Digital Clock With Sensor

Semicon_Academi 27 7조 / 조현철, 한지윤, 허현강, 황주빈

- 다양한 기능을 포함한 디지털 시계 설계
- IoT 시스템 설계













시간

온도

거리

목차

1. 개요

- 1-1. 목적
- 1-2. 기능 및 목표
- 1-3. FPGA Basys3

2. 프로젝트

- 2-1. 동작 영상
- 2-2. 전체 구조
- 2-3. Clock Sensor
- 2-4. 시스템 구성도 및 데이터 처리

3. UART

- 3-1. Block Diagram
- 3-2. Cmd 설명
- 3-3. Clock_timer simulation
- 3-4. DHT11 simulation
- 3-5. Conclusion&동작영상

4. DHT11

- 4-1. 구조
- 4-2. Simulation

5. UltraSonic_Sensor

- 5-1. 개요
- 5-2. 기초이론
- 5-3. Block Diagram & ASM Chart
- 5-4. Simulation 검증
- 5-6 동작검증

6. 결론

6.1 개선 사항 및 보완점



프로젝트 개요 / 목적



프로젝트 개요 / 기능 및 목표



핵심 기능

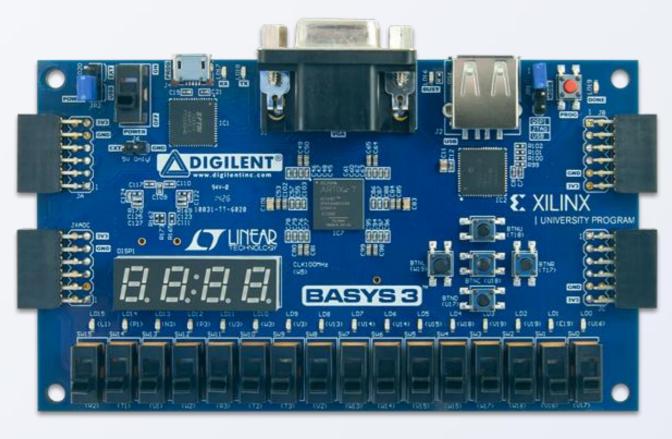
- ✓ 시계, 스톱워치, 타이머, 거리,온도 및 습도 측정
- ✔ PC와의 통신으로 정보 확인 가능
- ❷ 안정적인 데이터 전송



프로젝트 목표

- ✓ 시스템 설계의 안정적인 데이터 통신 기술을 구현
- ☑ 다양한 환경 데이터 정보를 수집
- ☑ 이후 스마트 IoT 환경 구축을 위한 발판

프로젝트 개요 / FPGA basys3



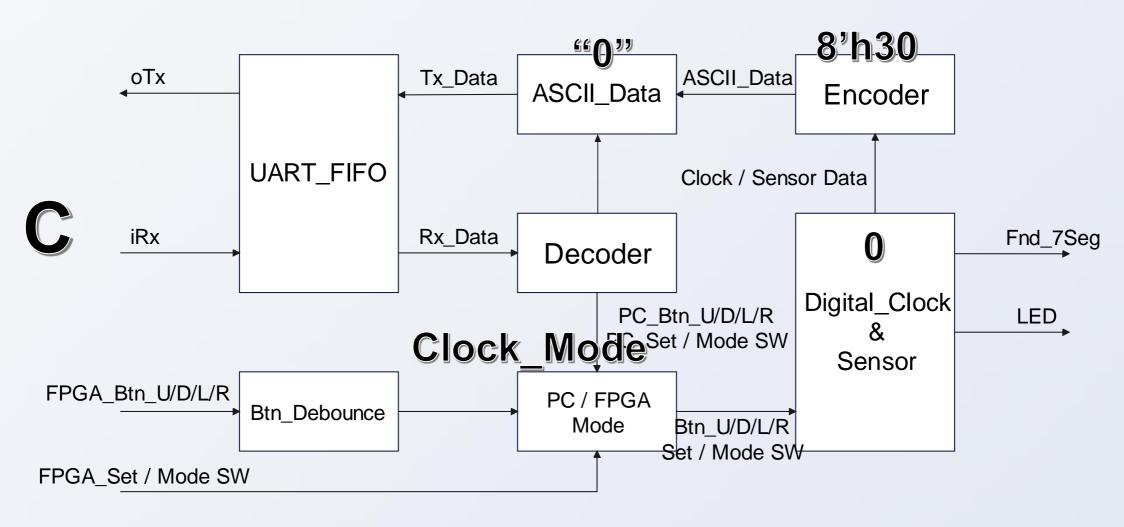




Used Board

- (L) 100MHz Input Clock
- (g) UART
- **9** 7 SW / 14LED
- ♦ 5 Button
- T 7 Segment
- ? 2 Pmod Connector

프로젝트 / 전체 구조 및 동작 방식



프로젝트 / 시스템 구성도 및 데이터 처리

입력부(Input Section)

중앙 처리부 (**Processing Section**) 출력 및 전송부 (Output & Transmission)



Digital Clock



MCU / FPGA



Display



DHT11 Sensor



FIFO Buffer



UART 통신



Ultrasonic Sensor

2단계:데이터 통합 및 처리



PC/외부 장치

3단계: 출력 및 전송

1단계: 데이터 수집

UART 통신

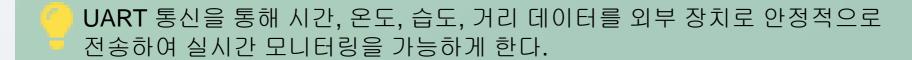


🧡 간결한 연결

2개의 신호선 (Rx / Tx)로 통신 가능 <u>|</u>|으 호환성

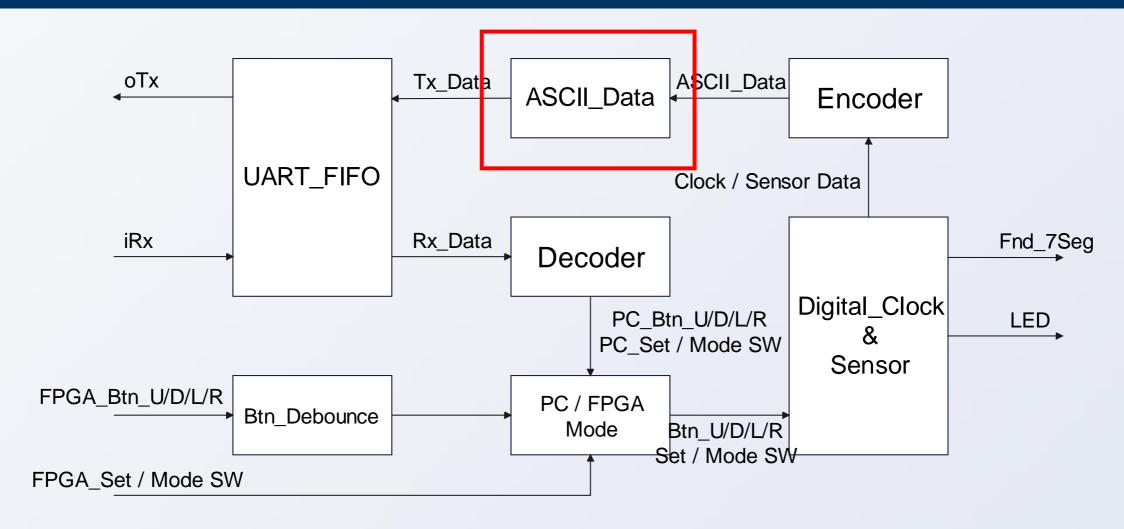
표준화된 통신 방식 다양한 장치와 쉽게 연결 데이터 안정성

FIFO 버퍼 사용 안정적인 데이터 전송

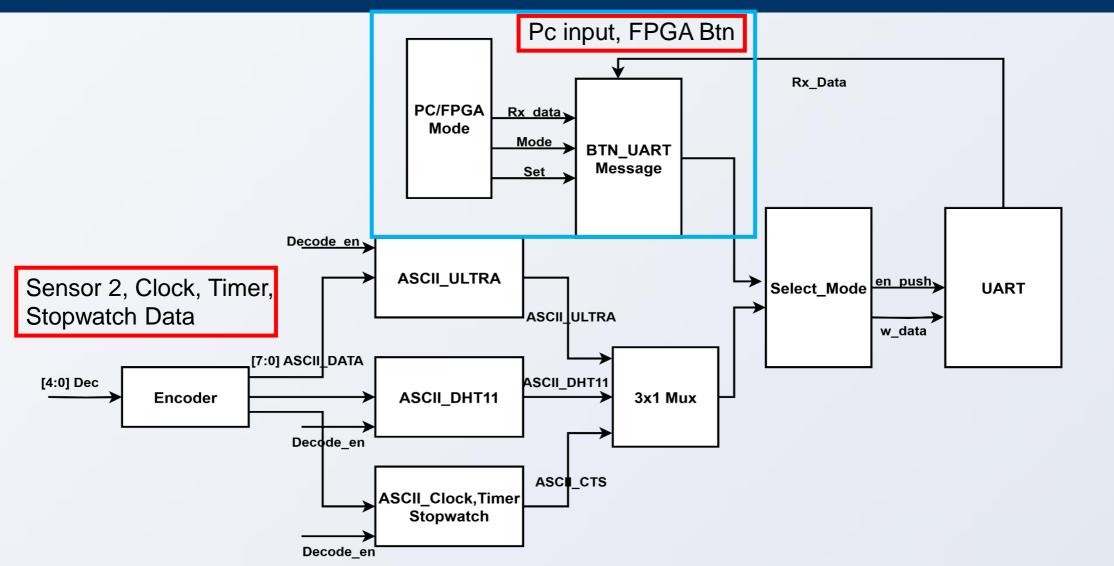


UART

Block Diagram



Block Diagram(specific)



UART cmd

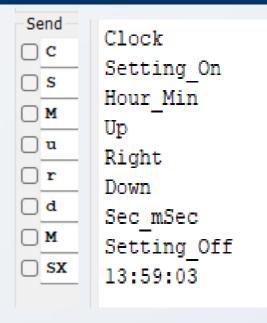
1. Cmd에 따른 Pc 출력 Send ❖ "W" - "Stopwatch" Clock 13:26:37 "U" - "Ultrasensor" Stopwatch ❖ "D" - " DHT11 " WX ❖ "X" - "선택된 모드에 따른 값 출력" 00:00:00 Timer 00:00:00 Ultrasensor DISTANCE : 0016cm 28.0'C40.0%

각 모드 별 cmd에 따른 Pc 출력



- "C" -> Clock
- "S" -> Setting_On/ Setting_Off
- "M" -> Hour_Min/ Sec_mSec
- "u,d,l,r" -> up, down, left, right
- "X" -> 13:59:27



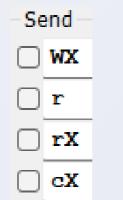


같은 문자에 서로 다른 문자열 출력?

- ✓ 내부적으로 카운터 사용
- ✓ 누른 횟수를 저장
- ✓ LSB만 읽어 홀짝에 따라 출력 전환

```
" Stopwatch "
```

- "W" -> Stopwatch
- "r" -> Stop_Watch_RUN/ Stop_Watch_STOP
- "c" -> Stop_Watch_Clear



Stopwatch 00:00:00 Stop_Watch_RUN Stop_Watch_STOP 00:04:75 Stop_Watch_Clear 00:00:00

각 모드 별 cmd에 따른 Pc 출력

- **☆** "Timer"
- "T" Timer
- "S" Setting_On, Setting_Off
- "u,d" Up, Down
- "r" Timer_Run, Timer_Stop
- " Ultrasensor&DHT11 "
- "U" "Ultrasensor"
- "D" "DHT11"

센서는 자동으로 값을 반환

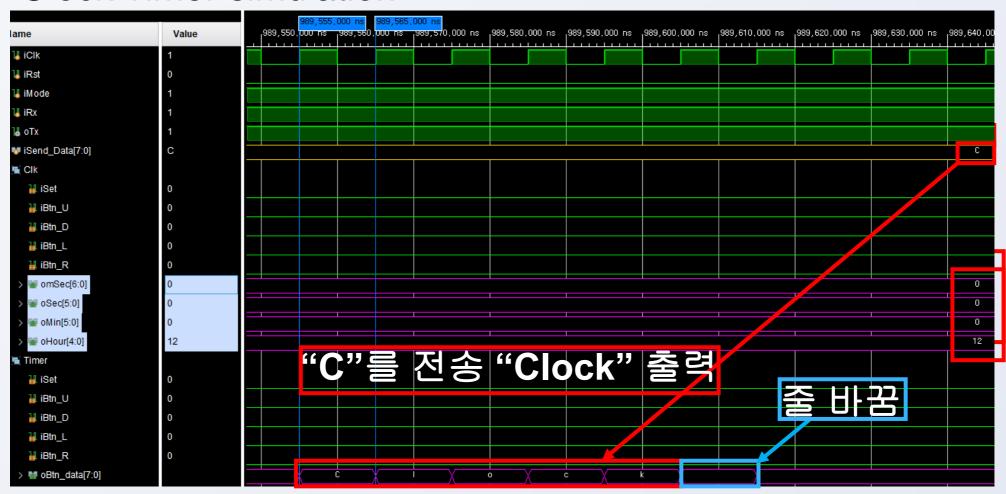
같은 문자가 기능에 따라 서로 다른 문자 출력?

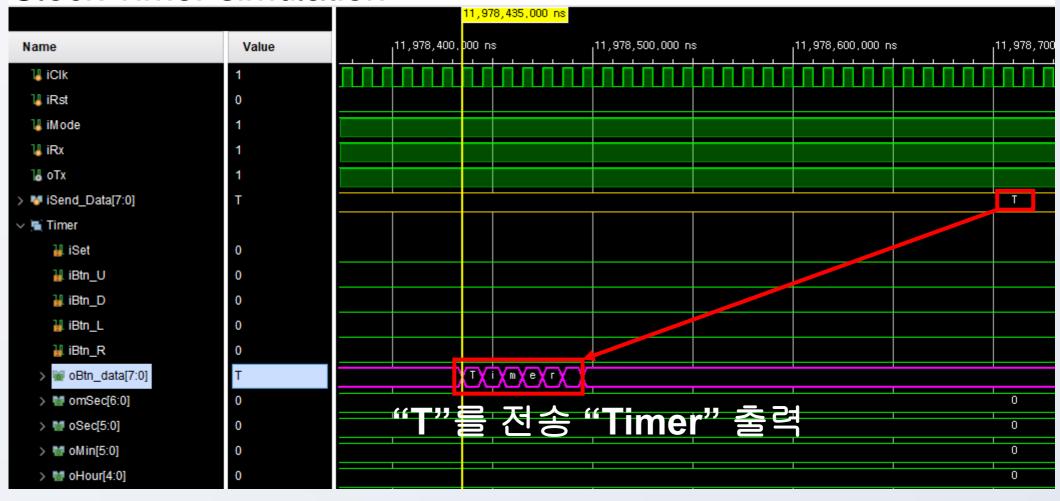
- ✓ case문을 사용하여 같은 문자가 들어와도
- ✓ 선택된 모드마다 다른 문자가 출력되도록 설계

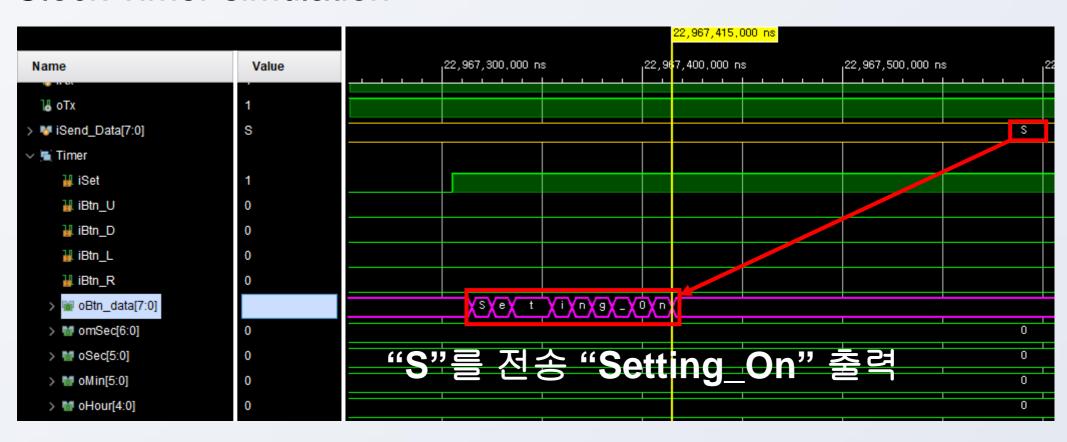
Send			
	TX		
	SX		
	u		
	SX		
	r		
	r		
	X		
	сX		
			Send

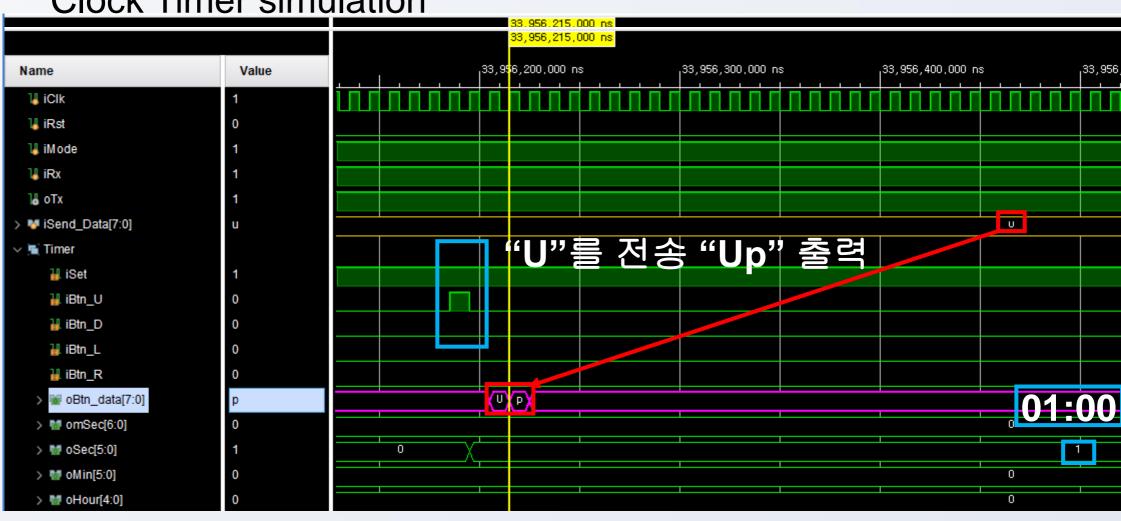
```
Timer
00:00:00
Setting On
00:00:00
Up
Up
Setting Off
00:00:02
Timer RUN
Timer STOP
00:00:01
Timer Clear
00:00:00
```

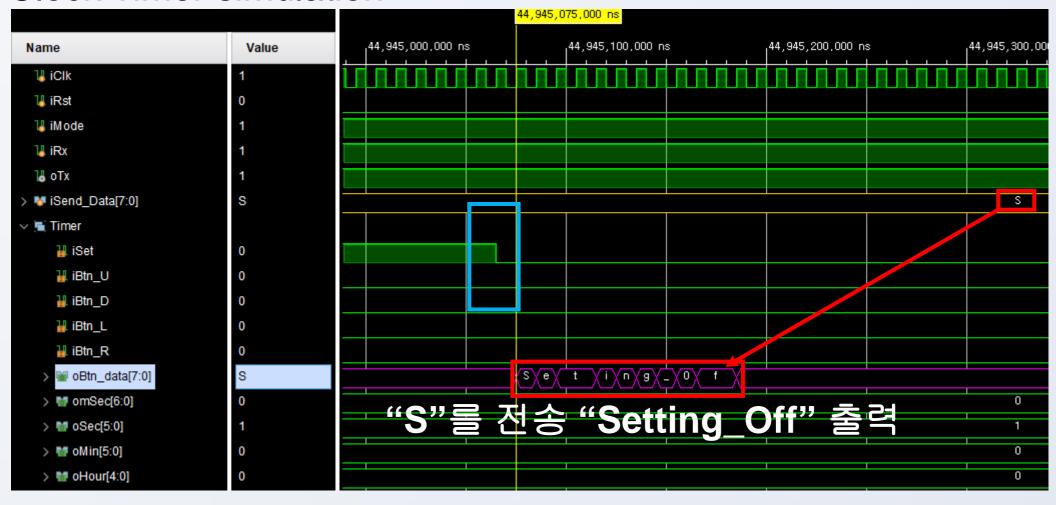
```
1. "C"를 전송 Clock 출력(Clock 모드)
2. "T"를 전송 Timer 출력 (Timer 모드)
3. "S" -> Setting_On출력 (세팅 모드)
4. "u" -> "Up" (1초 증가)
5. "S" -> "Setting_Off" (세팅 모드 종료)
6. "r" -> "Timer_Run" (Timer 시작)
```

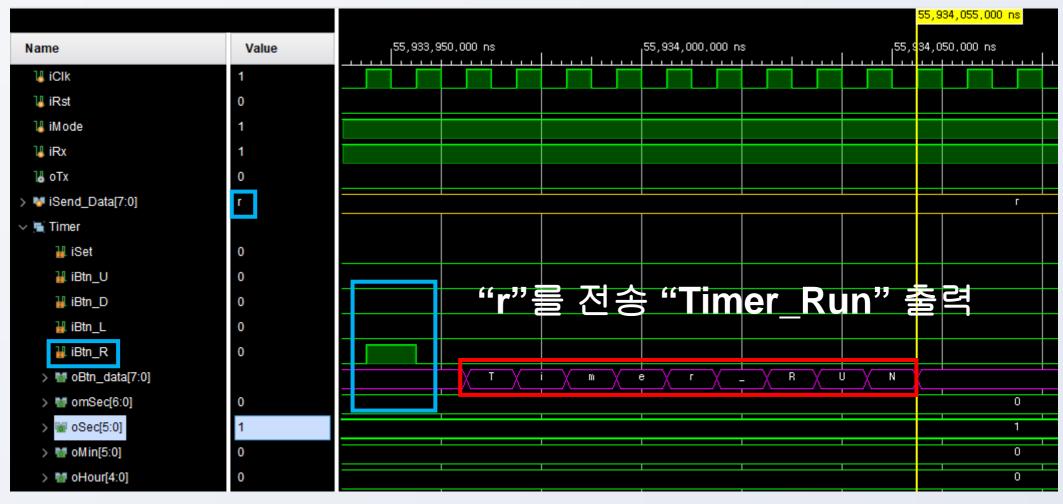




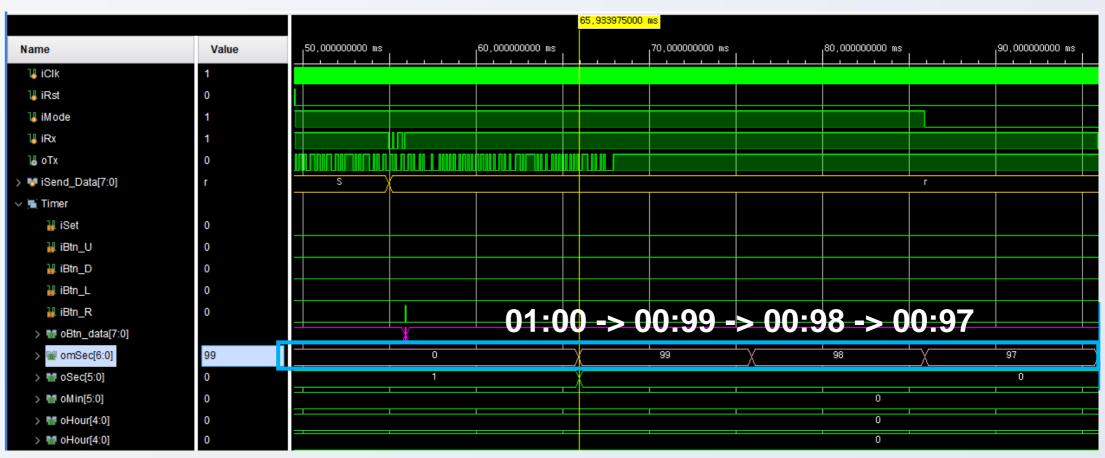








Clock Timer simulation

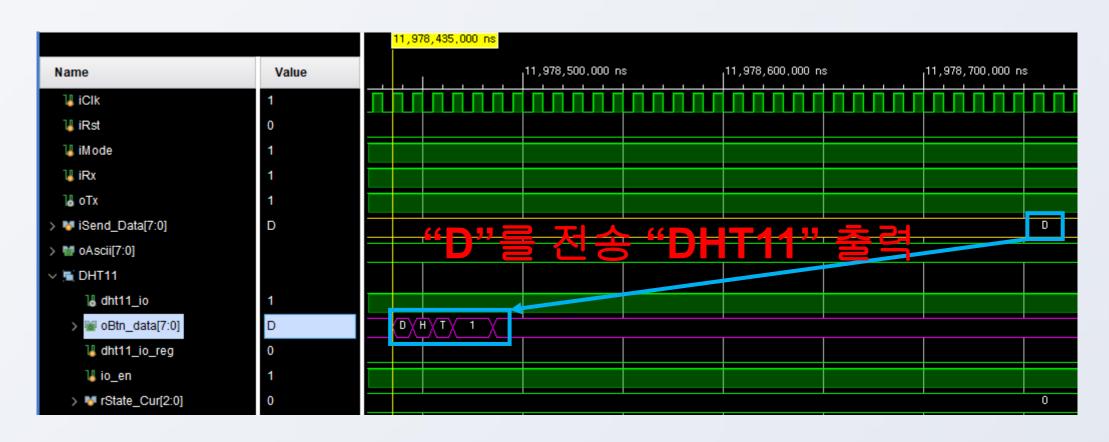


입력에 따라 원하는 명령어가 출력되고 Timer가 잘 동작함을 확인

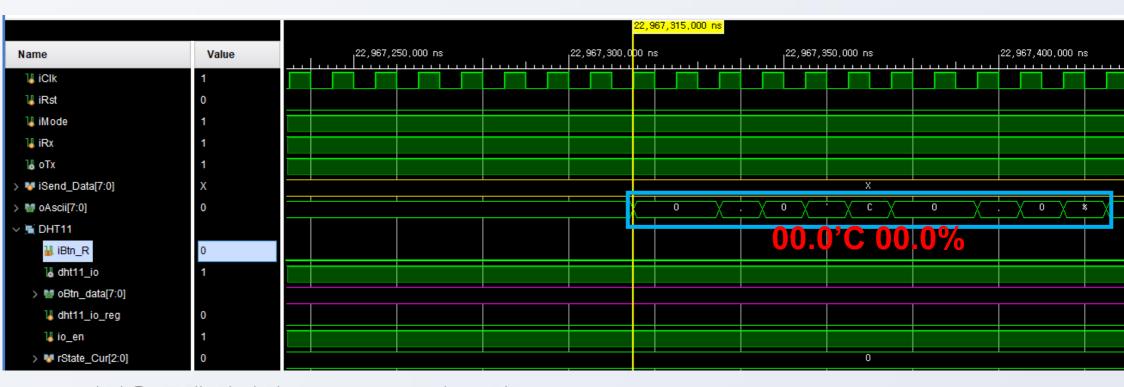
DHT11 simulation

- 1. "D"를 전송 DHT11 출력(DHT11 모드)
- 2. "X" -> 00.0'C, 00.0% (Start 신호 X)
- 3. Start 신호 인가 후 "X" -> 30.0'C 70.0%

DHT11 simulation

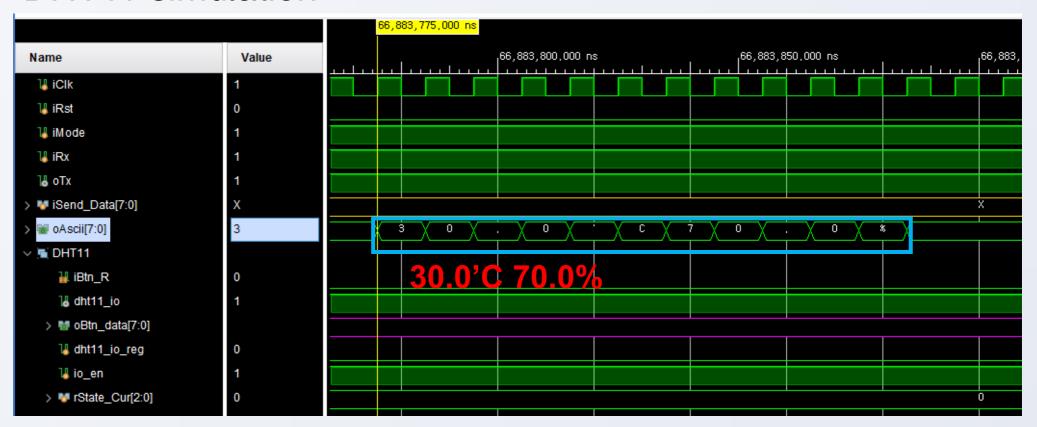


DHT11 simulation



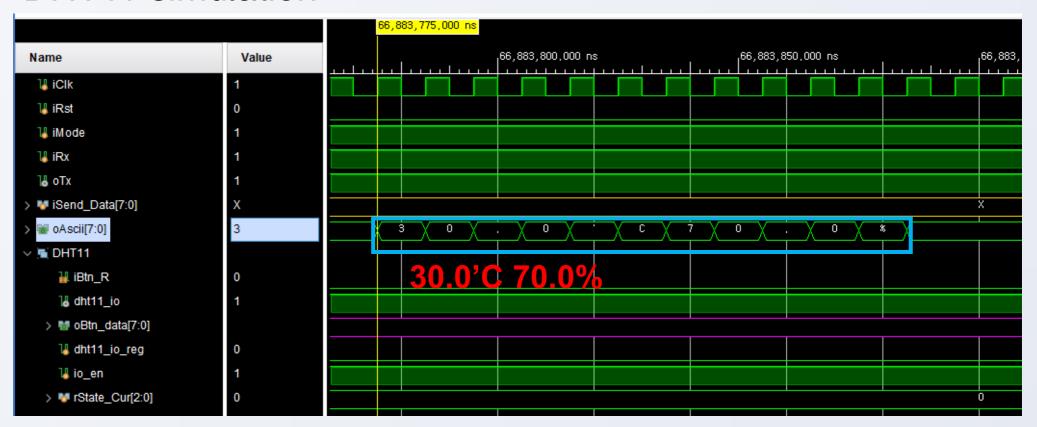
"X" 입력을 통해 확인한 DHT11 온도/ 습도 값 "D"입력 후 1초가 지나지 않았고 start 입력 x -> 00.0'C 00.0%

DHT11 simulation



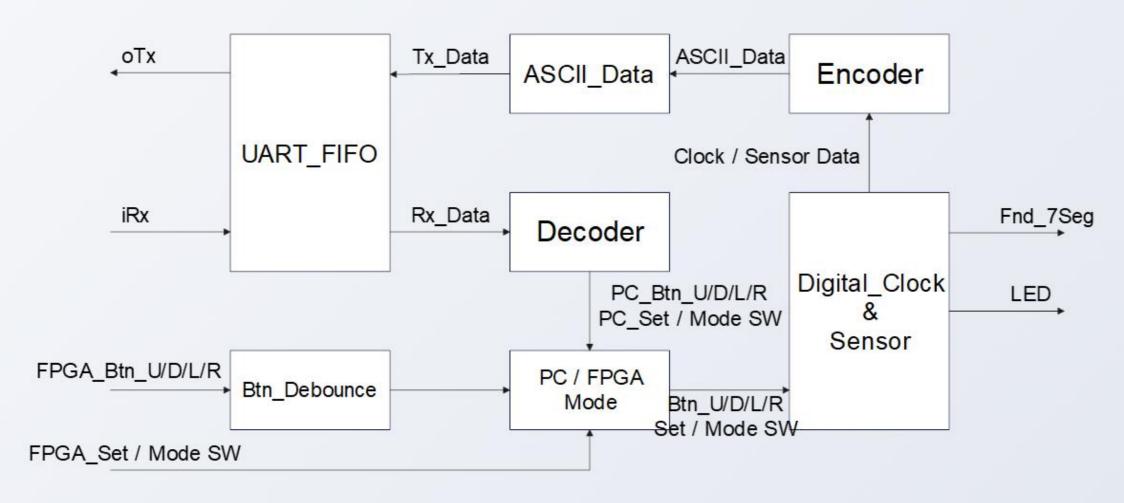
"r"(Start 신호)를 준 뒤 "X"하여 출력된 온/습도값 30.0'C 70.0% 온/습도는 1초에 한 번씩 값을 읽어오기에 따로 "r"입력 시 출력값을 내보내진 않게 설계

DHT11 simulation

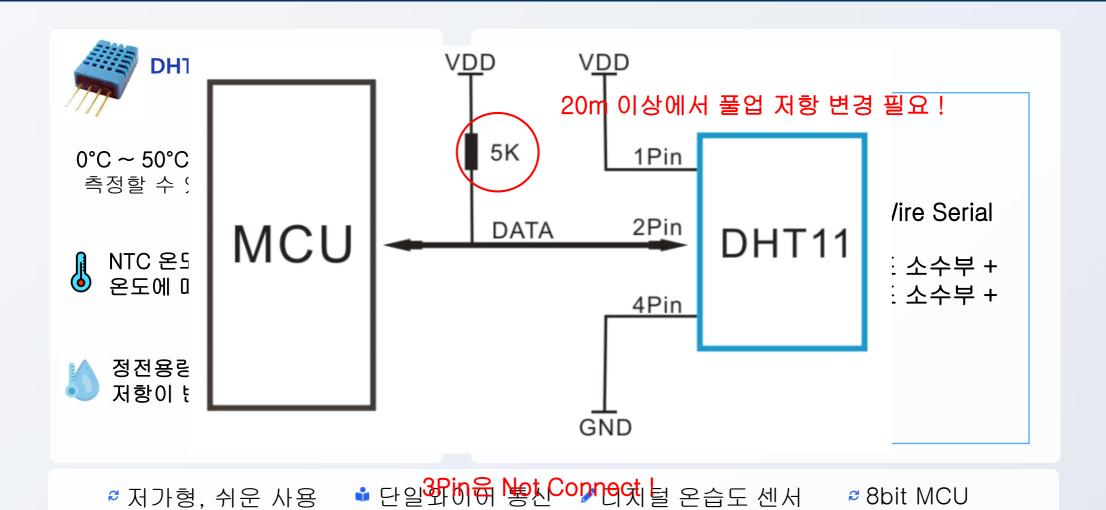


"r"(Start 신호)를 준 뒤 "X"하여 출력된 온/습도값 30.0'C 70.0% 온/습도는 1초에 한 번씩 값을 읽어오기에 따로 "r"입력 시 출력값을 내보내진 않게 설계

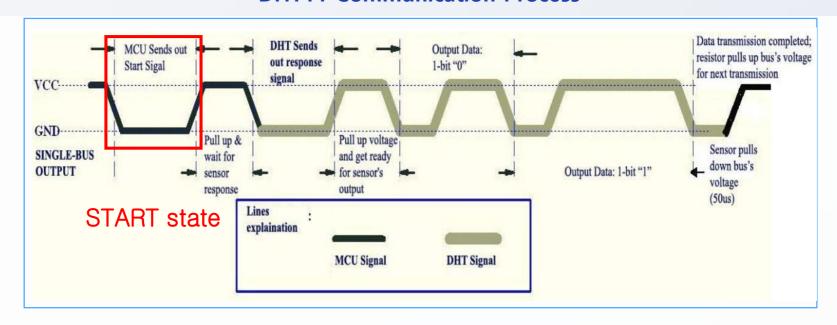
Conclusion



Temperature & Humidity Sensor

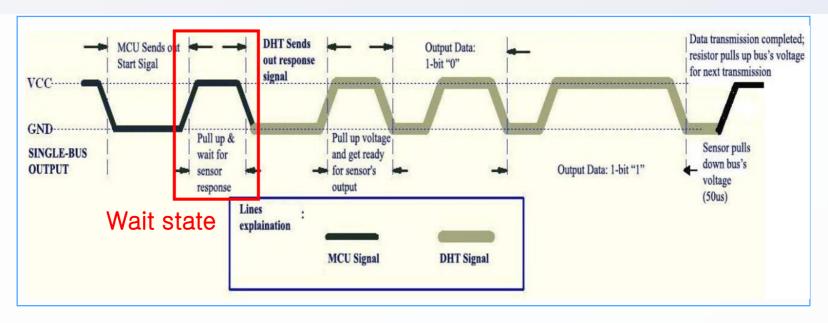


DHT11 Communication Process



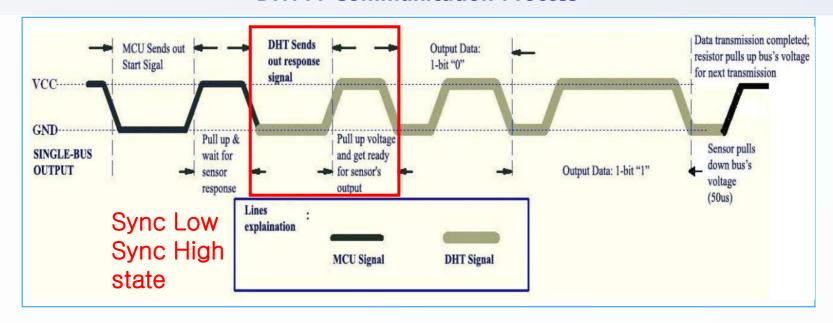
1. FPGA 보드에서 inout 출력으로 Start Signal (Low) 최소 18ms 인가

DHT11 Communication Process



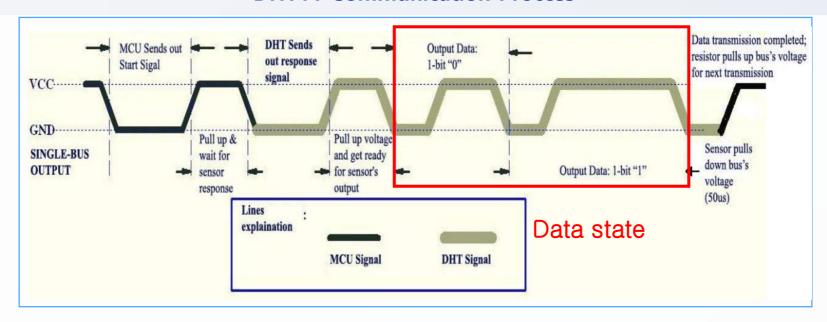
2. FPGA 보드에서 inout 출력으로 High Signal 20us~40us 인가 이때 30us 지점에서 tristate buffer enable off

DHT11 Communication Process



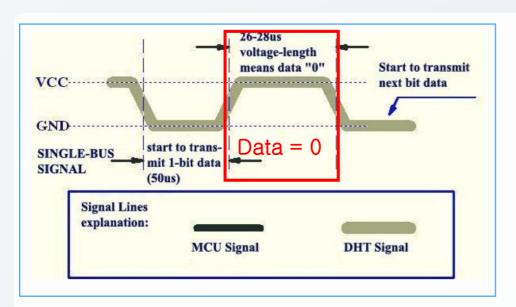
3. DHT11 센서가 Signal을 받았다는 신호로 Low, High Signal을 80us동안 전송

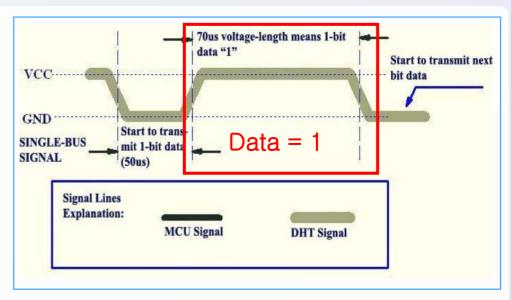
DHT11 Communication Process



4. DHT11이 40bit의 Data를 전송

DHT11 Communication Process

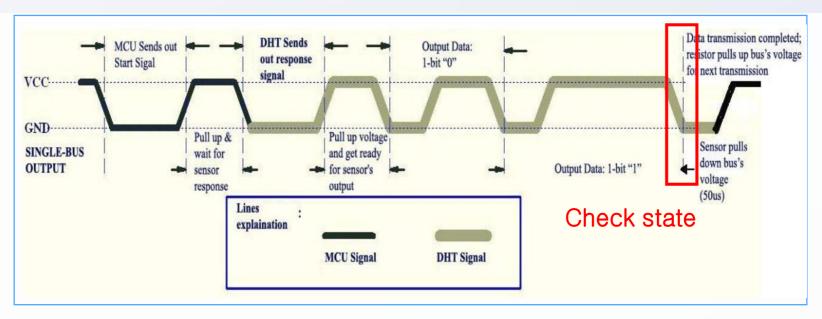




5. DHT11이 전송을 시작하기 전에 50us 동안 Low Signal을 보낸 후 Pull up한 시간에 따라 Data 1과 0 구분 가능

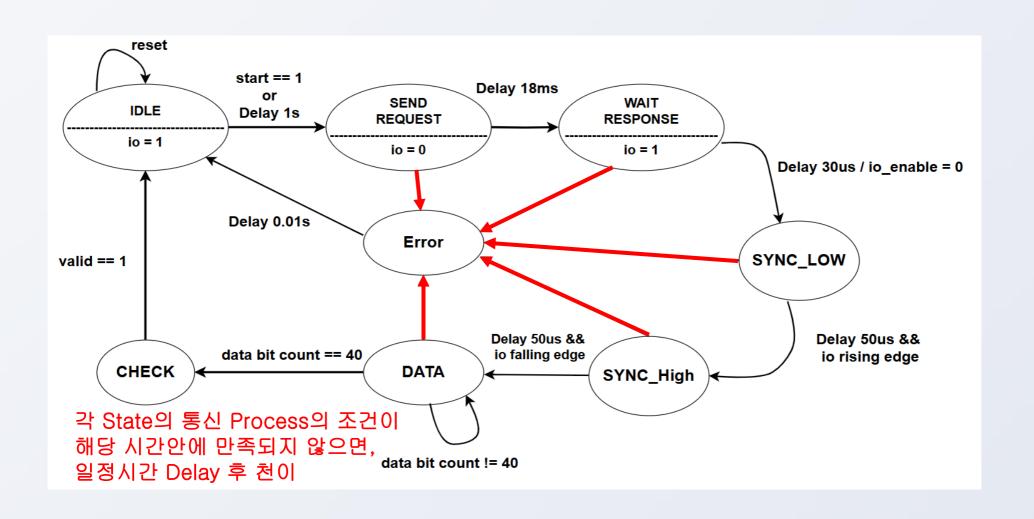
DHT11 Sensor

DHT11 Communication Process

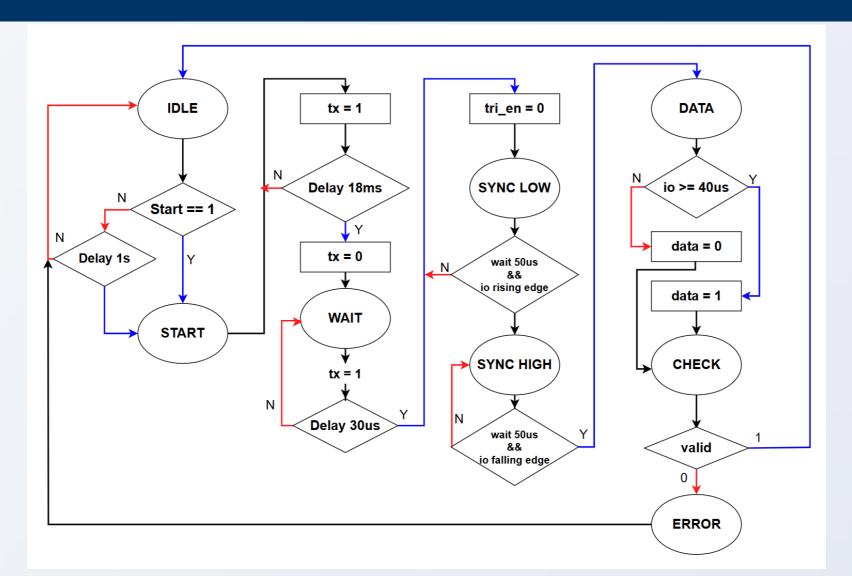


6. 수신한 40bit의 하위 8bit와 상위 32bit를 8bit씩 더한 값이 같다면 Valid 출력 1, 다르다면 0 출력

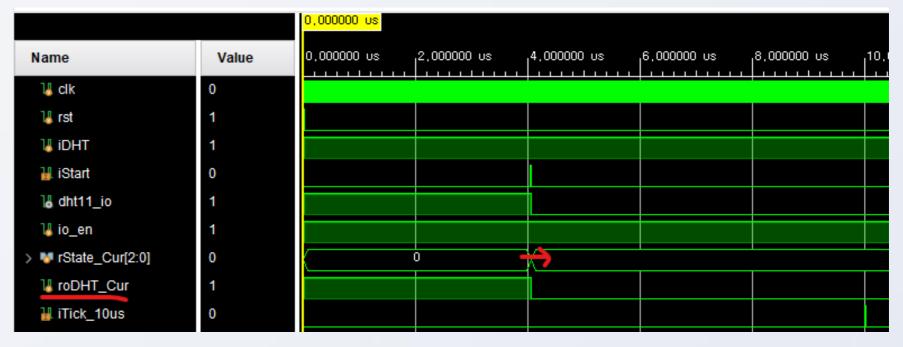
DHT11 Control Unit FSM



DHT11 Control Unit ASM



Simulation 검증 목표 : 수신한 40bit data에 따른 Valid 여부와 inout 동작 검증

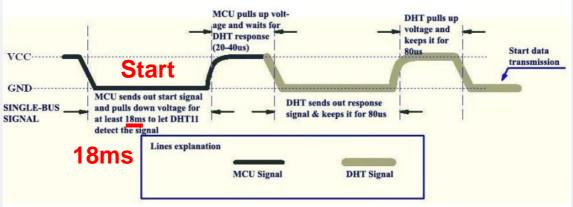


'ਊ' start 신호 감지 후 tristate buffer 출력으로 low 출력하면서 START 상태로 천이



Simulation 검증 목표 : 수신한 40bit data에 따른 Valid 여부와 inout 동작 검증

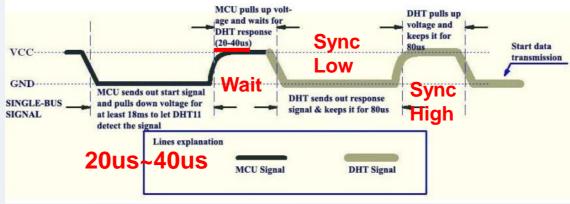






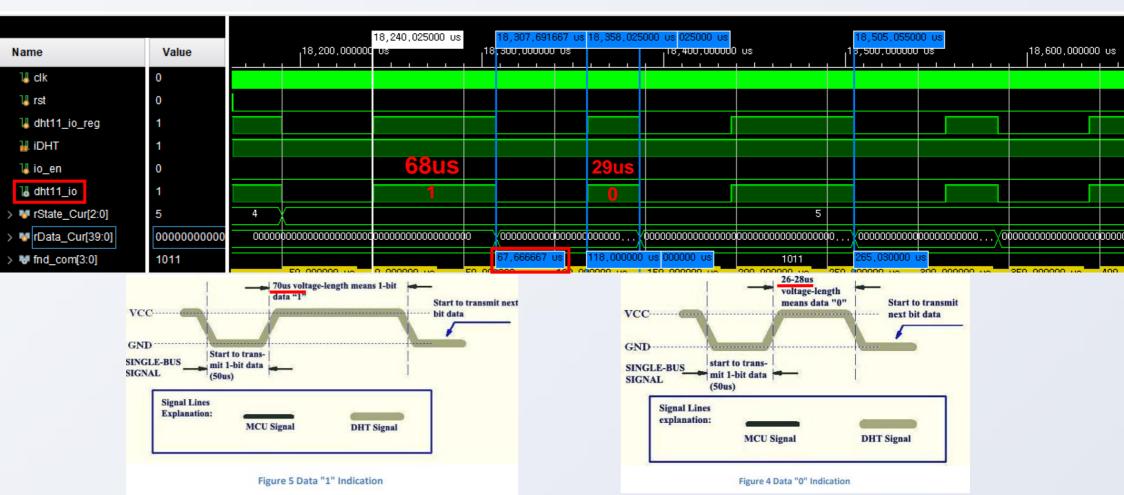
Simulation 검증 목표 : 수신한 40bit data에 따른 Valid 여부와 inout 동작 검증





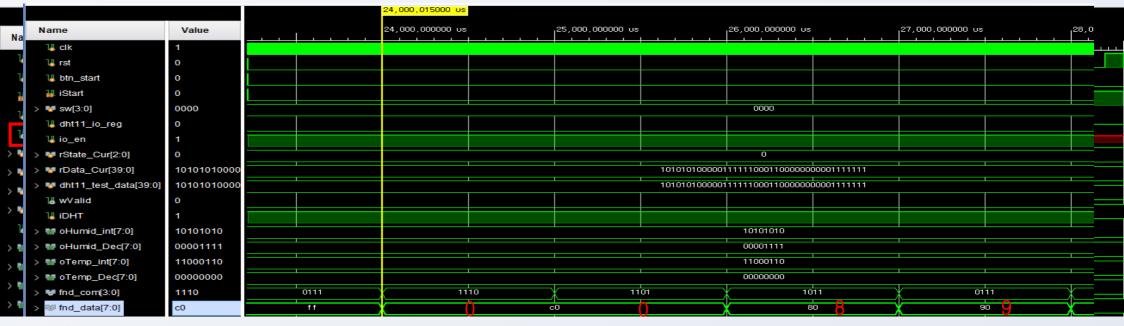


Simulation 검증 목표 : 수신한 40bit data에 따른 Valid 여부와 inout 동작 검증





Simulation 검증 목표 : 수신한 40bit data에 따른 Valid 여부와 inout 동작 검증





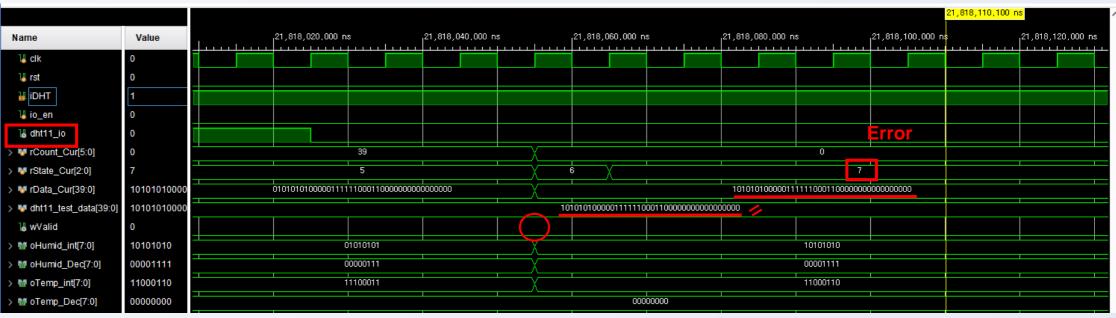
data register의 LSB에 DHT11 Sensor Data의 MSB부터 밀어 넣는다



Check State에서 수신된 valid data와 직접 상위 32bit를 더한 값을 비교하여 1 또는 0 출력



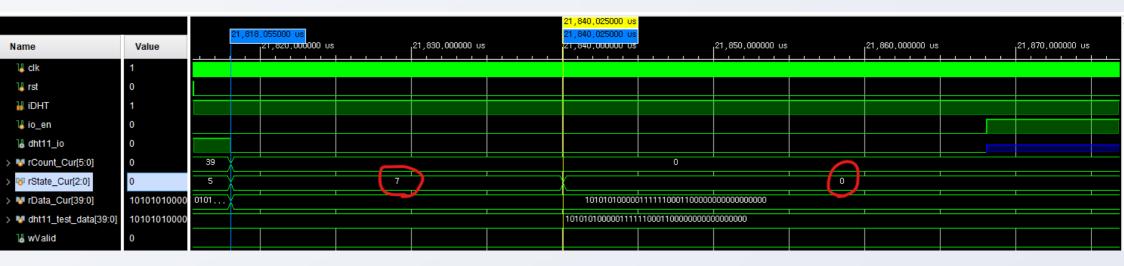
Simulation 검증 목표: 40bit Data는 잘 수신했는데 Valid Data가 틀린 경우





Check State에서 Valid가 0이고 Error State로 천이

Simulation 검증 목표 : 40bit Data는 잘 수신했는데 Valid Data가 틀린 경우

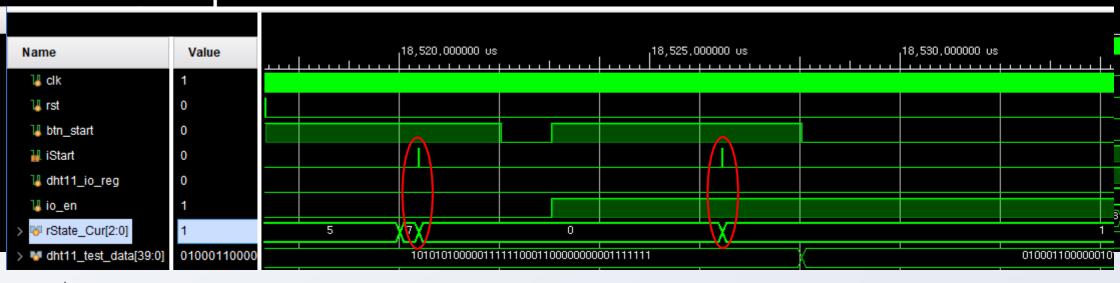




Error State에서 일정 시간이 지나면 IDLE로 천이하여 다음 측정 진행가능



Simulation 검증 목표 : Data bit가 중간에 끊길 경우 error 검출여부





DATA State에서 dht11 신호가 low로 130us가 지속되면 Error State로 천이

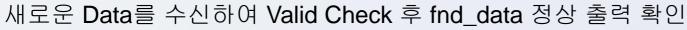


이후 Start 신호가 들어오면 계속해서 측정 가능



- 습도 70.05%, 온도 30.01도 data로 검증





DHT11 TroubleShooting



InOut Receive Data Noise 문제



1. Syncronizer로 해결

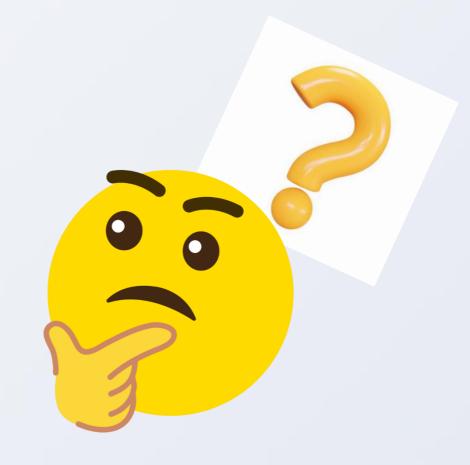


2. 각 State의 천이를 위한 시간 Count

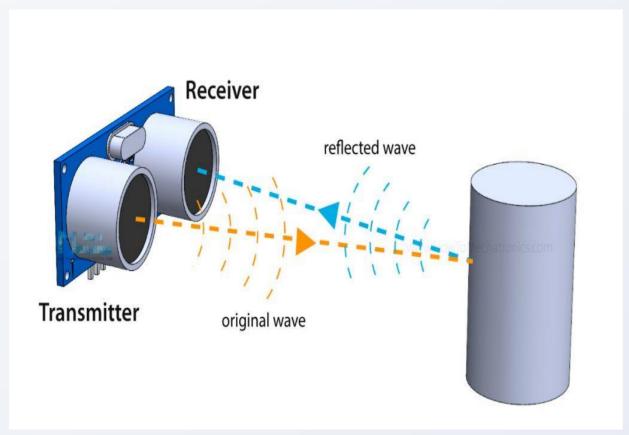
Ultra Sonic Sensor

Overview

How do you measure distance?



Overview





핵심 기능

- ❷ 초음파가 반사되는 시간으로 거리산출
- ☑ Trigger : 시작신호
- ☑ Echo: 초음파 반사시간 측정신호

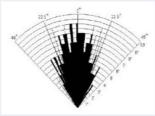
SPEC





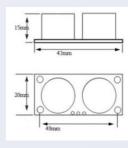
동작전원: +5 V DC

동작전류:<2mA



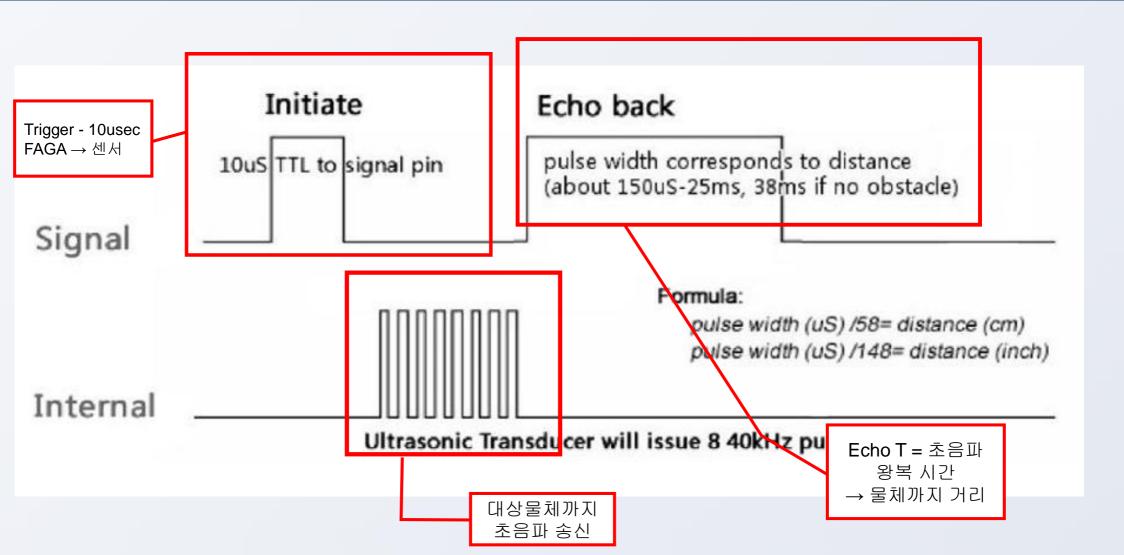
범위: 2cm ~ 400cm

각도:30°

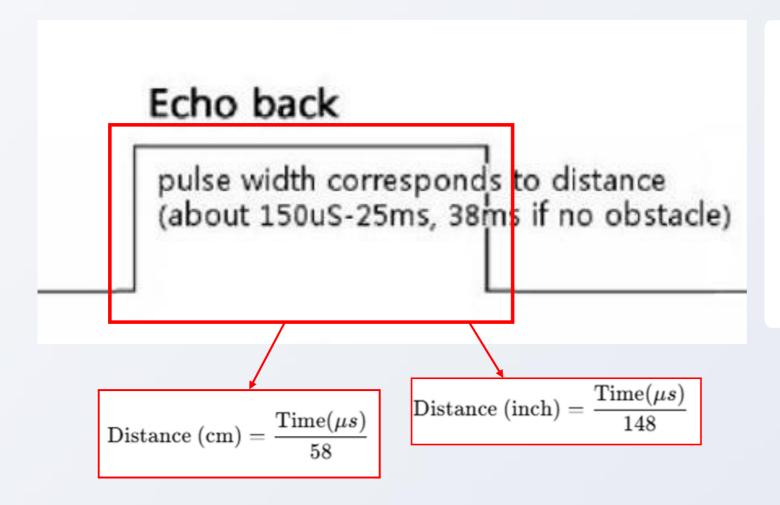


규격: 45mm x 20mm x 15mm

Basic Theory



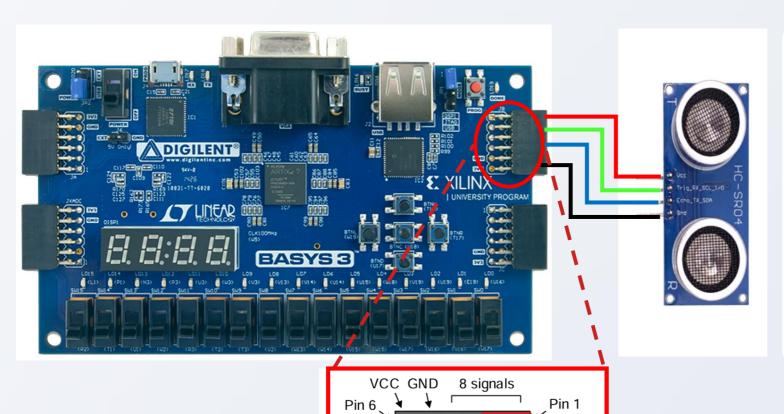
Basic Theory



Design Point

- 1. Tick = 1usec 설정
- 2. Only, Echo = high Tick Count!!

시스템 구성도

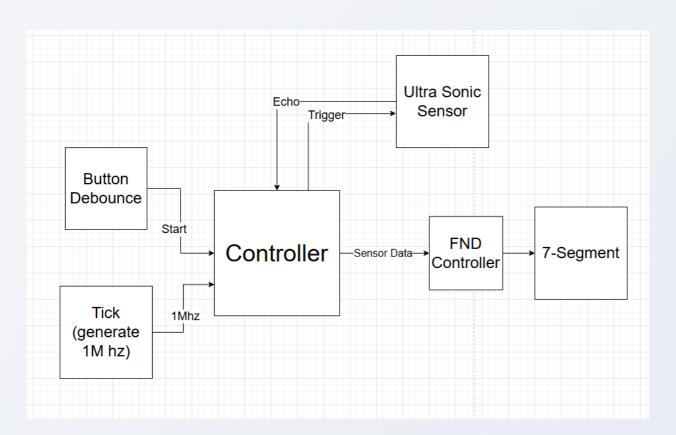


♥ VCC : 3.3V

⊘ Trig (Rx) : JB1

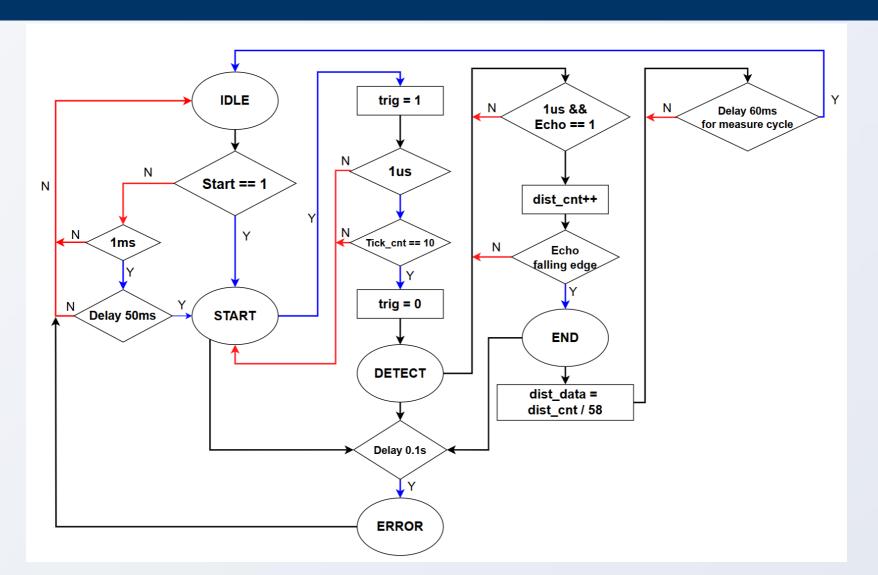
⊘ Echo (Tx): JB2

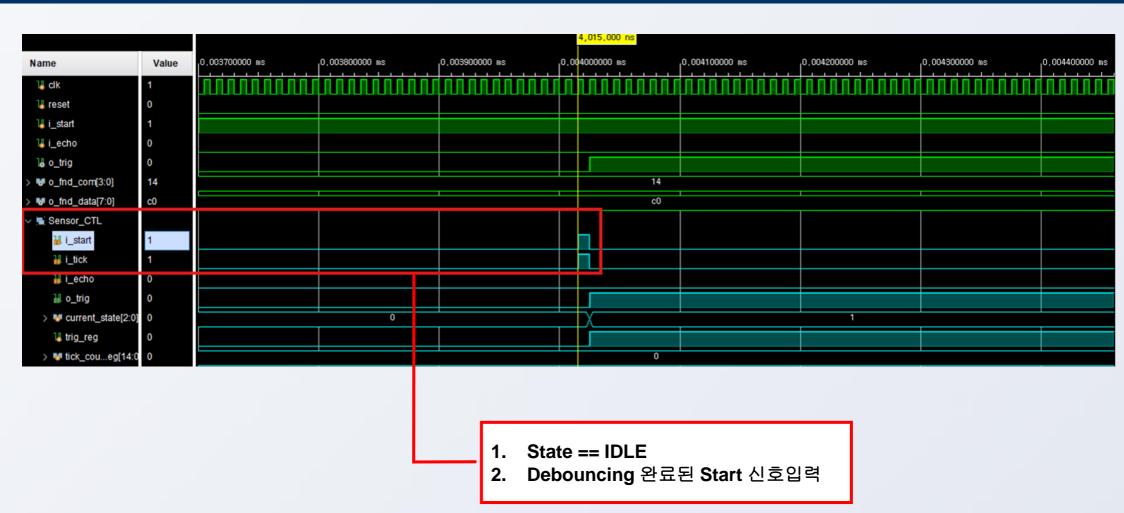
Block Diagram

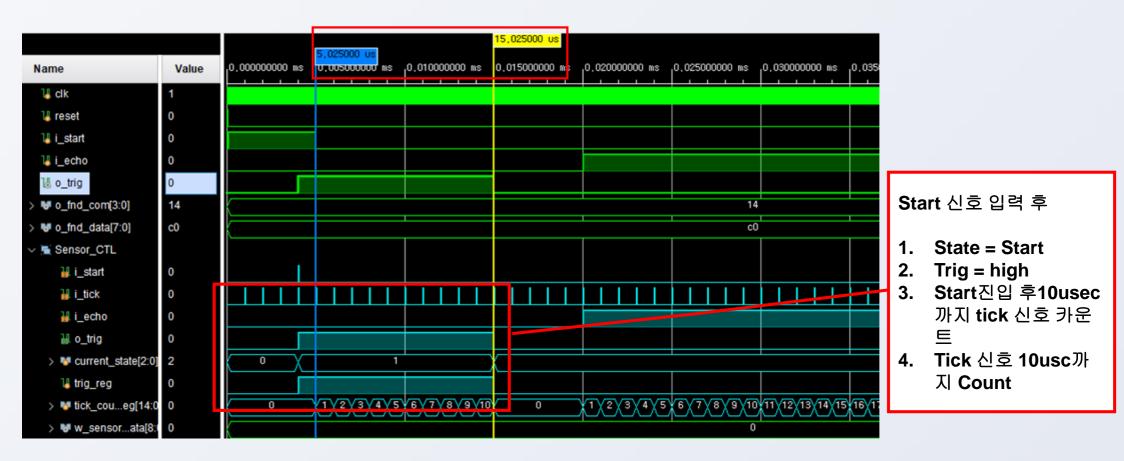


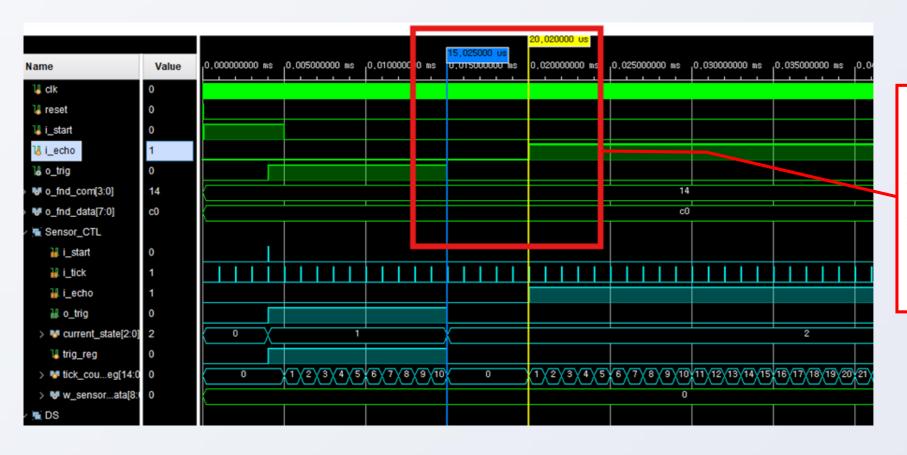
- Controller : 상태제어(FSM)
- Debouncing Prevent Chattering
- ▼ Trigger Tx Signal
- **♥** FND Controller
 - Digit Spliter
 - Mux

HC-SR04 Control Unit ASM







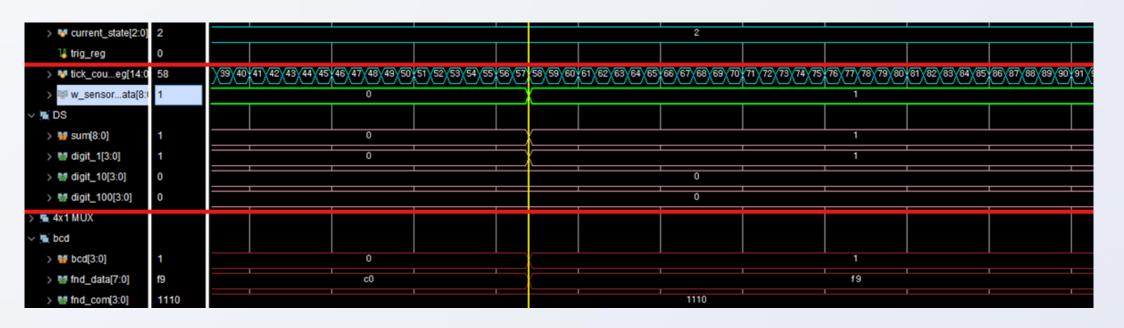


- 1. State = DETECT
- 2. 초음파 방출(200u sec)
- 3. Echo && tick Echo 신호 주기를 tick으로 카운트



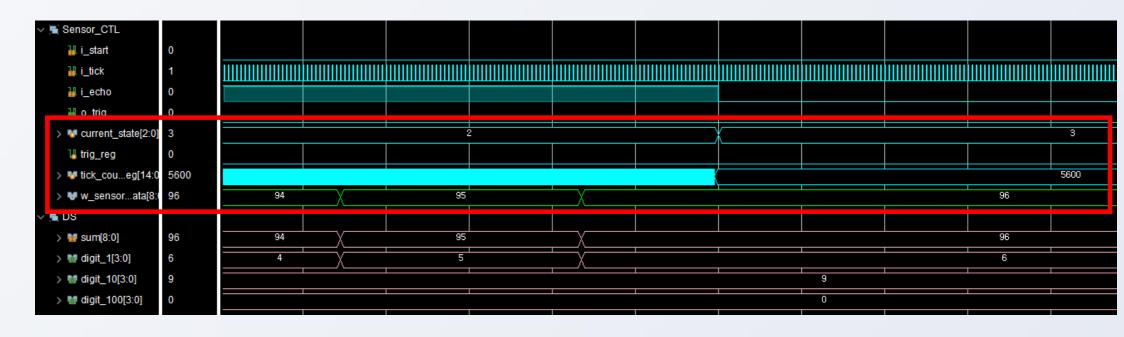
- 1. STATE == DETECT
- 2. Echo 신호 주기만큼 Tick 카운트
- 3. 초음파 신호의 왕복시간

$$\text{Distance (cm)} = \frac{\text{Time}(\mu s)}{58}$$



58까지 증가 후 거리출력 값이 1로 증가함.

$${\rm Distance}\;({\rm cm}) = \frac{{\rm Time}(\mu s)}{58}$$



Tick이 5600까지 카운트 되면 공식에 따라 거리가 96cm로 계산.

계산 후 State = End 천이

這是

개선 사항 및 보완점



개선 사항

- 초음파 센서를 이용한 추가적인 기능 추가
- ▼ 모든 동작을 PC 명령어를 통해 간결하게 수행

감사합니다.