

# *SoC Semiconductor Peripheral RTC Design Project*

---

*Reference :*

2024-06-17

# *SoC Semiconductor Peripheral RTC Design Project*

# *SoC Semiconductor Peripheral RTC Design Project*

- 프로젝트 배경
- 프로젝트 주제 및 내용
- 프로젝트 세부 내용
- 프로젝트 수행 결과물(예시)

# *SoC Semiconductor Peripheral RTC Design Project*

- 프로젝트 배경
- 프로젝트 주제 및 내용
- 프로젝트 세부 내용
- 프로젝트 수행 결과물(예시)

# SoC Semiconductor Peripheral RTC Design Project

.xpr

## 프로젝트 수행 배경

프로젝트 계획				
프로젝트명	<ul style="list-style-type: none"> <li>SoC Semiconductor Peripheral RTC Design Project</li> </ul>			
프로젝트 추진배경	<ul style="list-style-type: none"> <li>시스템 반도체 설계는 HDL 언어를 이용해 이루어짐 → 우리 나라 업체에서는 대부분 Verilog HDL 언어를 사용</li> <li>Verilog HDL 언어를 이용하여 반도체 설계 프로젝트를 수행함</li> <li>조합 논리 회로 순차 논리 회로를 이용하여 반도체 설계를 경험</li> <li>하드웨어의 소형화 저전력화가 이루어짐으로 SoC(System On Chip) 형태의 반도체가 증가하고 있음.</li> <li>ARM Core 기반의 특정 Peripheral IP 를 설계하여 SoC 설계를 진행</li> <li>ARM Core 기반의 SoC를 설계하기 위해서 AMBA BUS를 이해해야 함</li> <li>ARM Core 기반의 Peripheral IP를 설계함으로써 SoC 설계를 이해하고 시스템 반도체 설계 능력을 갖추.</li> </ul>			
프로젝트 세부사항	기준시간	70h	기준인원	4인
	활용툴	<ul style="list-style-type: none"> <li>Xilinx FPGA Board, VIVADO, Vitis , 오실로스코프 디지털멀티미터 전원공급기 함수발생기</li> </ul>		
	과제 결과물	<ul style="list-style-type: none"> <li>시스템 반도체 구조 설계서</li> <li>소프트웨어 구조 설계서</li> <li>시스템 반도체 설계 결과 파일</li> </ul>	<ul style="list-style-type: none"> <li>소프트웨어 실행 결과 파일</li> <li>최종 데모용 Board</li> <li>동작 영상 자료</li> <li>최종 보고서</li> </ul>	
프로젝트 직무역량	세부 직무역량			통합 직무역량
	<ul style="list-style-type: none"> <li>시스템 반도체 설계 능력</li> <li>ARM Core 활용 능력</li> <li>반도체 검증 능력</li> <li>C언어 프로그래밍</li> <li>DataSheet 작성 능력</li> <li>Timing, Waveform 독해 능력</li> <li>계측기 활용 능력</li> </ul>			<ul style="list-style-type: none"> <li>시스템 반도체 설계</li> <li>디지털 하드웨어 설계</li> <li>펌웨어 설계</li> <li>팀워크 수행 역량</li> </ul>
프로젝트 운영전략	<ul style="list-style-type: none"> <li>프로젝트 목표 및 일정, 조 편성, 주제선성, 산출물, 평가 등 사전 OT 진행</li> <li>학습자 역량, 성향 및 수준을 고려한 조 편성</li> <li>평가항목 : 아이디어, 기술구현, 프로젝트 관리, 프리젠테이션, 팀 역할분담</li> </ul>			

# *SoC Semiconductor Peripheral RTC Design Project*

- 프로젝트 배경
- 프로젝트 주제 및 내용
- 프로젝트 세부 내용
- 프로젝트 수행 결과물(예시)

# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 주제 및 내용

역량 요구사항	필요 기술(직무 역량)	인력양성 목표(훈련 목표)
<ol style="list-style-type: none"> <li>1. Verilog HDL 설계 이해</li> <li>2. ARM Architecture 이해</li> <li>3. AMBA BUS 이해</li> <li>4. Memory Map I/O 이해</li> <li>5. UART/RS485/SPI 통신protocol</li> <li>6. AHB / APB / AXI Bus 이해</li> <li>7. Module Level 설계</li> <li>8. Top Level 설계</li> <li>9. Memory 동작 이해</li> <li>10. 조합논리회로 설계 이해</li> <li>11. 순차논리회로 설계 이해</li> <li>12. C언어 활용 능력</li> </ol>	<ul style="list-style-type: none"> <li>• Verilog HDL 기술</li> <li>• Timing Chart 작성 기술</li> <li>• 활용 기술 EDA Tool</li> <li>• 계측기 사용 기술</li> <li>• 언어 활용 기술 C</li> <li>• 계측기 측정 기술</li> <li>• 설계 사양 기능별 구체적</li> <li>• 기술 능력</li> <li>• 설계 사양 비교분석 기술</li> <li>• 전체 설계 흐름도 작성 기술</li> <li>• 각 단계별 결과물 정리 및 보고서 작성 능력</li> </ul>	<p>참여자의 역량을 반영한 핵심 기술교육과 프로젝트 수행을 통해 Low Level 개발의 SoC 반도체 Peripheral RTC 설계 전문가 양성</p>



구분	프로젝트 주제	난이도
팀 프로젝트	SoC Semiconductor Peripheral RTC Design Project	상

# *SoC Semiconductor Peripheral RTC Design Project*

- 프로젝트 배경
- 프로젝트 주제 및 내용
- 프로젝트 세부 내용
- 프로젝트 수행 결과물(예시)



# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 주제 및 내용(1)

프로젝트 팀 주제	세부 수행 내용	습득 직무 역량
SoC Semiconductor Peripheral RTC Design Project	<ul style="list-style-type: none"> <li>➤ 반도체 설계 수행</li> <li>✓ 계층구조 설계</li> <li>✓ 연동 AXI4 Lite Channel</li> <li>✓ 디지털 IP Block Design</li> <li>✓ IP Module Test &amp; Verification</li> <li>✓ ARM Core 와 IP 연동</li> <li>✓ Display Control 반도체 설계</li> <li>✓ 메모리 설계</li> <li>✓ Timer/Counter 설계</li> <li>✓ FPGA 보드에서 검증</li> <li>✓ Verification</li> </ul>	<ul style="list-style-type: none"> <li>➤ 지식</li> <li>✓ Verilog HDL 언어 이해</li> <li>✓ 개발목표 및 사양에 대한 구체적 지식</li> <li>✓ 설계 흐름도에 맞춰 설계 장비를 선정하는 지식</li> <li>✓ 불 대수</li> <li>✓ 조합회로 순차회로 설계 및 해석</li> <li>✓ FSM 구조 및 해석</li> <li>✓ ARM System Architecture 이해</li> <li>✓ Memory Mapped I/O 이해</li> <li>✓ 메시지 송수신의 Protocol 설계 능력</li> <li>✓ 메모리 활용 능력</li> <li>✓ CPU 와 Peripheral BUS 간의 통신 이해 능력</li> <li>✓ 반도체 설계 검증 능력</li> <li>✓ AMBA BUS 이해</li> <li>✓ AXI4-Lite Channel 설계</li> <li>✓ 사용할 기능 블록과 IP 들에 대한 지식</li> <li>✓ SW 구조 설계 방법</li> <li>✓ FSM 구조 및 해석</li> <li>✓ 동기 비동기 시스템 구조</li> <li>✓ Timing and Waverform 해석</li> </ul>

# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 주제 및 내용(2)

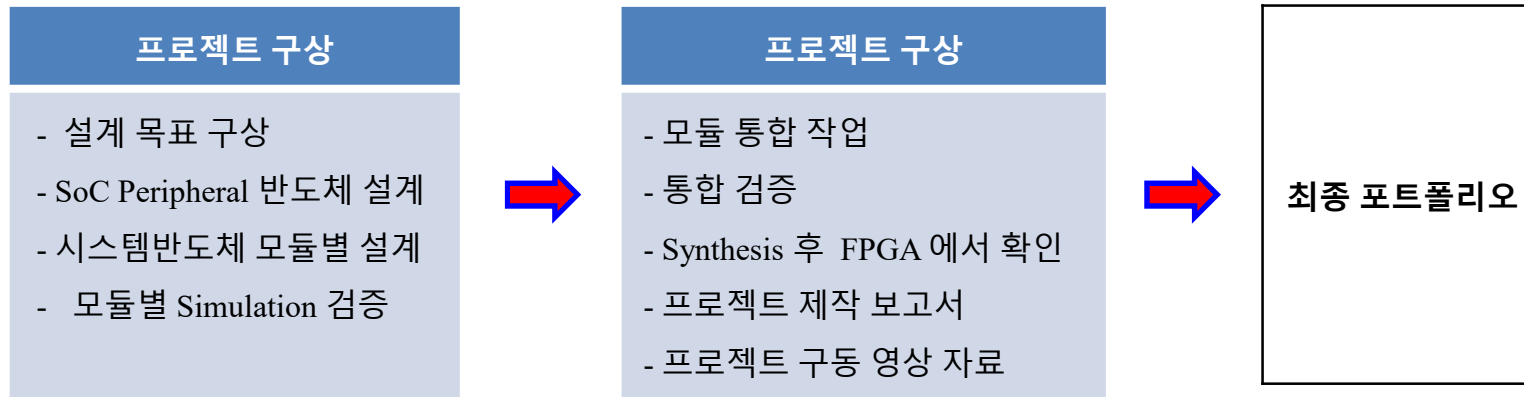
프로젝트 팀 주제	세부 수행 내용	습득 직무 역량
SoC Semiconductor Peripheral RTC Design Project	<ul style="list-style-type: none"> <li>➤ S/W 수행</li> <li>✓ SW 구조 설계</li> <li>✓ Machine Code 작성(C 언어)</li> </ul>	<ul style="list-style-type: none"> <li>➤ 기술                             <ul style="list-style-type: none"> <li>✓ Verilog HDL 활용 기술</li> <li>✓ 반도체 Timing Chart 검증 기술</li> <li>✓ C 언어 활용 기술</li> <li>✓ Memory 설계 기술</li> <li>✓ Timer/Counter 설계 기술</li> <li>✓ AXI4-Lite Channel 설계 기술</li> <li>✓ TestBench 작성 기술</li> <li>✓ 각 설계 블록별 주요 레지스터 맵 작성 기술</li> </ul> </li> <li>✓ 시스템 버스를 운영하는 소프트웨어 해석 능력</li> <li>✓ 소프트웨어를 포함한 시스템 레벨에서의 분석 및 설계 기술</li> </ul>
		<ul style="list-style-type: none"> <li>➤ 기술                             <ul style="list-style-type: none"> <li>✓ 목표 설계부터 구현 검증 버그 해결 그리고 결과물까지 전체 경험 Flow</li> <li>✓ 요구사항에 대한 분석 및 충분한 이해</li> <li>✓ 설계 진행상황을 구체적으로 파악하는 태도</li> <li>✓ 문제점 대응 방안을 수립하려는 노력</li> <li>✓ 현실적으로 구현 가능한 아키텍처를 능동 적으로 정의하는 태도</li> <li>✓ 반도체 제품 내부 시스템을 이해하려는 노력</li> <li>✓ 팀 커뮤니케이션 역량 습득</li> <li>✓ 결과 발표 역량 습득</li> </ul> </li> </ul>

# *SoC Semiconductor Peripheral RTC Design Project*

- 프로젝트 배경
- 프로젝트 주제 및 내용
- 프로젝트 세부 내용
- 프로젝트 수행 결과물(예시)

## ➤ 프로젝트 과제 (결과물 예시)

- 전공교과에서 학습한 내용과 연계하여 최종 포트폴리오에 활용할 수 있도록 기획
  - 참여기업에서 실제 진행 중이거나 진행한 프로젝트의 연관성을 고려하여 수행
  - 프로젝트별 주제에 참여한 기업 소속의 실무자가 멘토로 참여
  - 종합 실무 프로젝트의 산출물을 활용하여 최종 포트폴리오 완성



## ➤ 프로젝트 수행 결과물 (예시) [ 1 / 33 ] ➔ 설계 목표

### 프로젝트 포트폴리오

#### 프로젝트 결과물 예시

#### 프로젝트 내용

Part 1,

## 설계 목표

- MicroBlaze Soft Processor Core와 AXI4 Lite AMBA BUS를 활용한 Peripheral 설계 프로젝트를 진행한다.
- 개요
  - Microblaze로 세탁기 기능을 FSM을 이용해 설계
  - Microblaze & AXI4Lite AMBA BUS를 이용해 PWM, FND, RTC Peripheral IP 설계
  - 세탁기의 Auto, Select mode별 세탁, 헹굼, 탈수 기능별로 PWM 제어
  - Timer IP : 1second 단위로 INTERRUPT 발생
  - UART IP : Bluetooth 모듈을 이용해 세탁기 제어

목표 설정

# SoC Semiconductor Peripheral RTC Design Project

.xpr

## 프로젝트 수행 결과물 (예시) [ 2 / 33 ] → 설계 사양(1/3)

### 프로젝트 포트폴리오

#### 프로젝트 결과물 예시

#### 프로젝트 내용

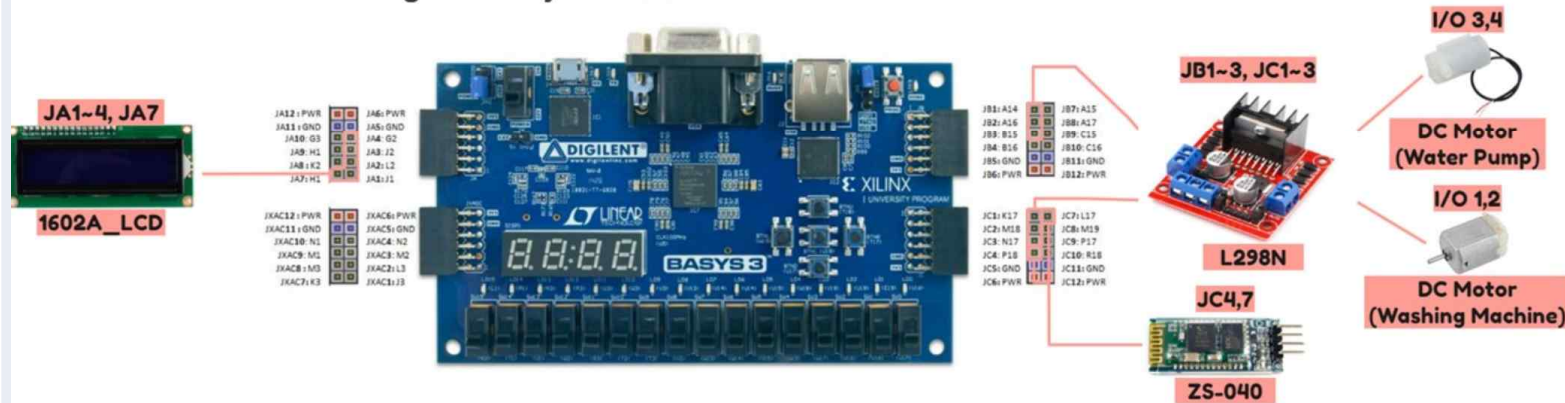
Part 2,

## 개발 환경 & PIN 배치

Xilinx VIVADO & VITIS



Digilent Basys 3™ Artix-7 FPGA Trainer Board



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 3 / 33 ] ➔ 설계 사양(2/3)

## 프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

### Micro Blaze Design Input / Output Spec

I/O	Signal Name	Bit	Description
Input	Sys_clock	1	Input 100MHz Clock Source
Input	reset	1	Input reset pushbutton
Input	i_btn_tri_io	1	Input Wash Machine control pushbutton
Input	usb_uart_rxd	1	UART Rx signal
Input	o_lcd_tri_io	7	Output LCD Data
Output	o_fnd_font	8	Output 7-Segment font data
Output	o_fnd_sel	4	Output 7-Segment select
Output	o_pump_en	1	Enable for pump
Output	o_motor_en	1	Enable for motor
Output	o_pump_pwm	2	Pump_pwm
Output	o_motor_pwm	2	Motor_pwm
Output	usb_uart_txd	1	UART Tx signal

➤ 설계 사양

➔ Input Spec

➔ Output Spec

## ➤ 프로젝트 수행 결과물 (예시) [ 4 / 33 ] ➔ 설계 사양(3/3)

### 프로젝트 포트폴리오

#### 프로젝트 결과물 예시

#### 프로젝트 내용

## IP 별 구현 기능

IP	세부 기능
RTC (Real Time Clock)	시간 설정이 가능한 시계
FND	각 세탁 모드와 남은 시간 표시
PWM	각 모드 별 세탁기 모터 제어 Water Pump 제어
UART	UART통신 Bluetooth 제어 휴대폰 입력 세탁기 제어 Blaze RTC 시간 출력
GPIO	LCD Data, Button
Timer	Interrupt를 이용해 1초를 count하는 down counter제작

#### ➤ 설계 사양

#### ➔ IP

- RTC(Real Time Clock)
- FND
- PWM
- UART
- GPIO
- Timer



# SoC Semiconductor Peripheral RTC Design Project

.xpr

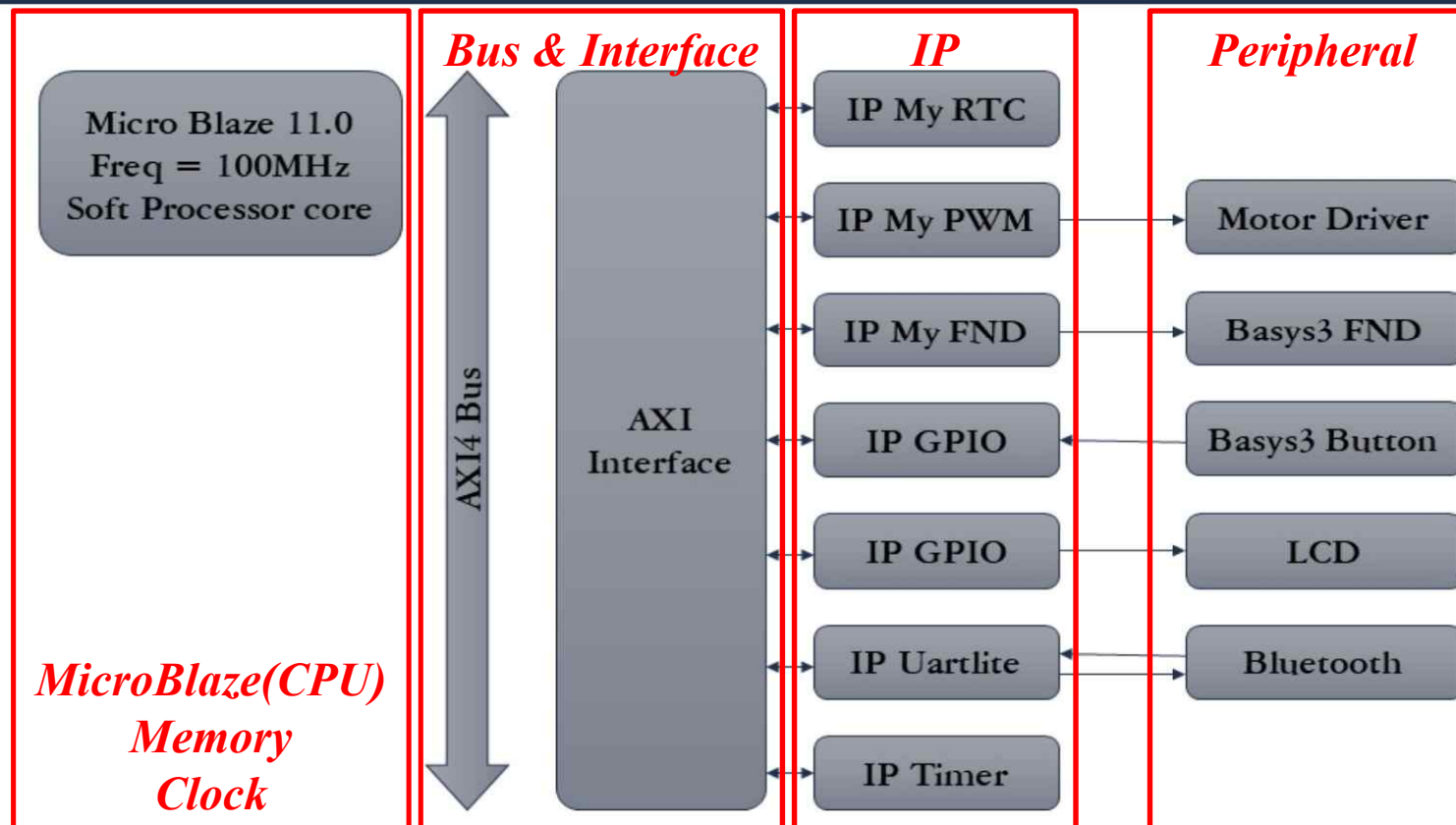
➤ 프로젝트 수행 결과물 (예시) [ 5 / 33 ] ➔ *Block Diagram*

프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

## Block Diagram



*Block Diagram*

➔ IP

- RTC(Real Time Clock)
- FND
- PWM
- UART
- GPIO
- Timer

➔ Peripheral

- Motor(Driver)
- FND(Basys3)
- Button(Basys3)
- LCD
- Bluetooth

# SoC Semiconductor Peripheral RTC Design Project

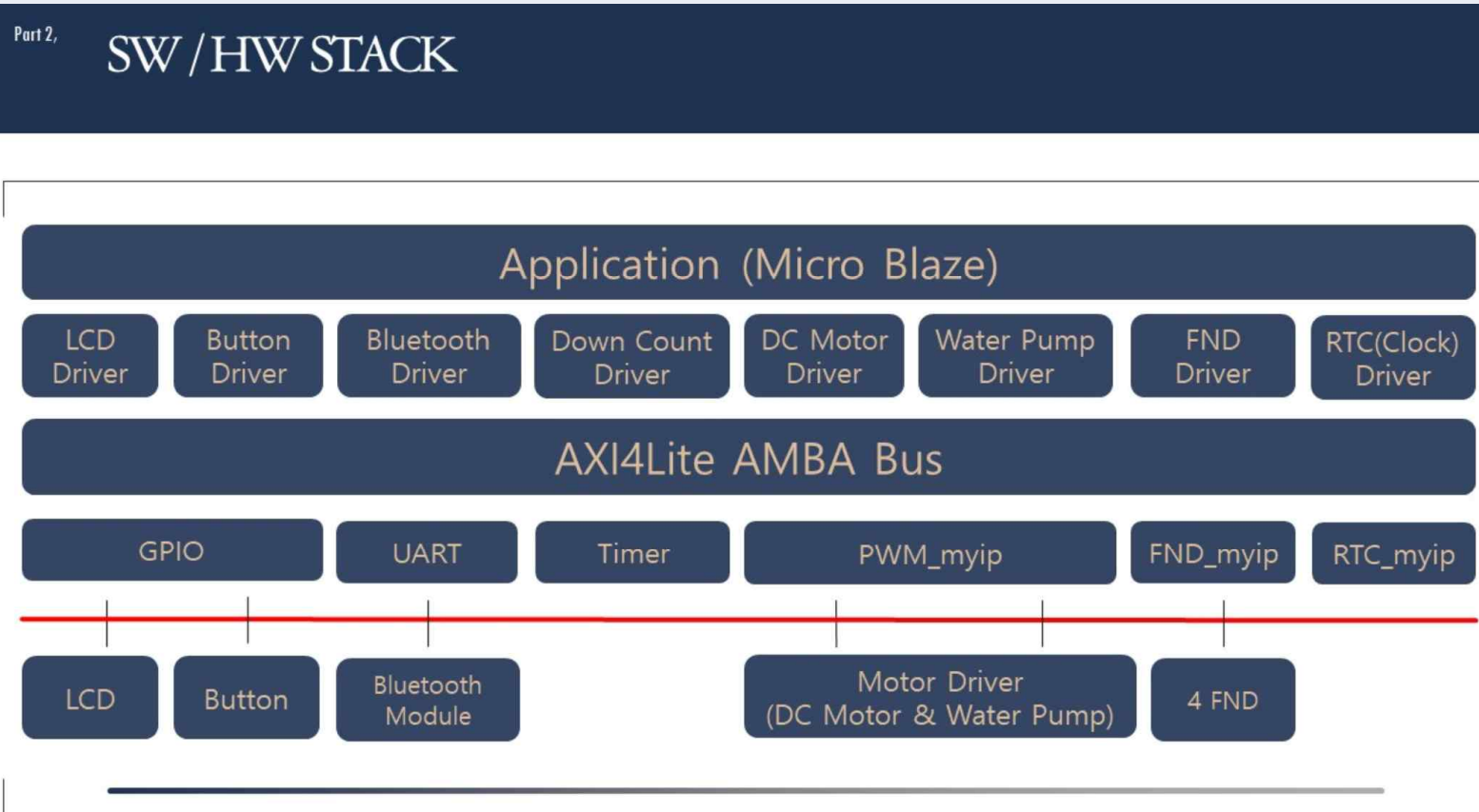
.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 6 / 33 ] → SW/HW STACK

### 프로젝트 포트폴리오

#### 프로젝트 결과물 예시

#### 프로젝트 내용



#### SW/HW STACK

##### → IP

- RTC(Real Time Clock)
- FND
- PWM
- UART
- GPIO
- Timer

##### → Peripheral

- Motor(Driver)
- DC Motor
- Water Pump
- FND(Basys3)
- Button(Basys3)
- LCD
- Bluetooth

# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 7 / 33 ] ➔ *FSM Mode & PWM*

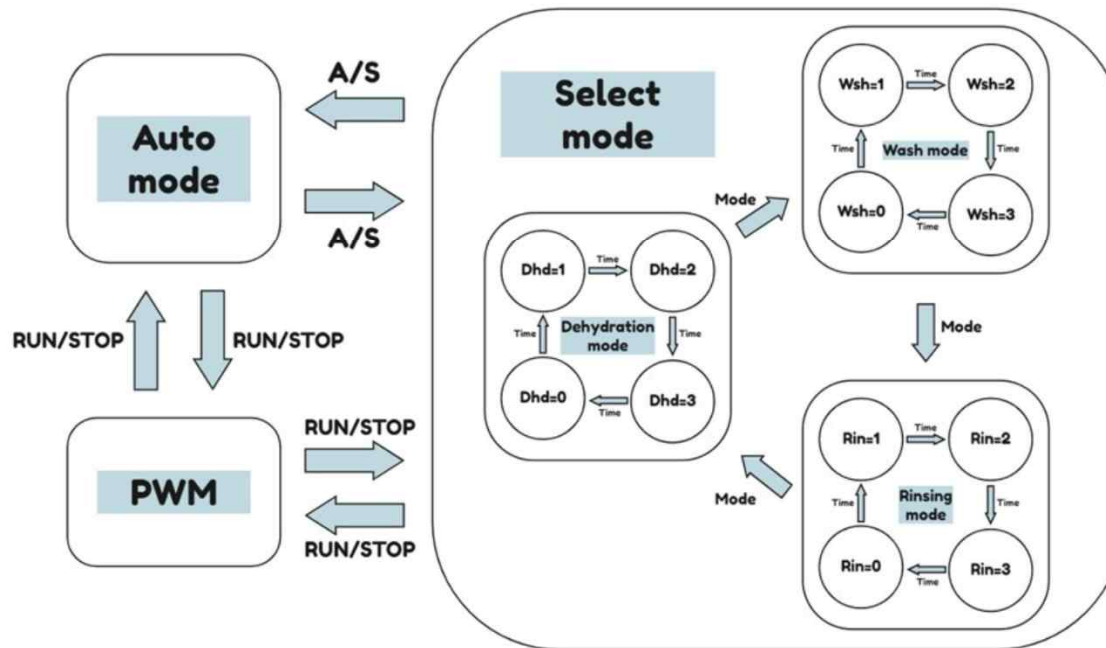
### 프로젝트 포트폴리오

#### 프로젝트 결과물 예시

#### 프로젝트 내용

Part 2,

FSM



*FSM*

➔ Auto Mode

➔ Select Mode

➔ Button Set

➔ PWM

# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 8 / 33 ] → *FSM\_PWM*

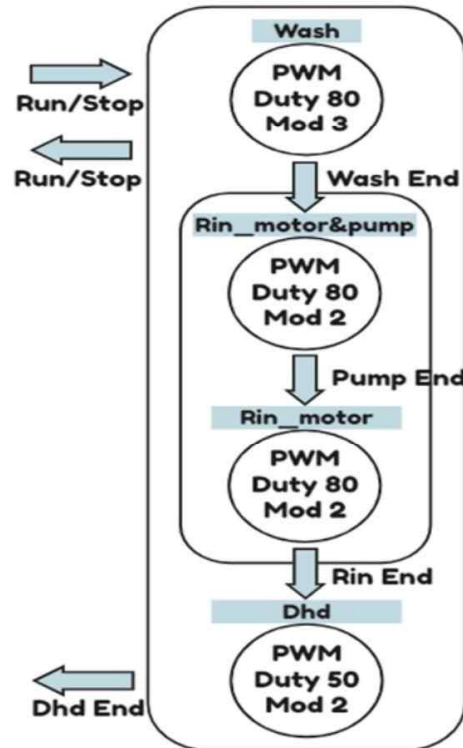
### 프로젝트 포트폴리오

#### 프로젝트 결과물 예시

#### 프로젝트 내용

Part 2,  
FSM

#### PWM FSM



#### PWM

Mode 0 : Stop

Mode 1 : Left

Mode 2 : Right

Mode 3 : Left → Stop → Right → Stop

Wash : Duty 80 양방향 5초 주기 회전(Mode 3)

Rinse : Duty 80 한방향(Mode2)  
+ Duty 80 Water Pump ON(Mode2)

Dehydration : Duty 50 한방향 (Mode2)

각 State 별 Run/Stop 가능

#### *PWM FSM*

→ PWM Mode

- 0 : Stop
- 1 : Left
- 2 : Right
- 3 : Left – Stop  
– Right – Stop

→ Wash

- Mode 3, 양방향
- Duty 80, 5초

→ Rinse

- Mode2, 한방향
- Duty 80
- Water Pump ON

→ Dehydration

- Mode2, 한방향
- Duty 50

# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 9 / 33 ] ➔ *IP Block Design*

## 프로젝트 포트폴리오

### 프로젝트 결과물 예시

### 프로젝트 내용

Part 3,

## IP Block Design

- PWM
- FND
- RTC
- GPIO
- Timer
- UART
- Project

### *IP Block Design*

➔ PWM

➔ FND

➔ RTC

➔ GPIO

➔ Timer

➔ UART

➔ Project

# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 10 / 33 ] ➔ IP Block Design ➔ PWM Generate

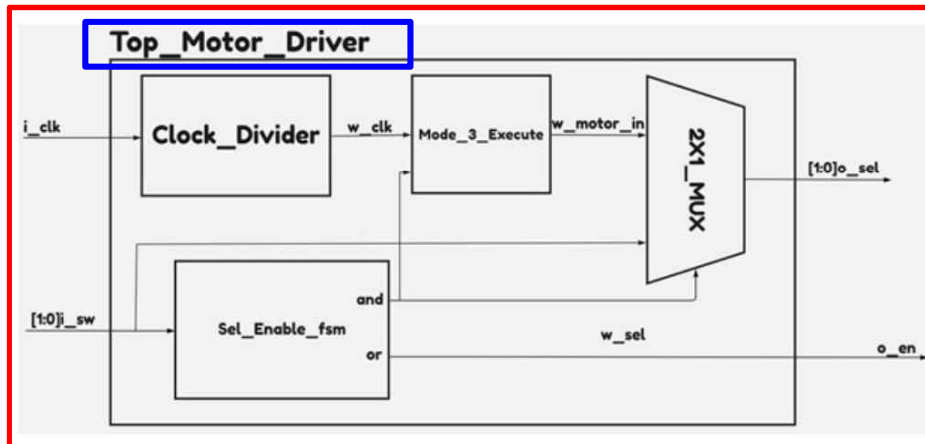
프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 3,

## PWM Driver Sub Module - RTL Design IP



Top\_Motor\_Driver

Clock\_Divider : 100MHz → 1Hz

Mode\_3\_Execute : 5sec mode change

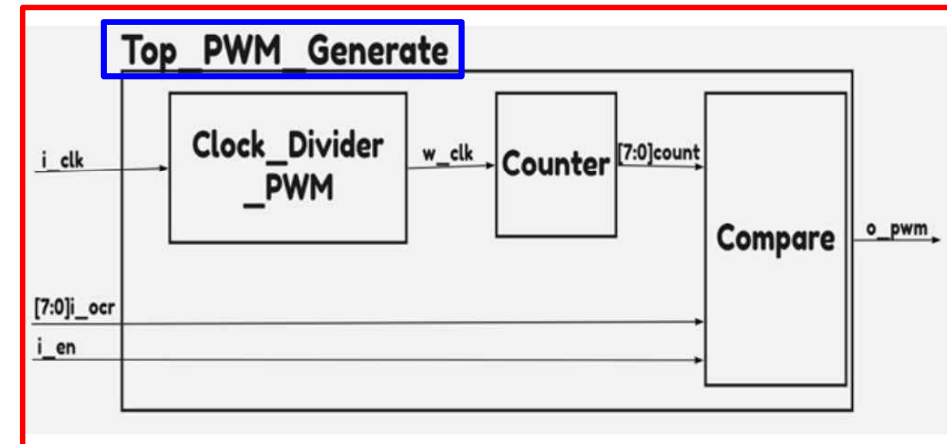
Mux Select pin에 Switch 2bit AND 할당

Mux Input A = input\_switch    입력 00, 01, 10    출력 = A

Mux Input B = mod\_3 execute    입력 11    출력 = B

11의 경우 Mod 3 Execute 동작 및 Mux Select 변경

→ 모터의 정지, 정방향, 역방향, 자동 방향 조절 모드 별 출력 신호 제어용



PWM\_Generator

Clock\_Divider\_PWM : 100MHz → 1MHz

Counter : 100

Compare : less than

1MHz(1us) 주기 → 100 count

10kHz(100us) 주기 → 10kHz PWM 출력

OCR 0~99 와 count 비교로 less than 출력 PWM 생성

o\_lt = (i\_en == 1'b1) ? (i\_a < i\_b) : 0

i\_en = 1일 때 i\_a(count) < i\_b(i\_ocr) 참이면 1 거짓이면 0

→ PWM 생성용

# SoC Semiconductor Peripheral RTC Design Project

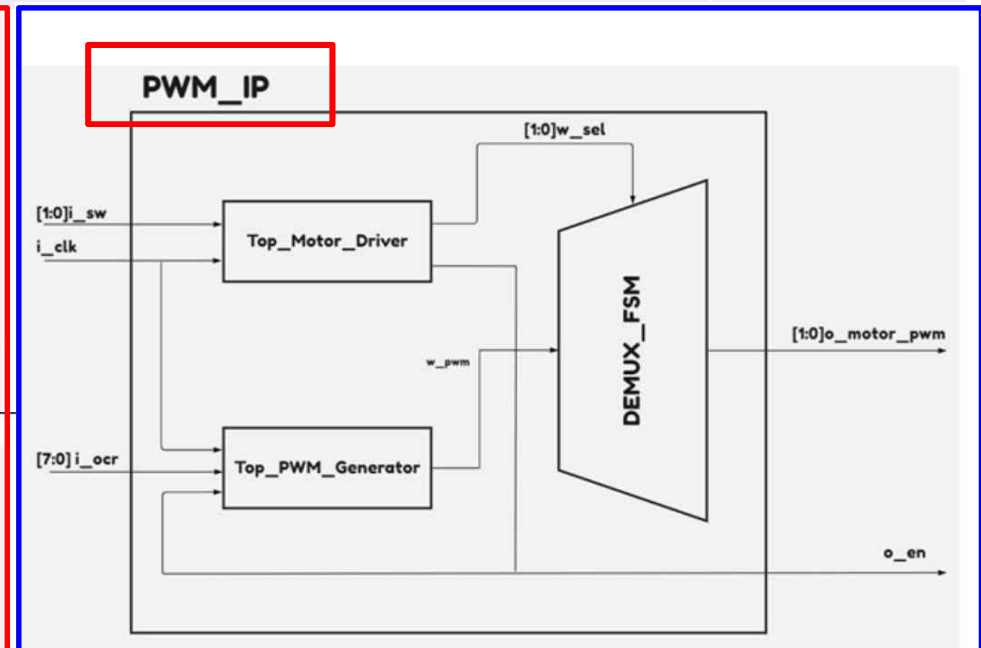
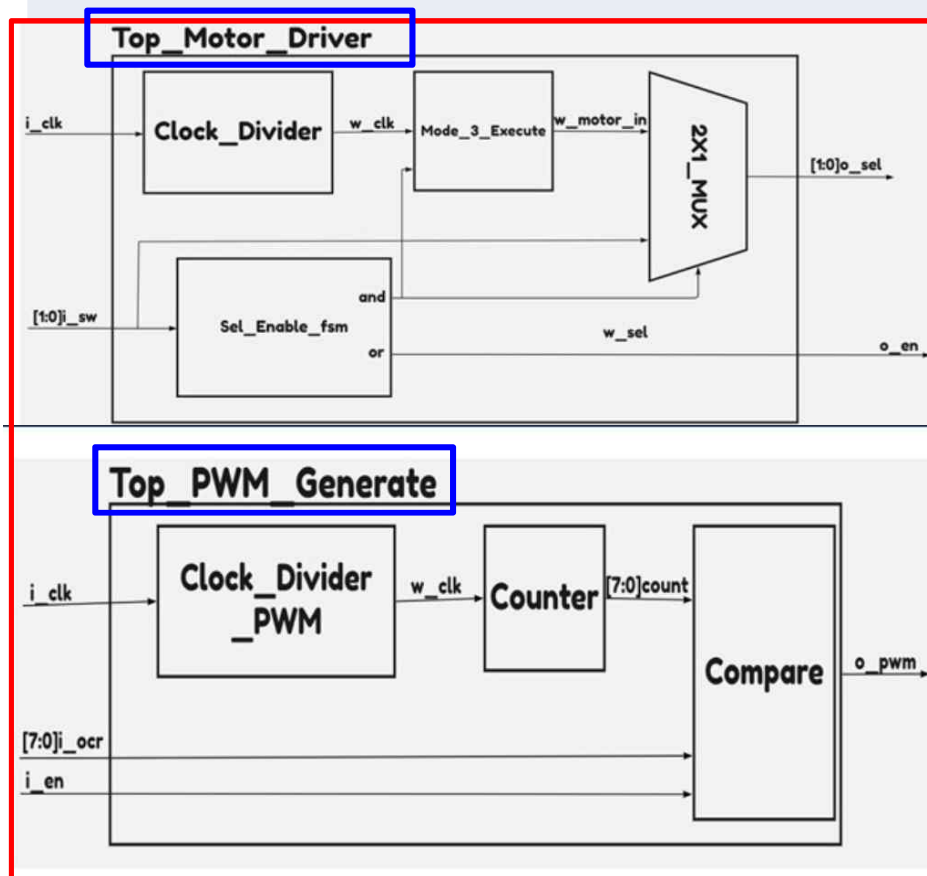
.xpr

➤ 프로젝트 수행 결과물 (예시) [ 11 / 33 ] ➔ IP Block Design ➔ PWM IP

## 프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용



PWM 양방향 제어 위하여  
H/W Motor Driver의 In1, In2에 들어가는 신호를 Demux로 변경

Duty 50



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 12 / 33 ] ➔ IP Block Design ➔ PWM IP & AXI 4 Lite

프로젝트 포트폴리오

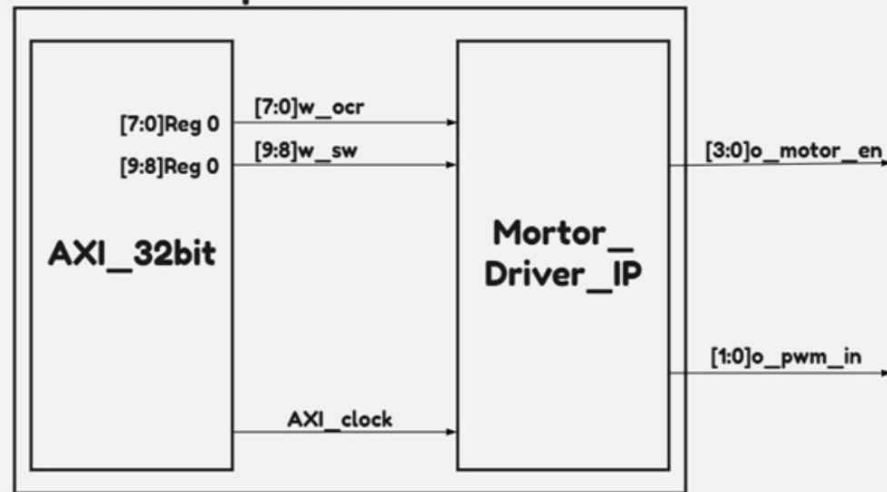
프로젝트 결과물 예시

프로젝트 내용

Part 3,

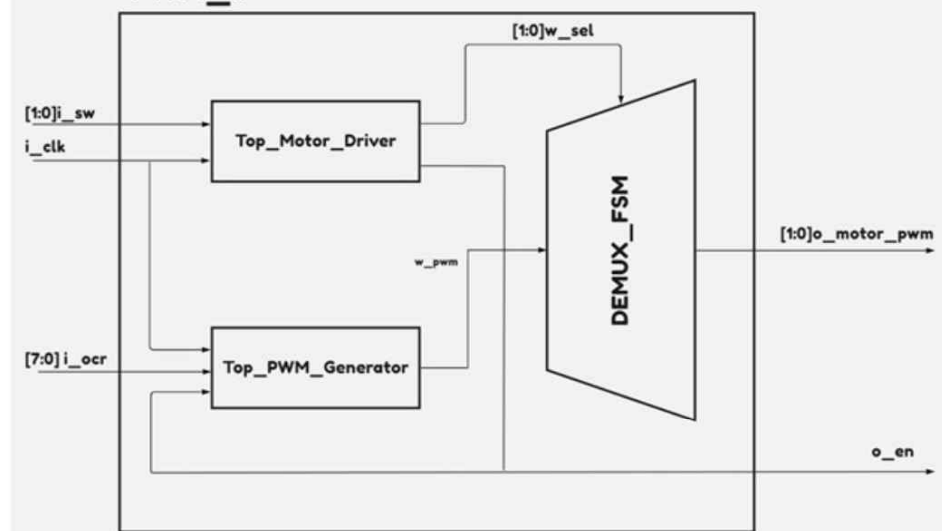
## PWM Driver Top Module - RTL Design IP

### AXI 4Lite Top PWM



Blaze에서 AXI Bus 이용  
Reg 0 8비트 Duty 제어 위한 OCR 설정 출력 IP 입력 받음  
Reg 0 2비트 Mode Data 출력 IP 입력 받음

### PWM\_IP



PWM 양방향 제어 위하여  
H/W Motor Driver의 In1, In2에 들어가는 신호를 Demux로 변경



# SoC Semiconductor Peripheral RTC Design Project

.xpr

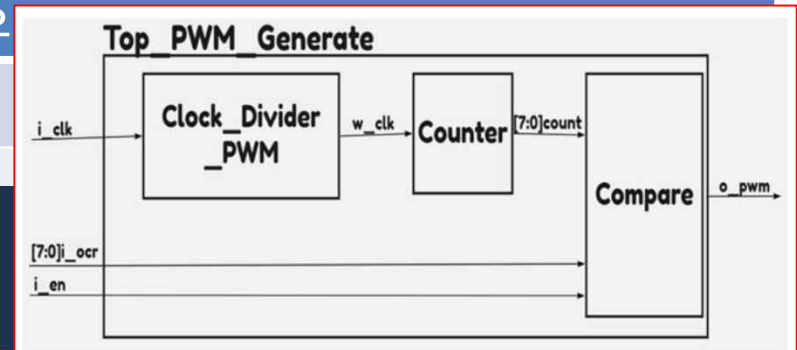
➤ 프로젝트 수행 결과물 (예시) [ 13 / 33 ] ➔ IP Block Design ➔ PWM Simulation

프로젝트 포트폴리오

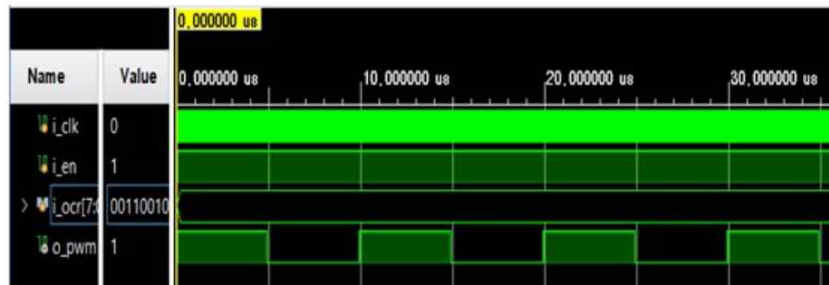
프로젝트 결과물 예시

Part 3,

## PWM Simulation - RTL Design IP



## TEST BENCH SIMULATION



OCR = 50

timescale 1ns / 1ps

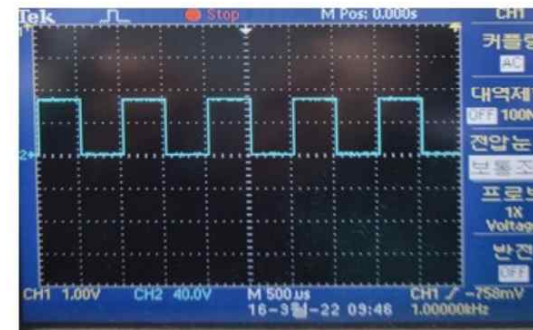
```

module pwm_test_tb(
);
    reg i_clk;
    reg i_en;
    reg [7:0] i_ocr;
    wire o_pwm;

    top_pwm_gen tpg(
        .i_clk(i_clk),
        .i_en(i_en),
        .i_ocr(i_ocr),
        .o_pwm(o_pwm)
    );

    initial begin
        i_clk = 0; i_en = 1; i_ocr = 50;
    end
    always #5 i_clk = ~i_clk;
endmodule
    
```

## Oscilloscope Waveform



OCR = 50  
→ Duty Rate = 50



OCR = 90  
→ Duty Rate = 90

T = 100us  
f = 100kHz

# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 14 / 33 ] ➔ IP Block Design ➔ PWM Driver RTL

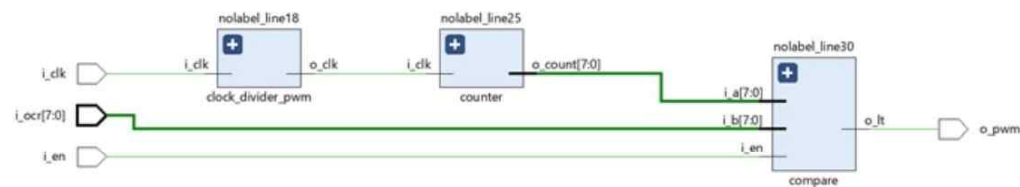
프로젝트 포트폴리오

프로젝트 결과물 예시

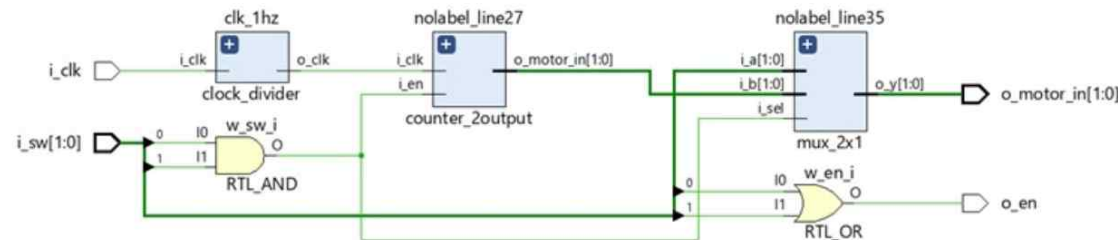
프로젝트 내용

Part 3,

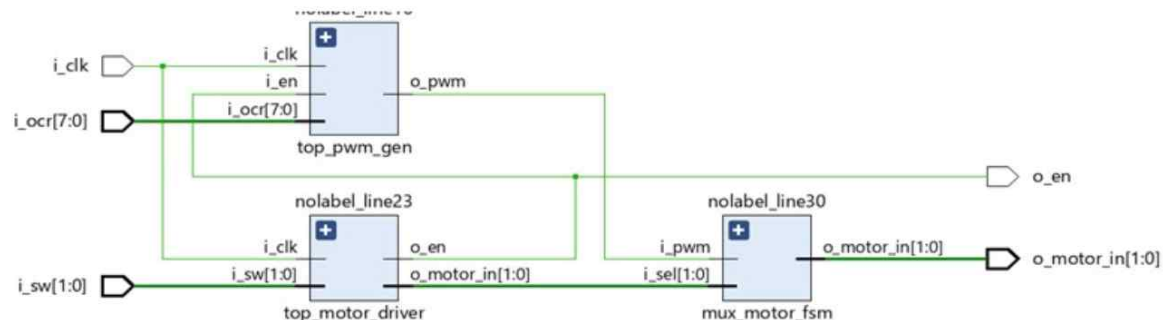
## PWM Driver RTL View- RTL Design IP



### Sub Module



### Top Module



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 15 / 33 ] ➔ IP Block Design ➔ PWM Driver AXI4 RTL & IP

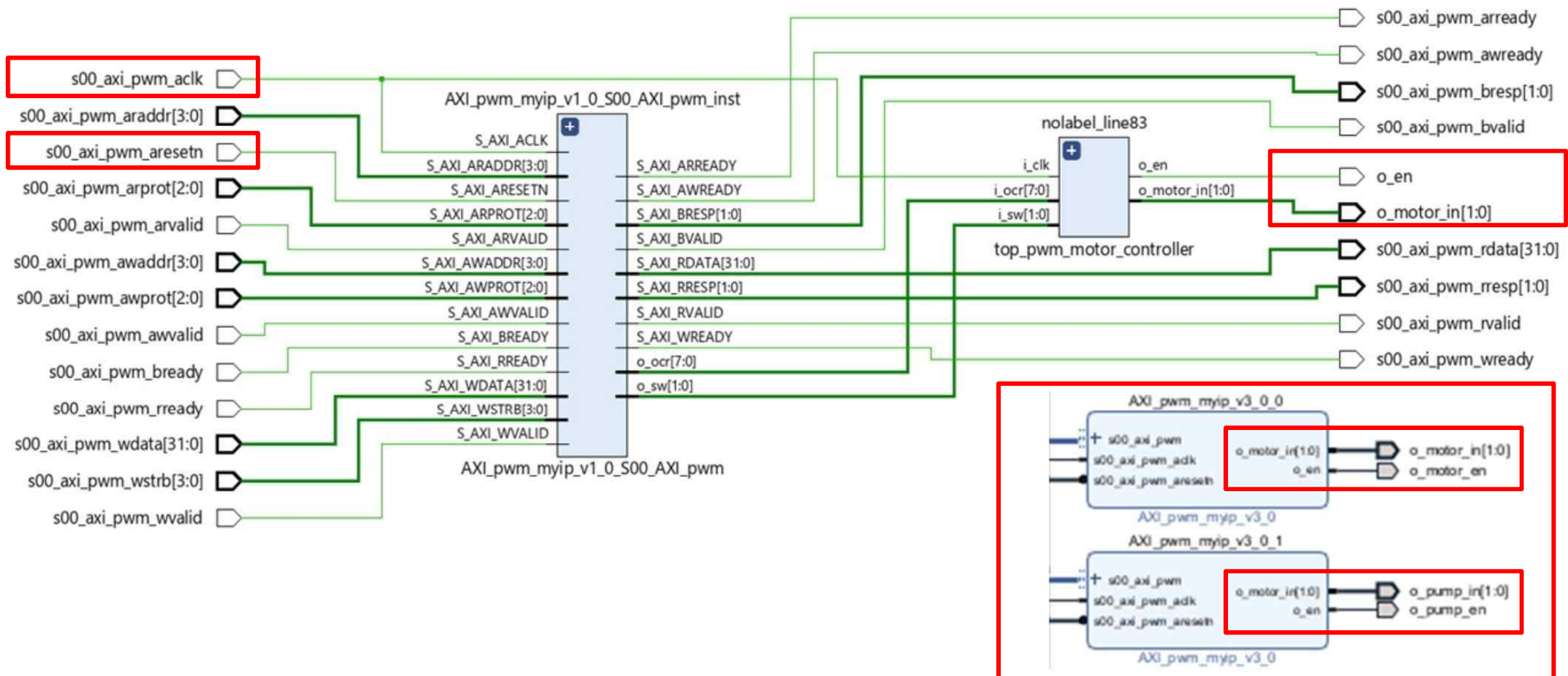
## 프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 3,

## PWM Driver AXI4 RTL View - RTL Design IP



# SoC Semiconductor Peripheral RTC Design Project

.xpr

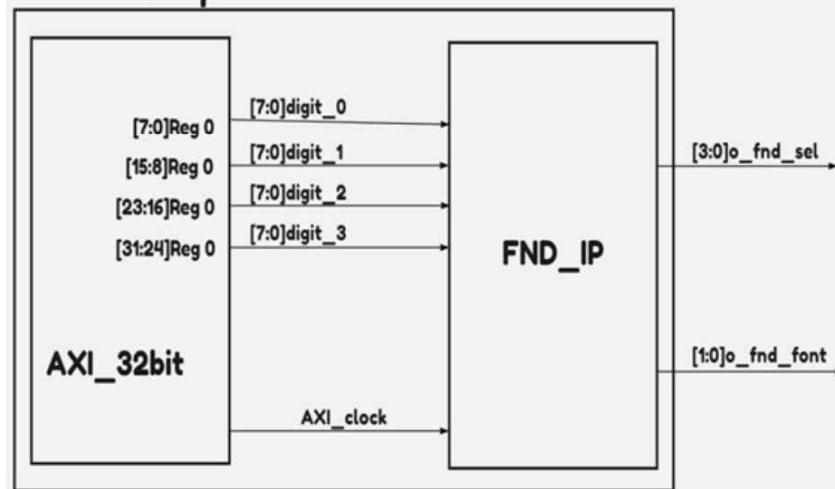
➤ 프로젝트 수행 결과물 (예시) [ 16 / 33 ] ➔ IP Block Design ➔ FND IP & AXI4Lite

프로젝트 포트폴리오

Part 3,

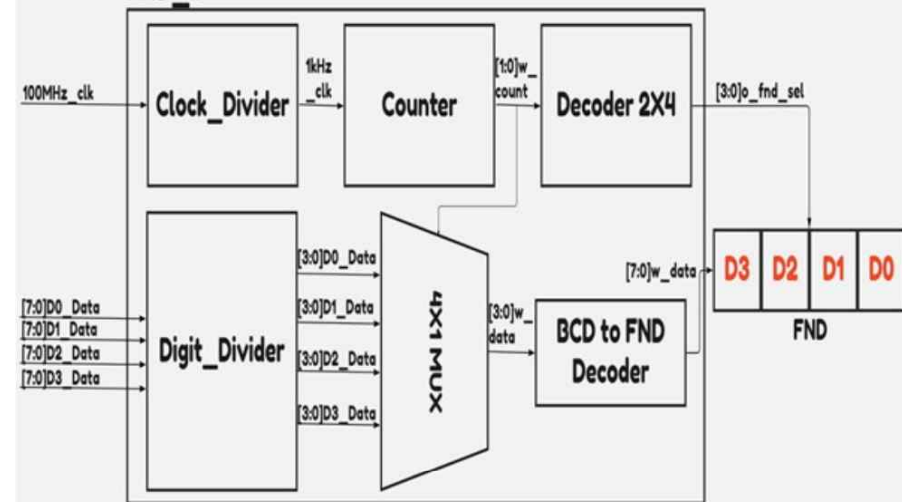
## FND Top Module - RTL Design IP

AXI 4Lite Top FND



Blaze에서 AXI Bus 이용  
Reg 0 32bit FND data 출력 FND 입력 받음

FND\_IP



Clock\_Divider : 100Mhz → 1khz(0.1sec)  
Counter : 2bit(0~3) counter  
Decoder\_2X4 : 2→4(bit), Select 1 FND of 4 FND  
Digit\_Divider : 8bit reg data→ time data  
4X1\_MUX : Select 1 FND FONT  
BCDtoFND\_Decoder : BCD(4bit) to FND FONT(8bit)



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 17 / 33 ] ➔ IP Block Design ➔ FND Simulation

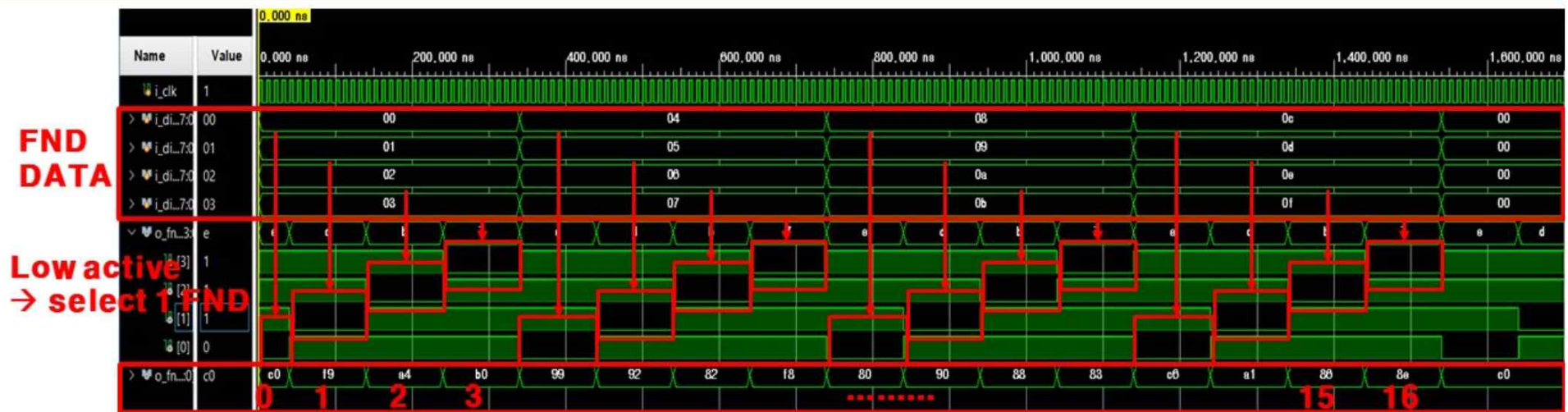
프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 3,

## FND Simulation - RTL Design IP



### testbench code

```
`timescale 1ns / 1ps
```

```
module fnd_test_tb(
);
    reg i_clk;
    reg [7:0] i_digit_0;
    reg [7:0] i_digit_1;
    reg [7:0] i_digit_2;
    reg [7:0] i_digit_3;
    wire [3:0] o_fnd_sel;
    wire [7:0] o_fnd_font;
```

```
    fnd_controller fc123(
        .i_clk(i_clk),
        .i_digit_0(i_digit_0),
        .i_digit_1(i_digit_1),
        .i_digit_2(i_digit_2),
        .i_digit_3(i_digit_3),
        .o_fnd_sel(o_fnd_sel),
        .o_fnd_font(o_fnd_font)
    );
```

```
initial begin
```

```
    i_clk = 1;
    i_digit_0 = 0; i_digit_1 = 1; i_digit_2 = 2; i_digit_3 = 3;
    #340 i_digit_0 = i_digit_0 + 4; i_digit_1 = i_digit_1 + 4; i_digit_2 = i_digit_2 + 4; i_digit_3
    = i_digit_3 + 4;
    #400 i_digit_0 = i_digit_0 + 4; i_digit_1 = i_digit_1 + 4; i_digit_2 = i_digit_2 + 4; i_digit_3
    = i_digit_3 + 4;
    #400 i_digit_0 = i_digit_0 + 4; i_digit_1 = i_digit_1 + 4; i_digit_2 = i_digit_2 + 4; i_digit_3
    = i_digit_3 + 4;
    #400 i_digit_0 = 0; i_digit_1 = 0; i_digit_2 = 0; i_digit_3 = 0;
    end
    always #5 i_clk = ~i_clk;
```

```
endmodule
```

# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 18 / 33 ] ➔ IP Block Design ➔ FND RTL

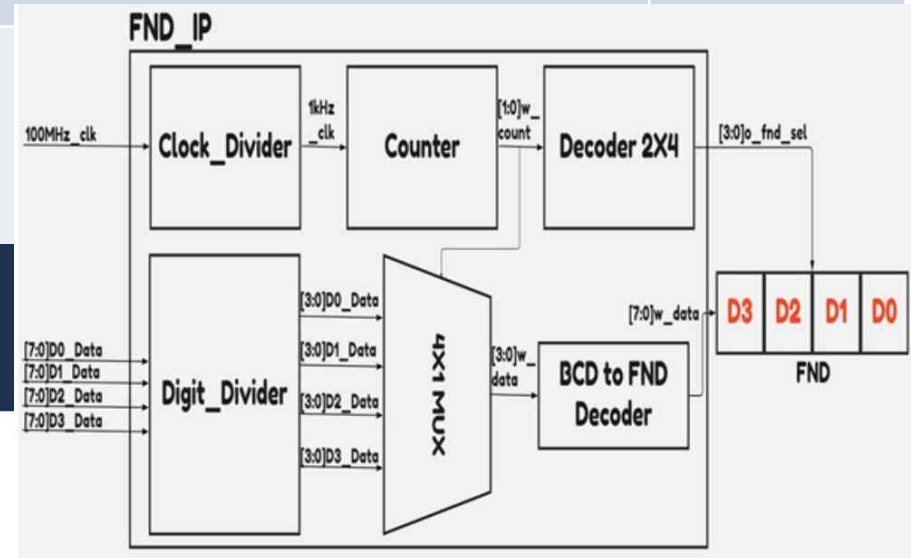
프로젝트 포트폴리오

프로젝트 결과물 예시

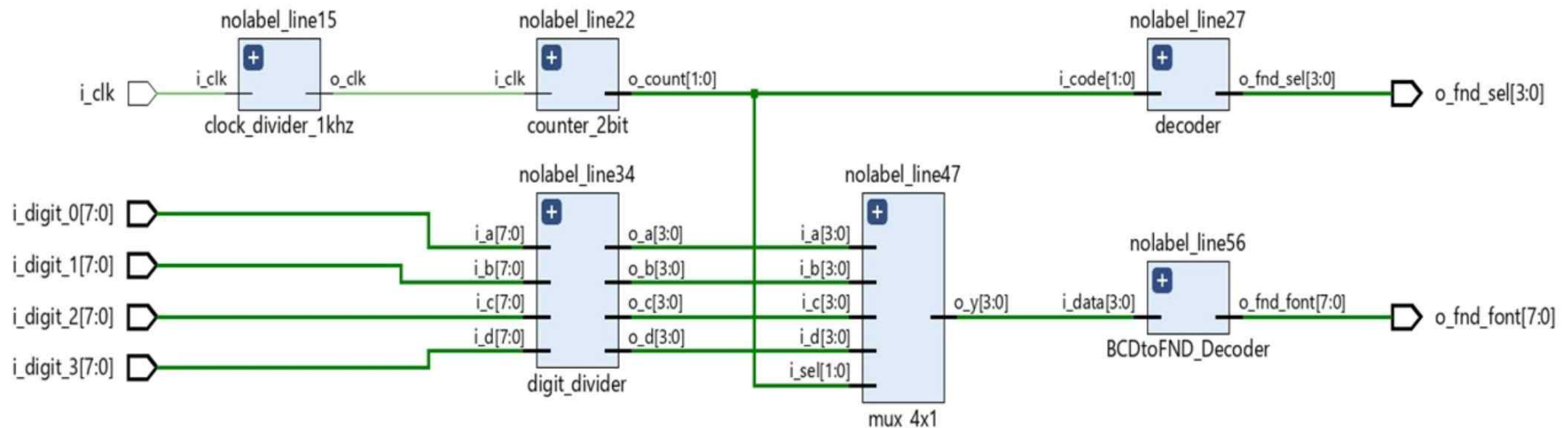
프로젝트 내용

Part 3,

## FND RTL View - RTL Design



### IP Top Module



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 19 / 33 ] ➔ IP Block Design ➔ FND AXI4 RTL & IP

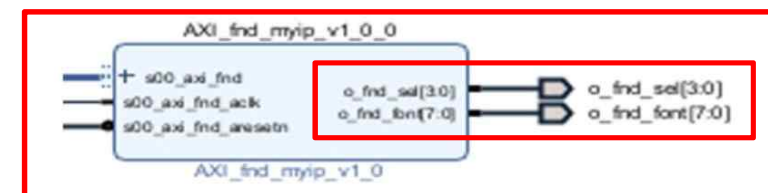
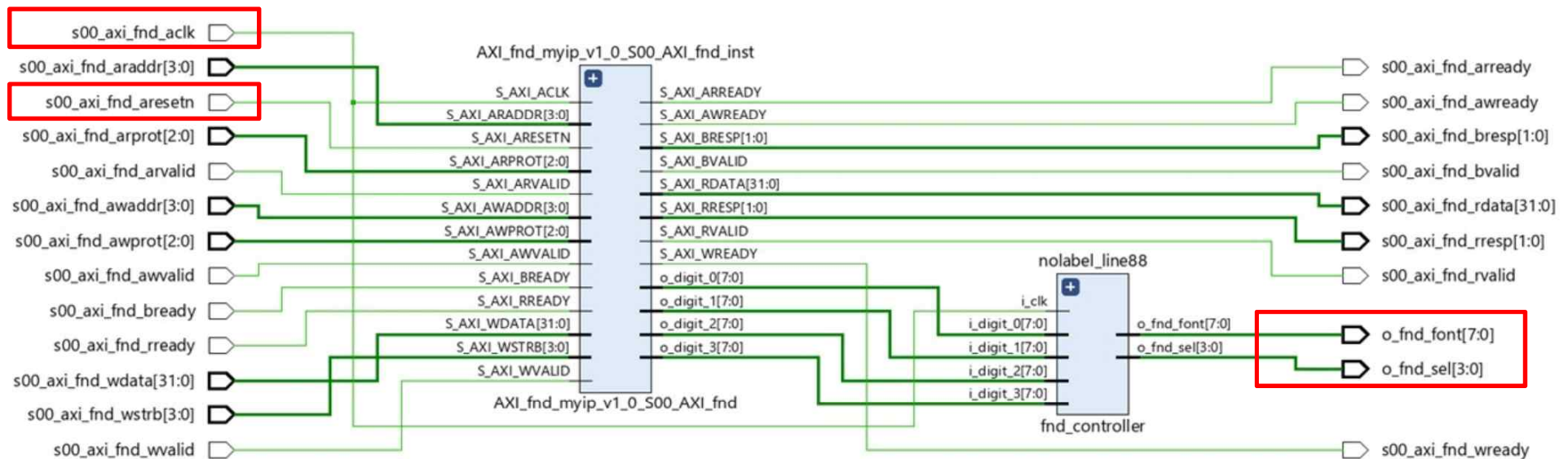
프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 3,

## FND AXI4 RTL View - RTL Design IP



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 20 / 33 ] ➔ IP Block Design ➔ RTC(Real Time Clock)

프로젝트 포트폴리오

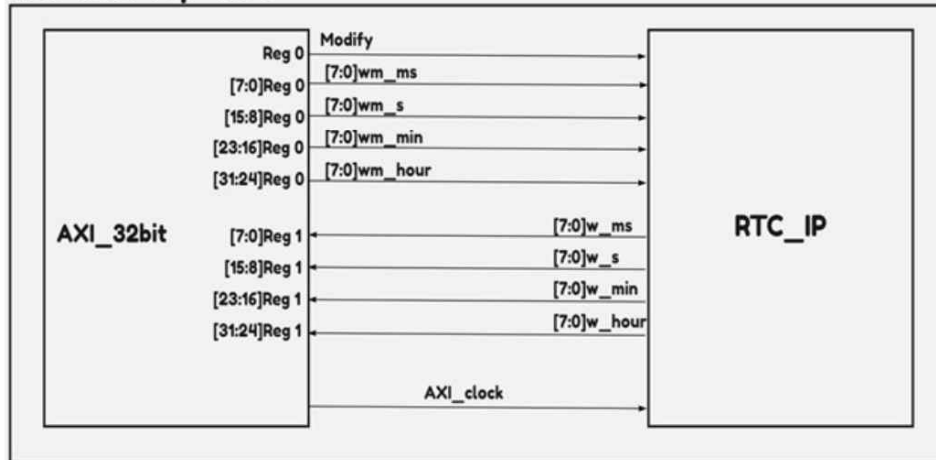
프로젝트 결과물 예시

프로젝트 내용

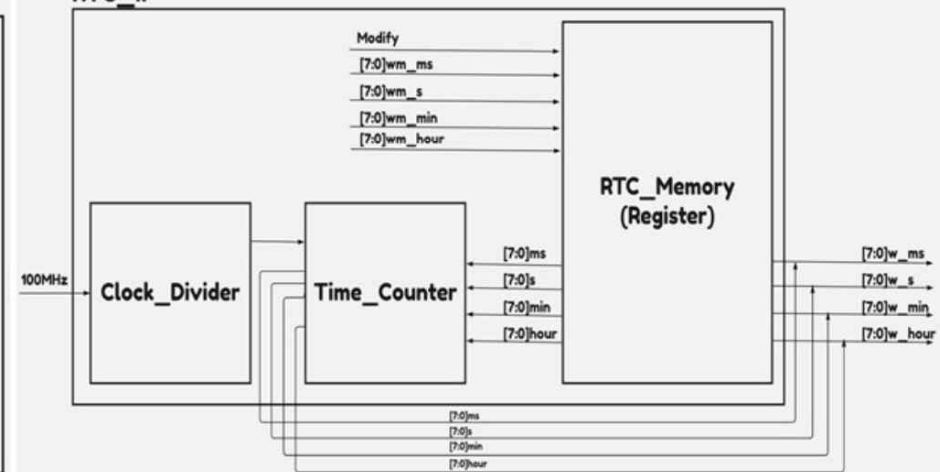
Part 3,

## RTC Top Module - RTL Design IP

### AXI 4Lite Top RTC



### RTC\_IP



Blaze에서 AXI Bus 이용

Reg 0 32bit Modify en & time data로 출력하여 IP 입력 받음

Reg 1 32bit IP 출력의 time data를 Blaze로 입력 받음

Clock\_Divider : 100MHz → 10Hz (1msec)

Time\_Counter : 시간에 대한 계산

RTC\_Memory(Register) :

modify 시간 적용 위한 Memory 역할



# SoC Semiconductor Peripheral RTC Design Project

.xpr

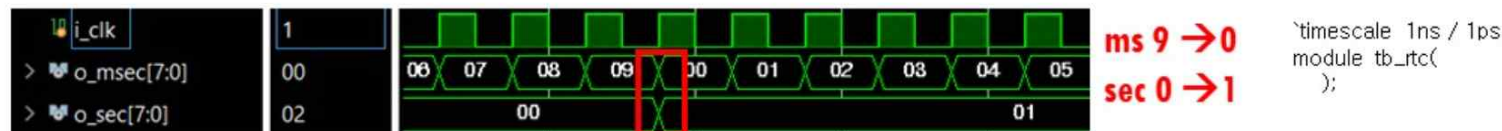
➤ 프로젝트 수행 결과물 (예시) [ 21 / 33 ] ➔ IP Block Design ➔ RTC IP Simulation

프로젝트 포트폴리오

프로젝트 결과물 예시

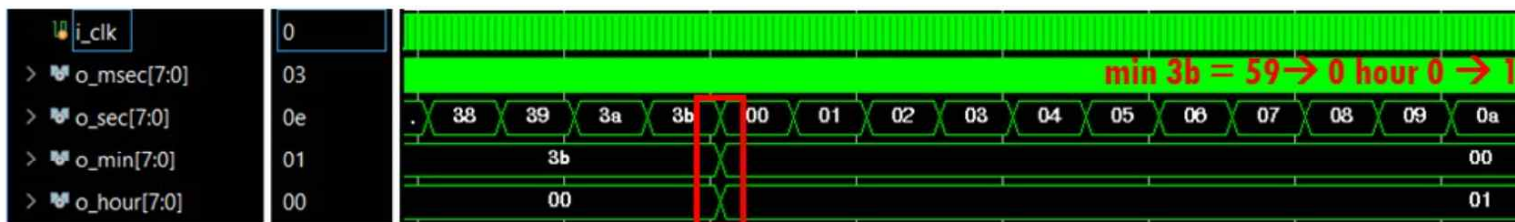
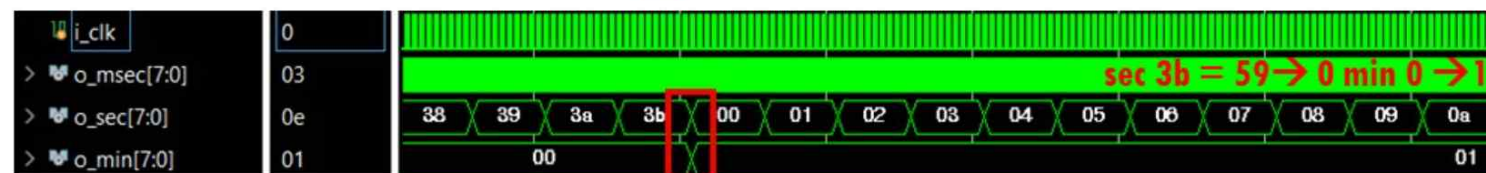
프로젝트 내용

## Part 3, RTC Simulation - RTL Design IP



```
reg i_clk;  
reg i_modify;  
reg [7:0] im_msec;  
reg [7:0] im_sec;  
reg [7:0] im_min;  
reg [7:0] im_hour;  
wire [7:0] o_msec;  
wire [7:0] o_sec;  
wire [7:0] o_min;  
wire [7:0] o_hour;
```

```
RTC_IP rtc(  
    i_clk(i_clk),  
    i_modify(i_modify),  
    im_msec(im_msec),  
    im_sec(im_sec),  
    im_min(im_min),  
    im_hour(im_hour),  
    o_msec(o_msec),  
    o_sec(o_sec),  
    o_min(o_min),  
    o_hour(o_hour)  
);  
initial begin  
    i_clk = 0;  
    im_msec = 0;  
    im_sec = 0;  
    im_min = 0;  
    im_hour = 0;  
    i_modify = 0;  
end  
always #5 i_clk = ~i_clk;
```



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 22 / 33 ] ➔ IP Block Design ➔ RTC RTL

프로젝트 포트폴리오

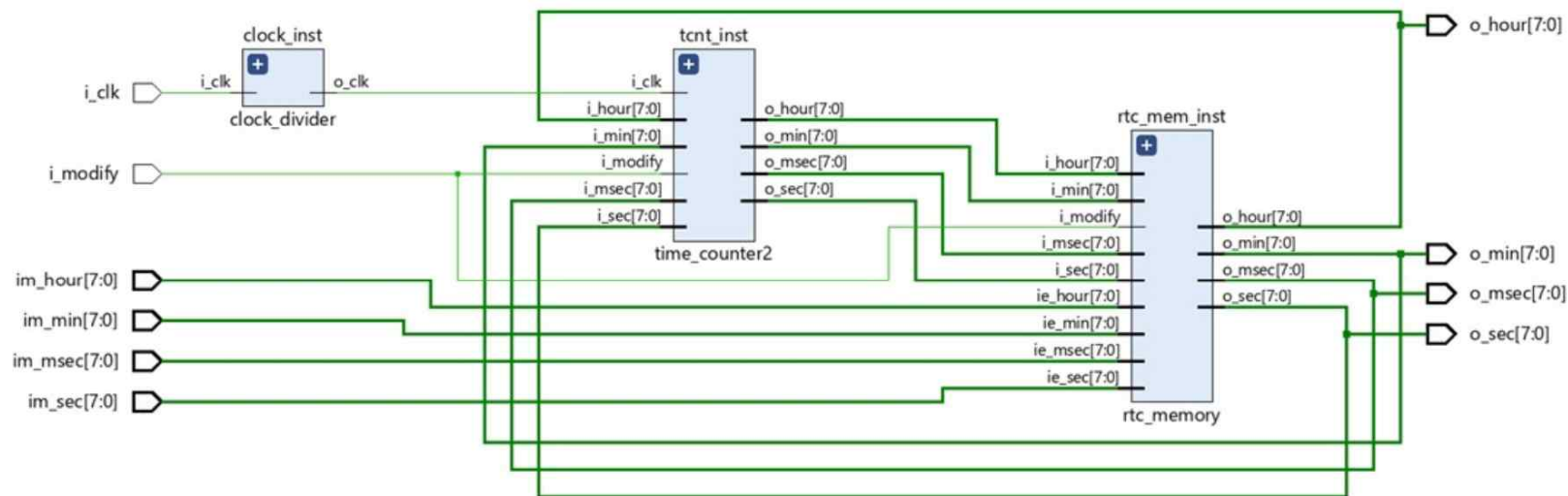
프로젝트 결과물 예시

프로젝트 내용

Part 3,

## RTC RTL View - RTL Design IP

### IP Top Module



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 23 / 33 ] ➔ IP Block Design ➔ RTC AXI4 RTL & IP

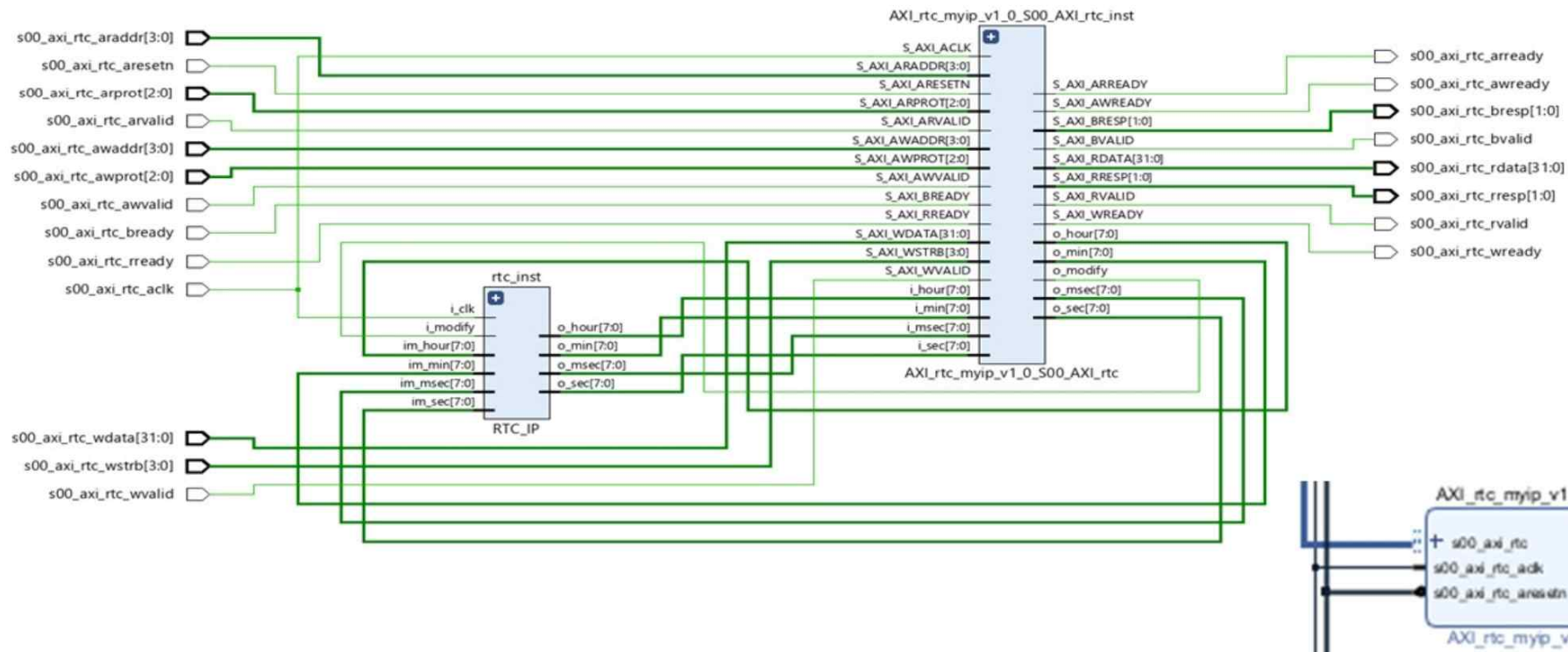
프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 3,

## RTC AXI4 RTL View - RTL Design IP



# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 24 / 33 ] ➔ IP Block Design ➔ GPIO & Timer & UartLite

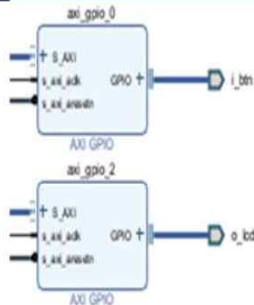
프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

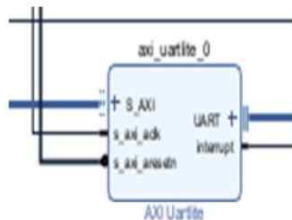
Part 3,

## GPIO & Timer & uartlite - RTL Design IP



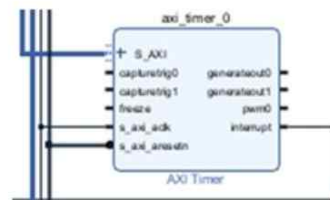
### GPIO

버튼 입력과 LCD 신호를 출력하기 위해 사용



### UART

DATA의 송/수신이 됐을 때, INTERRUPT로 호출될 Handler를 설정하고 instance에 대한 포인터를 콜백으로 지정해 이용



### Timer

100MHz의 clock에서 약 1초 간격으로 interrupt를 발생시켜 1초마다 downcount를 하도록 설정  
타이머 카운터 인스턴스를 0xFFFFFFFF로 설정해 32bit 까지 Upcount되고 overflow 발생  
→ 인터럽트 발생

100MHz의 clock

→ 약 1초 간격 interrupt 발생 code : `#define RESET_VALUE (0xffffffff - 100_000_000)`  
→ `XTmrCtr_SetResetValue(&TimerCounterInst, Timer_CNTR_0, RESET_VALUE)`





# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 26 / 33 ] ➔ *Project Result*

프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 4,

## 사진 & 영상

# SoC Semiconductor Peripheral RTC Design Project

.xpr

➤ 프로젝트 수행 결과물 (예시) [ 27 / 33 ] ➔ *Project Result*

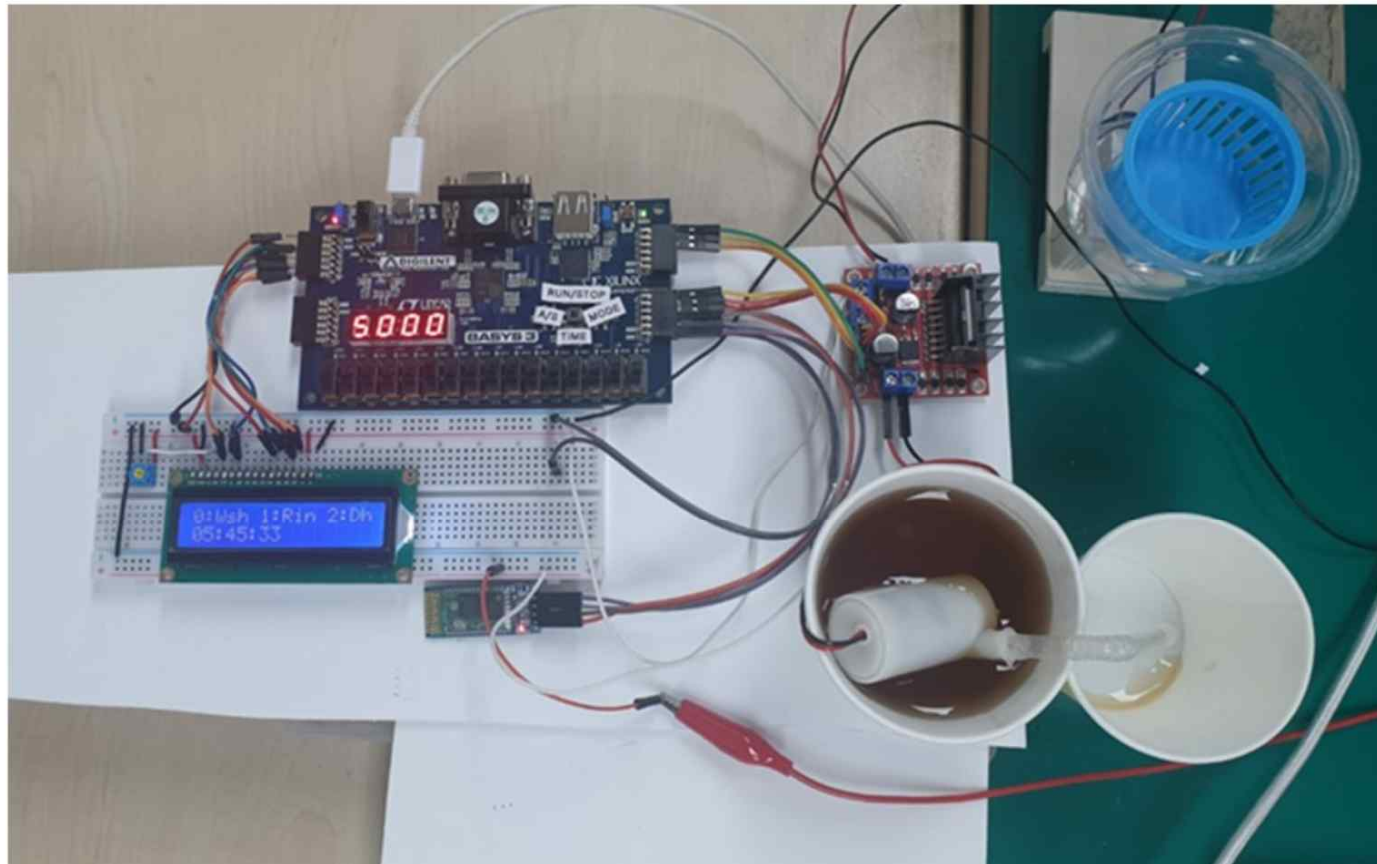
프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 4,

## 프로젝트 사진



# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 28 / 33 ] ➔ Project Result

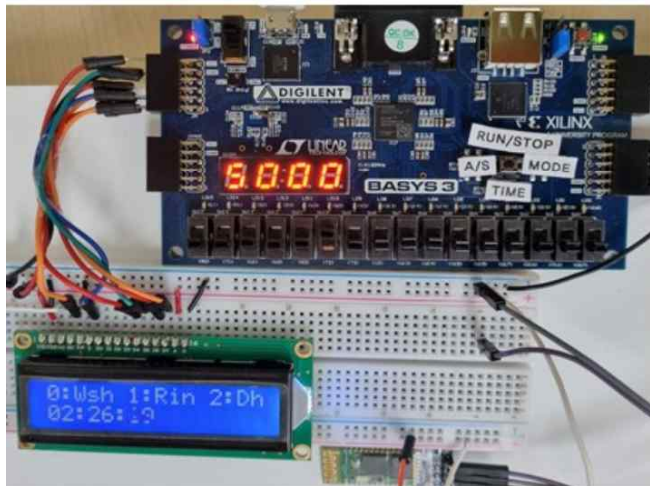
### 프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

Part 4

## 프로젝트 사진



Mode

Real TIME





# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 29 / 33 ] ➔ Project Result

### 프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

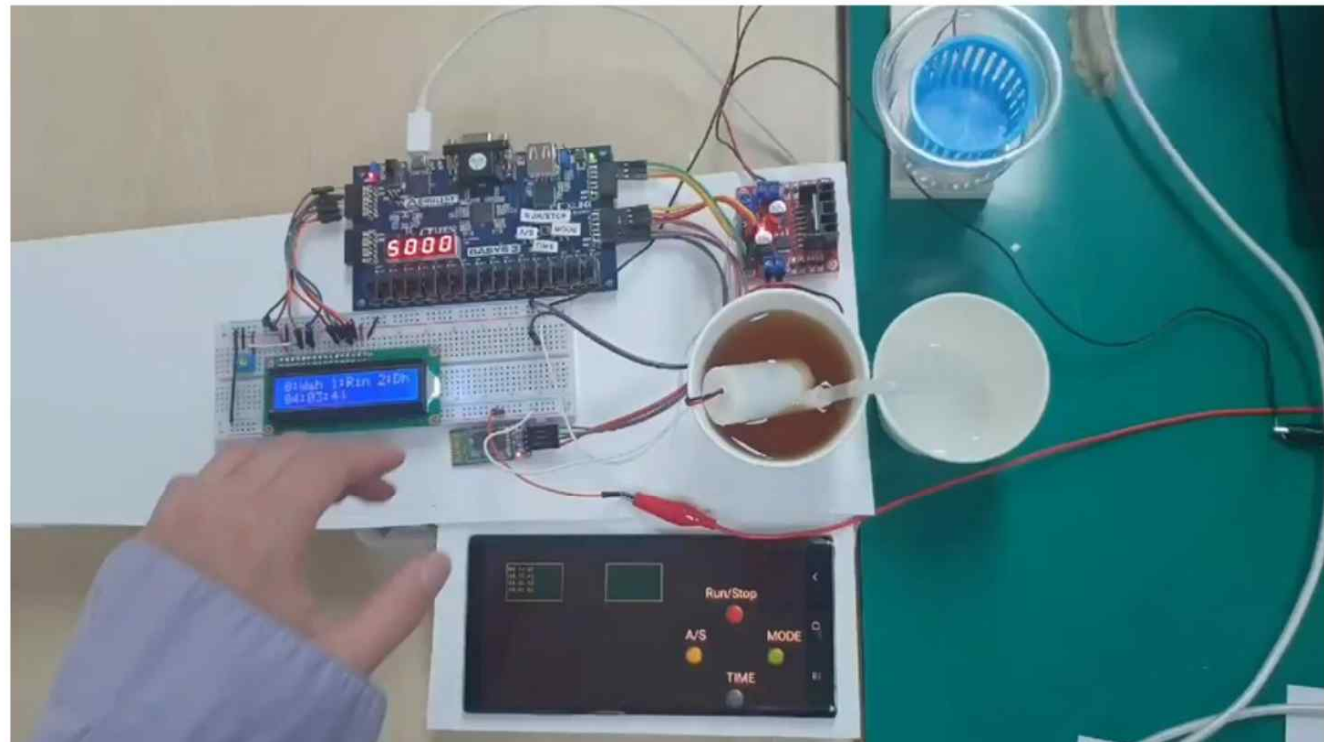
Part 4,

## 프로젝트 영상

Automode  
& 버튼을 이용하여 제어

총 30초 동안 세탁기 자동 동작

세탁 : 10초  
탈수 : 10초  
헹굼 : 10초  
총 : 30초



# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 30 / 33 ] ➔ Project Result

### 프로젝트 포트폴리오

프로젝트 결과물 예시

프로젝트 내용

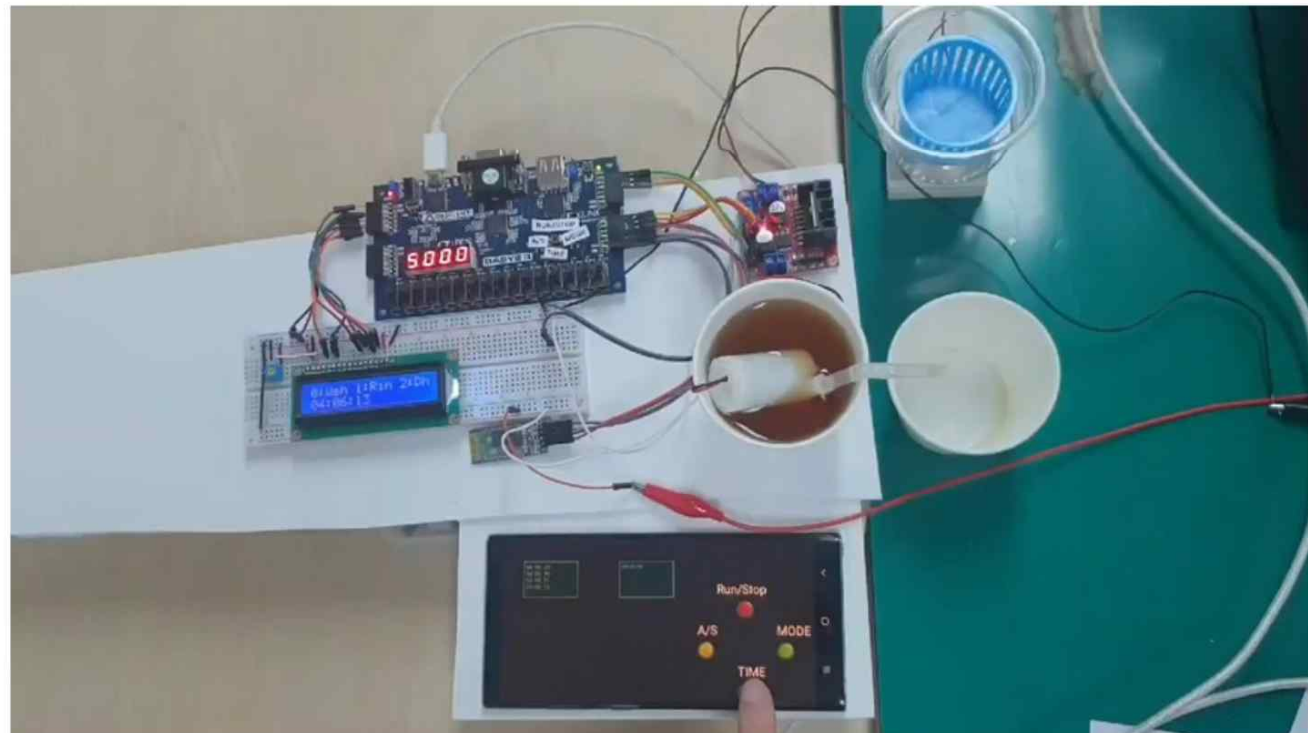
Part 4,

## 프로젝트 영상

Select mode  
& Bluetooth 제어

모드 별 10초씩 설정해  
총 30초 동안 세탁기 자동 동작

세탁 : 10초  
탈수 : 10초  
헹굼 : 10초  
총 : 30초



# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 31 / 33 ] ➔ Project Result

### 프로젝트 포트폴리오

프로젝트 결과물 예시

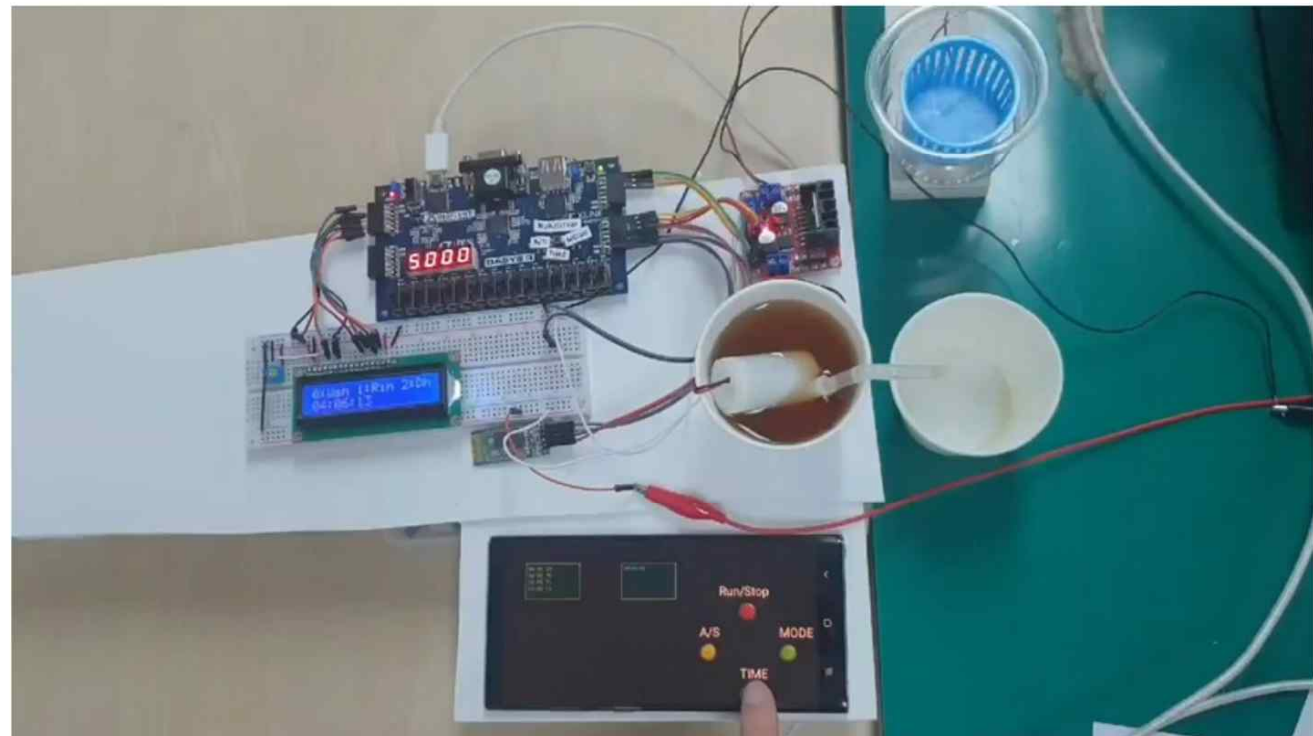
프로젝트 내용

Part 4,

## 프로젝트 영상

Wash mode  
& Bluetooth 제어

세탁 기능만 10초 동작  
세탁 : 10초



# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 32 / 33 ] ➔ Project Result

### 프로젝트 포트폴리오

프로젝트 결과물 예시

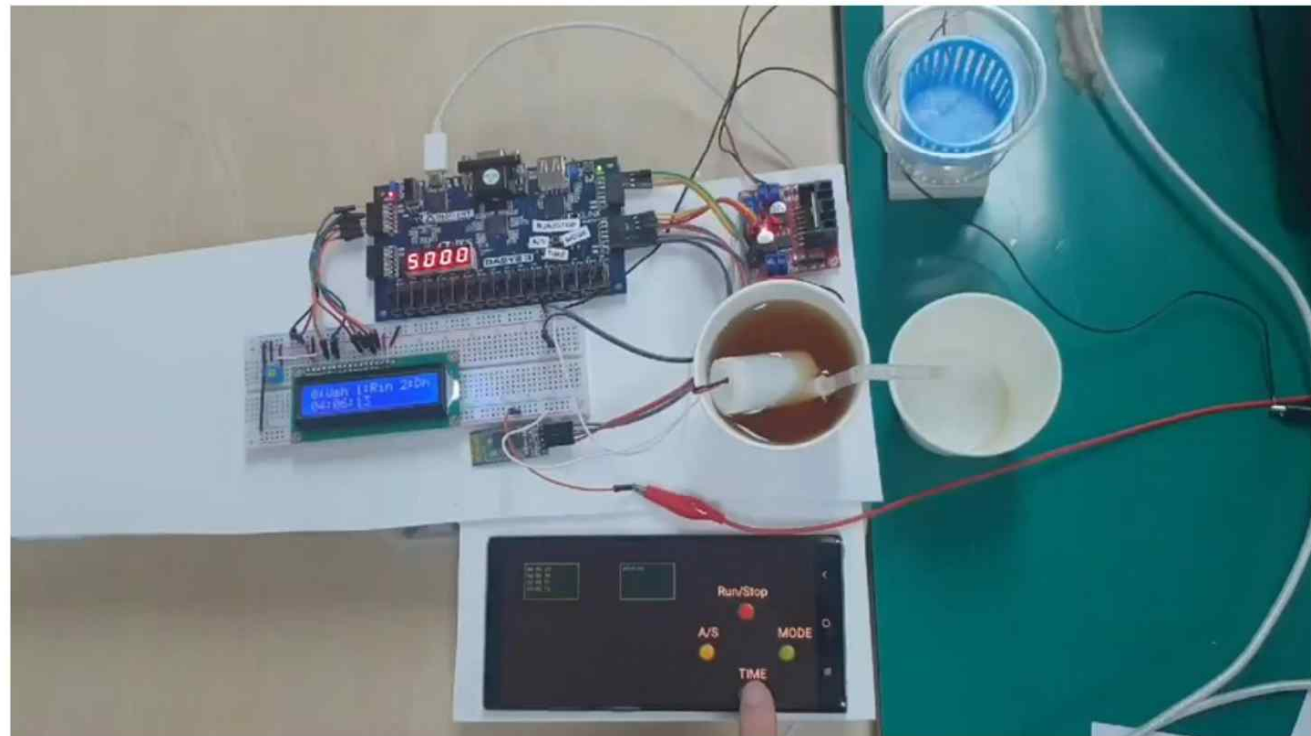
프로젝트 내용

Part 4,

## 프로젝트 영상

Rinse mode  
& Bluetooth 제어

행굼 기능만 10초 동작  
세탁 : 10초



# SoC Semiconductor Peripheral RTC Design Project

.xpr

## ▶ 프로젝트 수행 결과물 (예시) [ 33 / 33 ] ➔ Project Result

### 프로젝트 포트폴리오

프로젝트 결과물 예시

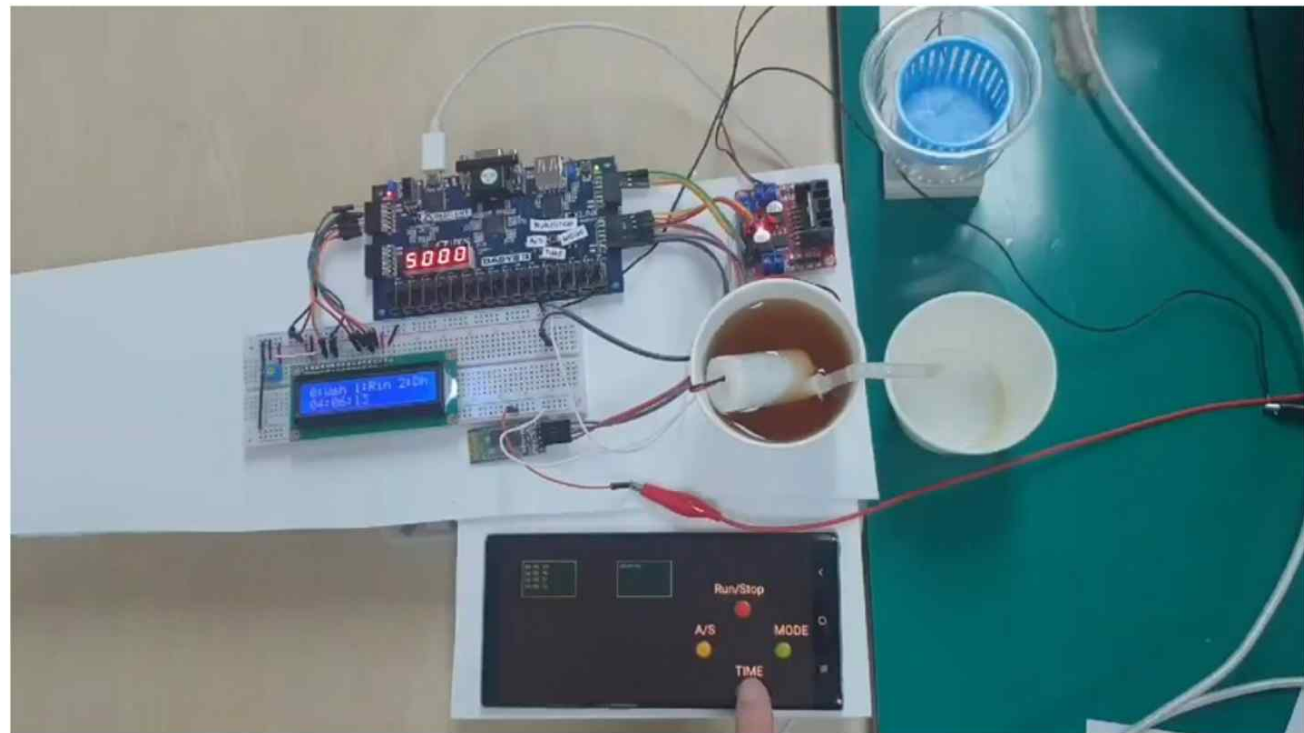
프로젝트 내용

Part 4,

## 프로젝트 영상

Dehydration mode  
& Bluetooth 제어

탈수 기능만 10초 동작  
세탁 : 10초



## *SoC Semiconductor Peripheral RTC Design Project*

- 프로젝트 배경
- 프로젝트 주제 및 내용
- 프로젝트 세부 내용
- 프로젝트 수행 결과물(예시)
- 결과 및 고찰



## ➤ 결과 및 고찰 [ 1 / ]

Part 5,

## 고찰 & 느낀점

## ➤ 결과 및 고찰 [ 2 / ]

Part 5

## 문제 해결

- Timer IP를 사용하여야 하는데 사용방법을 몰라서 코딩을 할수 없었음  
→ Xilinx에서 배포하는 Timer IP의 Datasheet에서의 시간 설정등을 참고하여 문제 해결  
<https://docs.xilinx.com/v/u/en-US/pg079-axi-timer>  
AXI Timer 2.0
  - When the counter is set to count down,  
$$TIMING\_INTERVAL = (TLRx + 2) * AXI\_CLOCK\_PERIOD$$
  - When the counter is set to count up,  
$$TIMING\_INTERVAL = (MAX\_COUNT - TLRx + 2) * AXI\_CLOCK\_PERIOD$$
  
where MAX\_COUNT is the maximum count value of the counter, such as 0xFFFFFFFF for a 32-bit counter.
- Xilinx에서 제공하는 Library의 사용의 생소함으로 인한 설정의 어려움으로 Timer Interrupt가 작동되는 도중에 Bluetooth이 작동되지 않는 문제  
→ Interrupt를 각자 실행시켜 문제 해결
- 세탁기 PWM 제어 시 각 모드의 Down Count 변수의 조건에서 변수가 0일 경우에 겹치는 부분으로 인한 문제  
행굼 단독 실행과 행굼을 뺀 실행에서 원하지 않는 동작이 출력됨  
→ 이유 : 시간에 대한 범위가 겹치는 경우가 발생  
→ flag로 제어 시도 → 문제 해결 X  
→ 겹치는 부분의 구체적 상황 설정을 위해 FSM을 다시 구성하고 and 연산 추가하여 문제 해결  

```
else if ((total_time < (set_time - (wsh*10)))&&(total_time>(set_time-(wsh*10)-4)))  
else if (((!(set_time-(wsh*10)) == (set_time-(wsh*10)-(rin*10))))&&(total_time < (set_time - (wsh*10)))&&(total_time>(set_time-(wsh*10)-4)))
```

추가
- Hardware적인 문제 :
  - Bluetooth, LCD 등의 모든 회로의 Ground의 연결을 고려하지 못함  
→ 모든 회로의 Ground를 일치시켜 줌



# SoC Semiconductor Peripheral RTC Design Project

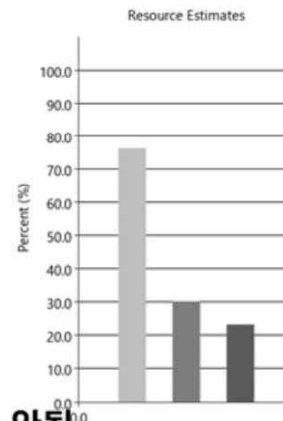
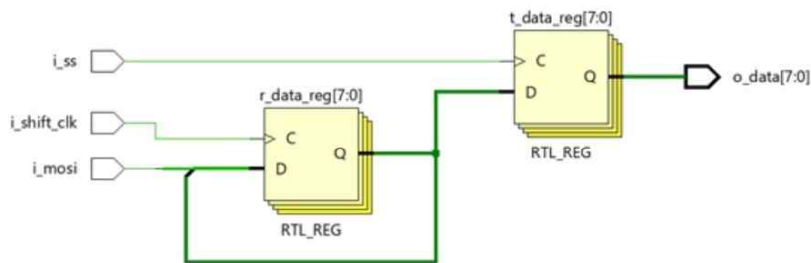
.xpr

## ➤ 결과 및 고찰 [ 3 / ]

Part 5

## 문제 해결 - SPI Module

Lorem Ipsum is simply dummy text of the printing and typesetting industry



## ➤ 결과 및 고찰 [ 4 / ]

Part 5,

### 느낀점

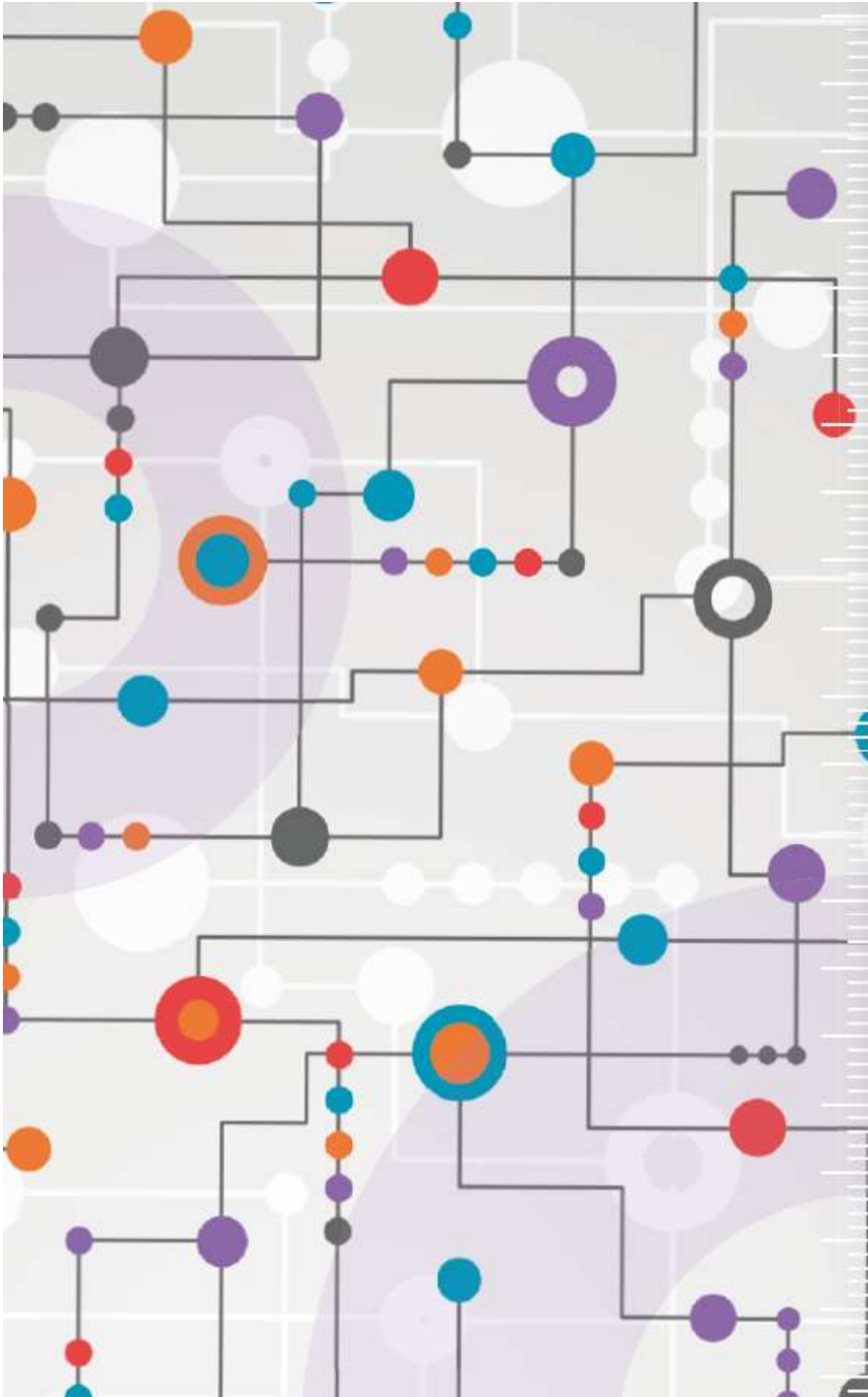
- Microblaze를 사용할 때, IP 사용에 문제가 발생하면 Bitstream과 Vitis를 실행시키는 과정에 많은 시간이 소요됨
  - 사전에 설계 단계에서 실수를 줄이기 위한 계획과 시뮬레이션 검증 과정의 중요성을 느낌
- 빠른 프로젝트의 수행을 위해서 역할 분담과 구체적인 계획의 필요성을 느낌
- 설계 전에 FSM과 Hardware Architecture등의 사전 작업 단계의 중요성을 느낌

## ➤ 결과 및 고찰 [ 5 / ]

Part 5,

### 느낀점

- Microblaze를 사용할 때, IP 사용에 문제가 발생하면 Bitstream과 Vitis를 실행시키는 과정에 많은 시간이 소요됨
  - 사전에 설계 단계에서 실수를 줄이기 위한 계획과 시뮬레이션 검증 과정의 중요성을 느낌
- 빠른 프로젝트의 수행을 위해서 역할 분담과 구체적인 계획의 필요성을 느낌
- 설계 전에 FSM과 Hardware Architecture등의 사전 작업 단계의 중요성을 느낌



수고하셨습니다.