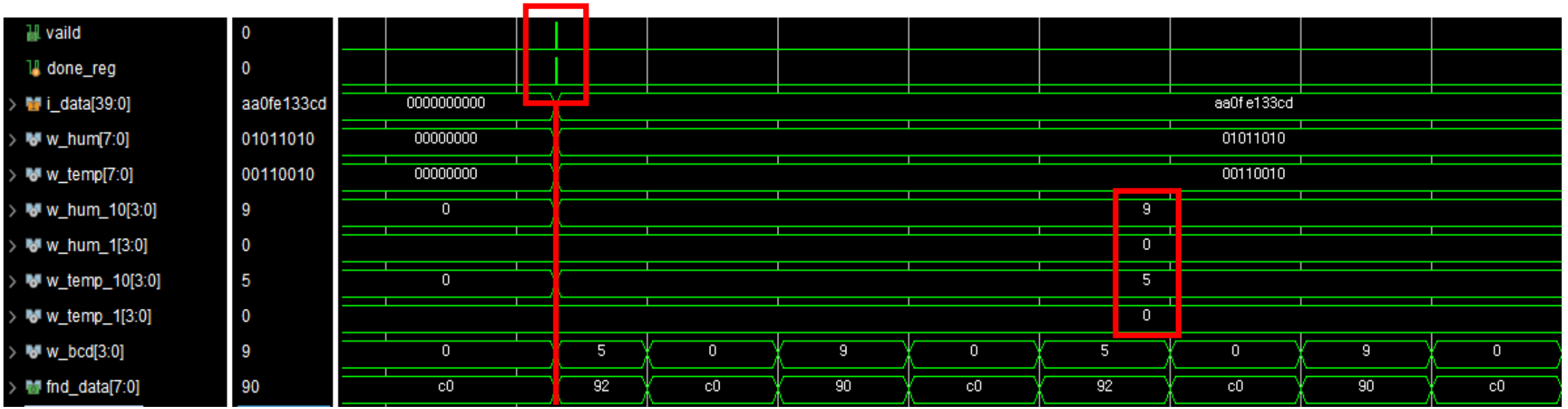
Simulation

```
Generated Random Data = 100001001000010011010110000010010110011
PASS (expected=100001001000010011010110000010010110011, received=100001001000010011010110000010010110011)
Generated Random Data = 000001101011100101111011000011011000 1101
PASS (expected=000001101011100101111011000011011000 1101, received=000001101011100101111011000011011000 1101)
Generated Random Data = 101100101100001010000100011001010001 0010
PASS (expected=101100101100001010000100011001010001 0010, received=101100101100001010000100011001010001 0010)
Generated Random Data = 000000001111001111100011000000100001101
PASS (expected=000000001111001111100011000000100001101, received=000000001111001111100011000000100001101)
Generated Random Data = 001110110010001111111000101110110001111101
PASS (expected=001110110010001111111000101110110001111101, received=001110110010001111111000101110110001111101)
```

```
if (received_data == data) begin
    $display("PASS (expected=%b, received=%b)", data, received_data);
end else begin
    $display("FAIL (expected=%b, received=%b)", data, received_data);
end
```

랜덤 데이터 테스트 통과

Simulation



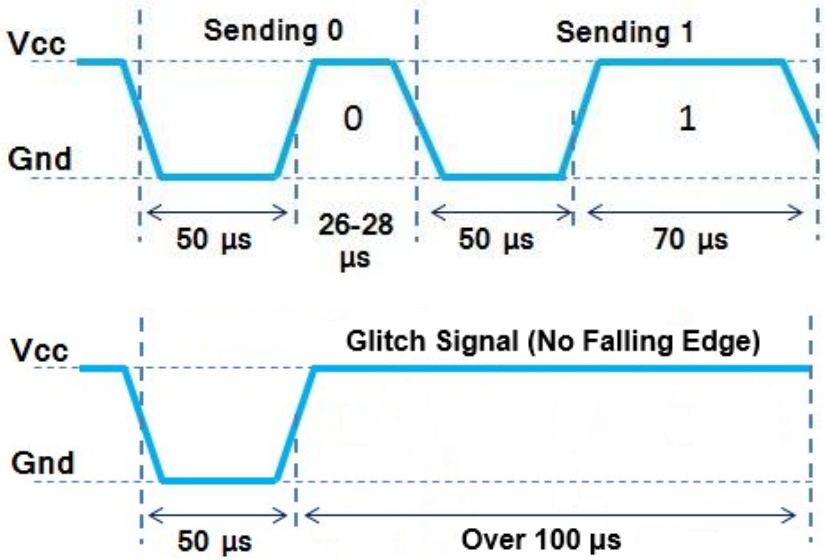
[39:0] i_data → data split → digit split → bcd_decoder → segment

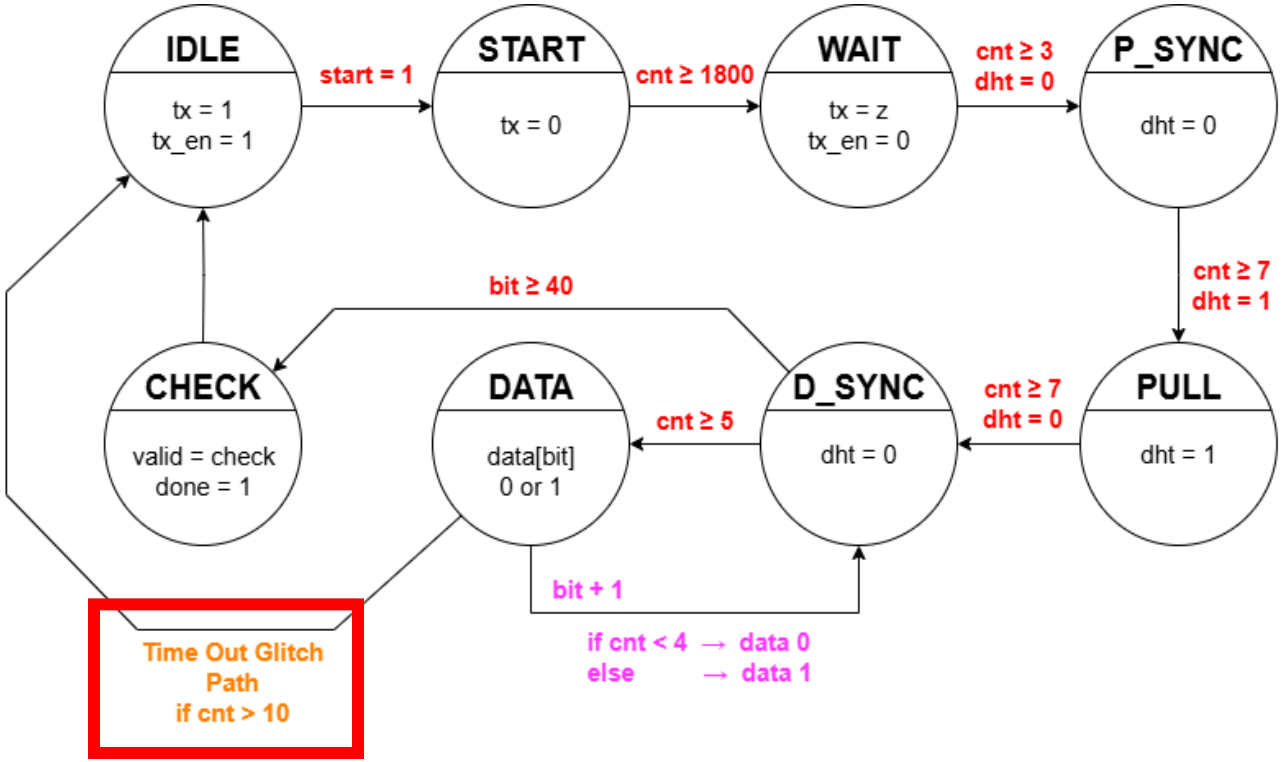
FND data = c0(0), 92(5), 90(9)

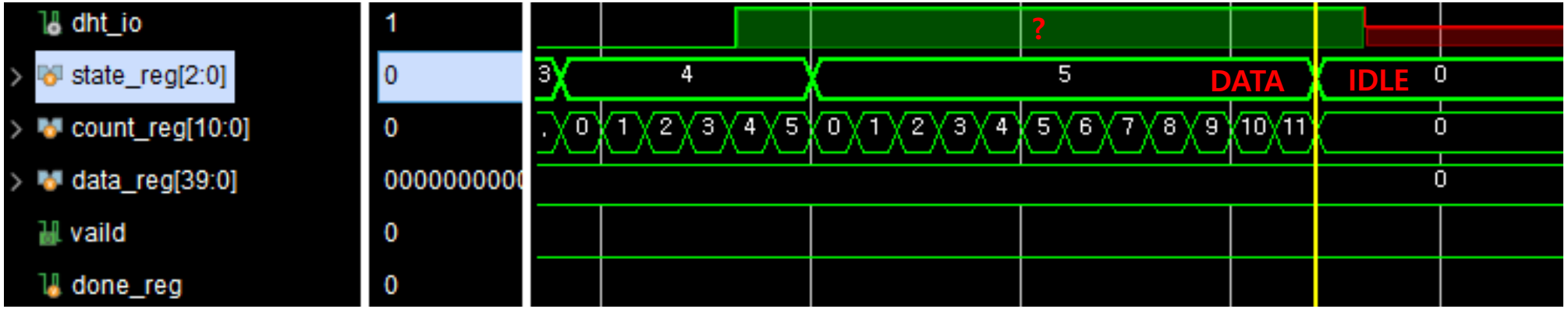
Trouble Shooting

센서 자체의 오류 신호

- FSM에 방어 코드 추가







데이터의 High 신호가 오래 유지될 때 리셋 후 IDLE 상태로 천이

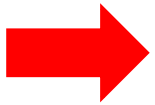
Trouble Shooting

타이밍 여유 개선

- 계산 과정을 분할함

```
CHECK_SUM: begin
    if (tick) begin
        if (count_reg >= 5) begin
            vaild_next = (total[7:0] == data_reg[7:0]);
            final_data_next = (vaild_next) ? data_reg : 0;
            done_next  = (vaild_next) ? 1 : 0;
            tx_next    = 1;
            en_next    = 1;
            count_next = 0;
            state_next = IDLE;
        end else begin
            count_next = count_reg + 1;
        end
    end
end
```

Worst Negative Slack (WNS):	2.428 ns
Worst Hold Slack (WHS):	0.036 ns



```
CHECK_SUM: begin
    if (tick) begin
        if (count_reg >= 5) begin
            vaild_next = (total[7:0] == data_reg[7:0]);
            state_next = DONE;
        end else begin
            count_next = count_reg + 1;
        end
    end
end

DONE: begin
    final_data_next = (vaild_reg) ? data_reg : 0;
    done_next  = (vaild_reg) ? 1 : 0;
    vaild_next = 0;
    tx_next    = 1;
    en_next    = 1;
    count_next = 0;
    state_next = IDLE;
end
```

Worst Negative Slack (WNS):	4.161 ns
Worst Hold Slack (WHS):	0.073 ns

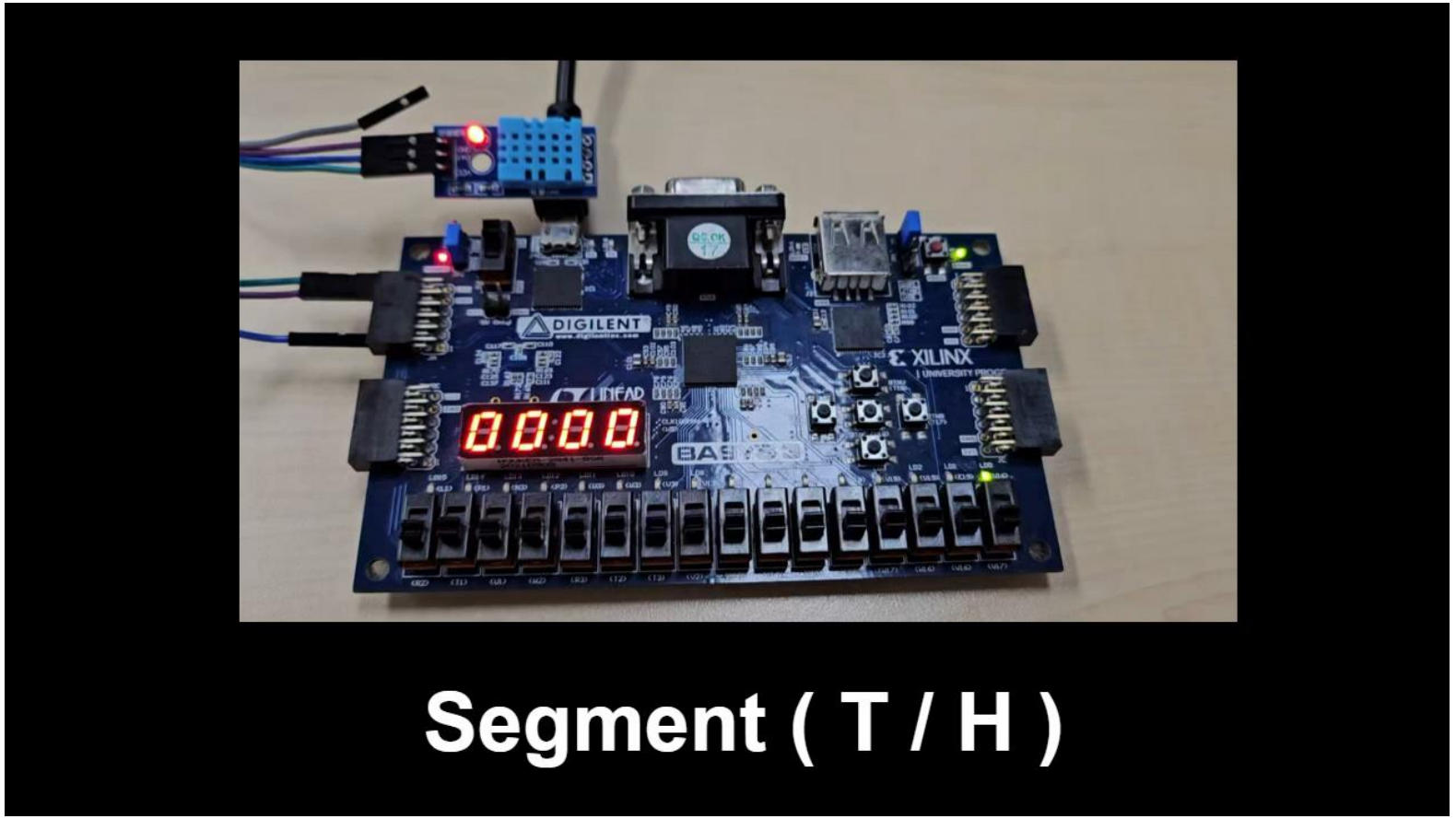
Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)
265	233	98	249

+ 1 증가

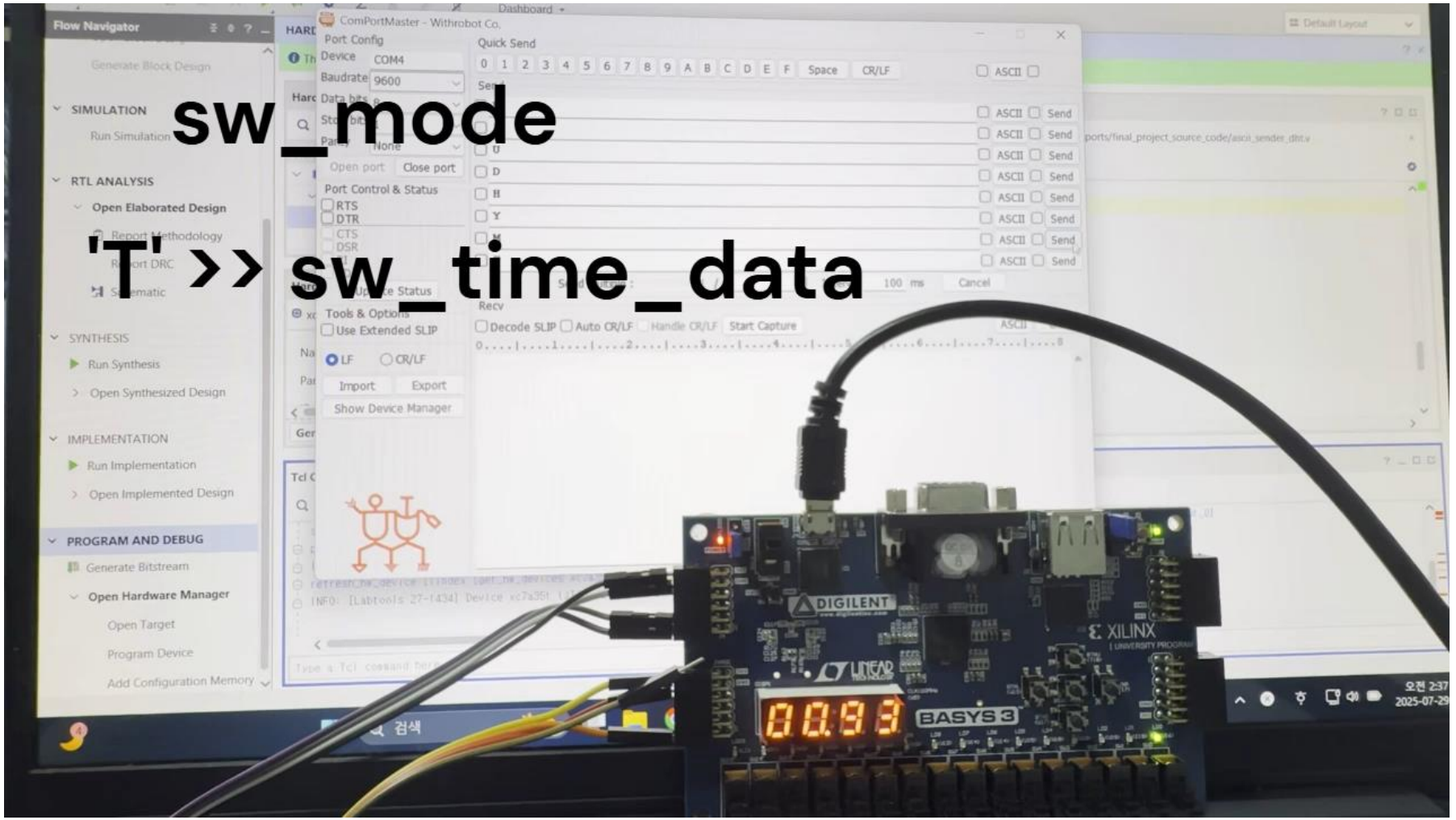
Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)
266	234	99	250

Good Trade-off

동작 영상



동작 영상


sw_mode
'T' >> sw_time_data
Flow Navigator
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
Open Target
Program Device
Add Configuration Memory
ComPortMaster - Withrobot Co.
Port Config
Device COM4
Baudrate 9600
Open port
Close port
Port Control & Status
RTS
DTR
CTS
DSR
Quick Send
Send
Recv
Decode SLIP
Auto CR/LF
Start Capture
Use Extended SLIP
LF
CR/LF
Import
Export
Show Device Manager
DIGILENT
XILINX
BASYS 3

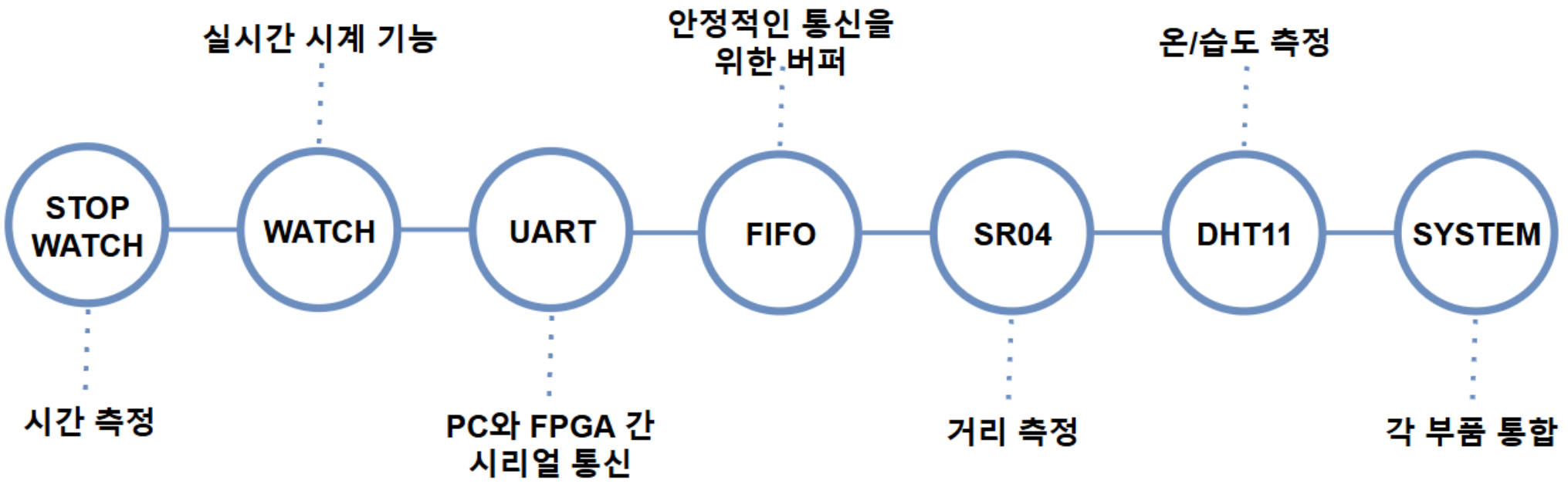
고찰

박찬호 : Setup time violation을 발생했을 때 Timing report에서 어느 부분에서 violation이 났는지 확인하고 발생 원인을 찾고 상황에 맞게 문제 해결을 경험할 수 있었습니다.

석경현 : 모듈 설계 시 timing 문제에 대해 항상 유의해야 하고 최대한 많은 case에 대응하여 반복적인 simulation을 통해 glitch가 발생하지 않는 시스템을 구현해보는 과정에서 많은 것을 배웠습니다.

Summary

Verilog HDL을 기반으로 StopWatch, Watch, 초음파 센서, 온/습도 센서 컨트롤러를 구현하고 UART, 버튼을 통해 제어 가능한 임베디드 시스템 구현



3조 TEAM_PROJECT

-End-