FPGA: CNN 추론 가속기

TEAM: S.P.E.K(김재민, 박윤주, 신홍민, 엄민호)

WHY: Cloud가 아닌 Edge 처리?

지연 (Latency)

방대한 영상/센서 데이터의
서버 왕복 지연은 실시간 응답을
불가능하게 만듭니다.

네트워크 (Network)

네트워크 환경에 따라 성능이
저하되거나 서비스가 중단될 수 있어,
현장에서의 단독 동작이 필수적입니다.

"EDGE 환경에서의 실시간 추론 요구가 빠르게 증가"

엣지 AI: CPU vs FPGA

전력 비효율
(Power Inefficiency)

범용 CPU 아키텍처는 AI 추론 연산에
전력 대비 효율이 낮아
발열과 소모량이 높습니다.

실시간 불안정
(Real-time Instability)

처리 지연(latency) 및 동기(sync)
불안정 발생 시, 프레임 깨짐이나 라인
아티팩트 등 치명적인 오류로 이어집니다.

메모리 고갈
(Memory Depletion)

고해상도 영상 프레임을 버퍼링하는
방식은 온칩 메모리(BRAM)를
순식간에 고갈시킵니다.

"엣지 AI는 전력/ 지연시간/ 메모리 제약이 성능을 좌우한다."

Solution

FPGA를 이용한 하드웨어 가속기 설계를 통하여, CPU 대비 압도적인 속도와 전력 효율을 달성할 수 있다.

본 프로젝트는 Zybo Z7-20 FPGA로

1) 카메라 입력부터 CNN 추론, 화면 출력을 처리하는 독립형 시스템을 구축하고,

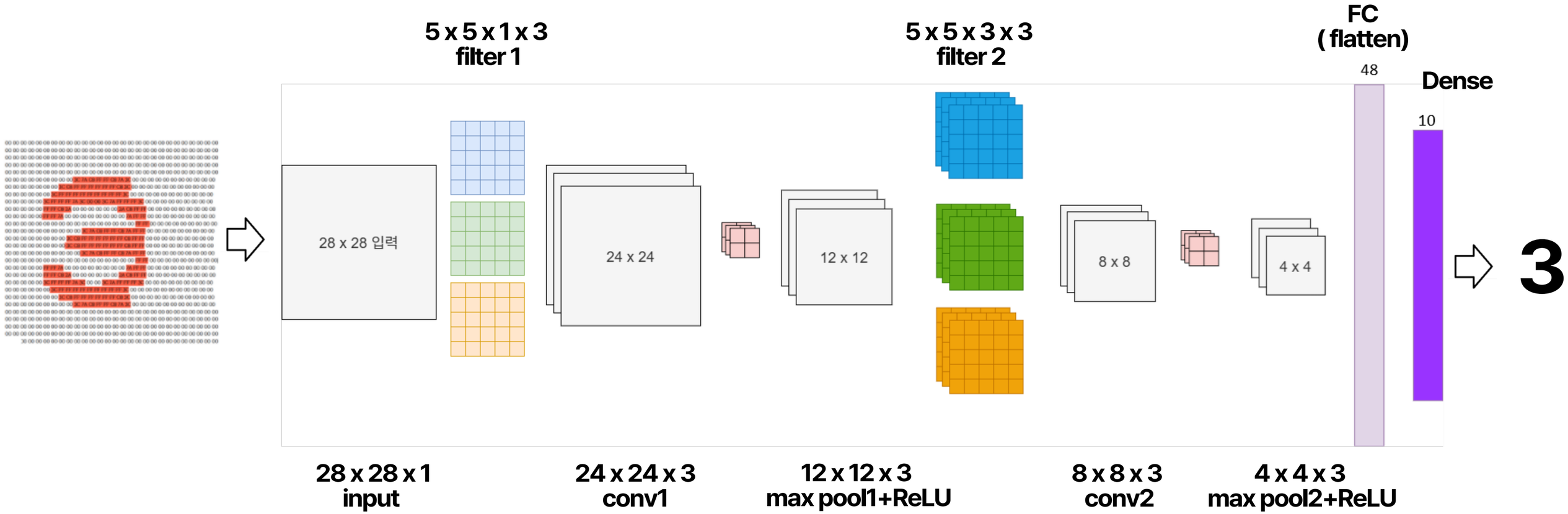
2) CPU 대비 성능을 수치적으로 증명하는 것이 목표

Target Archiecture

- 개발보드: Zybo-Z720
  - PL: CNN 추론·영상 처리 로직
  - PS: Boot only
- CNN CORE Target
  - stream line 구조
  - 125Mhz
  - Resource , Power 최소화
  - accuracy > 95%

CNN CORE 결과 비교

1)MNIST CNN 학습 모델 Architeture



Parameter :790개

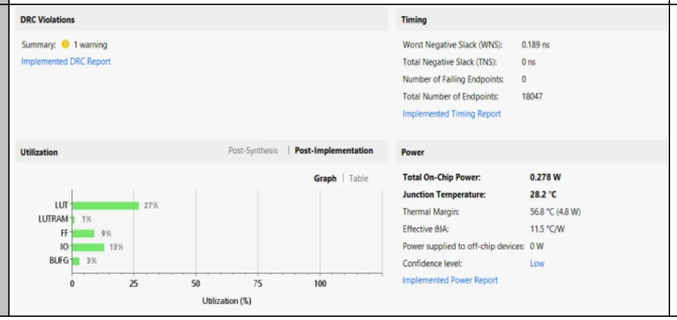
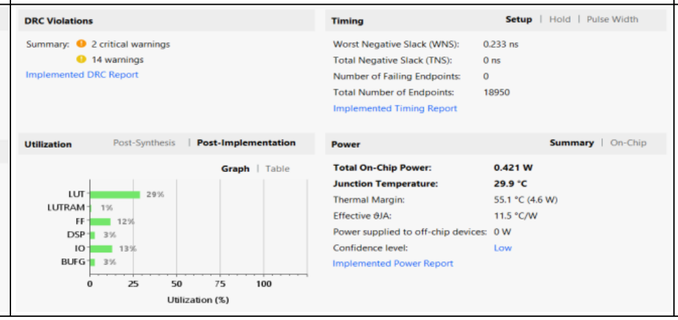
CNN CORE 결과 비교

2)CNN 구현방법 (QAT, PTQ)

비교 항목	가속기 설계 1팀 (Hardware-Aware QAT)	가속기 설계 2팀 (Standard PTQ)
장점	• 고정 Q-Format 기반 학습 Activation(Q0.8), Weight(Q1.7)을 반영해 FPGA 동작 일관성 확보 • 학습중 Fake Quantization 적용 Rounding/Clamping 반영으로 포화(Saturation) 강건성 및 HW 일치도 확보 • RTL 구현 복잡도 최소화 학습 단계에서 정수 가중치·바이어스 포맷 확정으로 HW 매핑 단순화	• 데이터 분포의 최대값(amax) 기반 스케일링으로 정밀한 표현력 확보 • Quantization 오차가 작고 정확도가 높음 • FP32 학습 모델을 다양한 하드웨어 타겟으로 변환 가능한 표준적 방식
단점	• 스케일을 2^n, Activation, Weight을 고정 Q-Format으로 제한하여 **표현 범위 감소** • Shift-only 제약 -> **학습 파라미터 튜닝이 까다로움**	• 레이어별 실수 스케일 처리로 **HW 구현 복잡도 증가** • Rescale에 곱셈 연산이 필요하여 **DSP 자원 필요** • **SW(Float)–HW(Int) 간 성능 괴리** 가능성 존재
설계 결론	Hardware Consistency & Reliability 중심 설계 (신뢰성 우선, 하드웨어 기반 설계)	Algorithm & Precision 중심 설계 (정확도 우선, 소프트웨어 기반 설계)

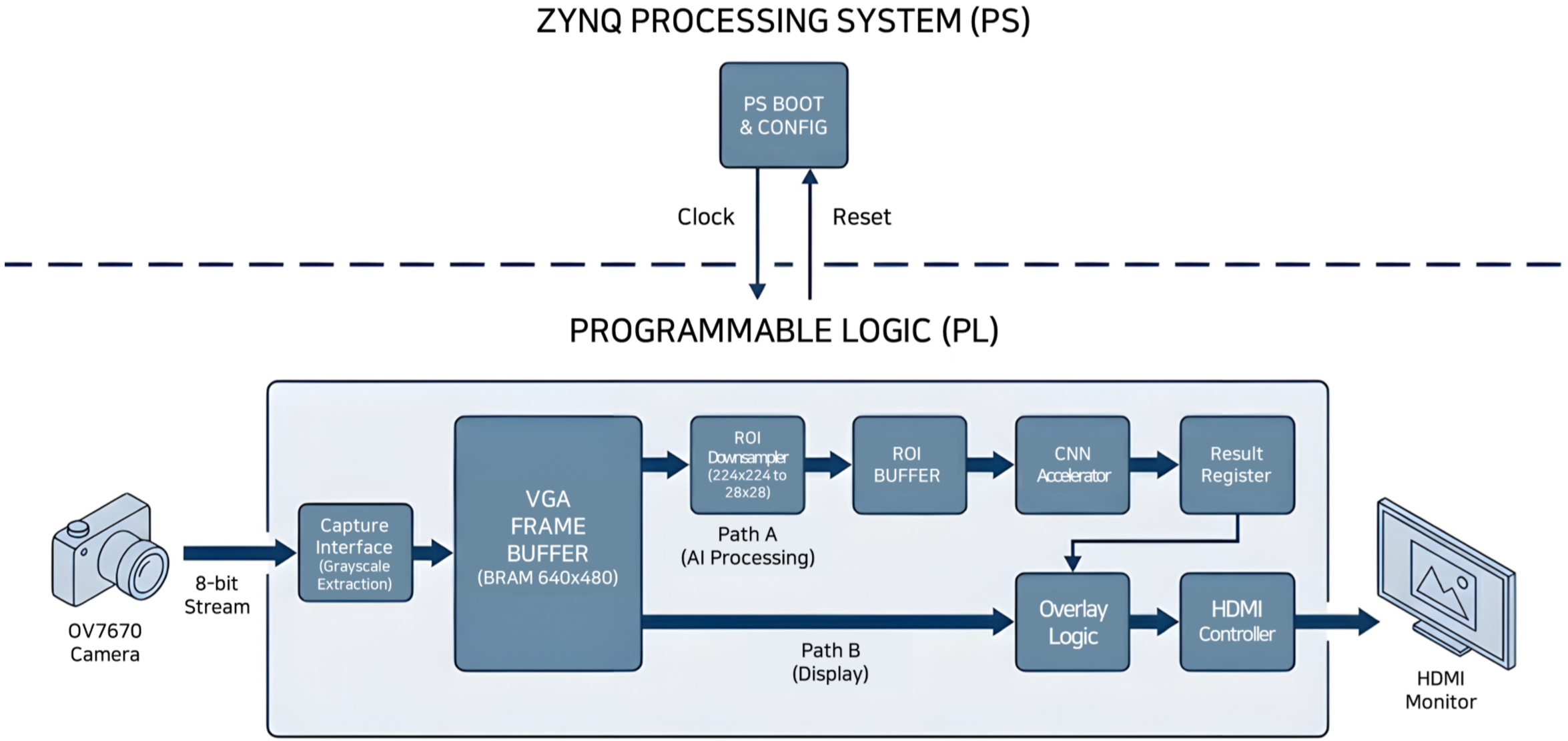
CNN CORE 결과 비교

3) 구현 결과: Summary

	QAT	PTQ	비교 요약
목표 주파수 (ZYBO Z7-20)	125 MHz 통과	125 MHz 통과	Timing 만족
소요 클럭 (1 inference)	811 clk (입력 후처리 27 clk)	816 clk (입력 후처리 32 clk)	QAT가 5 clk 빠름
LUT / FF 사용률	27% / 8.6%	29% / 11.7%	QAT 사용량 감소
DSP 사용량	0	6 (2.73%)	QAT 우세(DSP 미사용)
SW 추론 정확도	96.31%	96.93%	HW-SW 일치도 QAT 우세
FPGA 추론 정확도	96.32%	96.35%	
Total Power	0.278 W	0.421 W	QAT 약 34% ↓
VIVADO REPORT			TARGET 충족 HW 구현 검증 완료

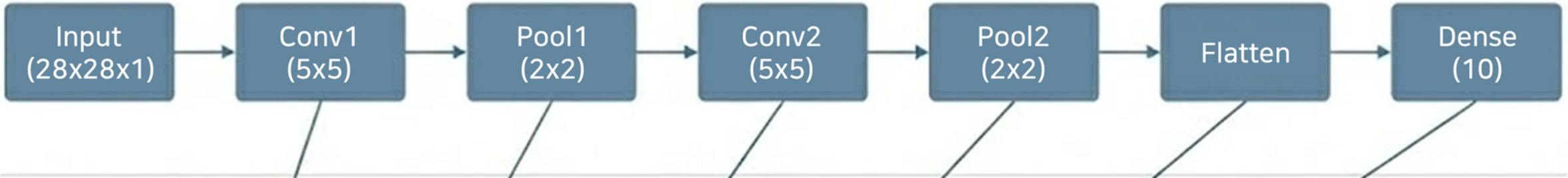
설계 및 검증

1) System Block Diagram



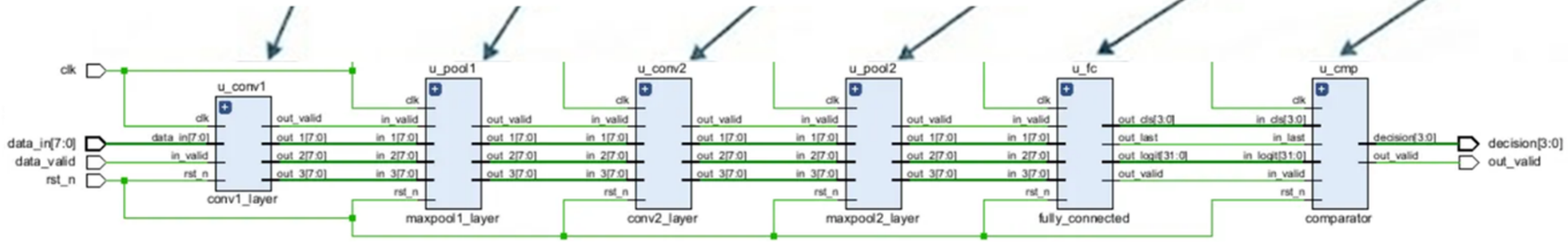
2)CNN RTL Pipeline Sturcture

Software Model



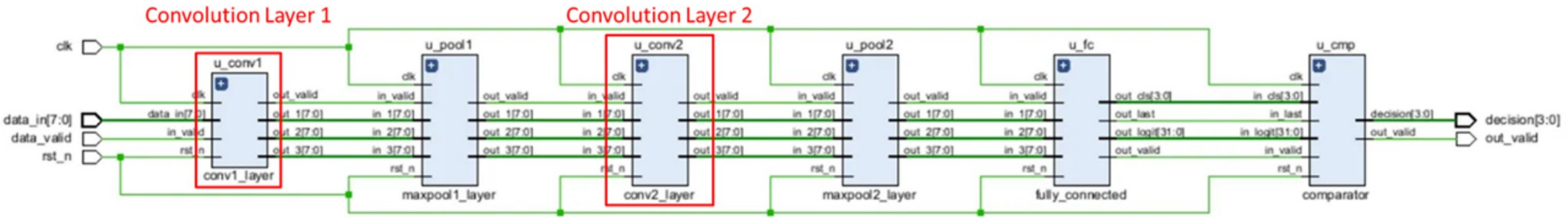
Hardware RTL

Software Model의 각 Layer는 독립적인 Verilog Module로 설계되었으며, 데이터가 각 단계를 순차적으로 흘러가는 스트리밍 파이프라인 구조로 연결




```
// cnn_core_top.v
conv1_layer u_conv1 (...);
maxpool1_layer u_pool1 (...);
conv2_layer u_conv2 (...);
maxpool2_layer u_pool2 (...);
fully_connected u_fc (...);
comparator u_cmp (...);
```

3)CNN Architecture : Rescale & Clamp



Activation(Q0.8) × Weight(Q1.7) = Result(Q1.15) → Rescale(Q0.8)

Conv 연산 결과를 다음 Activation 스케일에 맞게 Rescale 필요

```
wire signed [31:0] raw = (sum + 32'sd64) >>> 7;                                  # Rescale 진행
wire [7:0] clamp = (raw[31]) ? 8'd0 : (raw > 32'sd255) ? 8'd255 : raw[7:0]; # 8bit 저장, Clamp
```

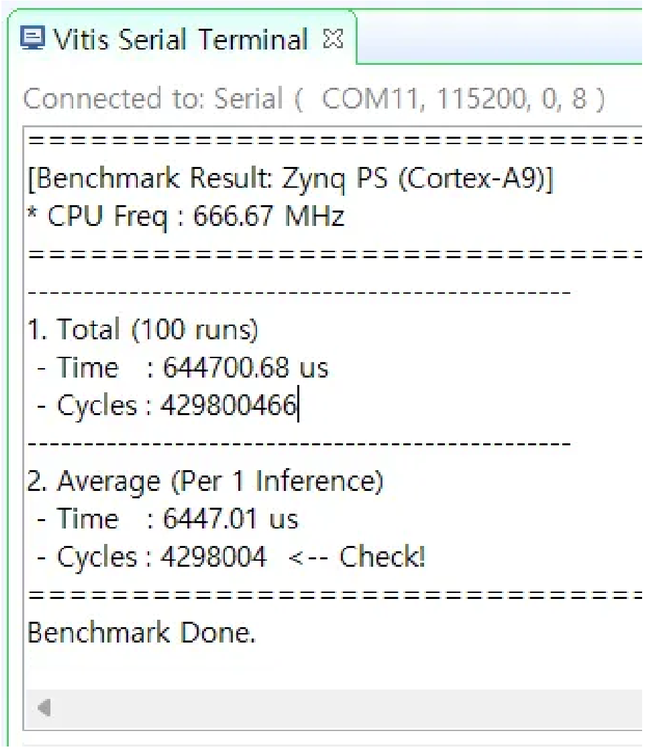
설계 및 검증

4) FPGA가속기 vs CPU (Coretax a9 cpu) = Zybo-Z720의 PL 연산 vs PS 연산

PL(verilog) 연산을 C로 1:1 레퍼런스 구현

Verilog -> **C언어**

SW-HW 결과를 동일 조건에서 검증



	FPGA	CPU (ARM-Cortec-A9)	비교
동작 주파수	125M Hz	667 MHz	CPU가 클럭 속도 5.3배 빠름
측정 방식	RTL sim	Bare-metal Measure	
소요 사이클	811 Cycles	4,298,004 Cycles	FPGA가 약 5,300배 효율적
소요 시간	6.488us	6447.01 us (6.47ms)	FPGA가 약 993배 더 빠름

설계 이슈 + 해결방안

CAMERA ISSUE

메모리 자원 부족 현상

- 컬러 HDMI 송출로 RGB565 프레임 버퍼 적용
- VGA 프레임 버퍼가 BRAM의 약 96% 점유

<원인>
RGB565는 컬러 정보 유지를 위해 프레임 버퍼 필요
▼
VGA 해상도 기준 프레임 버퍼 용량이 큼
(한 픽셀당, R(5)+G(6)+B(5) =>16bit)
▼
Zybo z7-20 내부 BRAM 용량 한계 초과

<해결방안>

1. YUV422 format 사용
   : 밝기 성분(Y)만 추출하여 Grayscale 실시간 처리구조
   - 한 픽셀당, Y(8bit) -> BRAM 점유율 약 50% 낮춤

CNN CORE ISSUE

CNN CORE 출력 이상

- 입력 이미지의 변화와 무관하게 decision 값이 특정 값으로 출력
- 시뮬레이션 파형 상 결과 값 변화가 정상적으로 반영되지 않음

<원인>
1) 가중치 매핑 구조 불일치
2) 이전 경로의 메모리 파일 참조 에러
3) 연산 파이프라인 지연 에 따른 제어신호 타이밍 불일치
▼
가중치 적용 오류 및 연산 타이밍 불일치로 인한 추론 결과 왜곡

<해결방안>

1. 가중치 매핑 정렬 (Row-Major) - > (Ch-Row-Col)
2. 메모리 파일 로컬 관리 및 최신 데이터 강제 로드
3. 쉬프트 레지스터 기반 제어신호 파이프라인화

최종결과

Onboard Edge AI

Video Processing + CNN Accelator

CNN 추론 결과가 프레임 주기 내에 반영되는 End-to-End 실시간 시스템 구현

QAT vs PTQ

1.하드웨어 친화적 모델
: 포화 및 스케일 불일치 문제 최소화
: Resource 사용량 감소, Power 감소

2. FPGA와 시너지
: 불필요한 연산 감소
: Pipeline 단순화

FPGA vs Coretax-A9

1.성능
: FPGA가 약 993배 빠른 추론 속도

2.구조적 효율성
: FPGA의 HW 병렬 구조로
약 5,300배 높은 연산 효율

기대효과

1. ASIC 구현시 성능 확장 가능성
: 기생 성분 감소 및 배선 최적화 가능
: 높은 주파수 동작 가능 -> 성능향상

2. 저전력 고속 추론 시스템
: INT8 중심 연산으로 전력 소모 down
: 엣지 디바이스, 임베디드 적용 가능성

URL Link

https://www.notion.so/2e665721b385805880a1f9499fbd1757?source=copy_link

QR Link