Education

창원대학교
전기전자제어공학부 전자전공 졸업
2017. 03 ~ 2024. 02

센텀고등학교 졸업
2014. 03 ~ 2017. 02

Certification

컴퓨터활용능력 1급
2022. 12. 02

지게차운전기능사
2019. 04. 27

운전면허 1종 보통
2017. 07

Experience

우리웍스 현장실습
2021. 12 ~ 2022. 01

패스트 캠퍼스
반도체 설계 기본 Verilog & FPGA(31H)
2023. 12

[HARMAN] 세미콘 아카데미 반도체 설계
2023. 07 ~ 2024. 01

Tools & Skills

OrCAD Capture
-아날로그 회로 설계 및 Simulation

Allegro PCB Editor
-PCB 회로 설계 및 거버 파일 추출

Xillinx Vivado
-Verilog 사용으로 Peripheral 설계 및 Simulation을 통한 검증

Cadence Virtuoso
-Schematic, Layout 등 Circuit 제작

STM32
-C/C++언어 기반의 임베디드 시스템 제작

Projects

① Analog 초음파 거리 측정기
마날로그 발진기, 증폭기, 검출회로 등으로 구성된 초음파 거리 측정기

② 자율 이동체 개발 프로젝트
STM32 보드를 이용한 자율 이동체 구현

③ Digital Clock
Verilog HDL을 기반으로 Digital Clock 구현

④ W5500을 활용한 TCP IP 구현
Verilog HDL을 기반으로 W5500을 이용하여 IP 구현

Analog 초음파 거리 측정기

[기 관] : [Harman] 세미콘 아카데미 반도체 설계 과정
[기 간] : 2023. 09. 07 ~ 2023. 09. 15

[사용 기술]
1. NE555 Timer를 이용한 발진 회로 설계
2. Op-Amp 증폭기 설계
3. 다이오드를 이용한 평활 회로 설계

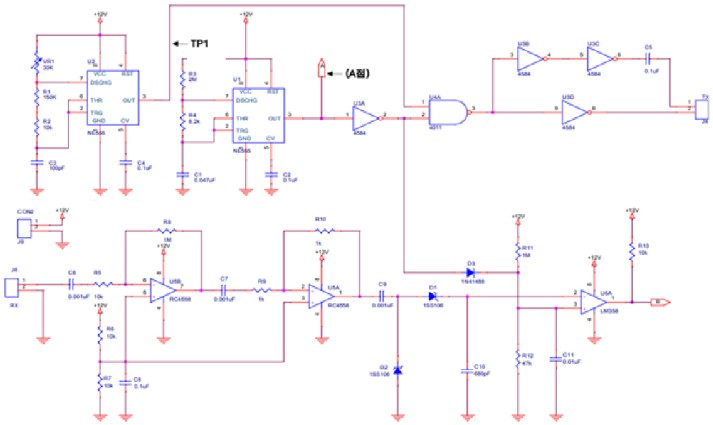
동작 영상

수행 목표

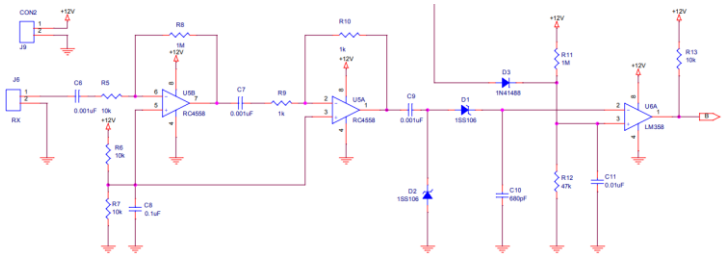
초음파 센서의 동작 주파수를 구현하여 실제 동작을 확인

회로 설계

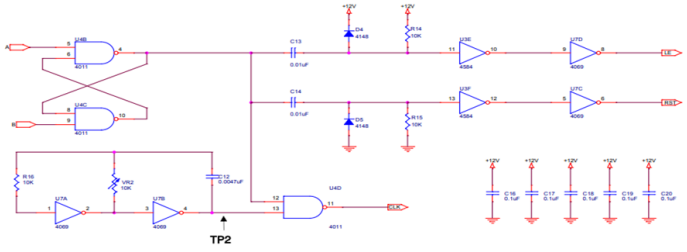
① 송신부



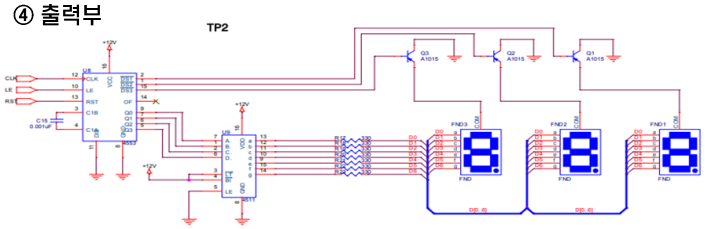
② 수신부



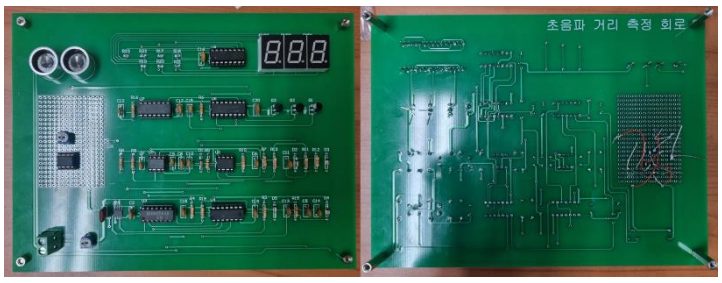
③ 시간 측정부



④ 출력부



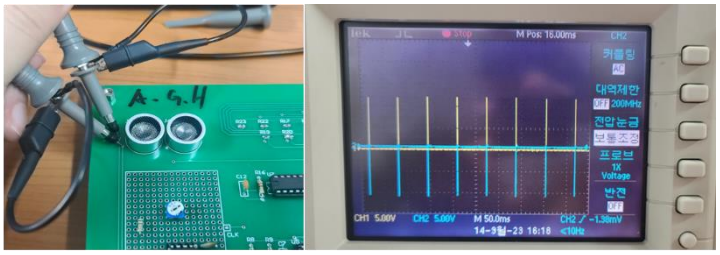
완성 사진



[전면부] [후면부]

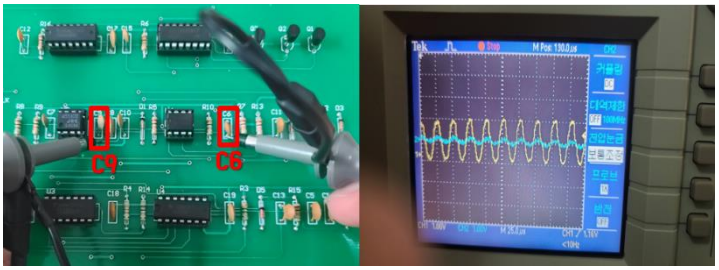
블록 다이어그램

동작 검증

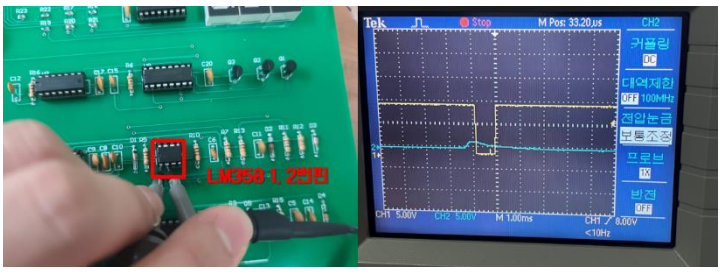


※ 초음파 센서 송신부분 측정

[초음파 센서 송신부]

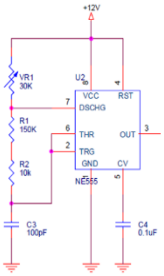


[초음파 센서 수신부]



[신호 검출 비교기]

디버깅 사례



$$\therefore f = \frac{1}{T} = \frac{1}{t_H + t_L} = \frac{1}{0.693\{(R_a + 2R_b)C\}} = \frac{1.44}{(R_a + 2R_b)C}$$

Ra = 가변 저항
Rb = R1 + R2

Problem

NE555 7번 단자 설계 오류로 인해 FND에 흘러가야하는 전류가 매우 작아 FND가 작동되지 않는 오류가 발생했음

Why?

Rb값이 매우 작아지고 또한, 전류가 FND로 가는 길에 SMD저항을 모두 거치게 되면서 전류값이 매우 낮아지게 됨

Result

위 식을 통해 가변저항의 값(약 17옴) 을 결정하여 수정하고 멀티미터를 통해 변경된 값을 확인하는 절차를 통해 해당 오류 수정

하만 커넥티드 자율 이동체 개발 프로젝트

[기 관] : [Harman] 세미콘 아카데미 반도체 설계 과정
[기 간] : 2023. 11. 13 ~ 2023. 12. 08
[역 할] : SW 설계 및 프로젝트 총괄

[사용 기술]
1. 운영체제 : Windows 11, Android
2. 개발도구 : STM32 Cube Ide
3. 개발언어 : C/C++

주행 영상

수행 목표

초음파 센서를 활용하여 장애물을 식별하고,
자율 주행 및 어플을 통한 수동 조작이 가능한 이동체 제작

부품 목록(BOM)

구성품	개수(EA)	설명	비고
STM32 보드	1	자율 이동체의 CPU 역할	
HCSR-04	1	장애물을 식별하기 위한 초음파 센서	
TT모터	4	자율 이동체 구동을 위한 모터	
L298N	1	모터를 제어하기 위한 모터 드라이버	
HC-06	1	블루투스 통신을 위한 블루투스 직렬포트 모듈	
9V 배터리	1	전원을 제공하는데 사용되는 배터리	
피에조 부저	1	피에조 방식을 이용하여 소리를 내는 모듈	
스위치	1	전원을 ON/OFF하기 위한 스위치	


[STM32 보드]


[HCSR-04]


[TT모터]


[L298N]

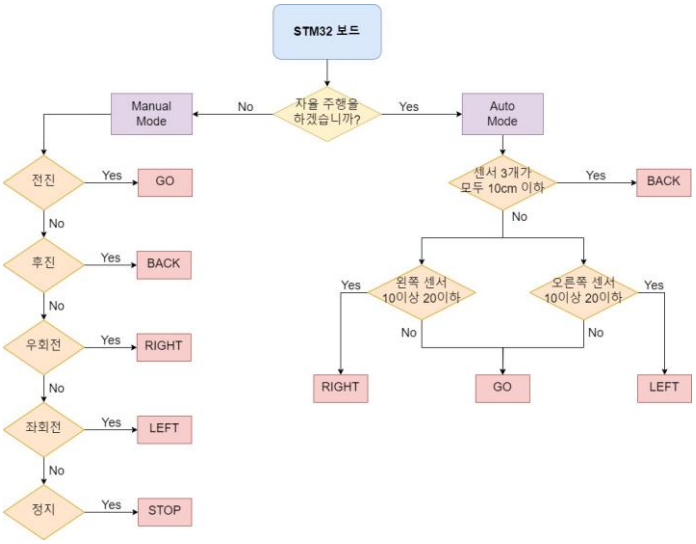

[HC-06]


[9V 배터리]


[피에조 부저]


[스위치]

동작 알고리즘



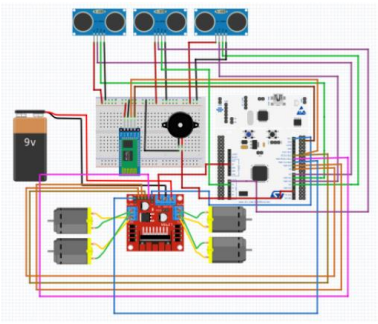
프로젝트 진행 시 유의 사항

① 다수의 테스트를 통해 이동체가 부딪히지 않는 일정 거리를 설정하여 알고리즘 구성 필요

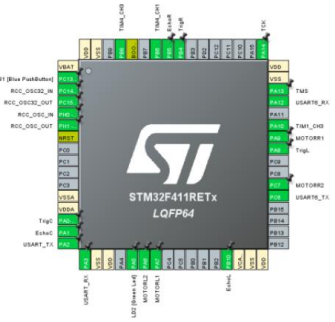
② 올바른 직진을 위해 양쪽 모터의 알맞은 pwm 값 설정

③ 원활한 블루투스 통신을 위해 어플을 제작하여 송·수신 데이터 확인 필요

Schematic & Pin Mapping



[Schematic]



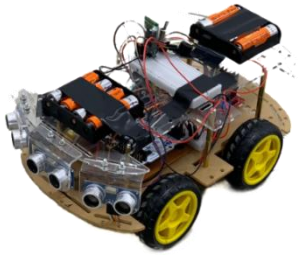
[Pin Mapping]

Main 동작 코드

```
/****************************** Auto Mode ******************************/
if(flag == 1)
{
    Sonic();    // 3개의 초음파 센서 거리 측정
    Forward();  // 일단 전진
    /******************** center 값에 따른 구동 조건문 ********************/
    if(cm_center < DIS_MIN || cm_right < DIS_MIN || cm_left < DIS_MIN)
    {
        Backward();
        HAL_Delay(DELAY_TIME);
    } // end of if Backward
    else if(cm_right < DIS_MID && cm_right >= DIS_MIN)
    {
        Rotate_L();
        HAL_Delay(DELAY_TIME);
    } // end of if Rotate_L
    else if(cm_left < DIS_MID && cm_left >= DIS_MIN)
    {
        Rotate_R();
        HAL_Delay(DELAY_TIME);
    } // end of if Rotate_R

/****************************** Manual Mode ******************************/
    else    // Manual Mode
    {
        switch(dum6) {
            case 'f' : Forward();          break;  // button click for Forward
            case 'b' : Backward();         break;  // button click for Backward
            case 'r' : Rotate_R();         break;  // button click for Rotate_R
            case 'l' : Rotate_L();         break;  // button click for Rotate_L
            case 's' : Stop();             break;  // button click for Stop
            case 'x' : flag = 1;           break;  // Auto mode On
            case 'a' : TIM1->ARR = 478;            // Sound ON -> Volume Up
                       TIM1->CCR3 = 478 / 2;
                       break;
            case 'c' : TIM1->CCR3 = 638;   break;  // Sound OFF
            case 'v' : PWM = 400;          break;  // button click for PWM 400
            case 'w' : PWM = 600;          break;  // button click for PWM 500
            case 'y' : PWM = 800;          break;  // button click for PWM 600
            case 'z' : PWM = 999;          break;  // button click for PWM 700
            default  : break;
        } // end of switch
    } // end of manual mode
```

결과 및 고찰


[결과물 사진]

[User Interface]

① 초음파 센서 함수 3개 사용 시 putty에 거리 값이 나오지 않음
-> 각 센서 함수 사이에 delay를 주어 동작을 수행하도록 해결

② 블루투스 통신 시 제대로 된 값을 받아오지 못함
-> 문자열이 아닌 문자 단위의 데이터를 전송하게 하여 간단하게 통신하여 해결

③ 초음파 센서 함수 구현 중 거리 값이 튀는 현상 발견
-> 값 30개 정도를 샘플링 하여 최빈값을 구해 제일 많이 측정된 값을 사용하여 이동체 구현

Digital Clock

[기 관] : [Harman] 세미콘 아카데미 반도체 설계 과정
[기 간] : 2023. 09. 25 ~ 2023. 10. 20

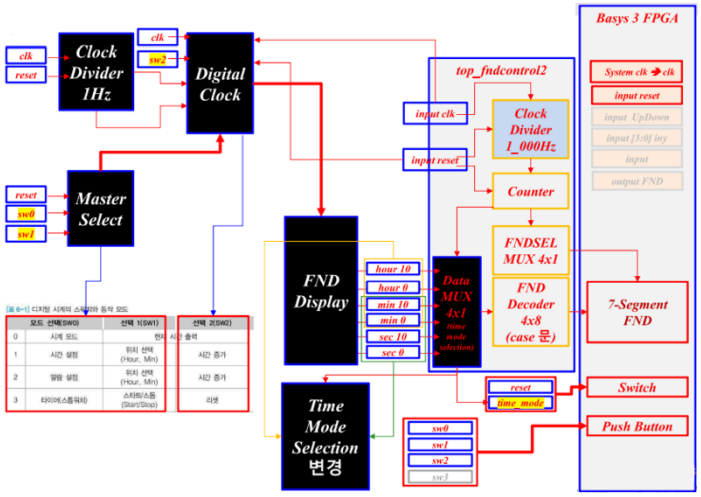
[사용 기술]
1. 개발 툴 및 언어 : Vivado, Verilog
2. 개발 보드 : Basys3
3. 사용 기술 : FSM (Finite State Machine)

동작 영상

수행 목표

Verilog 언어를 이용하여 각종 기능이 있는 디지털 시계를 설계한 후 제대로 동작하는지 Basys3 보드를 통해 검증

Block Diagram



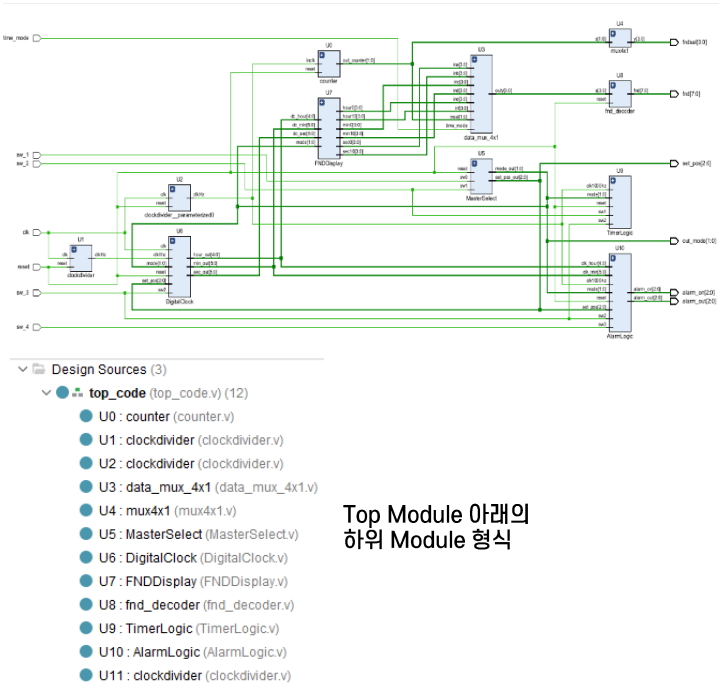
동작 모드

	모드 선택(SW0)	선택 1(SW1)	선택 2(SW2)	선택 3(SW3)
0	시계 모드	현재 시간 출력		알람 ON/OFF
1	시간 설정	위치 선택 (Hour, Min)	시간 증가	
2	알람 설정	위치 선택 (Hour, Min)	시간 증가	
3	타이머(스톱워치)	스타트/스톱 (Start/Stop)	리셋	

입출력 장치

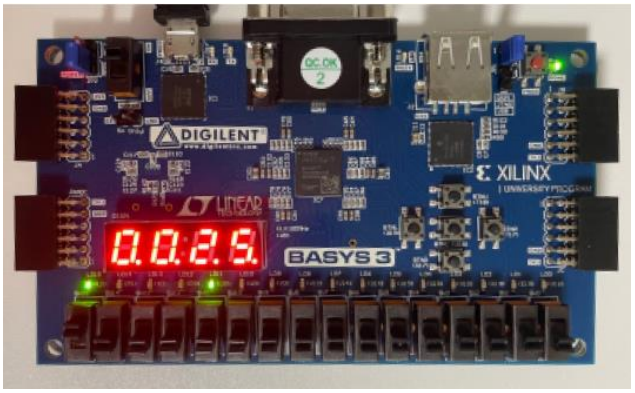
입출력	입출력 장치
clk	On board 1MHz OSC
reset	리셋 스위치
SW0~SW3	입력 키 스위치 Key0~Key3
시, 분, 초	FND3~FND1
알람 ON/OFF 상태	LED D1~D3
알람 신호	LED D5~D7
모드	LED D8~D9
시, 분, 초 설정 위치 표시	LED D13~D15

RTL ANALYSIS Schematic



Top Module 아래의
하위 Module 형식

결과 및 고찰



① 하나의 module 안에 기능을 모두 넣는 것보다 기능마다 module을 따로 만들어서 코드를 구현하면 보다 더 편리하다는 것을 알게 됨

② Verilog로 설계 시 FSM을 이용하여 코드를 구성하면 원하는 기능을 구현하는데에 있어 보다 더 쉽다는 것을 알게 됨

W5500을 활용한 TCP IP 구현

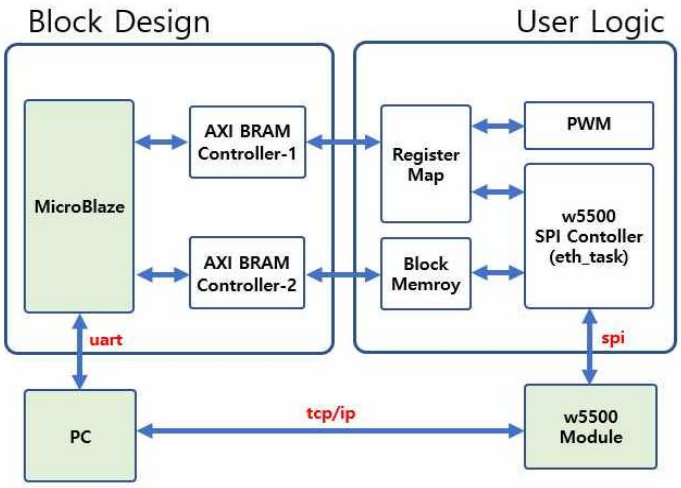
[기 관] : [Harman] 세미콘 아카데미 반도체 설계 과정
[기 간] : 2023. 11. 02 ~ 2023. 12. 01

[사용 기술]
1. 개발 툴 및 언어 : Vivado, Verilog
2. 개발 보드 : Basys3
3. 사용 부품 : W5500

수행 목표

Verilog 언어를 이용하여 SPI Controller를 통한 W5500 인터페이스 구현해
Basys3 보드를 통해 검증

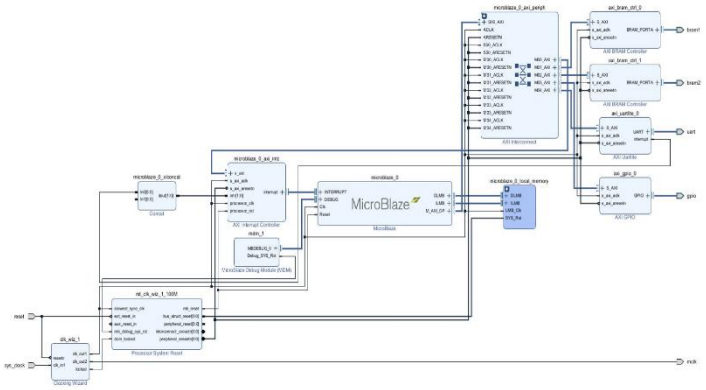
System Block



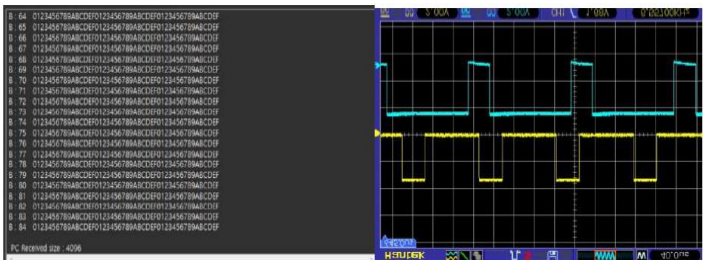
① 2개의 AXI BRAM Controller를 사용함

② 하나는 사용자 레지스터 맵을 구현하는 데 사용되며,
다른 하나는 W5500 인터페이스용 SPI 컨트롤러의 데이터 버퍼로 사용됨

Block Diagram



결과 및 고찰

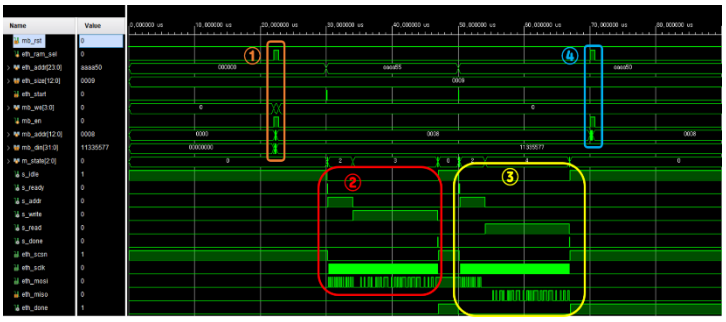


① TestBench에서의 작동 뿐만 아니라 실제 구현을 위해 Data Buffer를 사용함

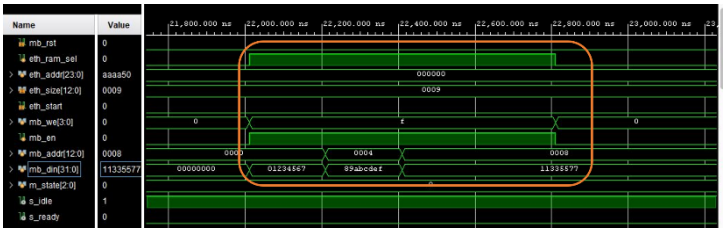
② 점퍼 케이블 사용으로 인해 SPI 통신 속도에 제한이 있음

③ FPGA에 BRAM을 올릴 때, Single Port x2 대신, Dual Port 구성으로 하면
더 많은 공간을 절약할 수 있을 것 같음

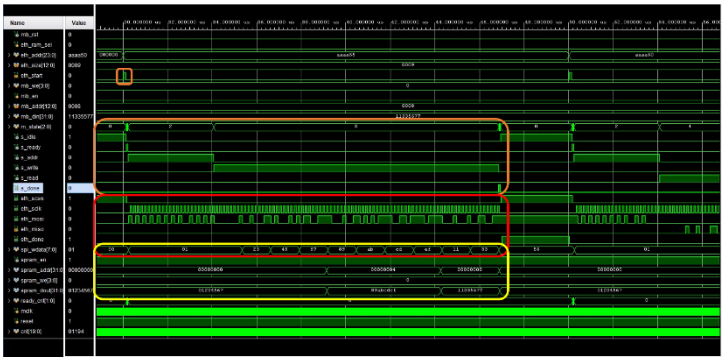
Simulation Analysis



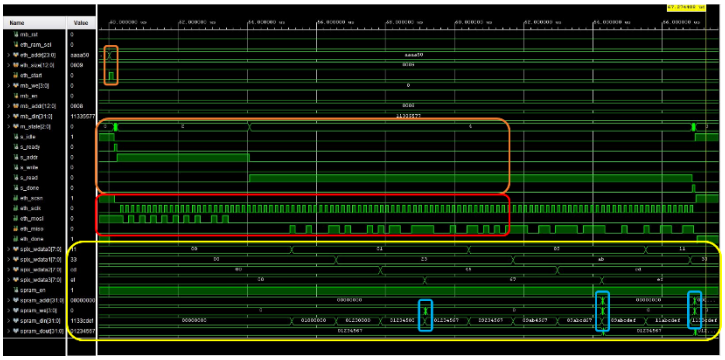
① Data Buffer에 0x01234567, 0x89abcdef, 0x11335577을 write



② SPI Write 구간



③ SPI Read 구간



④ Data Buffer에 저장된 데이터를 read 함