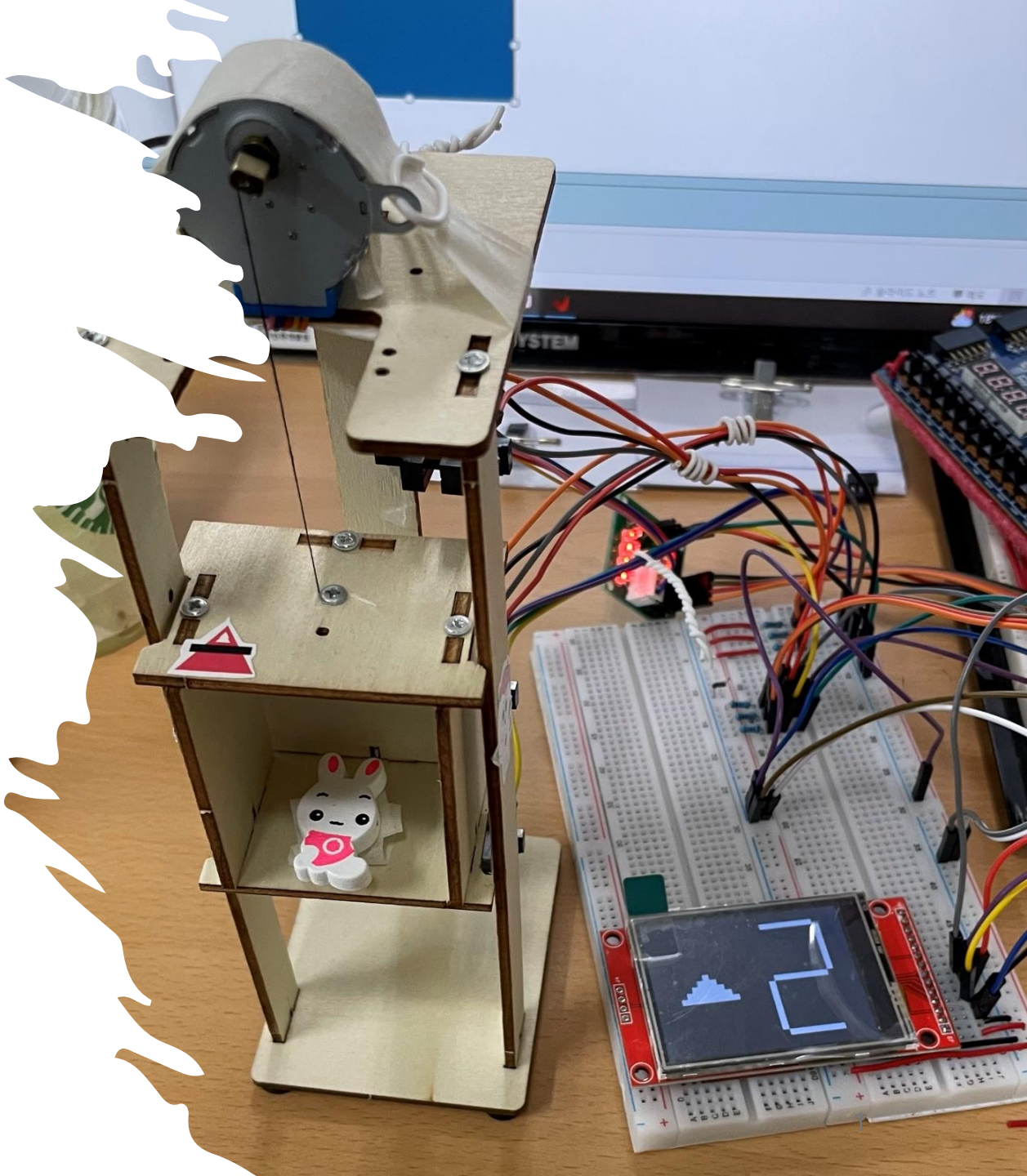
SoC Project

Elevator

Team Qualitas – 김현진, 이윤규 서정훈



2024-10-08

0. Table of Contents

1. Introduction

Objective

주 목표

- Soc 설계를 통한 엘리베이터 구현

부 목표

1. SPI 통신을 이용하여 Graphic LCD 활용
2. Timer/Counter로 아래의 기능 구현
    a. Interrupt로 소프트웨어 동작 최적화
    b. PWM 생성하여 LCD 밝기 제어
3. 블록 디자인
4. Embedded Software coding

1. Introduction

Member

팀장 김현진

- **코드 통합**
- **RTL 설계**
  - Timer/Counter
  - SPI 통신
- **IP Integration**
- **Embedded SW 설계 (총괄)**

이윤규

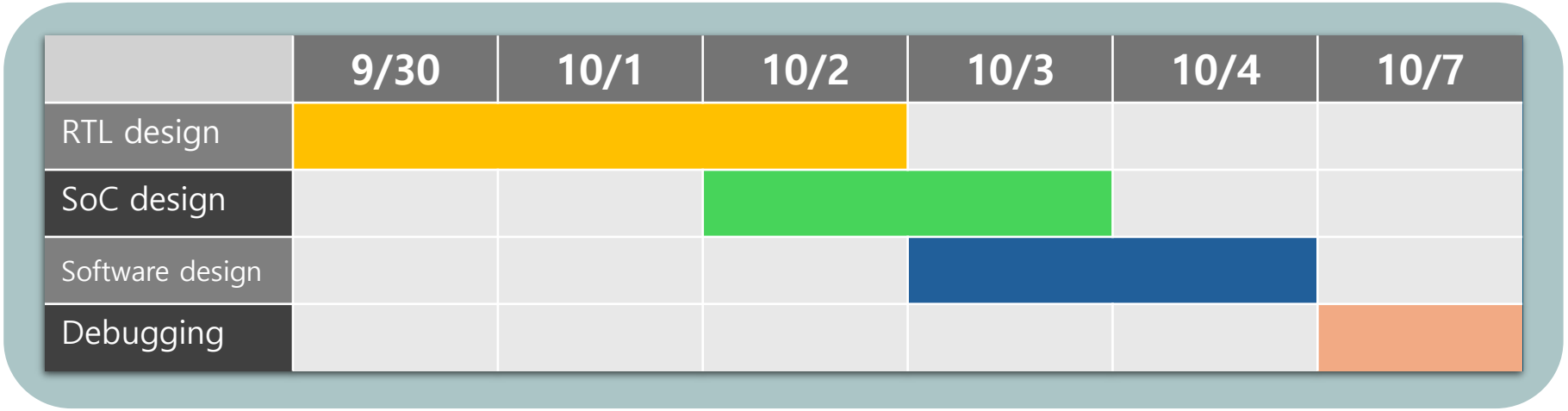
- **Embeded SW 설계 (Support)**
  - GPIO
  - External Interrupt

서정훈

- **RTL 설계**
  - Timer/Counter
- **IP Integration**
- **Embedded SW 설계**
  - Custom IP

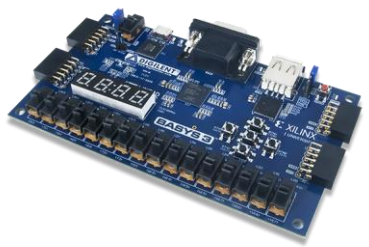
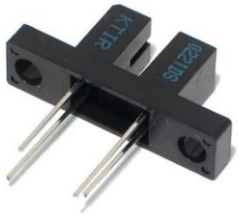
1. Introduction

Project Schedule

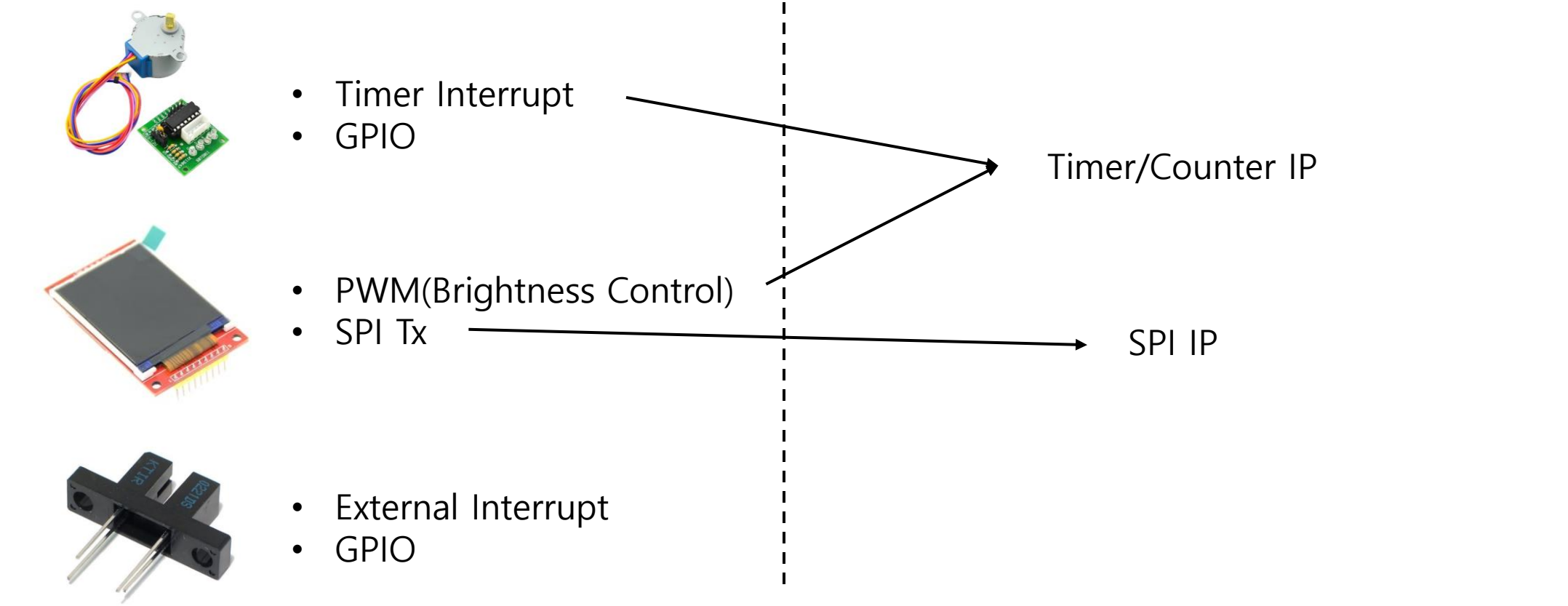


2. Overview

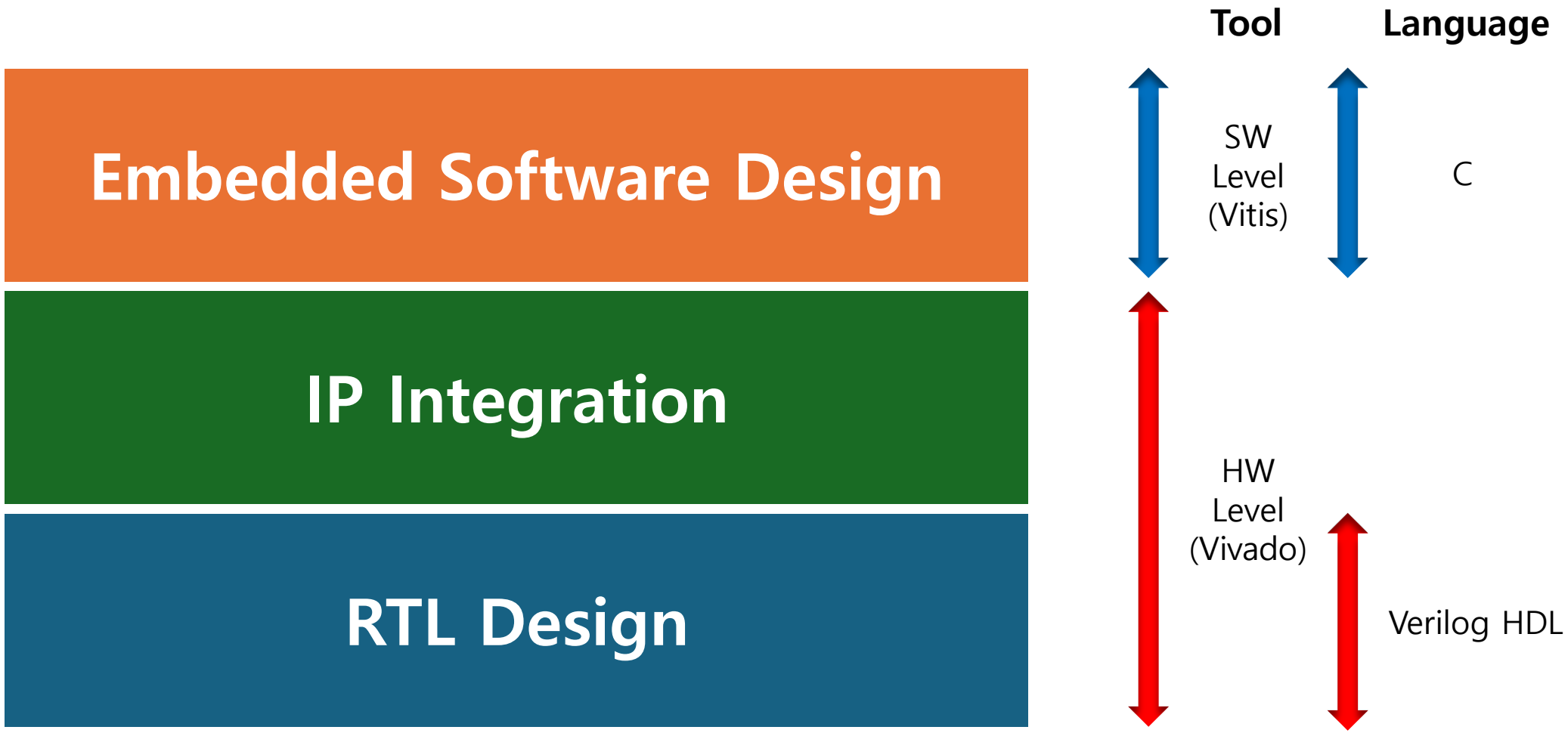
사용 기기들

FPGA Board	Stepper Motor	Photo Interrupter	Graphic LCD (with SPI)
			
Basys3 • 프로세싱 • 버튼 입력	**28BYJ-48(motor)** **ULN2003(driver)** • 엘리베이터 구동	**SG-23FF** • 층 이동 감지	**MSP2402** • 현재 층수 표시 • 이동 방향 표시

2. Overview


Software
Hardware
• Timer Interrupt
• GPIO
Timer/Counter IP
• PWM(Brightness Control)
• SPI Tx
SPI IP
• External Interrupt
• GPIO

3. Principles



RTL Design

Overall Flow

1. FPGA용 모듈 설계
2. 테스트
   - Testbench
   - Top Module Programming
3. FPGA 모듈 → IP용 모듈로 편집
4. 테스트
   - Testbench

❖ 설계를 2번 하는 이유?

처음부터 IP용 모듈을 만드니
디바이스에 프로그래밍해서 테스트하기 어려움

<Example>

```
module spi_tx
parameter SCL_FREQ = 15_000_000;
parameter integer SCL_PSC = `CLK_FREQ / SCL_FREQ;
```

```
module spi_tx_ip
input [7:0] prescaler,
```

parameter를
레지스터 입력으로 변경한 예시

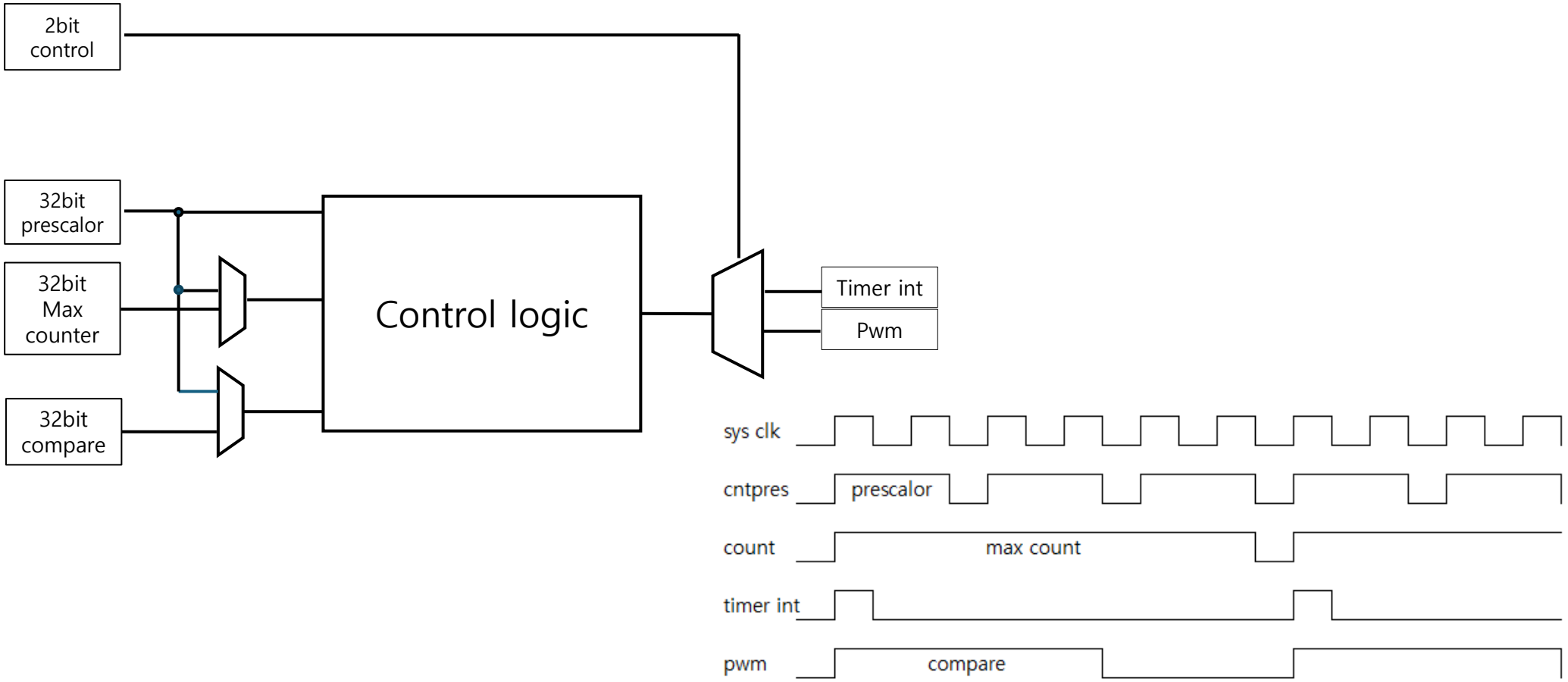
Timer/Counter

Functions

<table>
<tr><td colspan="2">Common</td></tr>
<tr><td colspan="2">1. Prescaler 설정
2. Count Period 설정 → 출력신호 주기 설정
3. 모드 선택</td></tr>
<tr><td>Overflow Interrupt mode</td><td>PWM mode</td></tr>
<tr><td>분주된 클럭이 max count까지 카운팅되면 인터럽트 신호 발생</td><td>분주된 클럭이 비교일치값보다
• 낮으면 high
• 높으면 low
출력</td></tr>
</table>

RTL Design

Timer/Counter



SPI - Tx

통신 규약(Master → Slave)

- **SCL, SDA**

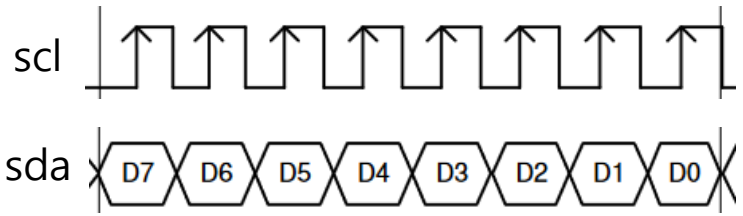
  통신 동기화 클럭 / 전송할 데이터(8-bit)

  sda로 양방향 통신

  본래 **CPOL**(polarity), **CPHA**(phase) 선택 가능

  여기서는 display driver에 맞게 0,0으로 고정

  → 유휴 상태에서 클럭 low 및, rising edge에서 데이터 샘플링



cpol = 0, cpha = 0

- **CS (Chip select, or Slave select)**

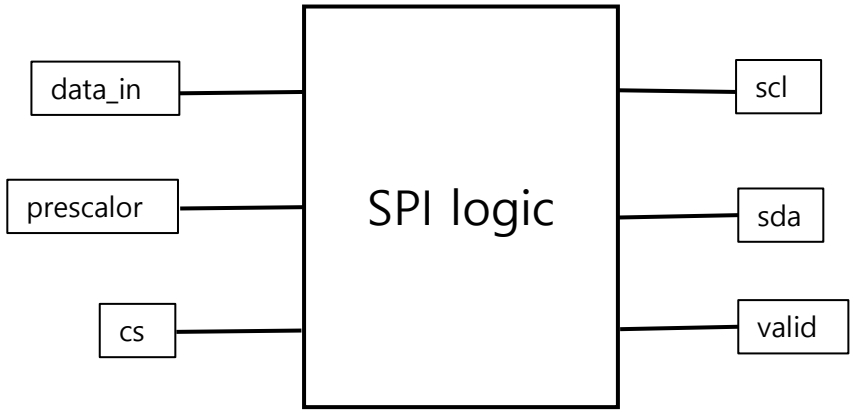
  Slave on/off 신호(active low)

  cs = 0이면 scl 생성

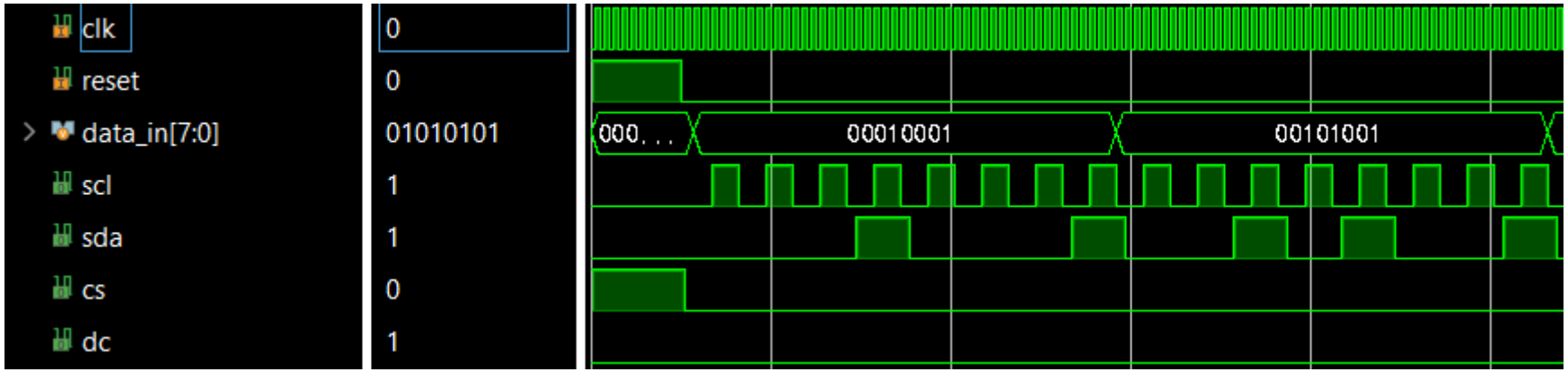
- **DC**

  커맨드/데이터 선택 신호

SPI - Tx

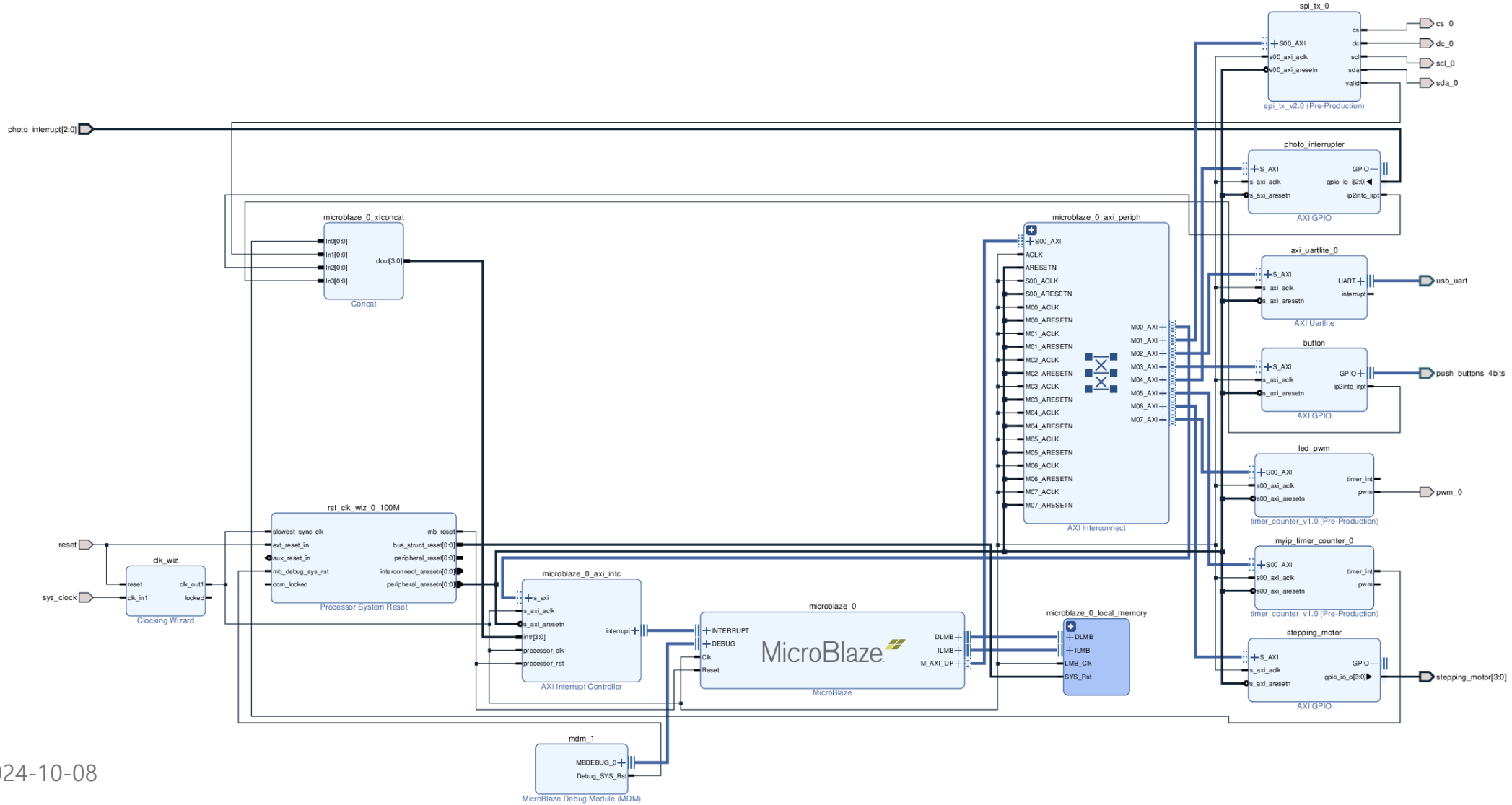


Logic

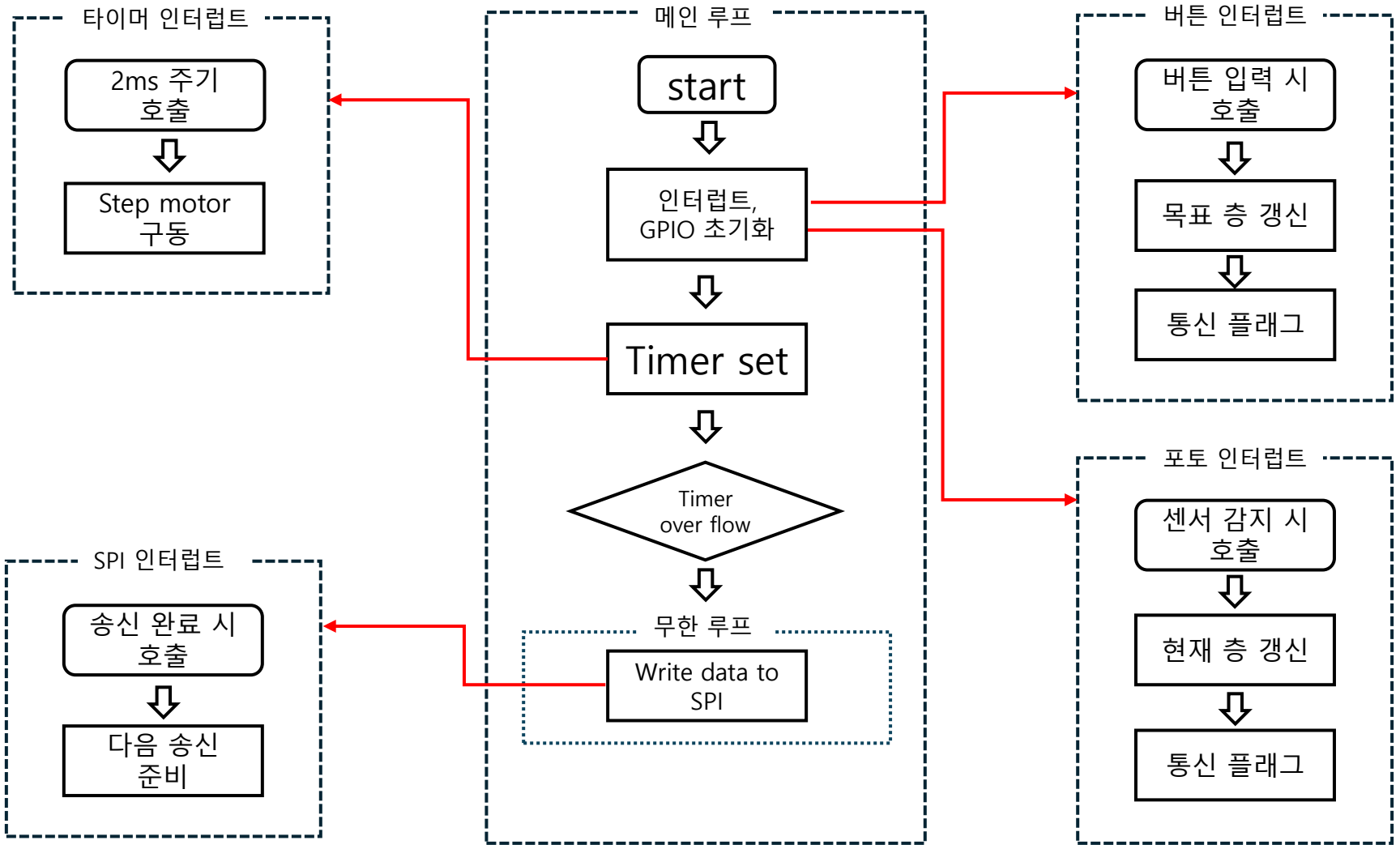


Timing Diagram

IP Integration



Embedded Software Design



SPI-LCD flow

SPI Init

SCL 주파수 설정

LCD Init

SendCommand(0x01) soft reset
전원 공급 및 리셋 후 자동으로 sleep mode 진입

SendCommand(0x11) sleep out

SendCommand(0x29) display on

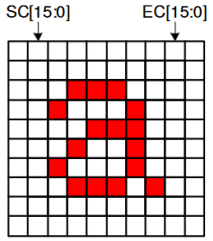
SendCommand(0x3a) RGB format setting 명령
SendData (0x55) 16bit(RGB:5/6/5)로 설정

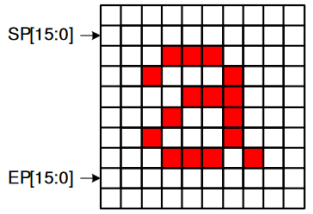
Send Data (RGB value)

SendCommand(0x2a) column addr. set
SendData x2 시작 주소 16-bit
SendData x2 끝 주소 16-bit

SendCommand(0x2b) page(row) addr. set
SendData x2 시작 주소 16-bit
SendData x2 끝 주소 16-bit

SendCommand(0x2c) memory(gram) write
SendData x2 데이터 전송(16-bit)
반복





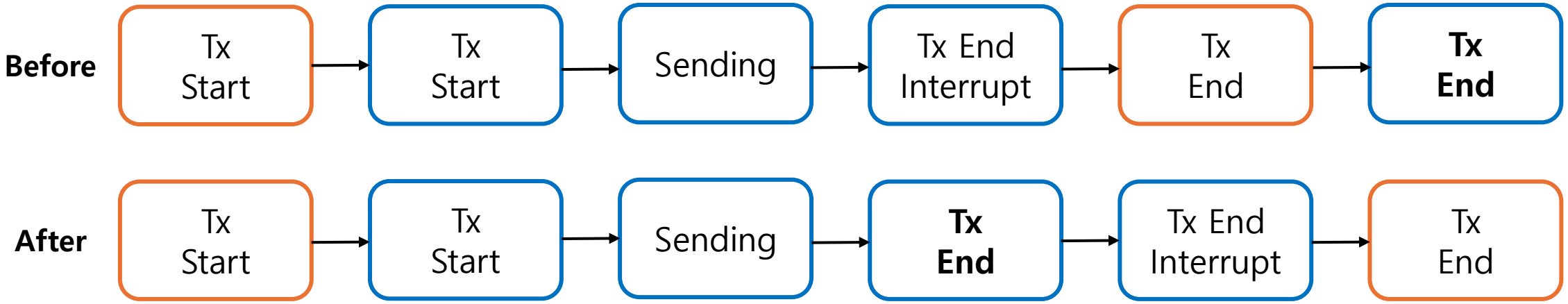
4. Demo Video

https://youtu.be/JZBbCFE_m-k

5. Conclusion

We have gained a better understanding of the operational differences between the FPGA and MicroBlaze.

When we converted the FPGA code directly into an IP, it did not function as expected.
This was because MicroBlaze takes time to execute interrupts, causing a delay in the completion of communication. We made the following modifications to address this issue.



However, due to these modifications, continuous data transmission became impossible, resulting in a slowdown in communication speed.

Fin.

Thank you for your time

GitHub Link
https://github.com/stardusk2nd/Qualitas

QR code