

# Skill-Portfolio

대표 기술

FPGA 설계

## 개인 정보

성명 : 채민석  
생년월일 : 2000.09.18.  
연락처 : 010-4708-6527  
E-mail : cms2789@gmail.com  
웹페이지 : [www.minseok.co.kr](http://www.minseok.co.kr)

## 최종 학력사항

재학기간	학교명 및 전공
2019.03. ~ 2025. 08.	상명대학교 시스템반도체공학과
2016.03. ~ 2019. 02.	단국대학교사범대학부속고등학교

## 교육 이수사항

교육명	내용	교육기관
[Haman] 세미콘 아카데미 설계/검증 과정	VerilogHDL/UVM, ARM Cortex-M, AMBA BUS	서울기술교육센터
Soc 설계 역량강화 교육 프로그램	Linux / VCS / RTL Synthesis / Design Compiler	EDA Elitech

## 자격 사항

자격증명	취득일	시행기관
TOEIC SPEAKING IH	2025.03.11.	YBM

## 전문설계 소프트웨어 능력

언어(툴)	활용능력
Verilog HDL	RTL 설계
SystemVerilog	RTL 설계 및 UVM 검증
Python	openCV를 이용한 영상처리
C	Firm Ware 소프트웨어 개발

## 프로젝트 수행사항

번호	프로젝트명	프로젝트 내용	수행기관
1	FPGA UART Multi-Sensor	CLOCK / UART / SR04 / DHT11 모듈 구현	대한상공회의소
2	RISC-V 기반 Single Core CPU 설계	RV32I기반 CPU 설계	대한상공회의소
3	AMBA APB Peripheral 설계 및 검증	APB Peripheral 구조 이해 및 C FirmWare 구현	대한상공회의소
4	AXI 기반 SPI / I2C 통신 모듈 설계	SPI 및 I2C 통신 기법 설계	대한상공회의소
5	VGA기반 뱀사다리 게임	이미지 처리를 통한 보드게임 설계	대한상공회의소
6	UVM기반 Linebuffer 검증	UVM기반 LineBuffer 모듈 검증	대한상공회의소
7	Soc기반 펜플로터 각인 서비스	FPGA, STM32를 이용한 펜플로터 구현	대한상공회의소

# 프로젝트 기술서

작 성 자

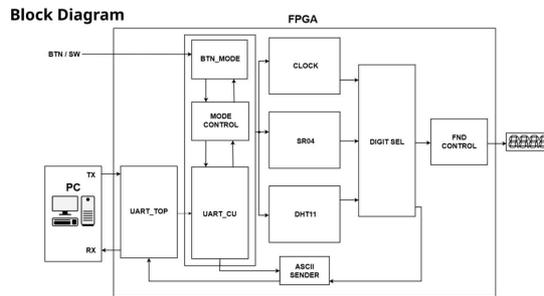
채민석

<b>1</b>	<b>프로젝트명 : FPGA UART Multi-Sensor (4인 프로젝트)</b>
수행기간	2025.08.20. ~ 2025.08.28.
담당역할	SR04 설계, StopWatch, Watch 설계, UART 설계
수행목표	Verilog HDL 기반의 계층적 구조 디지털 시스템 설계 UART 프로토콜을 이용한 FPGA-PC 간 데이터 송수신 로직 구축 FSM기반의 정밀 초음파 센서 제어 및 거리 측정 알고리즘 개발 외부 인터럽트와 UART 통신을 통합한 멀티 제어 시스템 확보
사용 기술 / 환경	Digilent Basys3, Verilog HDL, Xilinx Vivado, SR04, DHT11, UART

## 세부수행내용

### [프로젝트 요약]

FPGA(Basys3 보드)를 활용하여 시계, 스톱워치, 타이머의 기본 기능과 더불어 온/습도(DHT11), 초음파(SR04) 센서 데이터를 통합하여 측정하고, UART 통신을 통해 PC로 실시간 모니터링 및 제어가 가능한 IoT 시스템 설계 프로젝트



### [주요 모듈 설명]

1. UART\_TOP(RX, TX, FIFO) : 비동기 직렬 데이터 수신(RX) 및 송신(TX)을 처리하며, 데이터 손실을 방지하기 위해 FIFO (First-In First-Out) 버퍼 사용
2. UART\_Control : FIFO로부터 들어온 데이터를 분석하는 핵심 로직 유닛 'r', 's', 'm'와 같은 PC의 단일 문자 명령어를 다른 모듈에서 사용할 수 있는 1-tick 제어 신호로 변환
3. MODE CONTROL : 시계 또는 스톱워치 모드를 관리하는 FSM(Finite State Machine) PC 명령으로 장치의 상태를 제어하는 가상 스위치 역할
4. DIGIT SEL (CLOCK & SR04 & DHT11) : 각각 시계, 초음파센서, 온습도 센서의 핵심 기능을 담당하는 모듈입니다. 시간 계산, 카운팅, 수정, 센서 측정 등의 동작 처리
5. FND\_CNTRL: 현재 시간이나 스톱워치 값을 7세그먼트 디스플레이에 표시하는 제어 모듈

### [트러블 슈팅]

1. SR04 부분에서 Negative Slack이 발생하는 문제가 생김 -> 거리계산시 단순 나눗셈으로 계산하여 발생한 문제로 인식 -> 비트 연산자로 문제 해결

### [고찰]

외부 디바이스 제어 시 데이터시트를 분석하는 것에 대한 중요성을 알 수 있었고 하드웨어적 설계를 할 때는 FSM과 같은 상태 기반로직으로 체계적으로 구현하는 것이 필수적이라는 것을 알 수 있었음

# 프로젝트 기술서

작 성 자

채민석

2

프로젝트명 : RISC-V 기반 Single Core CPU 설계 (1인 프로젝트)

수행기간

2025.09.17. ~ 2025.10.02.

프로젝트 개요

RISC-V RV32I ISA 표준 기반의 Single Cycle CPU Core 설계 및 명령어 타입별 동작 검증 수행

수행목표

1. 오픈소스 ISA인 RISC-V의 RV32I 기반으로 Single Cycle CPU Core 설계
2. 시뮬레이션을 통해 각 명령어 타입별 기능이 정상적으로 동작하는지 검증

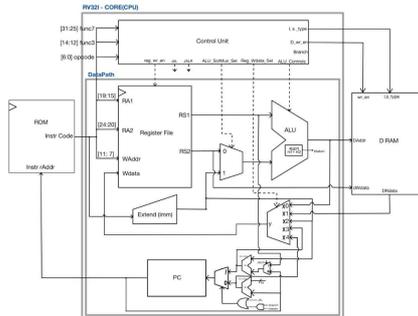
사용 기술 / 환경

Digilent Basys3, Verilog HDL / SystemVerilog, Xilinx Vivado, RISC-V(RV32I)

## 세부수행내용

### [프로젝트 요약]

오픈소스 ISA(명령어 집합 구조)인 RISC-V의 RV32I를 기반으로 Single Cycle CPU Core를 설계하고, 시뮬레이션을 통해 각 명령어 타입별 (R,I,S,B,U,J) 기능이 정상적으로 동작함을 직접 작성한 rom 코드로 기능 검증



### [주요 기능 설명]

1. Control Unit: 명령어의 Opcode, funct3, funct7 필드를 해석하여 Datapath의 각 요소(Mux, ALU, Register File 등)에 필요한 모든 제어 신호를 생성
2. Datapath: Program Counter, Instruction Memory, Register File, ALU, Data Memory 및 Mux들로 구성되어 실제 데이터의 연산과 흐름을 담당
3. Instruction/Data Memory: 각각 명령어와 데이터를 저장하는 메모리로, 본 설계에서는 ROM과 RAM으로 모델링
4. 명령어 타입별 테스트
  - R Type: add, sub, slt, sltu, sra, srl 등 레지스터 간의 모든 산술/논리/시프트 연산 동작 검증
  - I Type: addi, slti, ori 등 상수 연산과 lw, lb, lhu 등 다양한 Load 명령어의 부호/제로 확장 기능 검증
  - S Type: sw, sh, sb 명령어를 통해 메모리의 특정 주소에 워드, 하프워드, 바이트 단위로 정확히 저장 확인
  - B Type: beq, bne, blt 등 조건 분기 명령어가 ALU의 비교 결과에 따라 PC를 제대로 변경하는지 확인
  - U/J Type: lui, auipc를 이용한 32비트 상수 생성 및 주소 계산, jal/jalr을 이용한 함수 호출 및 복귀 과정 검증

### [고찰]

이번 RISC-V CPU 설계를 통해 컴퓨터 구조의 이론적 지식을 실제 하드웨어 로직으로 구현하며 깊이 있는 학습을 할 수 있었음

# 프로젝트 기술서

작 성 자

채민석

<b>3</b>	프로젝트명 : AMBA APB Peripheral 설계 및 검증 (1인 프로젝트)
수행기간	2025.10.17. ~ 2025.10.27.
프로젝트 개요	RISC-V Multi-Cycle CPU 환경 내 AMBA APB 인터페이스 기반 커스텀 주변장치 통합 시스템 구현
수행목표	1. RISC-V RV32I ISA를 기반으로 Multi-Cycle CPU Core를 구현 2. AMBA APB BUS에 UART Peripheral을 연동하여 PC와 FPGA 보드 간의 양방향 통신 구현
사용 기술 / 환경	Digilent Basys3, SystemVerilog, Xilinx Vivado, RISC-V(RV32I), AMBA APB
<b>세부수행내용</b>	
<b>[프로젝트 요약]</b> 기존 Single-Cycle 구조를 5단계(Fetch, Decode, Execute, Memory, Writeback) 로 분리, 각 단계 사이에 플립플롭(파이프라인 레지스터) 을 추가하여 Multi-Cycle 구조로 변경. UART를 APB Slave로 연결하여 시스템 확장성 확보. 이후 C언어로 작성된 테스트 코드를 CPU에서 실행하여 PC와 UART 통신이 정상적으로 동작되는지 FPGA 보드에서 검증	
<b>[주요 기능 설명]</b> 1. Multi Cycle CPU - R-Type: add, sub, sll, srl, sra, slt, sltu, xor, or, and - I-Type: addi, slti, sltiu, xori, ori, andi, slli, srli, srli - Load (l) Type: lw, lb, lh, lbu, lhu (Sign/Zero Extension 검증) - S-Type: sw, sh, sb (메모리 쓰기 검증) - B-Type: beq, bne, blt, bge, bltu, bgeu (분기 조건 및 PC 변경 검증) - U/J-Type: lui, auipc, jal, jalr (점프 및 Link Register 저장 검증) 2. UART Peripheral 검증 SystemVerilog Class 기반 UART 검증 환경 구축으로 송수신 신뢰성을 확보하고, Generator 기반 랜덤 데이터 생성 및 Driver/Monitor를 통한 APB/UART 프로토콜 유효성 체크 수행. Scoreboard를 활용한 데이터 일치 여부 및 무결성 검증 프로세스 자동화로 전체적인 검증 효율 향상. 3. C기반 FPGA 검증 C언어로 UART 송수신 및 LED 제어 프로그램 작성하여 동작 검증 - 0 ~ 7 입력시 해당 숫자의 LED 토글, 'A' 입력시 초기화 - LED 입력시 PC화면에 UART FeedBack 구현	
<b>[트러블 슈팅]</b> Verification시 TX Data Timing 불일치 -> wait 조건을 추가하여 event가 발생시 샘플링하도록 변경	
<b>[고찰]</b> APB BUS 기반으로 시스템을 구축하고 UART Peripheral을 연동을 공부하면서 표준화된 bus 인터페이스가 시스템 확장성과 IP 재사용성에서 뛰어난 이유를 체감 할 수 있었음.	

# 프로젝트 기술서

작 성 자

채민석

<b>4</b>	프로젝트명 : AXI 기반 SPI / I2C 통신 모듈 설계 (1인 프로젝트)
수행기간	2025.11.07. ~ 2025.11.18.
프로젝트 개요	AXI4-Lite 인터페이스 기반 SPI 및 I2C 직렬 통신 프로토콜 마스터/슬레이브 모듈 설계 및 동작 구현
수행목표	1. SPI/I2C 통신 Protocol을 이해하고 AXI4-Lite Interface와 연동 가능한 Module 구현 2. 설계한 통신 Module에 대해 Synopsys Tools(VCS, Verdi)를 이용하여 UVM 환경 구성 및 검증
사용 기술 / 환경	Digilent Basys3, SystemVerilog, Xilinx Vivado, VCS, Verdi, AXI-Lite, SPI, I2C
<b>세부수행내용</b>	
<b>[프로젝트 요약]</b> SPI와 I2C 통신 프로토콜의 메커니즘을 분석하고, AMBA AXI4-Lite 인터페이스와 연동할 수 있는 통신 IP를 SystemVerilog로 설계함. 설계한 모듈은 Synopsys의 VCS 및 Verdi를 이용해 UVM 기반 검증 환경에서 기능적 무결성을 확인했으며, 최종적으로 Basys 3 FPGA 보드에 올려 실제 하드웨어 동작을 검증함.	
<b>[주요 기능 설명]</b> 1. SPI (Serial Peripheral Interface) 모듈 - SCLK, MOSI, MISO, SS의 4개 선을 사용하는 Full-Duplex 동기식 직렬 통신 구현 - CPOL(Clock Polarity) 및 CPHA(Clock Phase) 설정을 통해 4가지 SPI 전송 모드 완벽 지원 - UVM 환경 내에서 Scoreboard를 구축하여 tx_data와 rx_data의 실시간 일치 여부 검증 - Master와 Slave 간의 상태 머신(ASM/FSM) 설계를 통해 안정적인 데이터 트랜잭션 수행 2. I2C (Inter-Integrated Circuit) 모듈 - SDA, SCL 2개 선과 Open-Drain 구조를 활용한 Half-Duplex 동기식 통신 구현 - 7비트 주소 체계를 통해 다중 Slave 연결이 가능하며, START/STOP 조건 및 ACK/NACK 체크 프로토콜 설계 - MicroBlaze 프로세서와 AXI4-Lite 인터페이스를 연동하여 C언어 어플리케이션으로 I2C 제어 레지스터를 핸들링하고 FND에 데이터를 출력하는 시스템 구축 3. 로직 분석기(Logic Analyzer)를 통해 실제 통신 파형을 캡처하여 프로토콜 규격 준수 여부 확인	
<b>[트러블 슈팅]</b> Master에서 주소 전송 이후 Slave로부터 ACK를 받지 못하고 NACK 상태가 지속되는 현상 발생. ->분석을 통해 ACK 구간에서 SDA가 여전히 High를 유지하고 있음을 확인. Master가 ACK 수신 시점에 SDA 드라이브를 High로 유지하고 Slave가 SCL이 High인 구간에서만 SDA를 Low로 당기도록 수정	
<b>[고찰]</b> 하드웨어 설계 전 프로토콜의 세부 규격과 UVM 검증 구조를 명확히 정의하는 것이 설계 오류를 줄이는 핵심임을 체감함. 현재 I2C Read 동작 시 특정 상황에서 발생하는 버그를 수정하기 위해 시퀀스를 규격 기준으로 재정리하고 있으며, 이를 통해 검증 신뢰도를 한층 더 높일 계획임.	

# 프로젝트 기술서

작 성 자

채민석

<b>5</b>	<b>프로젝트명 : VGA기반 뱀사다리 게임 (4인 프로젝트)</b>
수행기간	2025.11.24. ~ 2025.12.05.
담당역할	- 보드게임 Core Logic 구현 - Game Effect 효과 구현 - Board 및 말 모듈 구성 - 모듈 결합 구현
수행목표	1. OV7670, VGA, 게임 로직을 단일 FPGA에 통합하여 개구리 말을 사용하는 뱀사다리 게임 시스템 구현 2. 디지털 회로, 영상 처리, 임베디드 시스템 통합 하드웨어 설계 역량 향상
사용 기술 / 환경	Digilent Basys3, SystemVerilog, Xilinx Vivado, OV7670
<b>세부수행내용</b>	
<b>[프로젝트 요약]</b> OV7670 카메라 영상 처리와 게임 로직을 결합한 뱀사다리 게임 시스템 구현. 카메라 캡처 기능을 통한 플레이어 등록과 VGA 기반의 실시간 게임 인터페이스 출력을 통합한 시스템 구축	
<b>[주요 기능 설명]</b> 1. Game Control & Snake Core (담당) - 버튼 입력 기반 턴 제어 및 LFSR을 활용한 주사위 난수 생성, 뱀·사다리·잡기 등 게임의 핵심 이동 규칙 로직 설계 - 각 플레이어의 위치 정보(49칸)를 실시간으로 연산하고 상태 머신(FSM)을 통해 안정적인 게임 프로세스 구축 2. Board Image & Effect (담당) - 49칸 보드 배경 이미지와 도트(DOT) 기반 플레이어 말·표정 모듈 설계 및 리소스 배치 - 뱀 도달 시 보드 주소 오프셋 조절을 통한 Shake 효과 및 잡기 이벤트 발생 시 카메라 영상에 Red 필터를 씌우는 시각적 피드백을 구현 3. 모듈 통합 및 시스템 인터페이스 (담당) - 카메라 영상, 보드 오브젝트, 게임 UI를 최종 합성하여 VGA 출력, 개별 모듈 간 타이밍 조율 및 인터페이스 통합 수행 4. 영상 캡처 및 사운드 제어 - OV7670 카메라 설정 및 Memory에 이미지 데이터를 저장하여 게임 내 플레이어 얼굴 및 실시간 영상 소스 확보	
<b>[트러블 슈팅]</b> 게임 코어 내에 효과 로직과 상태 변화를 동시에 처리하면서 발생하는 타이밍 충돌로 출력이 불안정해지는 문제 발생. -> 전용 모듈을 물리적으로 분리하고 모듈 간 입출력 인터페이스와 타이밍을 재정의하여 동기화 오류를 해결하고 동작 안정성을 확보	
<b>[고찰]</b> 단순한 기능 구현을 넘어 모듈 간의 Sync 조율과 기능 분할 설계가 시스템 전체 안정성에 미치는 중요성을 깊이 체감함. 특히 서로 다른 로직(카메라, 게임 코어)을 하나의 TOP 모듈로 통합하는 과정에서 하드웨어적 인터페이스 설계 역량을 기를 수 있었음.	

# 프로젝트 기술서

작 성 자

채민석

6	프로젝트명 : UVM기반 Linebuffer 검증 (3인 프로젝트)
수행기간	2025.12.15. ~ 2025.12.22.
담당역할	Agent, Sequence, ScoreBoard 설계
수행목표	1. 설계자의 검증 요구사항에 맞게 UVM 검증 시나리오 작성 2. 기능 커버리지 100% 달성을 통한 설계 신뢰성 확인
사용 기술 / 환경	SystemVerilog, EDA Playground, UVM
세부수행내용	
<b>[프로젝트 요약]</b> 기업에서 제시한 산업용 IP 사양에 따라 영상 처리용 Line Buffer의 동작을 검증하기 위해 UVM 환경 구축. 설계 사양 대비 로직의 정확성을 확인하고 가변 해상도 환경에서의 데이터 무결성 확인.	
<b>[주요 기능 설명]</b> 1. Sequence Item 및 Sequence 설계 (일부 담당) - Sequence Item: 기업 요구 규격인 8-bit 픽셀 데이터와 유효성 제어 신호를 포함하는 트랜잭션 패킷 정의. 랜덤 제약 조건을 활용하여 다양한 입력 데이터 패턴을 생성할 수 있도록 구성. - Sequence: 라인 버퍼의 3x3 윈도우 생성 로직 검증하기 위해 시나리오에 따른 데이터 입력 흐름 제어. 임의 해상도 및 데이터 스트림 환경을 모사하여 DUT에 트래픽 인가. 2. Scoreboard 기반 자동 비교 로직 구현 (담당) - Reference Model: 윈도우 슬라이딩 메커니즘을 참조 모델로 구현하여 입력 데이터에 따른 이론적 기대 값 산출. - 데이터 비교: Monitor에서 수집된 실제 DUT 출력과 Reference Model의 결과값 대조. 에러 리포트: 데이터 불일치 발생 시 즉각적인 에러 로그를 발생시켜 분석 효율을 높임. 3. 기업 사양 기반 검증 환경 통합 Driver, Monitor, Agent 등의 컴포넌트를 통합하여 설계자의 요구 규격에 부합하는 통합 테스트벤치 완성	
<b>[트러블 슈팅]</b> 사양서에 명시된 초기화 레이턴시와 실제 설계 동작 간의 미세한 시간 차로 인해 데이터 샘플링 불일치 발생. -> 유효 신호(Valid)의 활성화 시점을 기준으로 Scoreboard의 비교 동작을 동기화하여 검증 정확성 확인.	
<b>[고찰]</b> 실제 산업용 IP 사양을 분석하여 이를 UVM 검증 컴포넌트 단위로 구체화하는 실무 역량을 기름. 엄격한 사양 준수의 중요성을 체감하고 가변적인 입력 환경에서도 신뢰할 수 있는 자동화 검증 환경을 구축할 수 있는 경험을 얻음.	

# 프로젝트 기술서

작 성 자

채민석

<b>7</b>	<b>프로젝트명 : Soc기반 펜플로터 커스텀 각인 서비스 (7인 프로젝트) (최우수상)</b>
수행기간	2026.01.05. ~ 2026.01.22.
담당역할	1. FPGA 이미지 전처리 2. 이미지 처리 성능 최적화 3. FPGA Module 최적화
수행목표	1. SoC 기반 하드웨어-소프트웨어 통합 실시간 각인 시스템 구축 및 실무 역량 확보 2. 입력 모드별 데이터 패스 설계 및 FPGA 이미지 가속기 구현을 통한 성능 극대화 3. Python 경로 최적화 및 STM32 RTOS 연동을 통한 각인 품질 확보 및 출력 시간 단축
사용 기술 / 환경	Digilent Zybo Z7-20, STM32, SystemVerilog, Python, C, OV7670
<b>세부수행내용</b>	
<b>[프로젝트 요약]</b> 카메라 영상, 이미지 파일 업로드, 직접 그린 스케치 등 3가지 입력 모드를 지원하는 각인 시스템 구축. 카메라 입력 데이터는 FPGA 보드 내에서 이미지 필터를 거쳐 Python으로 전송하며, 업로드 및 스케치 데이터는 PC에서 FPGA로 데이터를 송신하여 필터링 처리 후 다시 PC로 회신하는 데이터 흐름 설계. 이후 Python 단계에서 드로잉 경로를 최적화하고, 최종적으로 STM32가 전달받은 주소 좌표값에 따라 모터를 제어하여 실시간 각인 수행	
<b>[주요 기능 설명]</b> <b>FPGA (담당)</b> 이미지 전처리 가속: Canny Edge 알고리즘(Gaussian, Sobel, Hysteresis 등) RTL 설계 및 실시간 경계 검출 하드웨어 가속기 구현. 데이터 패스 최적화: 카메라 입력(Board -> Filter -> Python) 및 PC 입력(PC -> FPGA -> Filter -> PC) 경로 이원화 설계 및 [0xAA] 시작 트리거 프로토콜 기반 데이터 동기화 제어. 모듈 최적화: 3x3 윈도우 연산을 위한 라인 버퍼(Line Buffer) 구조 및 파이프라이닝 기법 적용을 통한 연산 병목 해결 및 처리 성능 극대화. <b>PYTHON</b> 경로 최적화: FPGA 수신 이진 데이터 기반 OpenCV 윤곽선 추출 및 효율적 각인을 위한 G-code 경로 계획 알고리즘 탑재. <b>STM32</b> 모터 정밀 제어: RTOS 기반 태스크 관리 및 브레젠햄 알고리즘을 활용한 2축 스테핑 모터의 실시간 동기 구동 시스템 완성.	
<b>[트러블 슈팅]</b> (Canny Edge 연산 단계별 지연으로 인한 실시간 출력 깨짐현상 발생 -> 각 연산 단계를 독립적인 스테이지로 분리한 하드웨어 파이프라인 설계 및 라인 버퍼 주소 제어 로직 고도화를 통해 연산 지연 없는 스트리밍 처리 환경 구축.	
<b>[고찰]</b> 여러 디바이스 간의 효율적인 데이터 인터페이스 설계 및 커스텀 프로토콜 정의 역량 확보. 하드웨어 가속기가 전체 시스템의 실시간성과 각인 품질에 기여하는 핵심적 역할을 체감할 수 있었음.	