LeNet Accelerator

2022142255 SeokChan Jeong

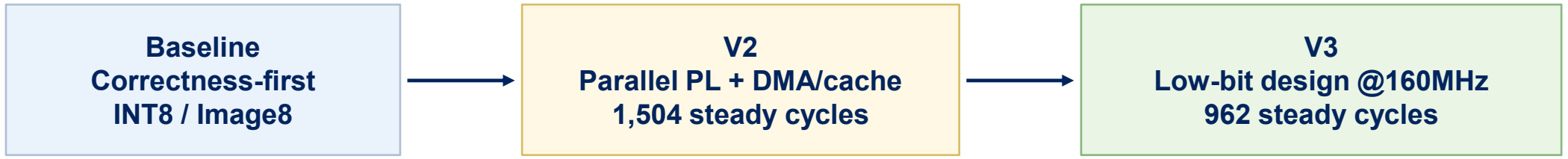
Table of Content

Abstract

Abstract

LeNet FPGA accelerator across three generations

The accelerator is measured on the full PS → PL → PS path for 1,000 MNIST images



Core Result V3 preserves accuracy while sharply reducing steady-state PL latency and end-to-end execution time.

98.30%	6.0125 µs	2.001×	814.6×
V3 accuracy 983 / 1000	V3 steady PL latency 962 cycles @160MHz	V2 → V3 PL latency speedup	E2E speedup vs true baseline

Benchmark Summary

Accuracy is maintained around 98%, while PL cycles drop from 434,330 cycles/image in the baseline to 962 steady cycles in V3.

Version	Accuracy	PL Cycles	PL Latency	E2E Total	Main role
Baseline	98.40% 984 / 1000	434,330 cycles/image	8,686.60 µs avg / image @50MHz	8,760.471 ms 1000 images	Correctness-first PS→PL→PS flow
V2	98.60% 986 / 1000	1,504 steady cycles	12.032 µs @125MHz	22.589 ms 1000 images	Parallel PL + DMA/cache timing
V3	98.30% 983 / 1000	962 steady cycles	6.0125 µs @160MHz	10.754 ms 1000 images	Low-bit design + clock improvement

99.78%
PL cycle reduction
vs Baseline

36.04%
V2 → V3 cycle
reduction

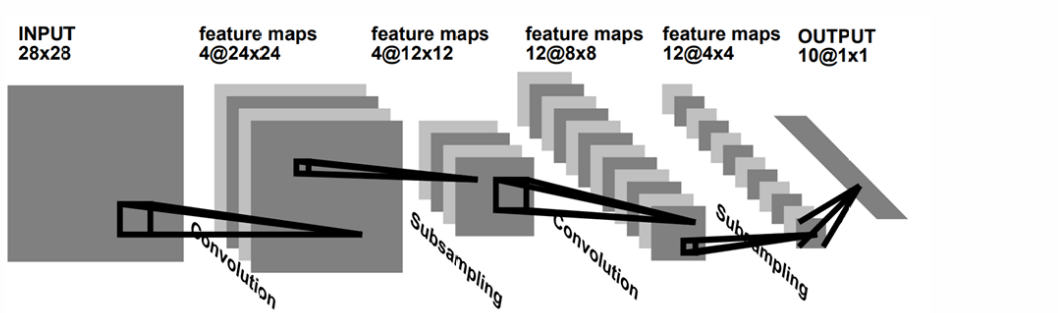
-0.10 pp
V3 accuracy change
vs Baseline

95.24%
Non-PL total
reduction vs Baseline

Abstract message: V3 achieves the best latency-efficiency point, delivering a 2× steady PL latency speedup over V2 while staying within 0.3 pp of V2 accuracy.

LeNet Baseline

What is LeNet?

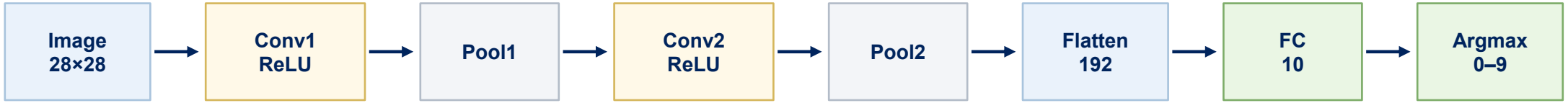


- **CONV1** : 4×(5×5) kernel, stride=1, no padding or bias → **Output**: 4×24×24
- **POOL1** : 2×2 max-pool, stride=2 → **Output**: 4×12×12
- **CONV2** : 12×(5×5) kernel, stride=1, no padding or bias → **Output**: 12×8×8
- **POOL2** : 2×2 max-pool, stride=2 → **Output**: 12×4×4
- **FC** : Flatten (12×4×4=192), no bias → **Output**: 10 classes

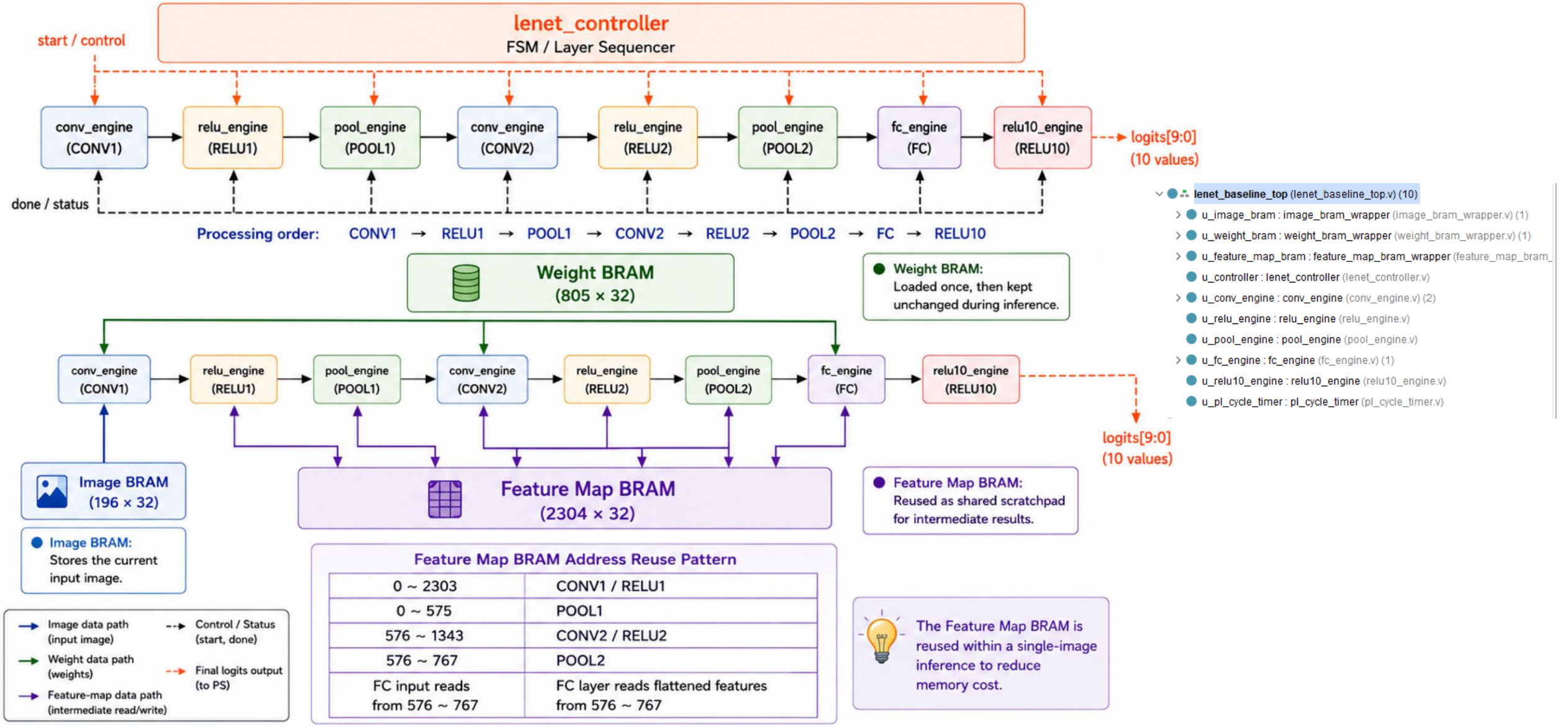
Stage	Operation	Parameters	Output Shape
Input	Image	uint8, 1 channel	1 × 28 × 28
Conv1	Valid conv + ReLU	4 filters, 5×5, stride 1, no bias	4 × 24 × 24
Pool1	MaxPool	2×2, stride 2	4 × 12 × 12
Conv2	Valid conv + ReLU	12 filters, 5×5, stride 1, no bias	12 × 8 × 8
Pool2	MaxPool	2×2, stride 2	12 × 4 × 4
FC	Fully connected + ReLU	192 → 10, no bias	10 logits
Output	Argmax	select max activated logit	digit 0–9

Input Data and Weight Setup

Header File	Parsed Type	Expected Size	Tensor Shape
conv1_arr.h	int8	4×1×5×5 = 100	4 × 1 × 5 × 5
conv2_arr.h	int8	12×4×5×5 = 1,200	12 × 4 × 5 × 5
fc1_arr.h	int8	10×192 = 1,920	10 × 192
test_images_arr.h	uint8	N×28×28	N × 28 × 28
test_labels_arr.h	uint8	N labels	N

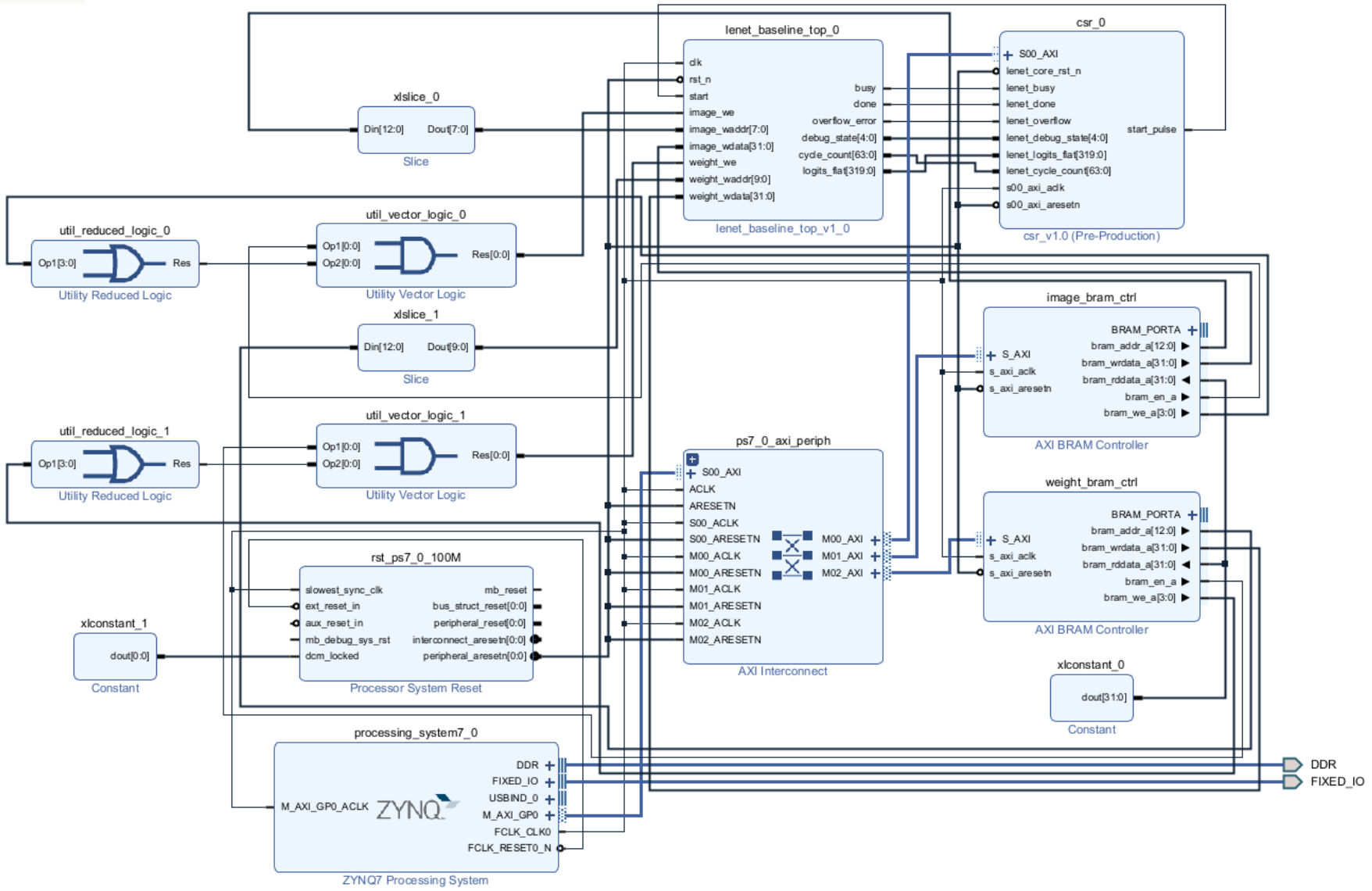


PL Architecture

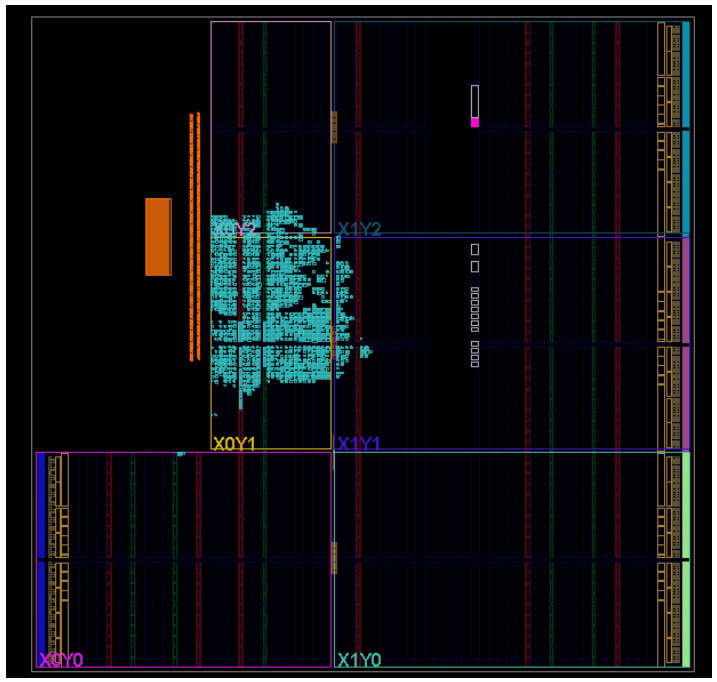


PL Result and Block Design

```
============================================================
LeNet RTL dataset test start
Images             : 1000
Clock              : 100 MHz
Weight words loaded: 805
============================================================
MISCLASSIFIED image[20]: RTL=8, LABEL=1, cycles=434330
MISCLASSIFIED image[59]: RTL=1, LABEL=2, cycles=434330
MISCLASSIFIED image[61]: RTL=9, LABEL=4, cycles=434330
MISCLASSIFIED image[94]: RTL=8, LABEL=2, cycles=434330
Progress 100/1000: correct=96, wrong=4, running_acc=96.00%
MISCLASSIFIED image[110]: RTL=9, LABEL=7, cycles=434330
MISCLASSIFIED image[128]: RTL=8, LABEL=1, cycles=434330
Progress 200/1000: correct=194, wrong=6, running_acc=97.00%
MISCLASSIFIED image[246]: RTL=5, LABEL=3, cycles=434330
MISCLASSIFIED image[273]: RTL=9, LABEL=0, cycles=434330
MISCLASSIFIED image[278]: RTL=4, LABEL=0, cycles=434330
MISCLASSIFIED image[287]: RTL=0, LABEL=6, cycles=434330
Progress 300/1000: correct=290, wrong=10, running_acc=96.67%
MISCLASSIFIED image[316]: RTL=2, LABEL=7, cycles=434330
Progress 400/1000: correct=389, wrong=11, running_acc=97.25%
Progress 500/1000: correct=489, wrong=11, running_acc=97.80%
MISCLASSIFIED image[508]: RTL=5, LABEL=3, cycles=434330
MISCLASSIFIED image[520]: RTL=9, LABEL=4, cycles=434330
MISCLASSIFIED image[527]: RTL=9, LABEL=4, cycles=434330
Progress 600/1000: correct=586, wrong=14, running_acc=97.67%
Progress 700/1000: correct=686, wrong=14, running_acc=98.00%
Progress 800/1000: correct=786, wrong=14, running_acc=98.25%
Progress 900/1000: correct=886, wrong=14, running_acc=98.44%
Progress 1000/1000: correct=986, wrong=14, running_acc=98.60%

============================================================
FINAL DATASET RESULT
============================================================
Correct                    : 986 / 1000
Wrong                      : 14
Classification accuracy    : 98.60%
Unknown/X/Z logits         : 0
Overflow observed          : 0

Total PL busy cycles       : 434330000
Average PL cycles/image    : 434330.00
Minimum PL cycles/image    : 434330
Maximum PL cycles/image    : 434330
Pure PL compute time       : 4343.300 ms
Average PL latency/image   : 4343.300 us

Weight-loading modeled time: 8.060 us
Dataset modeled time       : 4345.290 ms
Total modeled time         : 4345.298 ms
============================================================
TEST COMPLETED
```



Block Design Analysis



Tcl Console | Messages | Log | Reports | Design Runs | DRC | Methodology | **Power** | Timing

Settings
Summary (1.43 W, Margin: N/A)
Power Supply
Utilization Details
- Hierarchical (1.295 W)
- Clocks (0.009 W)
- Signals (0.011 W)
  - Data (0.01 W)
  - Clock Enable (0.001 W)
  - Set/Reset (<0.001 W)
- Logic (0.008 W)
- BRAM (0.008 W)
- DSP (0.004 W)
- PS7 (1.256 W)

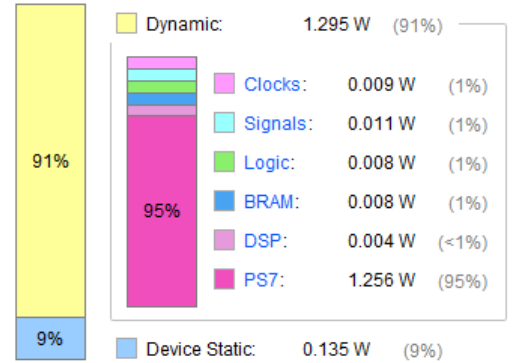
Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**1.43 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**41.5°C**
Thermal Margin:	43.5°C (3.6 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power



impl_1 (saved)

Tcl Console | Messages | Log | Reports | **Design Runs** | DRC | Methodology | Power | Timing

Name	Constraints	Status	WNS	WHS	THS	TNS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	BRAM	FF	URAM	DSP	Start
synth_1 (active)	**constrs_1**	**synth_design Complete!**												**0**	**0**	**0**	**0**	**0**	**8/1/26, 5:12 AM**
impl_1	constrs_1	write_bitstream Complete!	2.461	0.023	0.000	0.000		0.000	1.410	0				2434	4	2661	0	4	8/1/26, 5:12 AM
Out-of-Context Module Runs																			
design_1		Submodule Runs Complete																	8/1/26, 4:57 AM

PS Result

```
================================================
 LeNet Baseline Accelerator Benchmark
 PS image -> PL inference -> PS result
================================================

CSR     : 0x43c00000
IMAGE   : 0x40000000
WEIGHT  : 0x42000000
VERSION : 0x00010000
STATUS  : 0x00000008

[PS -> PL -> PS] 1000 images
mismatch image[10]: PL=0 label=3
mismatch image[11]: PL=2 label=4
mismatch image[20]: PL=8 label=1
mismatch image[59]: PL=1 label=2
mismatch image[61]: PL=9 label=4
mismatch image[94]: PL=8 label=2
mismatch image[110]: PL=9 label=7
mismatch image[128]: PL=8 label=1
mismatch image[246]: PL=5 label=3
mismatch image[273]: PL=9 label=0
mismatch image[278]: PL=4 label=0
mismatch image[287]: PL=0 label=6
mismatch image[316]: PL=2 label=7
mismatch image[508]: PL=5 label=3
mismatch image[520]: PL=9 label=4
mismatch image[527]: PL=9 label=4

==================== Summary ====================
Processed                   : 1000/1000
Accuracy                    : 984/1000 = 98.40%
Wrong                       : 16
Weight upload once          : 297 us
PS->PL->PS E2E total        : 8760471 us (8.760 s)
E2E + initial weight upload : 8760768 us (8.760 s)
E2E average/image           : 8760.47 us
Pure PL compute total       : 8686600 us
Pure PL average/image       : 8686.60 us
PS/AXI/control overhead     : 73871 us
PL total busy cycles        : 434330000
PL min/max cycles           : 434330 / 434330
PL overflow count           : 0
PL timeout count            : 0
================================================

Test completed.
```

Item	Arty Z7 Board	RTL Simulation
Test Images	1,000 images	1,000 images
Accuracy	98.40%	98.60%
Correct / Wrong	984 / 16	986 / 14
Operating Clock	50 MHz	100 MHz
PL Cycles / Image	434,330	434,330
Total PL Cycles	434,330,000	434,330,000
Pure PL Compute Time	8.6866 s	4.3433 s
Total Processing Time	8.760 s	4.345 s
Overflow	0	0
Timeout / X-Z	0	0

Total Processing Time	Pure PL Compute Time	PS / AXI / Control Overhead	PL Cycles / Image
8.760 s PS image → PL inference → PS result	8.6866 s FPGA computation only	0.0734 s E2E - PL time	434,330 single-image latency

Time Breakdown

V2 Accelerator

Where the Baseline Loses Cycles

The baseline is functionally correct, but its datapath repeatedly waits on memory, reuses a small MAC path, and cannot overlap frames.

01

Shared Feature BRAM

Problem

Conv, ReLU, and Pool repeatedly read/write the same feature-map BRAM.

v2 direction

Fuse Conv + ReLU + Pool into one streaming module.

02

Low MAC Throughput

Problem

A single MAC-style path serializes convolution and FC operations.

v2 direction

Allocate dedicated parallel engines using 160 DSPs.

03

Redundant Data Access

Problem

Adjacent 5×5 windows reread data that is already available.

v2 direction

Use line buffers and decouple window generation from MAC issue.

04

Memory Waiting Gaps

Problem

Single image / Pool1 buffers force the next stage or frame to wait.

v2 direction

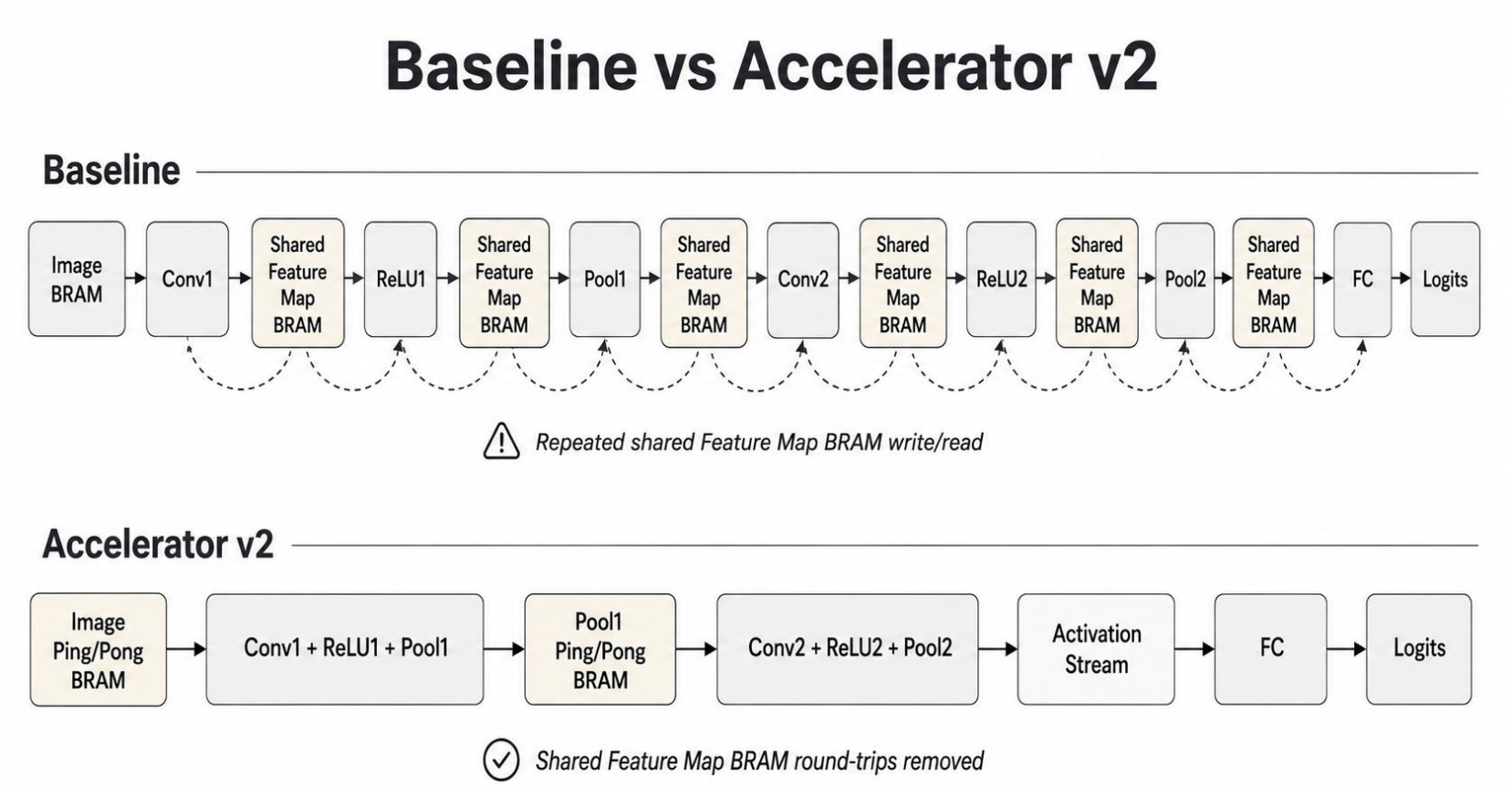
Use image / Pool1 ping-pong buffers and row-ready handshaking.

Core strategy: reduce PL cycles by increasing parallel compute, feeding MAC lanes continuously, and overlapping memory stages.

Bottleneck 1: Repeated Feature-Map BRAM Read / Write

B1

Baseline stores intermediate feature maps between Conv, ReLU, and Pool, causing repeated memory traffic.



Takeaway

The first bottleneck is not arithmetic itself. It is repeated movement of the same feature map through shared BRAM between small operators.

Upgrade rule

Keep data on-chip and streaming whenever possible.

Bottleneck 2: Low Arithmetic Throughput from a Reused MAC Engine

B2

The model requires many MAC operations per image. A sequential MAC path turns this into a large cycle count.

Layer	Output shape	Multiplications / output	Total multiplications
Conv1	4 × 24 × 24	5 × 5 = 25	57,600
Conv2	12 × 8 × 8	4 × 5 × 5 = 100	76,800
FC	10 classes	192 inputs / class	1,920
Total	-	-	136,320

Main issue

One image already requires 136,320 multiplications before address control, accumulation, and memory latency.

Sequential output-channel processing

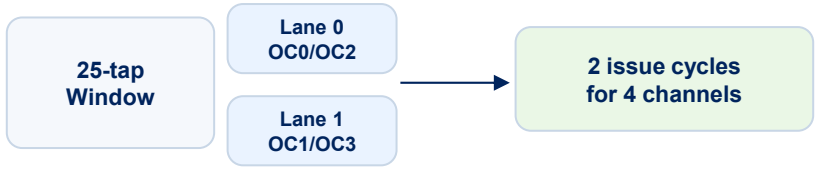
The input window is reused, but the baseline consumes it through one compute path. Cycles grow with output channels and kernel taps.

v2 Response: Dedicated Parallel Engines

Accelerator v2 replaces the shared convolution path with dedicated Conv1, Conv2, and FC engines, using 160 DSPs in total.

Conv1 Engine	Conv2 Engine	FC Engine	Total
50 DSP	**100 DSP**	**10 DSP**	**160 DSP**
2 dot25 lanes × 25 multipliers	4 output lanes × 25 taps	10 class lanes in parallel	parallel datapath allocation

Conv1 issue schedule



24×24 windows × 2 channel groups = 1,152 issue cycles

Conv2 partial parallelization

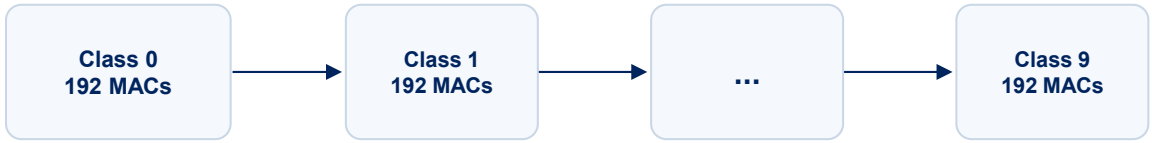
12 output channels = 3 groups; 4 input channels per group → 12 issue cycles per spatial position.

Parallel engines raise hardware utilization: each layer receives a datapath sized to its arithmetic demand instead of sharing one serial MAC path.

FC Engine: Broadcast One Activation to 10 Class Lanes

The fully connected layer is smaller than convolution, but class-parallel MAC lanes remove sequential class processing.

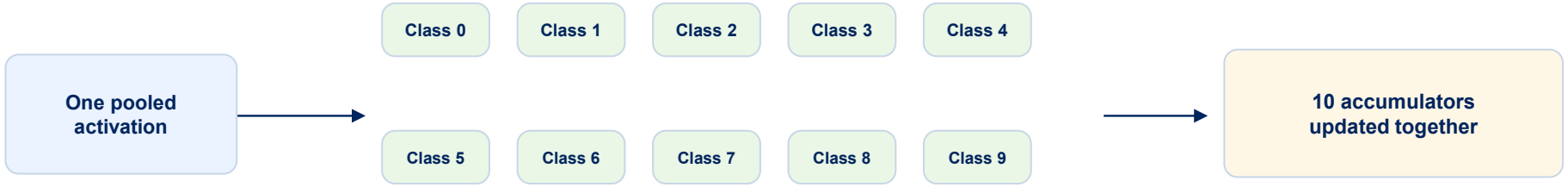
Baseline FC order



Key numbers

192 input activations
× 10 output classes
= 1,920 multiplications

Accelerator v2 FC order



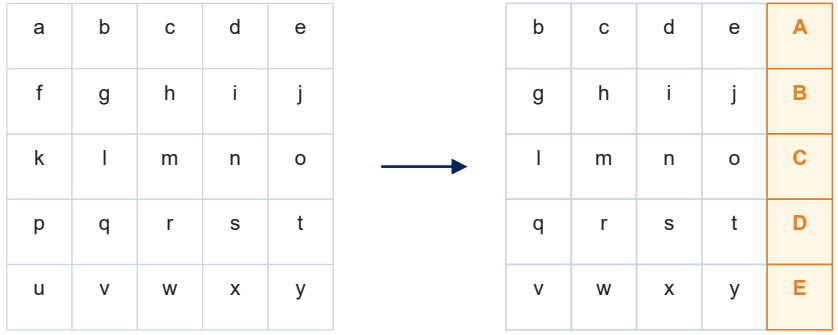
The RTL uses 10 independent weight banks and broadcasts one pooled activation to all class MAC lanes.

Bottleneck 3: Redundant Data Access Between Adjacent Windows

B3

Adjacent convolution windows share most pixels, but independent BRAM reads request the same data again.

Two adjacent 5×5 windows



20 / 25 pixels are reused

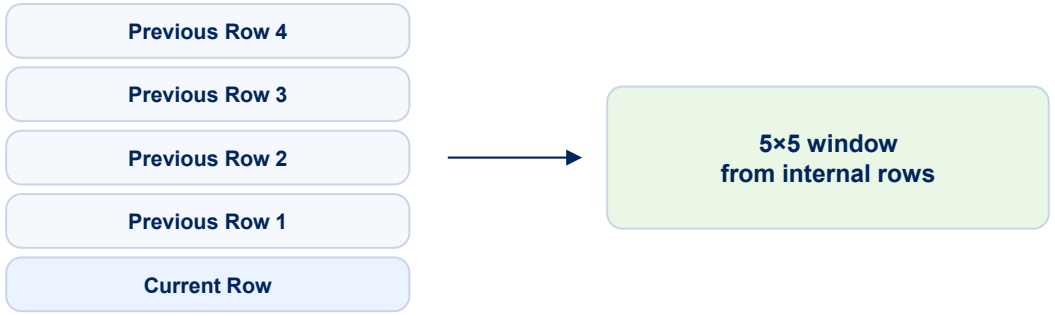
Naïve access

Window 0: 25 reads
Window 1: 25 reads
Window 2: 25 reads

Reuse-aware access

Next window needs only 5 new pixels; 20 pixels can be reused internally.

Upgrade direction: line buffer



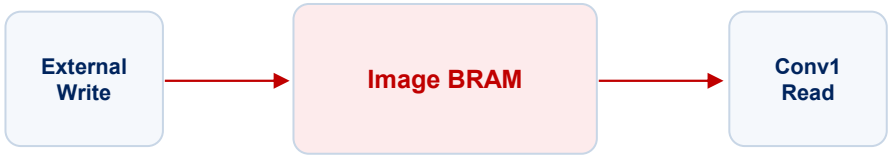
Line buffers retain previous rows so the accelerator does not reread them from image or Pool1 BRAM.

Bottleneck 4: Image Memory Waits Between Frames

B4

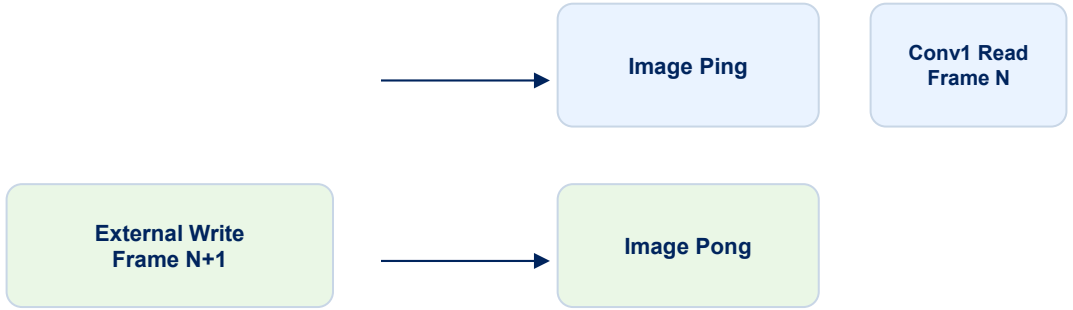
A single image BRAM cannot safely receive the next image while Conv1 is reading the current one.

Single image BRAM



Frame N load → Conv1 reads Frame N → Frame N+1 load

Image ping-pong buffer



Control signals

load_bank: bank for the next external image write
c1_active_bank: bank for the current Conv1 read
image_ready = 1 when the load bank is empty

Ping-pong buffering overlaps image transfer with computation, eliminating frame-load gaps before Conv1.

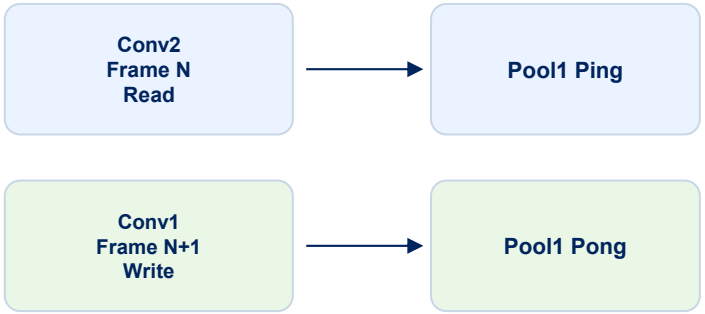
Bottleneck 4 Continued: Pool1 Buffer Prevents Layer Overlap

Conv1 writes Pool1 results while Conv2 reads Pool1 as input. With one buffer, the two stages conflict.

Single Pool1 BRAM conflict

Even with dedicated Conv1 and Conv2 engines, one intermediate buffer serializes them.

Pool1 ping-pong buffer



Frame pipeline

Pool1 ping-pong enables Conv1 of the next frame to run while Conv2 + FC process the current frame.

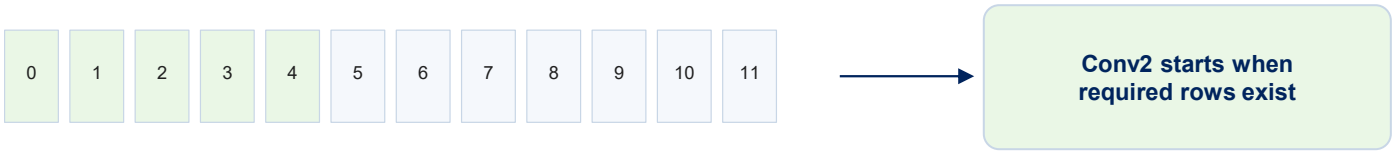
Pool1 Row-Ready: Start Conv2 Before the Whole Feature Map is Done

Pool1 ping-pong solves inter-frame conflict, but row-ready also reduces same-frame waiting between Conv1 and Conv2.

Without row-ready

Conv2 waits until the entire 12×12 Pool1 map is complete.

With row-ready bits



row_ready[0..11]

RTL-level idea

Each Pool1 bank has a 12-bit row_ready status. When Conv1 finishes writing channel group 2–3 for the last column of a row, the corresponding ready bit is set.

Conv2 checks whether the rows needed for its 5×5 window are ready before reading.

Result: Conv2 does not wait for the full Pool1 feature map; it can begin as soon as enough rows are available.

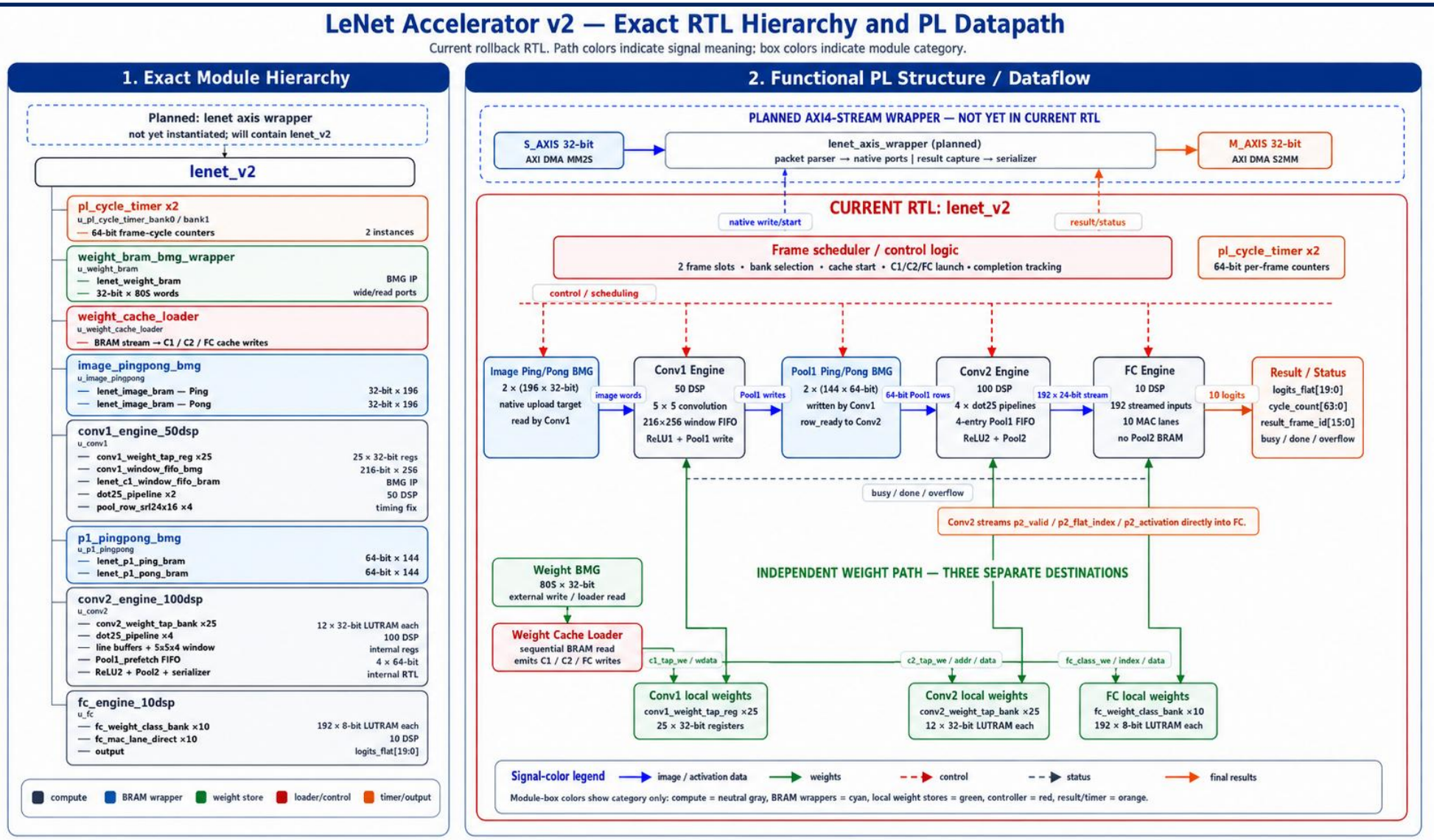
Upgrade Summary: From Sequential Baseline to Pipelined v2

The v2 design attacks the dominant PL bottleneck through module fusion, parallel engines, data reuse, and overlapped buffering.

Bottleneck	Baseline limitation	v2 hardware direction	Expected effect
B1 Feature BRAM	Conv / ReLU / Pool repeatedly access shared feature-map BRAM	Fuse Conv + ReLU + Pool into one module	Lower intermediate memory traffic
B2 MAC throughput	One MAC path serializes 136,320 multiplications / image	Conv1 50 DSP, Conv2 100 DSP, FC 10 DSP	More MACs completed per cycle
B3 Data access	Adjacent 5×5 windows reread mostly identical pixels	Line buffer, BRAM banking, decoupled window generation	Keep DSP lanes supplied
B4 Waiting gaps	Single Image / Pool1 memories prevent frame and layer overlap	Image / Pool1 ping-pong + row-ready	Overlap load, Conv1, Conv2, and FC

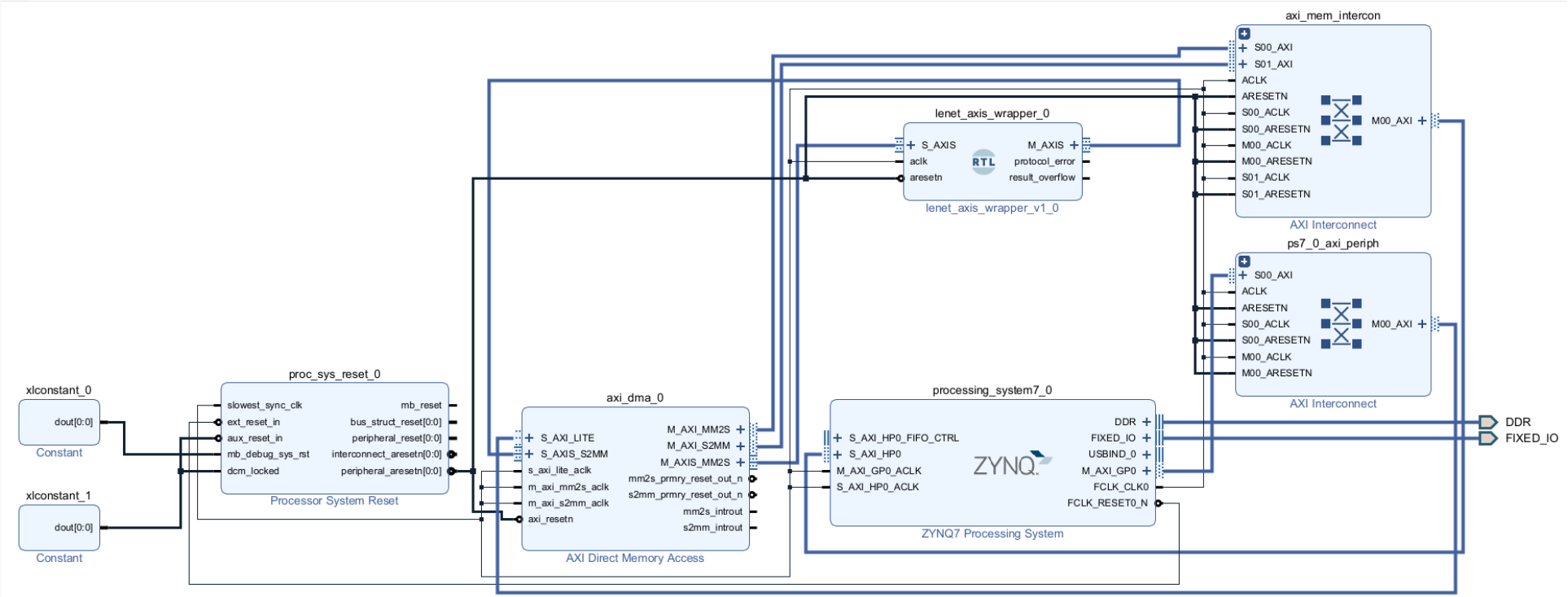
Main message: the baseline is not inaccurate; it is underutilized. v2 improves latency by restructuring the PL datapath around parallel compute and memory overlap.

PL Architecture

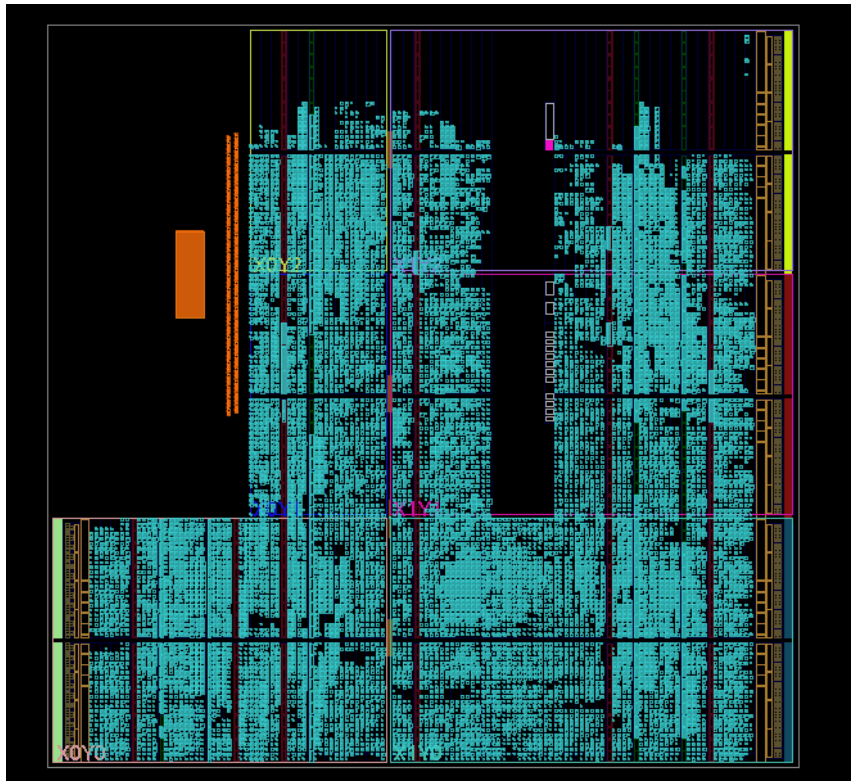


PL Result and Block Design

```
============================================================
 LENET AXI WRAPPER DATASET SIMULATION
 Clock        : 125 MHz
 Images       : 1000
 Weight words : 805
============================================================
Sending weight packet...
Weight packet complete.
MISCLASSIFIED image[20]: AXIS=8 LABEL=1 cycles=1504
  logits: 9780711 11545767 9437760 0 4010856 0 9950556 0 17199342 0
MISCLASSIFIED image[59]: AXIS=1 LABEL=2 cycles=1504
  logits: 0 28227836 25024248 5319674 0 0 764578 0 21092369 0
MISCLASSIFIED image[61]: AXIS=9 LABEL=4 cycles=1504
  logits: 0 4055719 3850217 118952 21014173 0 0 6845841 4193318 21375826
MISCLASSIFIED image[94]: AXIS=8 LABEL=2 cycles=1504
  logits: 0 18584919 22537120 605850 0 2747534 0 0 29510764 5696451
Progress 100/1000 correct=96 wrong=4
MISCLASSIFIED image[110]: AXIS=9 LABEL=7 cycles=1504
  logits: 0 11933352 8593986 193046 0 0 0 18369555 6645029 22399519
MISCLASSIFIED image[128]: AXIS=8 LABEL=1 cycles=1504
  logits: 0 12106443 7702000 7257941 0 6581200 10859137 0 19017769 0
Progress 200/1000 correct=194 wrong=6
MISCLASSIFIED image[246]: AXIS=5 LABEL=3 cycles=1504
  logits: 998177 0 651997 34147954 0 40750429 0 4998677 13649668 10781099
MISCLASSIFIED image[273]: AXIS=9 LABEL=0 cycles=1504
  logits: 26988731 0 10928520 0 0 0 0 4711667 19511658 33636187
MISCLASSIFIED image[278]: AXIS=4 LABEL=0 cycles=1504
  logits: 12919759 0 1525014 0 13133380 663104 10114426 0 10636819 6586348
MISCLASSIFIED image[287]: AXIS=0 LABEL=6 cycles=1504
  logits: 19933705 0 13277917 4396604 6072181 1267200 13916804 0 8264651 0
Progress 300/1000 correct=290 wrong=10
MISCLASSIFIED image[316]: AXIS=2 LABEL=7 cycles=1504
  logits: 8919249 0 34695982 19546819 0 0 0 33686993 31595567 18464547
Progress 400/1000 correct=389 wrong=11
Progress 500/1000 correct=489 wrong=11
MISCLASSIFIED image[508]: AXIS=5 LABEL=3 cycles=1504
  logits: 9434400 11681118 19607773 22333924 0 22395236 0 517716 0 2782209
MISCLASSIFIED image[520]: AXIS=9 LABEL=4 cycles=1504
  logits: 0 21457211 0 0 29922282 0 0 11291617 28372228 37671399
MISCLASSIFIED image[527]: AXIS=9 LABEL=4 cycles=1504
  logits: 747080 5838908 8393276 0 17302415 8432803 0 11283259 6941694 18286512
Progress 600/1000 correct=586 wrong=14
Progress 700/1000 correct=686 wrong=14
Progress 800/1000 correct=786 wrong=14
Progress 900/1000 correct=886 wrong=14
Progress 1000/1000 correct=986 wrong=14

============================================================
 AXI WRAPPER SIMULATION RESULT
============================================================
Correct             : 986 / 1000
Wrong               : 14
Accuracy            : 98.60 %
Frame ID errors     : 0
Protocol errors     : 0
Core overflow count : 0
Protocol sticky     : 0
Queue sticky        : 0
Total cycles        : 1510442
Average cycles      : 1510.44
Min cycles          : 1504
Max cycles          : 7946
============================================================
```



Block Design Analysis



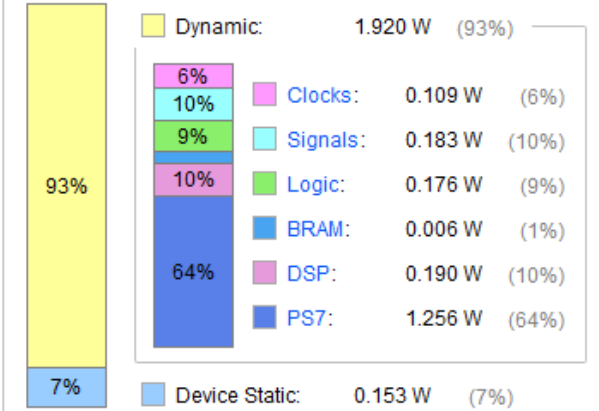
Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**2.073 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**48.9°C**
Thermal Margin:	36.1°C (3.0 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power



Name	Constraints	Status	WNS	WHS	THS	TNS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	BRAM	FF	URAM	DSP	Start	Elapsed	Run Strategy
synth_1 (active)	**constrs_1**	**synth_design Complete!**												**0**	**0**	**0**	**0**	**0**	**8/10/26, 3:58 AM**	**00:00:34**	**Vivado Synthesis Defaults* (Vivado Synthesis 2024)**
impl_1	constrs_1	write_bitstream Complete!	0.458	0.020	0.000	0.000		0.000	2.073	0	5123 Warn			21209	9	40128	0	160	8/10/26, 3:59 AM	00:05:22	Performance_NetDelay_high (Vivado Implementation 2024)
Out-of-Context Module Runs																					
lenet_image_bram_synth_1	lenet_image_bram	synth_design Complete!												0	0.5	0	0	0	8/2/26, 2:21 AM	00:00:56	Vivado Synthesis Defaults (Vivado Synthesis 2024)
lenet_weight_bram_synth_1	lenet_weight_bram	synth_design Complete!												0	1	0	0	0	8/2/26, 2:22 AM	00:00:39	Vivado Synthesis Defaults (Vivado Synthesis 2024)
lenet_p1_pong_bram_synth_1	lenet_p1_pong_bram	synth_design Complete!												0	1	0	0	0	8/2/26, 2:24 AM	00:00:39	Vivado Synthesis Defaults (Vivado Synthesis 2024)
lenet_c1_window_fifo_bram_synth_1	lenet_c1_window_fifo_bram	synth_design Complete!												0	3	0	0	0	8/2/26, 4:23 AM	00:00:40	Vivado Synthesis Defaults (Vivado Synthesis 2024)
design_1		Submodule Runs Complete																	8/9/26, 9:37 PM	06:20:51	
design_1_xbar_0_synth_1	design_1_xbar_0	synth_design Complete!												286	0	353	0	0	8/9/26, 9:37 PM	00:01:03	Vivado Synthesis Defaults (Vivado Synthesis 2024)
design_1_processing_system7_0_0_synth_1	design_1_processing_system7_0_0	synth_design Complete!												24	0	0	0	0	8/9/26, 9:37 PM	00:00:52	Vivado Synthesis Defaults (Vivado Synthesis 2024)
design_1_lenet_axis_wrapper_0_0_synth_1	design_1_lenet_axis_wrapper_0_0	synth_design Complete!												18726	0	36692	0	160	8/10/26, 3:56 AM	00:01:46	Vivado Synthesis Defaults (Vivado Synthesis 2024)
design_1_axi_dma_0_0		Using cached IP results																			

PS Result

```
============================================================
 LeNet-v2 D-CACHE ON / PS+PL TIMING ANALYSIS
============================================================
D-cache          = ENABLED with explicit DMA cache maintenance
DMA address      = 0x40400000
Result packet    = 13 words / 52 bytes
Images           = 1000
PL clock         = 125000000 Hz
Global timer     = 333333333 Hz (fixed for ~666.666 MHz Cortex-A9)

PL time          = FPGA computation cycle_count only
PS+PL FULL time  = weight setup/DMA + 1000 complete inferences
UART output      = excluded from measured region
============================================================

============================================================
 MISCLASSIFICATIONS
============================================================
image[20]: label=1 PL=8
image[59]: label=2 PL=1
image[61]: label=4 PL=9
image[94]: label=2 PL=8
image[110]: label=7 PL=9
image[128]: label=1 PL=8
image[246]: label=3 PL=5
image[273]: label=0 PL=9
image[278]: label=0 PL=4
image[287]: label=6 PL=0
image[316]: label=7 PL=2
image[508]: label=3 PL=5
image[520]: label=4 PL=9
image[527]: label=4 PL=9

============================================================
 FINAL SUMMARY
============================================================
Processed                    : 1000 / 1000
PL accuracy                  : 986 / 1000 = 98.60 %
Misclassifications           : 14
Frame ID errors              : 0
Protocol errors              : 0
Core overflows               : 0
Queue overflows              : 0

---------- PL TIME ----------
Total PL cycles              : 1510442
Average PL cycles            : 1510
Min PL cycles                : 1504
Max PL cycles                : 7946
Total PL computation time    : 12.083536 ms
Average PL computation time  : 12.083 us
Steady-state PL avg (1..999) : 12.032 us
Min PL computation time      : 12.032 us
Max PL computation time      : 63.568 us

---------- PS+PL FULL TIME ----------
PS+PL FULL total time        : 22.589361 ms
PS+PL FULL avg / image       : 22.589 us

---------- OVERHEAD ----------
Total non-PL time            : 10.505825 ms
Average non-PL / image       : 10.505 us
Full throughput              : 44268.627 images/s
Pure PL steady throughput    : 83111.702 images/s
PL share of full time        : 53.49 %
Non-PL share of full time    : 46.50 %
End-to-end / PL time ratio   : 1.869 x
============================================================
TEST FINISHED
```

Metric	Baseline	LeNet-v2	Change
Processed images	1000 / 1000	1000 / 1000	same
Correct predictions	984	986	+2
Wrong predictions	16	14	−2
Frame / protocol errors	N/A	0 / 0	clean protocol
Core / queue overflow	0	0 / 0	no overflow

PL cycles / image

434,330 cycles → **1,504 cycles**

288.8× fewer cycles · 99.65% reduction

PL latency / image

8,686.60 us → **12.032 us**

722.0× faster PL latency

E2E average / image

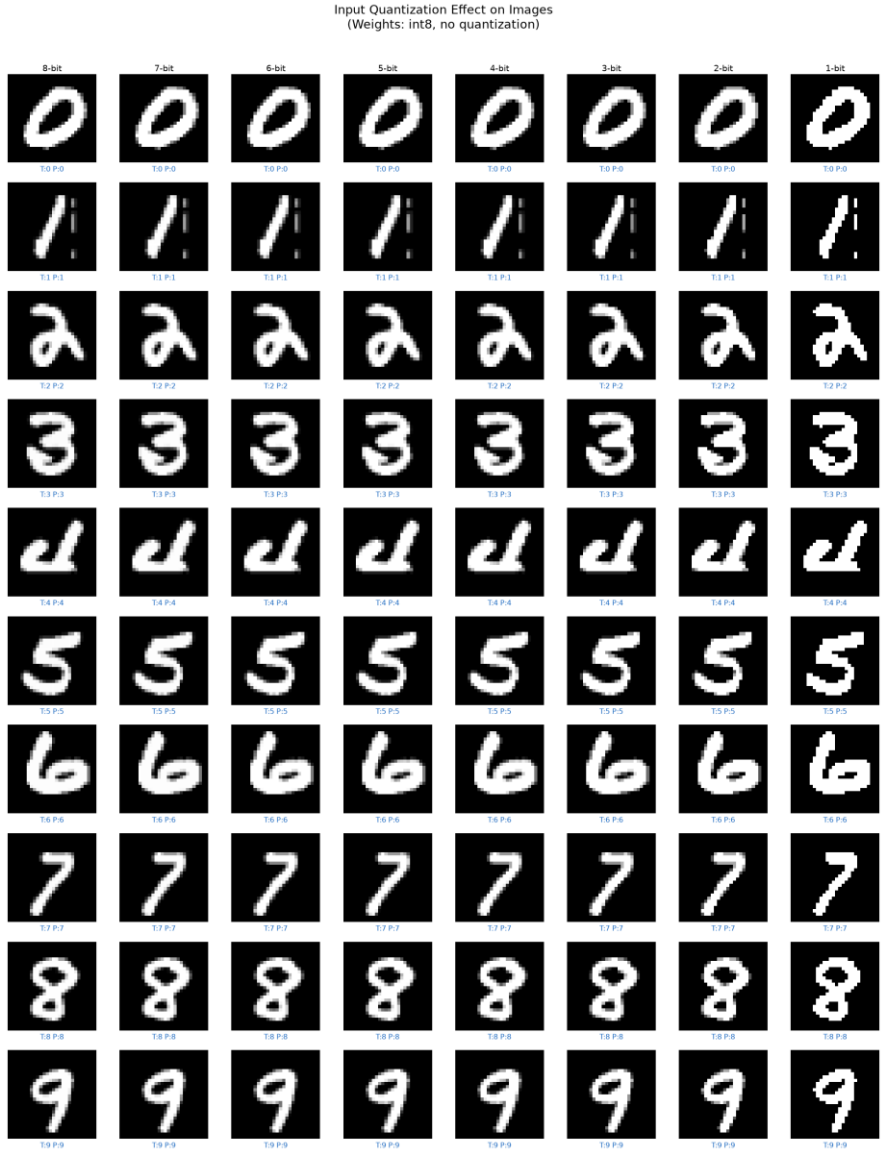
8,760.47 us → **22.589 us**

387.8× faster end-to-end

Architecture-normalized view: cycles / image

V3 Accelerator(Quantize+BitMoD)

Why 1-bit Image Input?

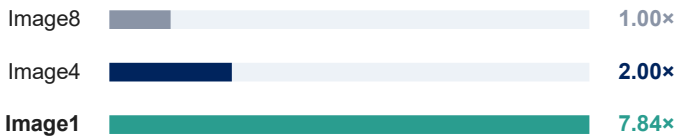



```
============================================================
  Step 1/5 : Evaluating accuracy for each bit-width ...
============================================================
   8-bit  acc= 98.60%  (986/1000)  10.7s
   7-bit  acc= 98.60%  (986/1000)  10.7s
   6-bit  acc= 98.70%  (987/1000)  9.9s
   5-bit  acc= 98.80%  (988/1000)  10.1s
   4-bit  acc= 98.70%  (987/1000)  9.9s
   3-bit  acc= 98.70%  (987/1000)  9.8s
   2-bit  acc= 98.60%  (986/1000)  10.0s
   1-bit  acc= 98.40%  (984/1000)  10.0s
```

Input-bit accuracy sweep

Image bits	8	6	4	3	2	1
Accuracy (%)	98.60	98.70	98.50	98.40	98.20	98.40
Imem compression	1.00×	1.33×	2.00×	2.65×	4.00×	7.84×

Imem compression by image precision



Input	Result
0	0
1	passes through

Why BitMoD FP4

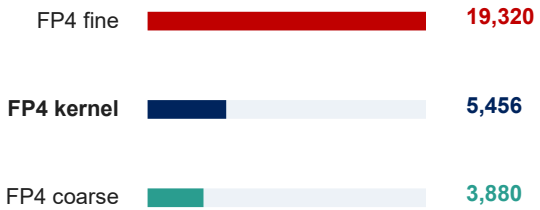
FP3 gives stronger compression and compute proxy, but the accuracy drop is too large. FP4 keeps the model inside the target loss budget.

Scheme	Bits	PG preset	Img	Acc.	ΔAcc	Wmem	Imem	Comp	GrpOps	HWscore
BitMoD-FP3 coarse	3	coarse [25,50,48]	1-bit	96.40%	−2.20 pp	2.49×	7.84×	3.47×	3,880	3.322
BitMoD-FP3 kernel	3	kernel [25,25,24]	1-bit	96.20%	−2.40 pp	2.35×	7.84×	3.47×	5,456	3.028
BitMoD-FP4 coarse	4	coarse [25,50,48]	1-bit	98.00%	−0.60 pp	1.90×	7.84×	2.61×	3,880	2.758
BitMoD-FP4 kernel	**4**	**kernel [25,25,24]**	**1-bit**	**98.10%**	**−0.50 pp**	**1.81×**	**7.84×**	**2.61×**	**5,456**	**2.525**
BitMoD-FP4 fine	4	fine [5,10,16]	1-bit	98.50%	−0.10 pp	1.66×	7.84×	2.61×	19,320	1.518

Reason	Weight Scheme	Img	Acc
Best Accuracy	INT4-Sym-PG-kernel [25,25,24]	4	99.00%
Safe Efficiency (loss ≤ 0.3pp)	INT3-Asym-PG-fine [5,10,16]	1	98.30%
Balanced Efficiency (loss ≤ 0.5pp) / RTL-friendly PG	INT3-Asym-PG-coarse [25,50,48]	1	98.10%
Aggressive Efficiency (loss ≤ 1.0pp)	INT3-Sym-PG-coarse [25,50,48]	1	97.60%
Best BitMoD (loss ≤ 0.5pp)	BitMoD-FP4-Adaptive-PG-kernel [25,25,24]	1	98.10%

Accuracy at Image1

Group rescale pressure



Choice	Accuracy	ΔAcc	Wmem	Comp	GrpOps	Role
INT3-Asym coarse	98.10%	−0.50	2.28×	3.47×	3,880	Best balanced proxy
INT3-Sym coarse	97.60%	−1.00	2.40×	3.47×	3,880	Aggressive / lower accuracy
BitMoD-FP4 kernel	**98.10%**	**−0.50**	**1.81×**	**2.61×**	**5,456**	**v3 co-design target**

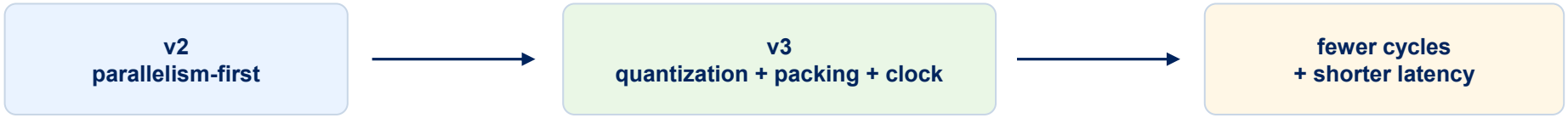
Chosen rationale: FP4 gives acceptable accuracy at 1-bit input, while kernel/coarse grouping avoids the very high group-rescale pressure of fine-grained PG.

v2 → v3: Method-Level Changes

v3 changes the data representation and the system path around the accelerator, not only the layer schedule.

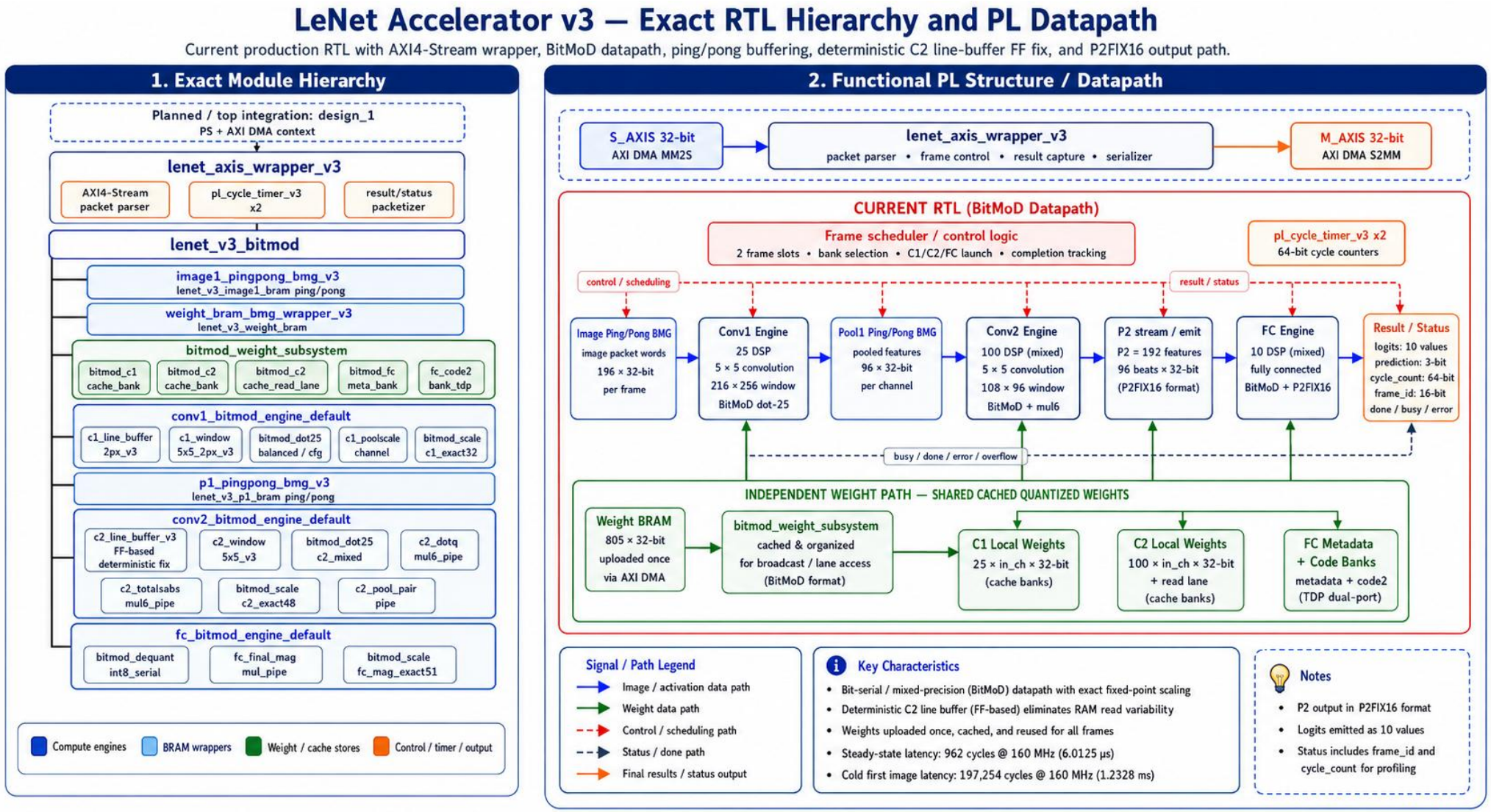
Area	LeNet-v2	LeNet-v3	Purpose
Input image	**8-bit image input**	**1-bit quantized + packed image**	**Reduce PS→PL transfer and image memory**
Weights	**INT8 weights**	**BitMoD FP4 adaptive per-group weights**	**Reduce weight storage and operand workload**
Datapath	Parallel INT8-style engines	Low-bit packed / BitMoD-aware compute path	Make quantization visible to hardware
Clock	125 MHz	160 MHz	Exploit shorter critical path after optimization
Transfer/control	DMA + cache maintenance	Prebuilt TX + bulk cache maintenance + direct MMIO	Lower non-PL overhead
Benchmark policy	D-cache ON timing analysis	Cold included + steady-state split	Separate cold start from steady throughput

Conceptual transition



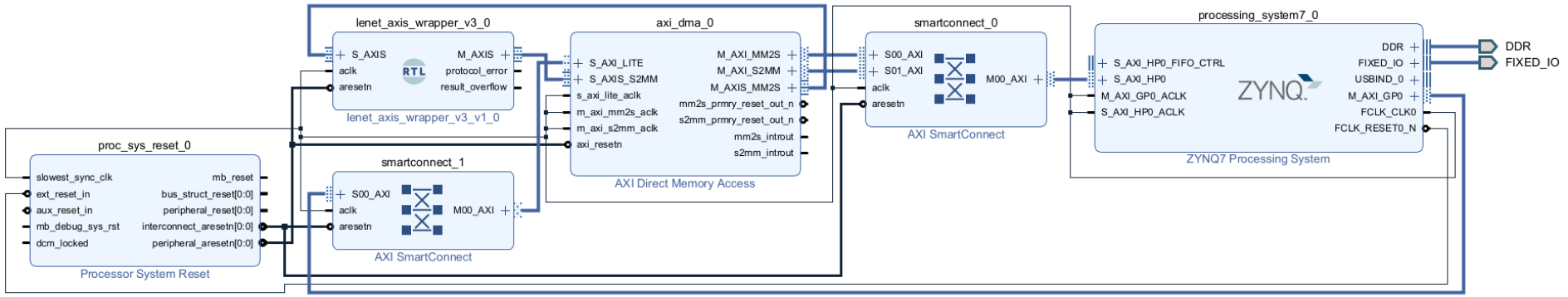
The important presentation point is that v3 makes the low-bit representation an architectural feature, not just an offline model-compression result.

PL Architecture

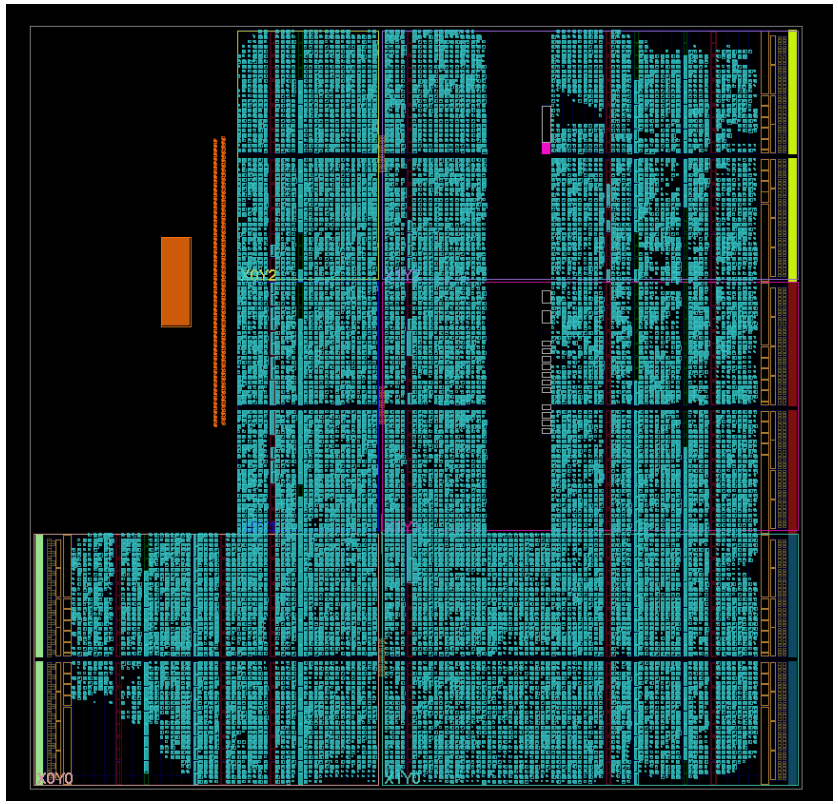


PL Result and Block Design

```
============================================================
LeNet V3 BitMoD PARALLELISM COMPARISON
C2_IC_PAR       = 1
FC_PAR          = 2
DOT25_PIPELINED = 1
Clock period    = 6.667 ns
Clock target    = 149.993 MHz
Images          = 1000
============================================================
[TB] 805 original INT8 weight words loaded.
[QPASS] exact BitMoD cache match
[IMG 0] label=4 pred=4 expected=4 cycles=841253 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FC
[IMG 1] label=9 pred=9 expected=9 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 2] label=9 pred=9 expected=9 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 3] label=7 pred=7 expected=7 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 4] label=1 pred=1 expected=1 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 5] label=1 pred=1 expected=1 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 6] label=9 pred=9 expected=9 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 7] label=0 pred=0 expected=0 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 8] label=7 pred=7 expected=7 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 9] label=8 pred=8 expected=8 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov=
[IMG 99] label=1 pred=1 expected=1 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCov
[IMG 199] label=0 pred=0 expected=0 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 299] label=5 pred=5 expected=5 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 399] label=6 pred=6 expected=6 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 499] label=6 pred=6 expected=6 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 599] label=9 pred=9 expected=9 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 699] label=8 pred=8 expected=8 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 799] label=1 pred=1 expected=1 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 899] label=9 pred=9 expected=9 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
[IMG 999] label=0 pred=0 expected=0 cycles=961 | C1=415 C2=830 FC=959 C1/C2ov=415 C2/FCo
============================================================
FUNCTIONAL RESULT
Prediction golden mismatches : 0 / 1000
Classification correct       : 983 / 1000
Exact cache mismatches       : 0
Exact logit mismatches (10)  : 0
Overflow flag                : 0
------------------------------------------------------------
PERFORMANCE RESULT
Cold frame cycles            : 841253
Steady frame cycles          : 961
Clock target                 : 149.993 MHz
Steady latency               : 6.407 us
Theoretical frame throughput : 156079.6 image/s
V2 reference                 : 1504 cycles
Cycle speedup vs V2          : 1.565x
------------------------------------------------------------
PREDICTION FUNCTIONAL PASS
BIT-TRUE EXACT PASS
============================================================
```



Block Design Analysis

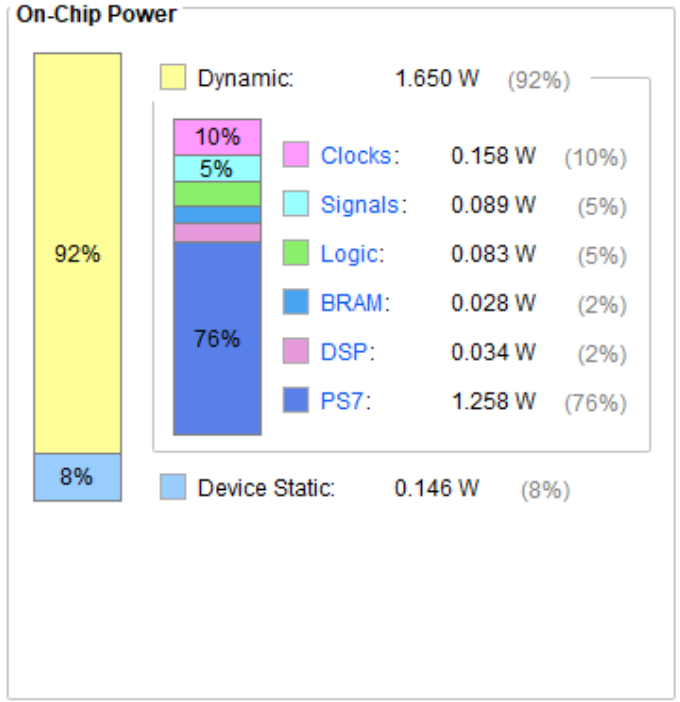


Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**1.796 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**45.7°C**
Thermal Margin:	39.3°C (3.3 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity



Name	Constraints	Status	WNS	WHS	THS	TNS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	BRAM	FF	URAM	DSP	Start	Elapsed
✓ **synth_1** (active)	**constrs_1**	**synth_design Complete!**												**0**	**0**	**0**	**0**	**0**	**8/18/26, 7:12 AM**	**00:00:36**
✓ impl_1	constrs_1	write_bitstream Complete!	0.003	0.016	0.000	0.000		0.000	1.796	0				35345	16	45869	0	167	8/18/26, 7:13 AM	00:10:11

Evaluation

Overall Benchmark Summary

All three accelerators were evaluated on 1,000 test images. The table separates model quality from PL computation and PS+PL system latency.

Metric	Baseline	LeNet-v2	LeNet-v3
Accuracy	98.40%	98.60%	98.30%
Correct / Wrong	984 / 16	986 / 14	983 / 17
PL steady cycles / image	434,330	1,504	962
PL latency / image	**8,686.60 µs**	**12.032 µs**	**6.0125 µs**
E2E latency / image	**8,760.47 µs**	**22.589 µs**	**9.386 µs steady**
Pure PL throughput	≈115 img/s	83,112 img/s	166,320 img/s
Key method	single sequential path	parallel engines + buffering	1-bit input + BitMoD FP4 + 160 MHz

PL cycle reduction

99.78%

Baseline → v3

v2 → v3 PL

2.001×

steady latency speedup

Accuracy cost

−0.30 pp

v3 vs v2

E2E throughput

106k img/s

v3 steady state

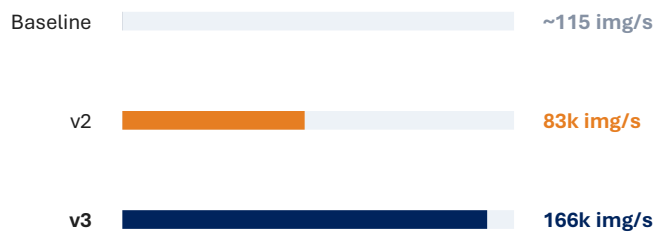
Accuracy–Performance Trade-off

v3 intentionally trades only a small accuracy difference for a large reduction in PL latency and cycle count.

Baseline	LeNet-v2	LeNet-v3	v3 penalty
98.40%	98.60%	98.30%	−0.30 pp
reference board run	best accuracy	low-bit final design	vs v2 accuracy

PL latency per image

PL throughput



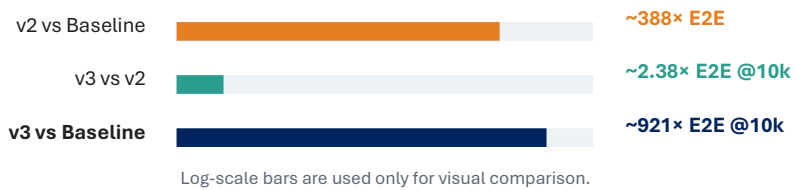
Evaluation takeaway: v3 keeps the classifier within the same accuracy band while cutting the steady PL latency by 2.001× compared with v2.

If the Benchmark Runs on 10,000 Images

This is an extrapolation from the measured 1,000-image benchmark. It assumes the same accuracy rate and similar steady-state timing behavior.

Projected result	Baseline	LeNet-v2	LeNet-v3
Expected correct	9,840	9,860	9,830
Expected wrong	160	140	170
Projected E2E total	**87.605 s**	**225.894 ms**	**95.086 ms**
Projected PL total	**86.866 s**	**≈120.320 ms**	**61.352 ms**
Average E2E / image	**8.760 ms**	**22.589 µs**	**9.509 µs**
Main note	accuracy reference	high accuracy, higher overhead	fastest system-level projection

Speedup overview



Log-scale bars are used only for visual comparison.

Projection method

v3 uses one cold-start image plus 9,999 steady-state images. That is why its 10k average improves compared with the 1,000-image cold-included average.

Projected 10k result: v3 would process 10,000 images in about 95 ms while still keeping an expected 98.30% accuracy.

Evaluation Conclusion

The three versions show a clear design evolution: correctness → parallelism → quantized hardware co-design.

Version	Main purpose	Strength	Remaining issue
Baseline	Verify the LeNet accelerator path	Stable PS → PL → PS behavior	Very large PL cycle count
LeNet-v2	Remove major computation bottlenecks	High accuracy and large PL speedup	PS+PL overhead still visible
LeNet-v3	**Make low-bit representation visible to hardware**	**Fastest PL and E2E result with small accuracy loss**	**Needs larger-scale validation and power/resource study**

One-sentence result

LeNet-v3 demonstrates that 1-bit input and BitMoD-style low-bit weights can preserve near-baseline accuracy while substantially reducing measured PL latency.

How to present it

Do not claim the 10k projection as measured data. Present it as an expected result if the same accuracy rate and steady-state timing hold.

Q&A