TOSHIBA

MPSoC 2019

DNN-Accelerated Multi-core SoC for Automotive Applications

Takashi Miyamori
Toshiba Electronic Devices & Storage, Kawasaki, Japan

Outline

1 Background

2 Architecture of the SoC

3 Functional Safety

4 Implementation Results

5 Conclusion

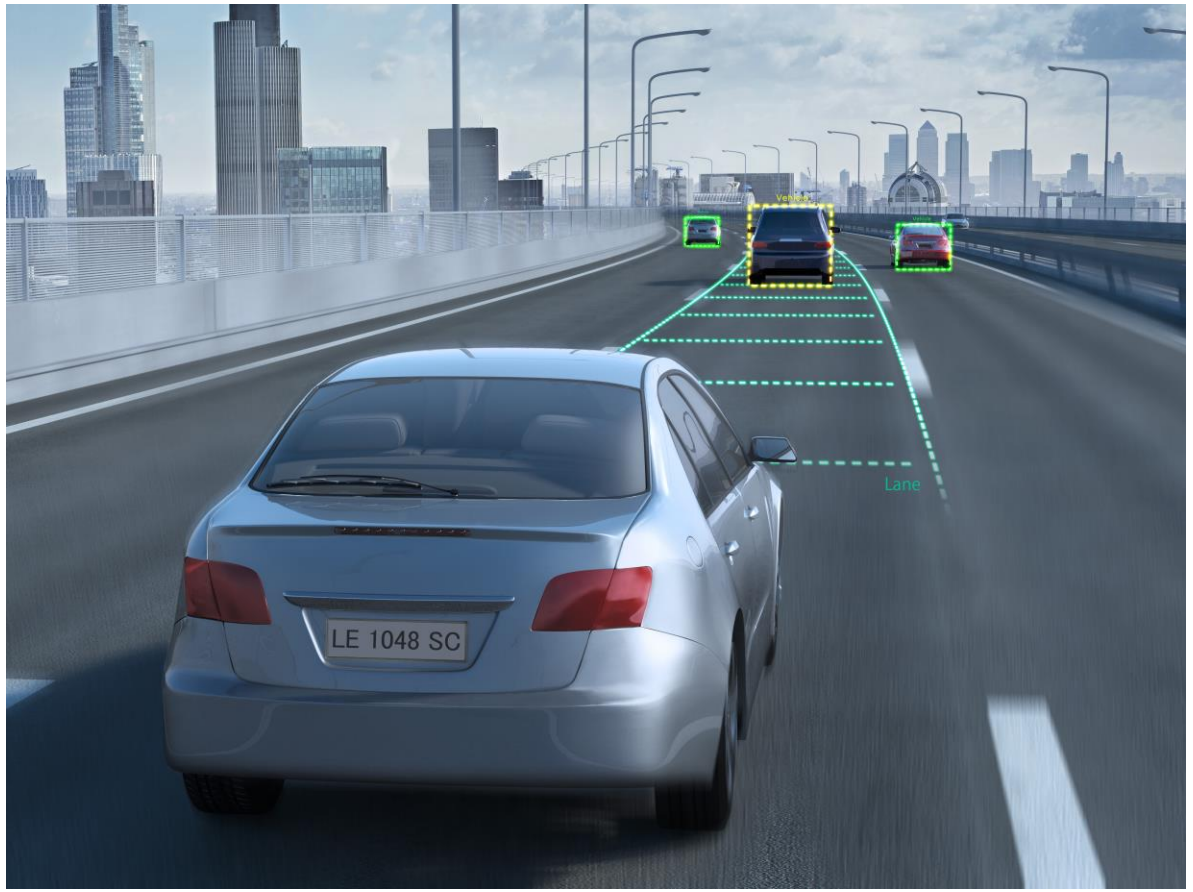
Outline

Demand for ADAS and Automated Driving System (ADS)

ADAS / ADS systems can reduce serious traffic accidents

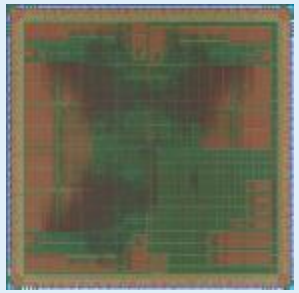
Number of traffic accidents in US
- 350 per billion km

Number of deaths in US
- 7 per billion km

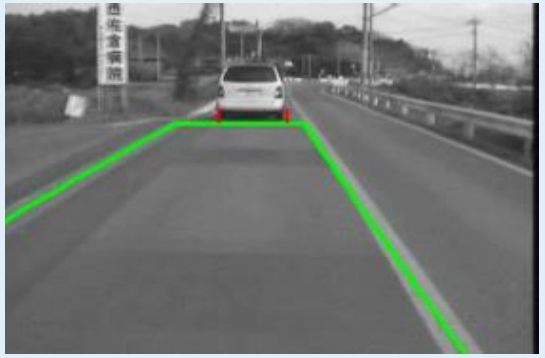


Toshiba's SoCs for ADAS

1st Gen. T5BG3XBG [CICC 2004]



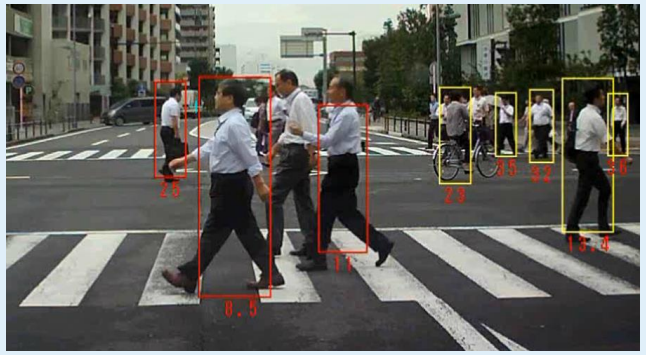
0.13μm CMOS
1W@1.5V
Core x3 + HWA x1



Lane Detection

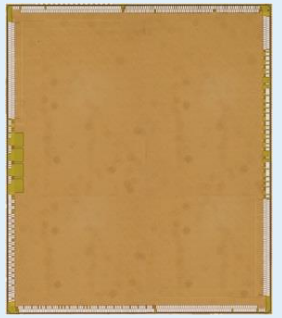
2nd Gen. MPV7506XBG [ISSCC 2012/MPSoC 2012]

40nm CMOS
1W@1.1V
Core x4 + HWA x6
464GOPS
617GOPS/W

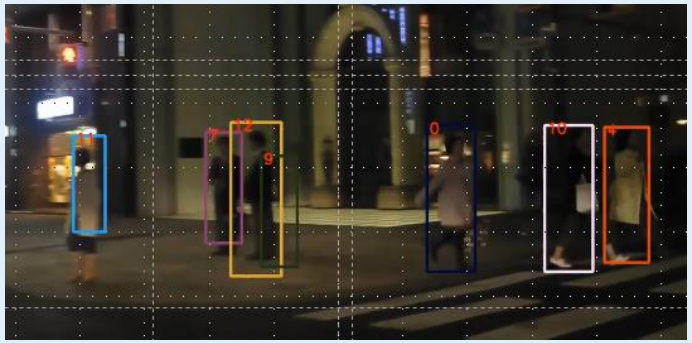


Pedestrian Detection(PD)

4th Gen. MPV7608XBG [ISSCC 2015/MPSoC2015]



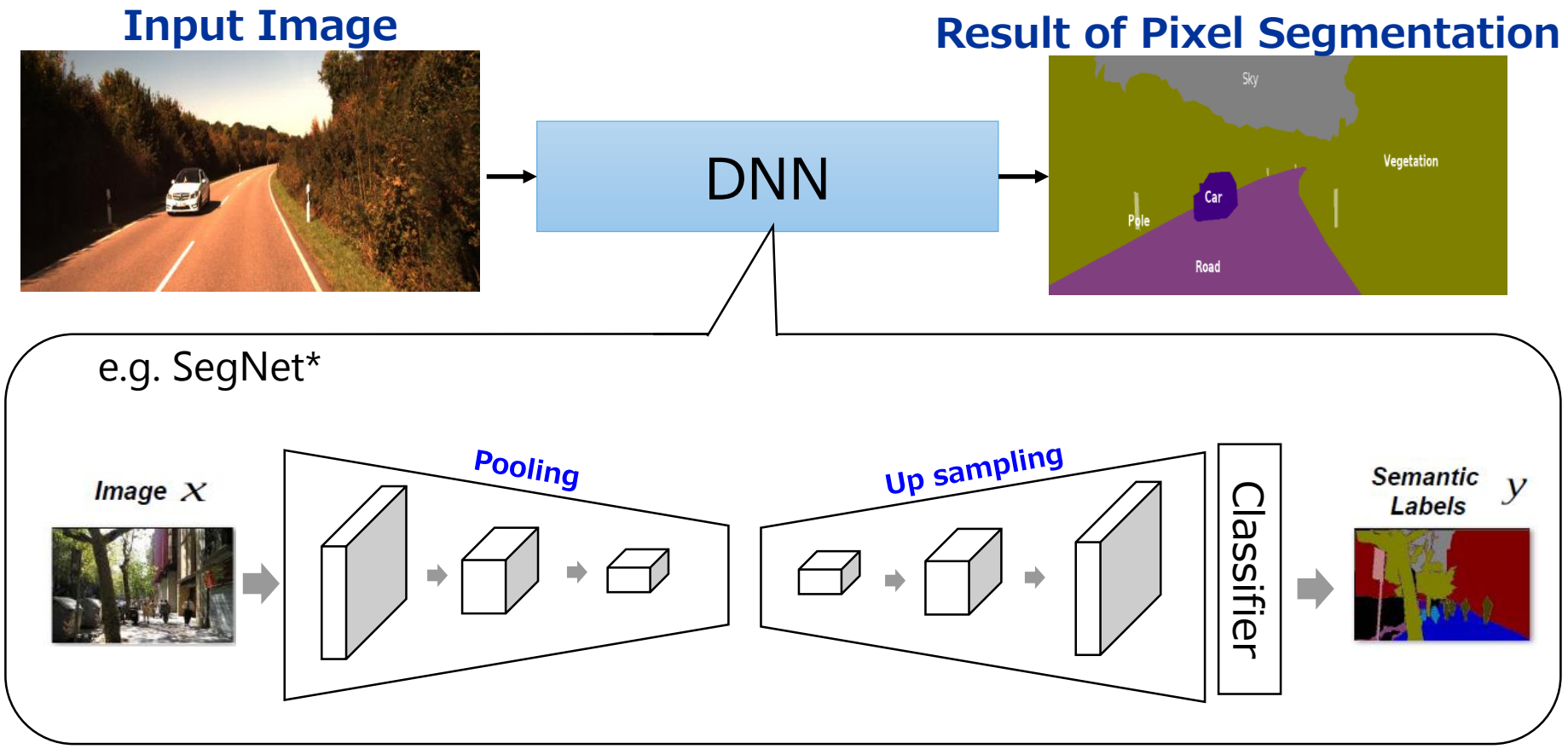
40nm CMOS
3W@1.1V
Core x8 + HWA x14
1900GOPS
564GOPS/W



PD at night-time

Semantic Segmentation by DNN

- **Classify objects in each pixel**



*) Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)

Outline

1 Background

2 **Architecture of the SoC**

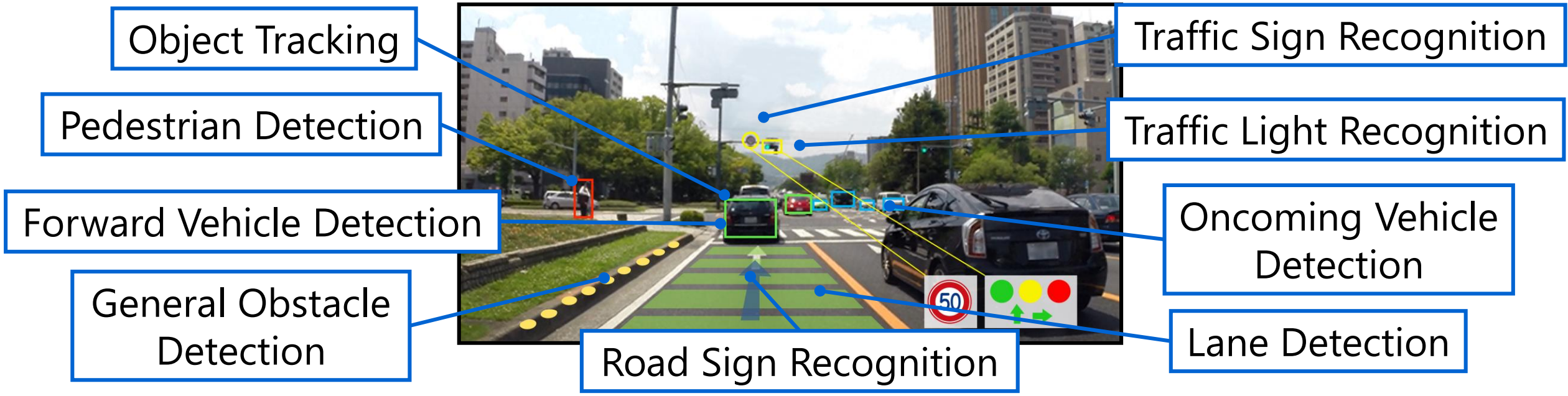
3 Functional Safety

4 Implementation Results

5 Conclusion

Requirement #1 : High Performance with Low Power Consumption

- SoCs for ADAS perform **many image recognition applications**
- To improve safety, more computing power is needed
  - More accurate, more object detection & tracking, and more detail
- Low power need keep performance stable in vehicle



Design Concept of Image Recognition SoCs

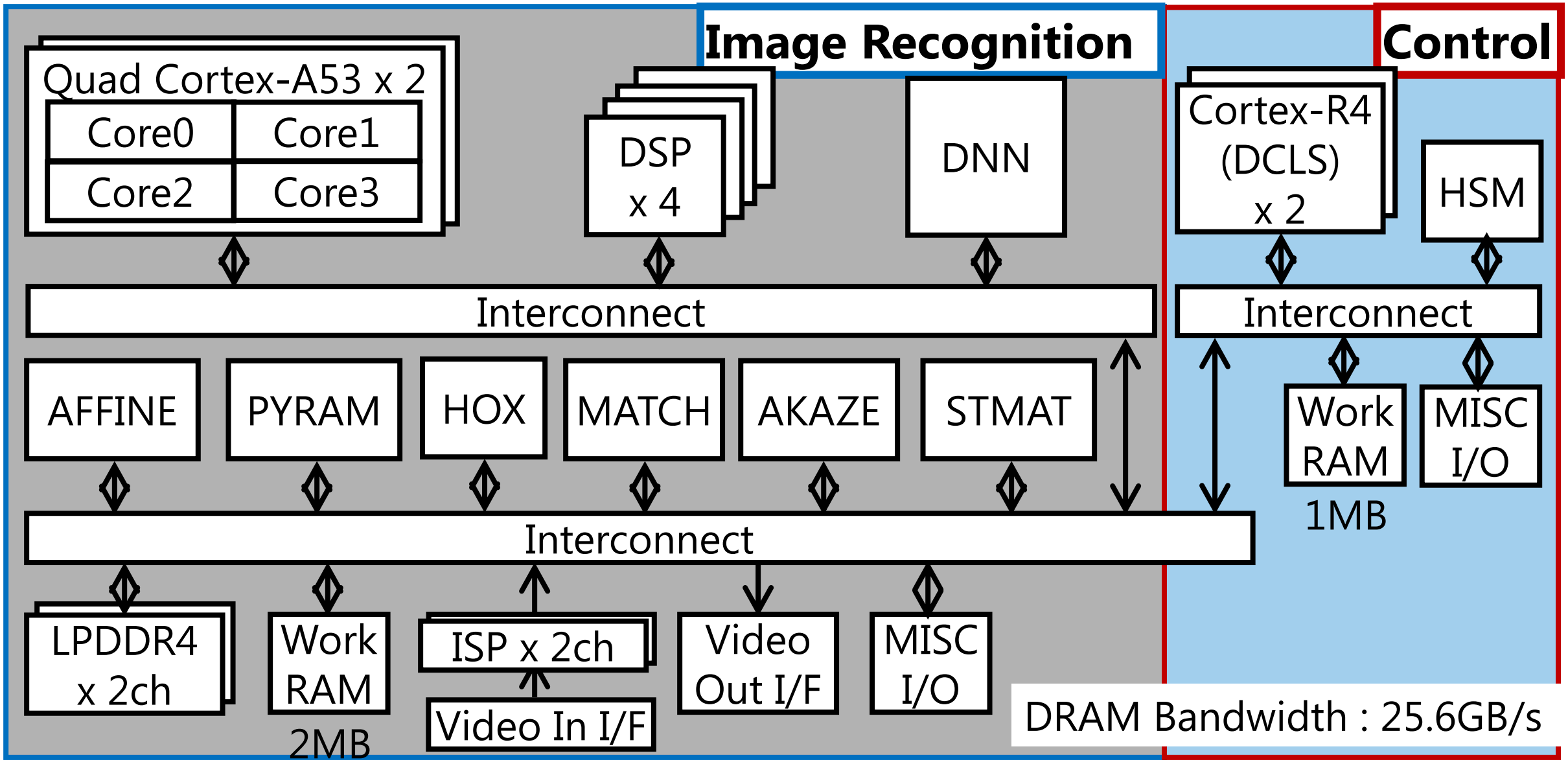
- Requirements:
  - High performance with low power consumption
  - High accuracy of object recognition

Heterogeneous Multi-core Architecture

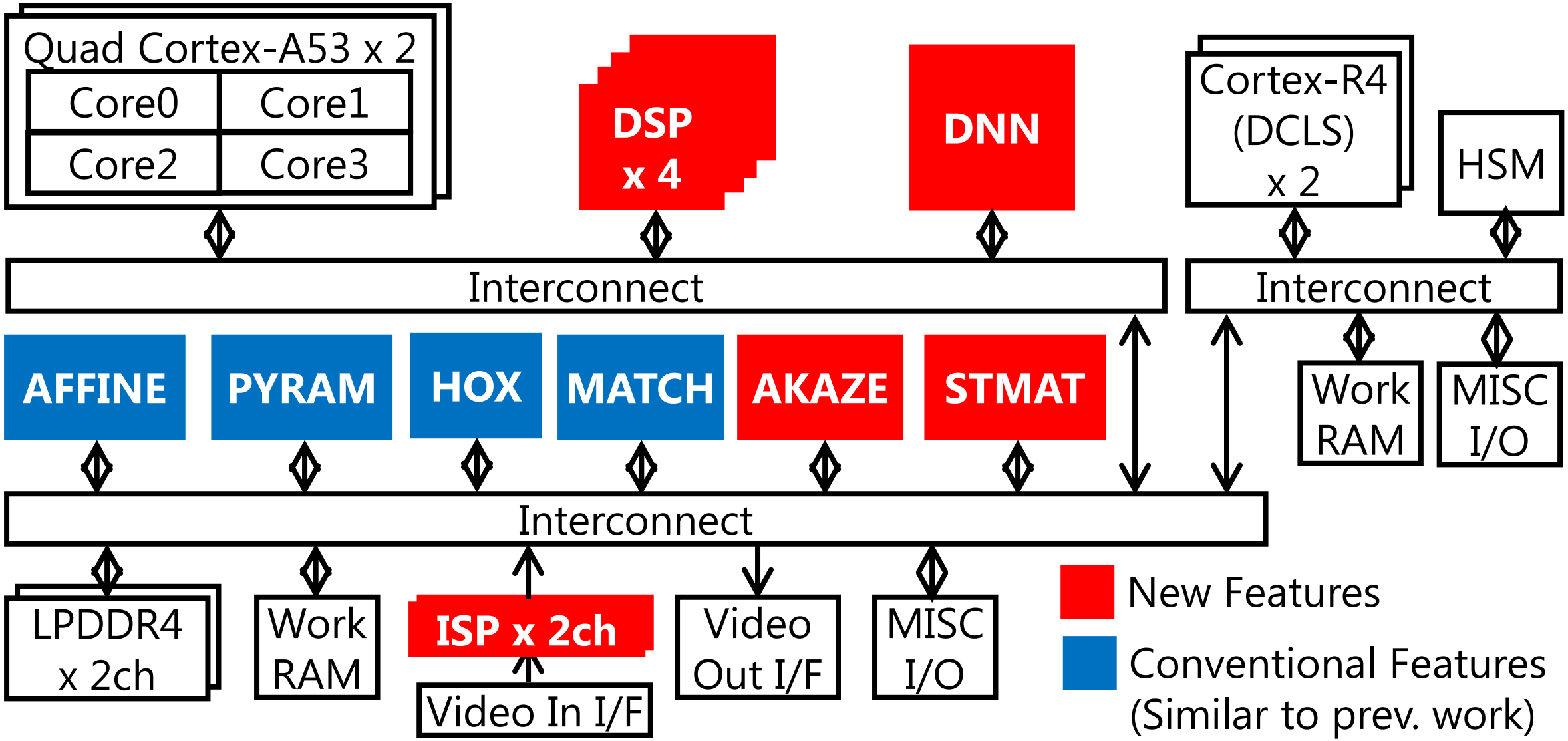
- Energy efficient multi cores and Multi DSPs
- Hardware accelerators
  - Performance bottlenecks and frequently used tasks
  - High accurate image recognition features

Adopted "Highly parallelized" approach rather than "High clock frequency" approach

Architecture of the SoC



Features Updated from Our Previous Work



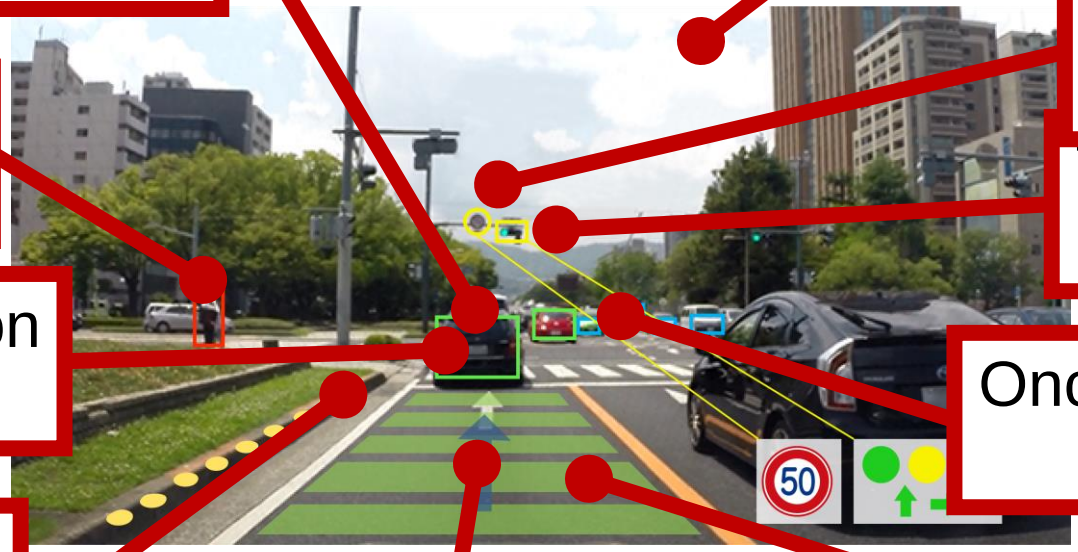
Corresponding Accelerators

Object Tracking
MATCH + CPU →
AKAZE + MATCH + CPU

Pedestrian Detection
HOX → HOX + DNN

Forward Vehicle Detection
HOX → HOX + DNN

General Obstacle Detection
MATCH(SfM) → STMAT



Road Sign Recognition
CPU + Filter → DNN

ISP Higher Resolution
Higher Dynamic Range

Traffic Sign Recognition
HOX → HOX + DNN

Traffic Light Recognition
HOX → HOX + DNN

Oncoming Vehicle Detection
HOX → HOX + DNN

Lane Detection
CPU + Filter → DNN

Hardware Accelerators to Improve Safety

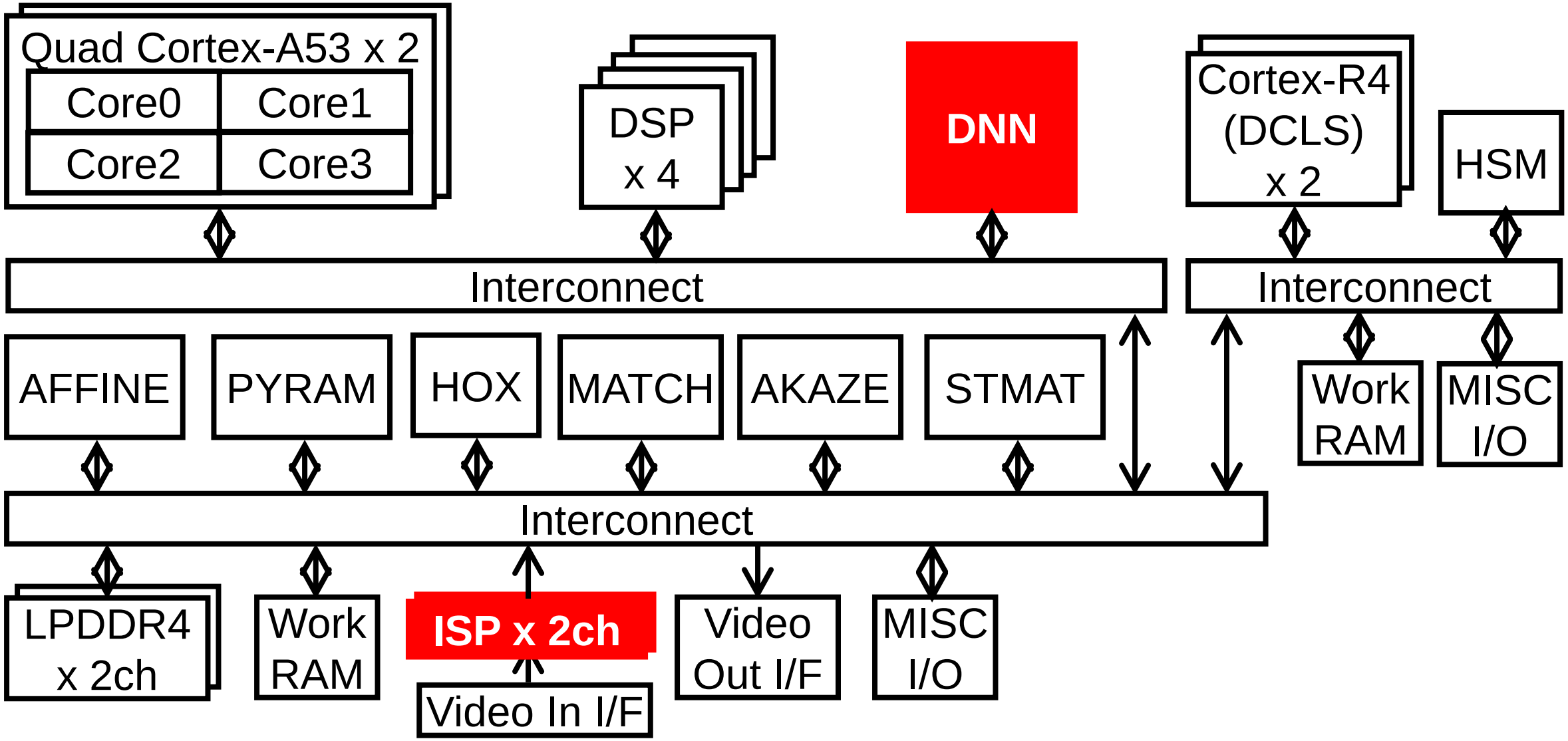
- **More accurate / More object detection & tracking**
  - Introducing advanced algorithm for detection & tracking
  - Increase computing power

➡ **Accelerators with advanced algorithm**
  - DNN for object detection & semantic segmentation
  - STMAT for depth map generation
  - AKAZE for tracking a lot of objects

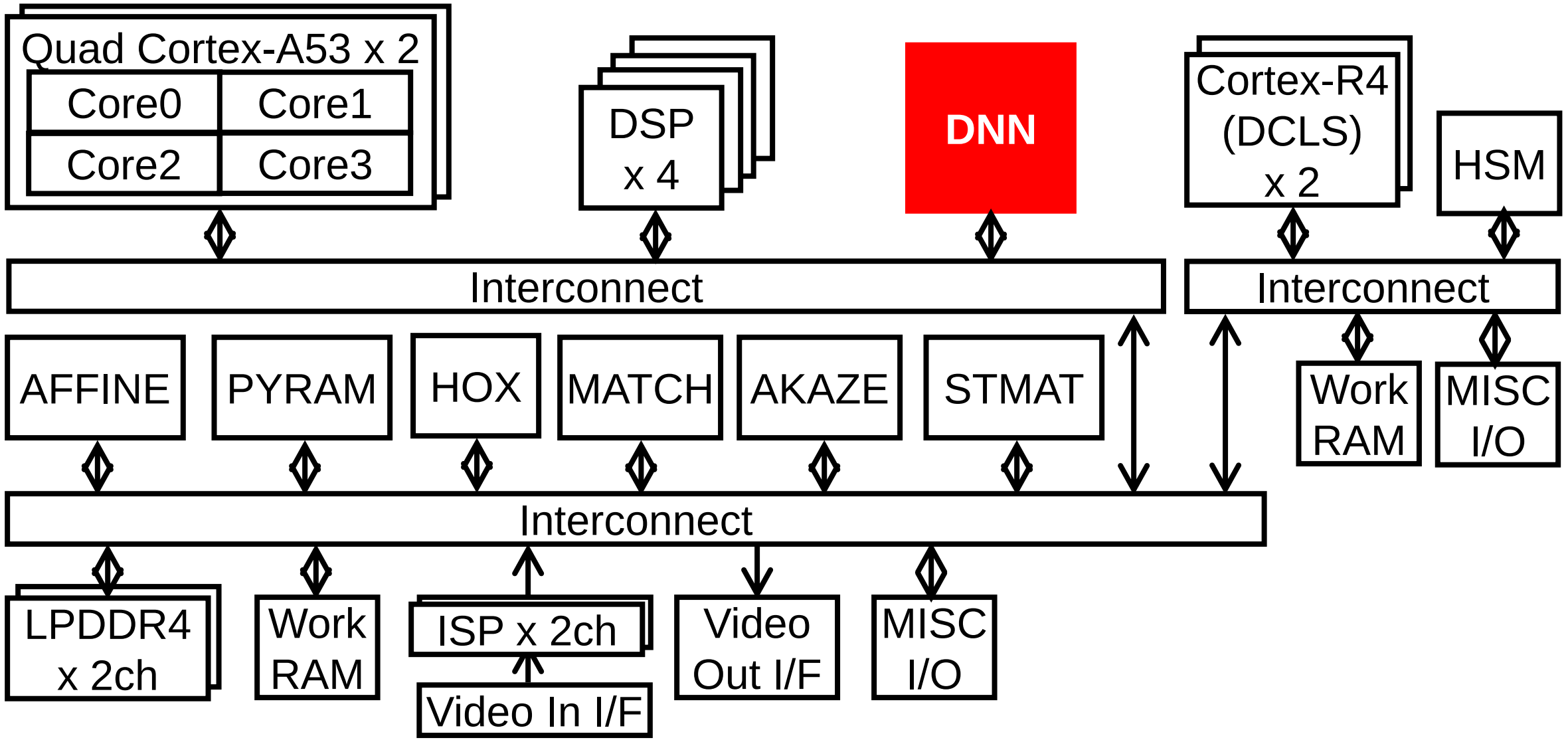
- **More detail**
  - Support high resolution and high dynamic range

➡ **Image Signal Processor (ISP)**

Hardware Accelerators (HWAs)



DNN Accelerator

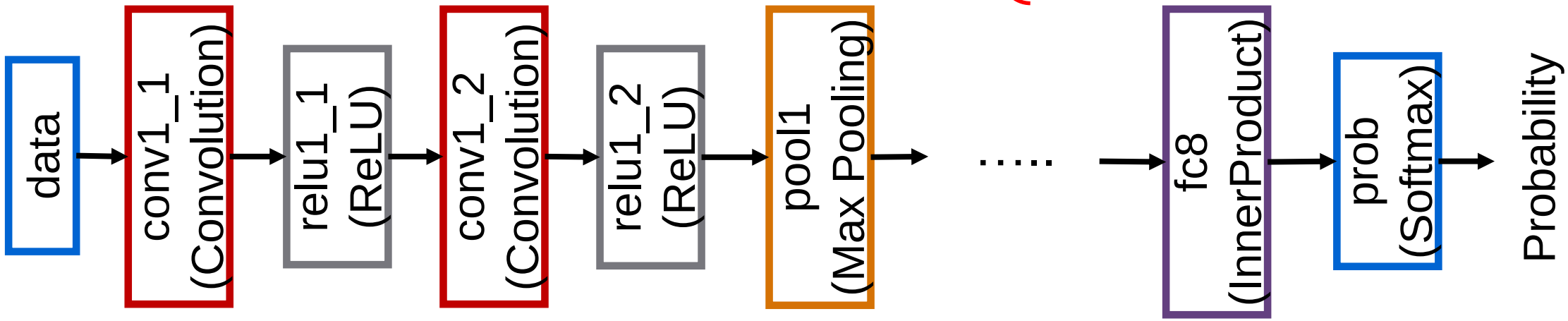


Deep Neural Network (DNN)

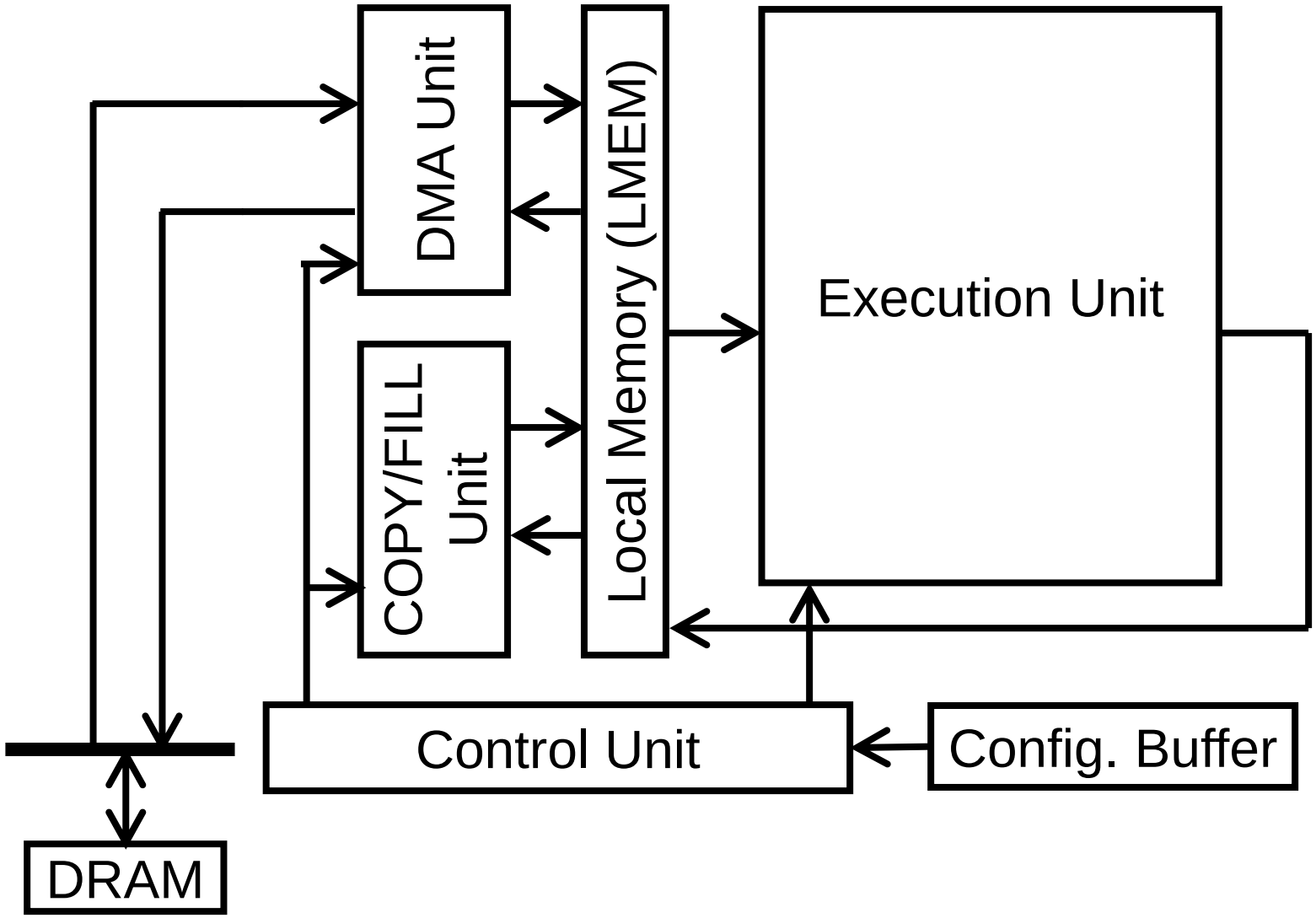
- One technology of machine learning
  - High accuracy for Computer Vision
  - Operations require a lot of MAC calculations and large bandwidth

VGG-16

15.5G MAC operations
138M weight parameters
224x224x3ch input features



Architecture of DNN Accelerator



Features for Low Power Consumption

Execution Unit

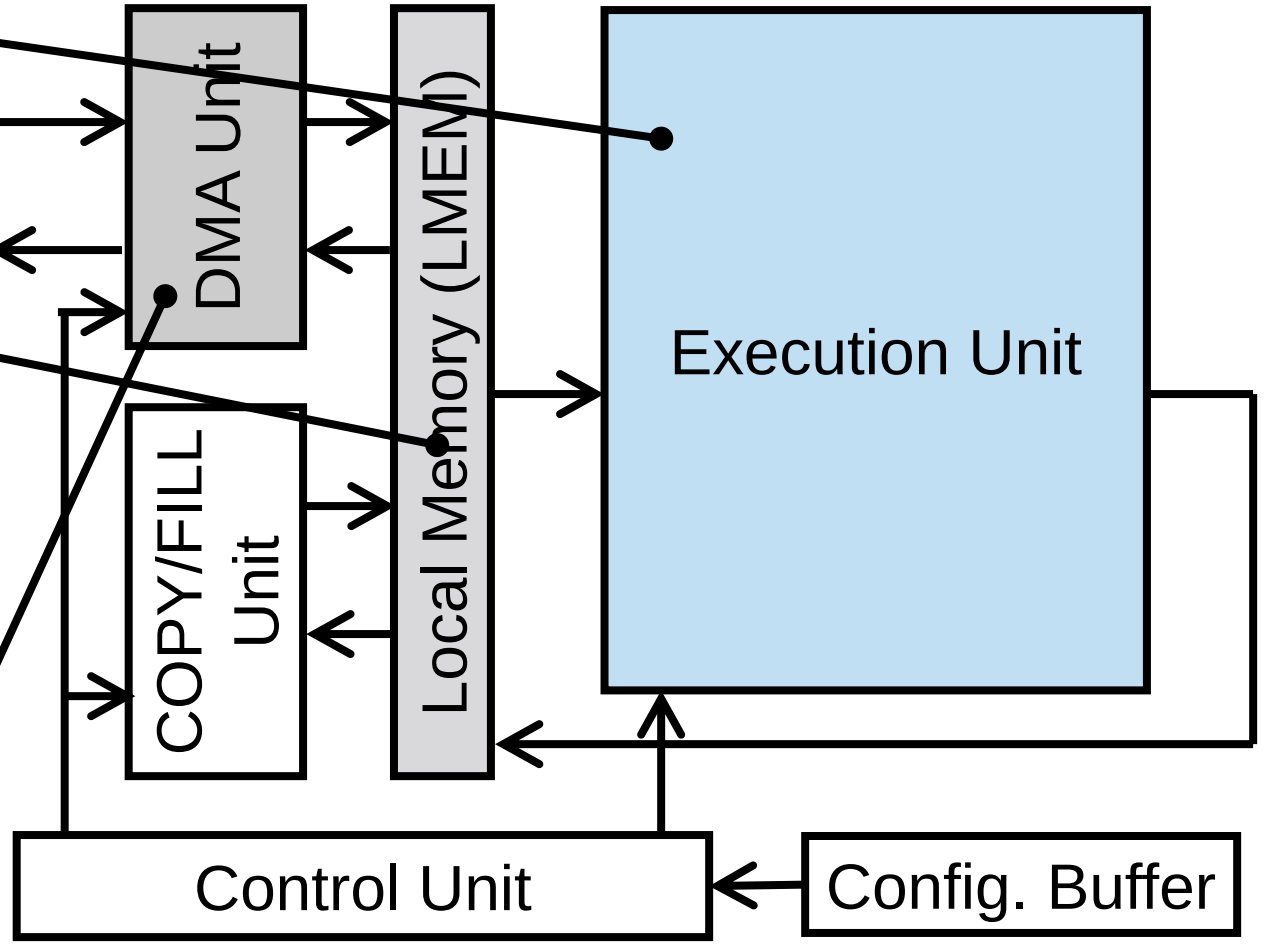
- 4 x 256 FP16 MAC units
- Pipelined layer structure

Local Memory

- 1MB capacity
- Share/reuse load data & intermediate data

DMA Unit

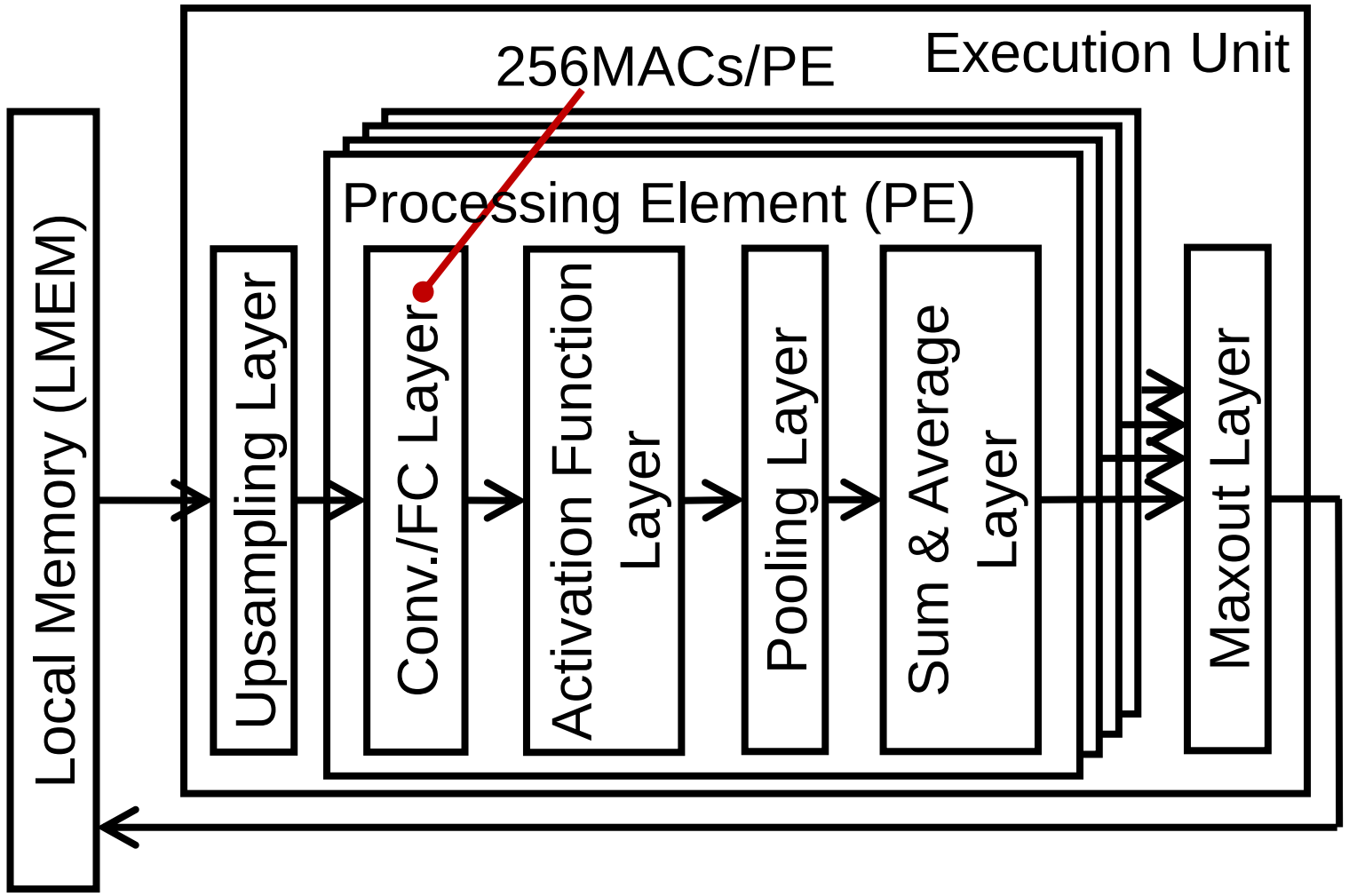
- Decompression unit to load compressed weight parameters (zero run-length / quantization)



Structure of Execution Unit

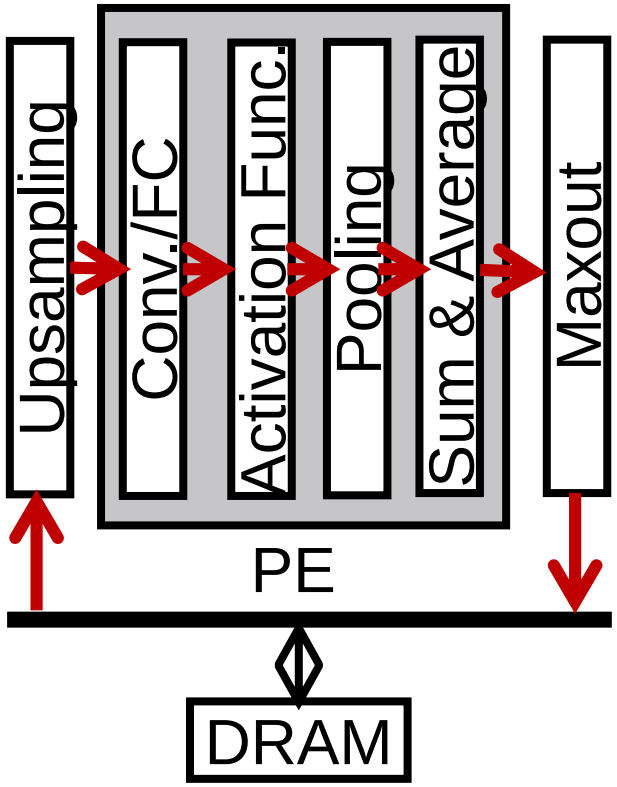
Pipelined layer structure can perform a series of DNN operations in one read and write local memory access

4 PEs share same input feature maps and simultaneously compute different channels for output feature maps

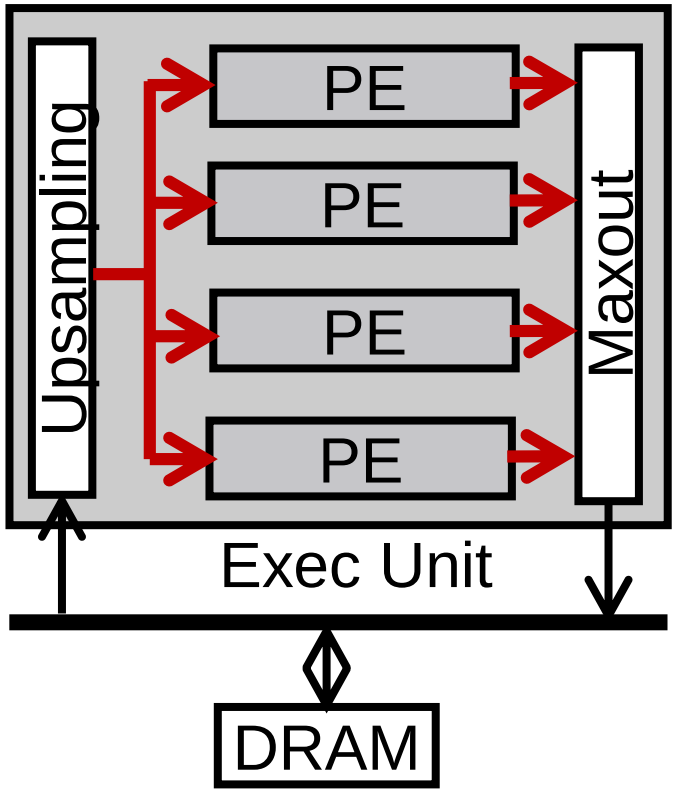


DRAM Bandwidth Reduction

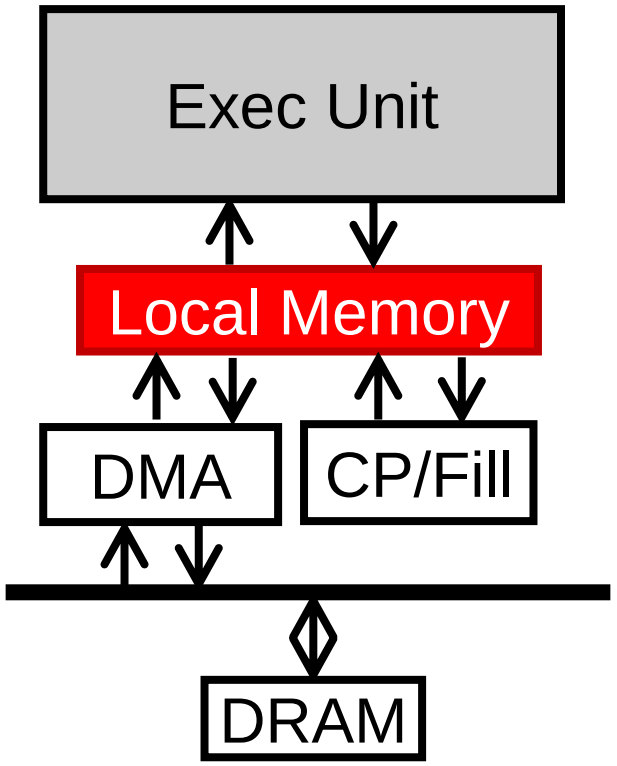
Pipelined Structure
(w/o Local Memory)
48GB/s



Input Feature Map Sharing
(w/o Local Memory)
17GB/s



Proposed
Local Memory
w/ Optimization by Tool
3.7GB/s



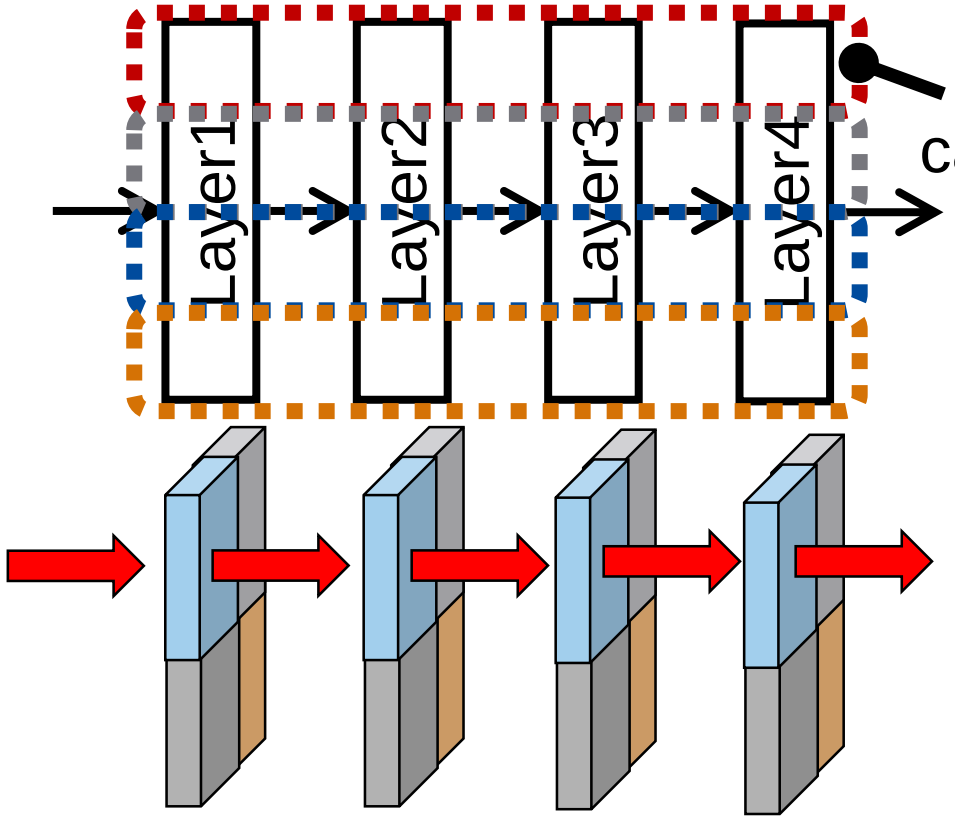
Use case : VGG-16

DNN Optimization by Tool

Two methodologies for DNN optimization

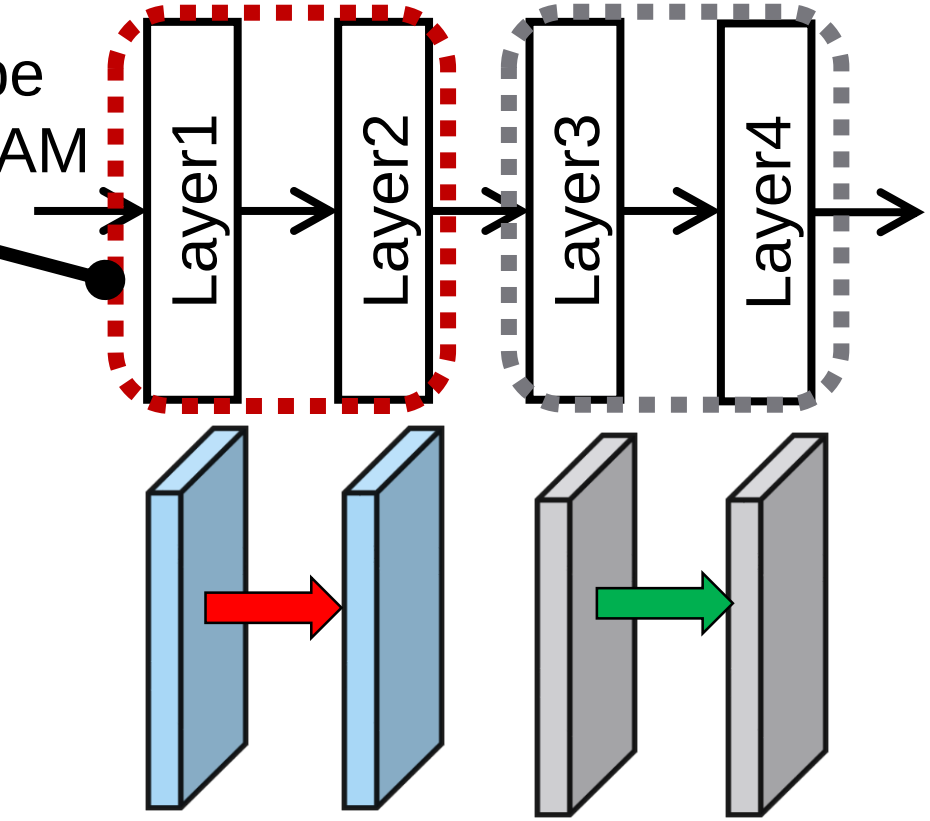
More MAC/LMEM utilization and Less DRAM access

Spatial partitioning



Each block can be calculated w/o DRAM accesses

Layer partitioning



Both of them can be applied simultaneously

Image Signal Processor (ISP)

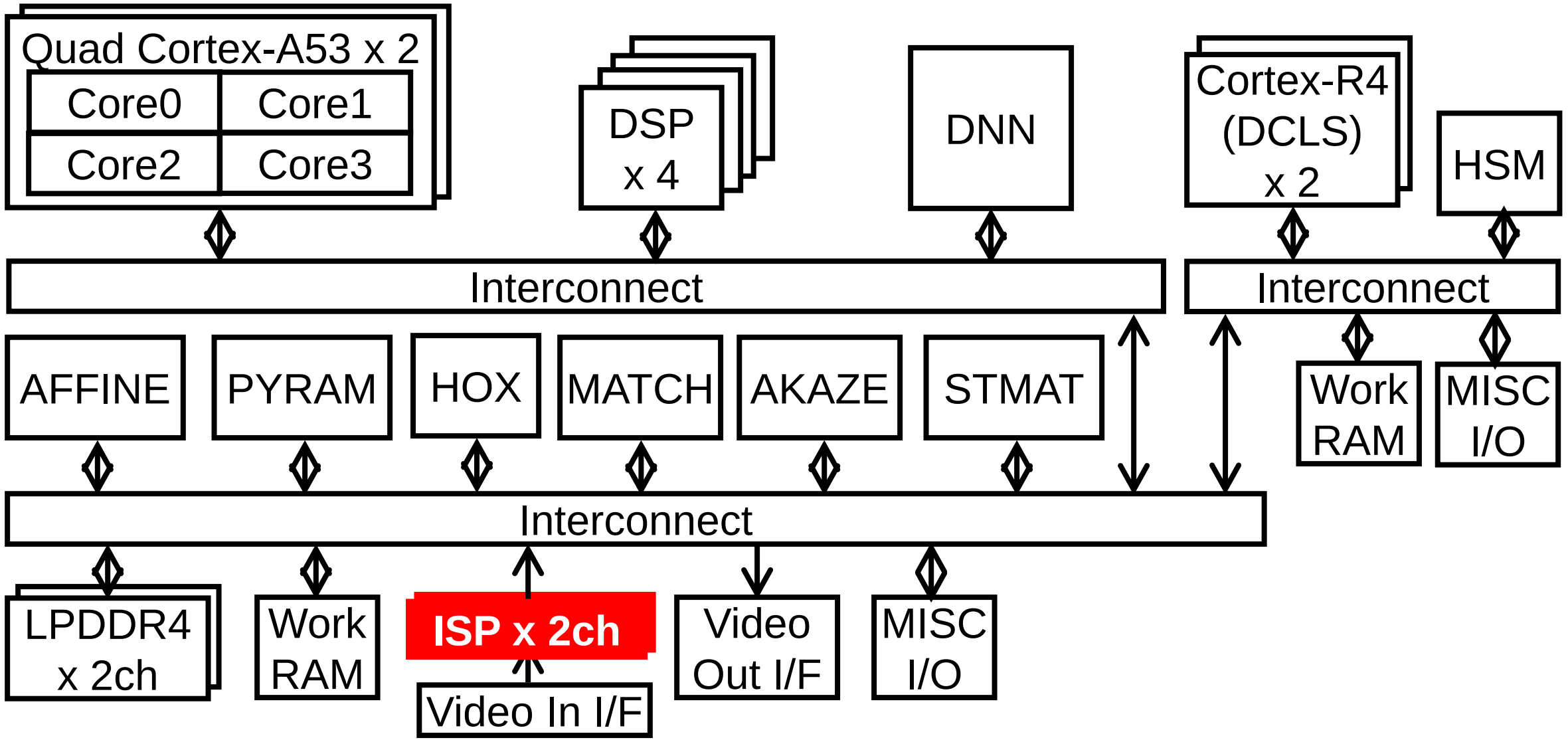
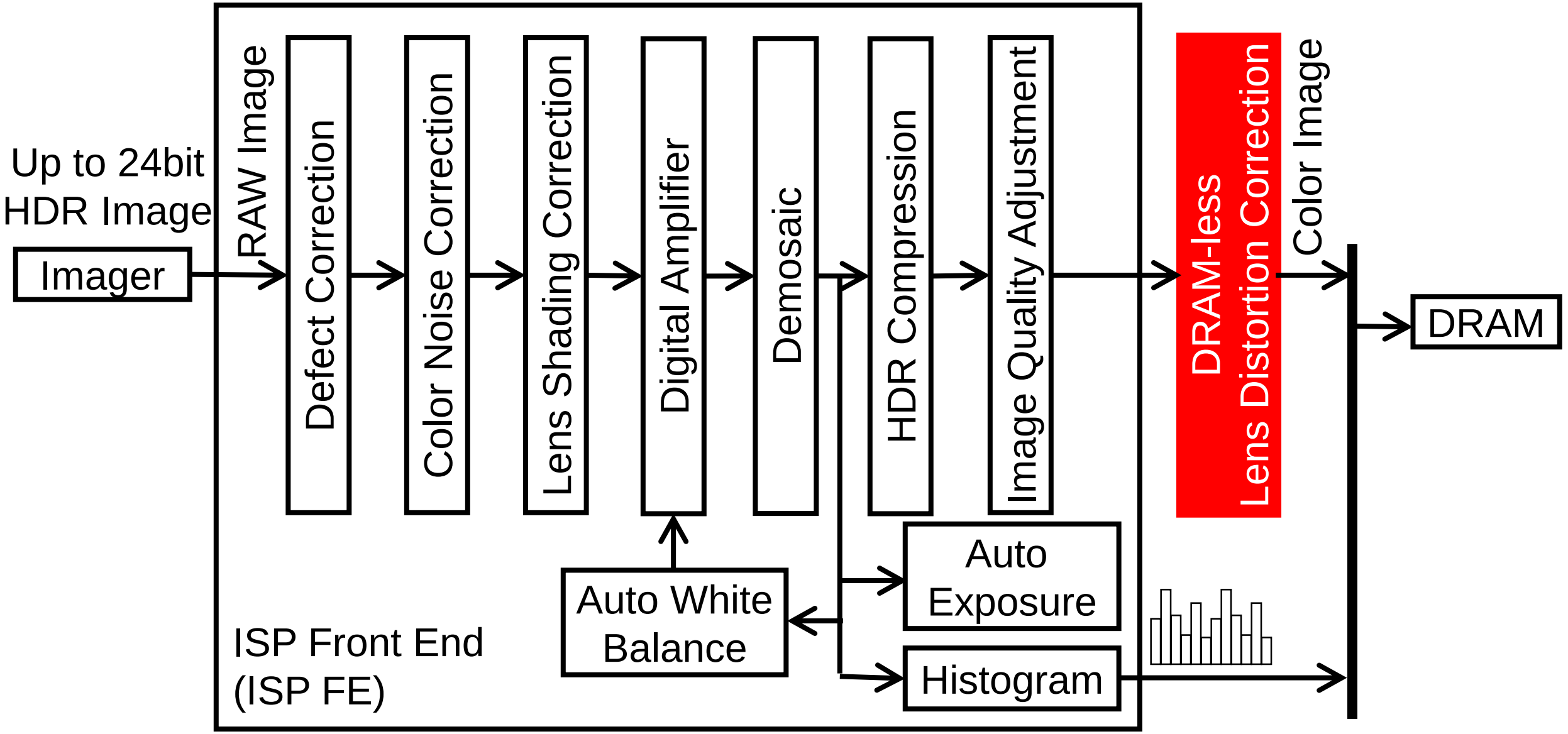


Image Signal Processor (ISP)



Lens Distortion Correction (LDC) Unit

Conventional (w/ DRAM)

LDC cannot connect ISP FE directly

Tiling

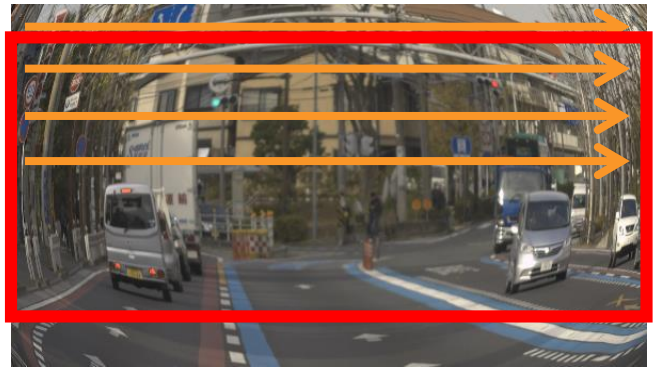
LDC

Tiling

Proposed (w/o DRAM)

LDC can connect ISP FE directly

Raster



LDC

Includes Larger Buffers

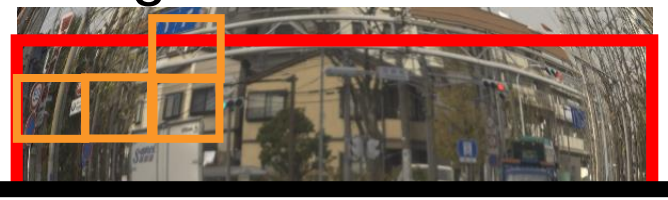
Random

Lens Distortion Correction (LDC) Unit

Conventional (w/ DRAM)

LDC cannot connect ISP FE directly

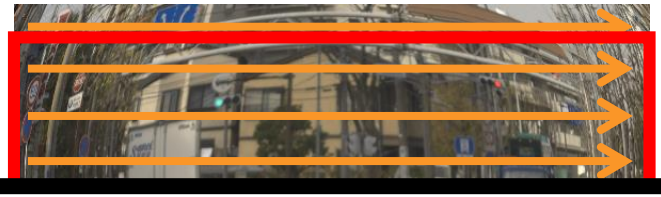
Tiling



Proposed (w/o DRAM)

LDC can connect ISP FE directly

Raster



Proposed LDC need not load input image from DRAM and can reduce bandwidth by 5.14GB/s

(Use Case: ISP 2 ch. / 2880 x 1860 RGB color image @ 40fps)

Tiling

Random

Larger Buffers

Performance Evaluation

	Performance [GOPS]	Power [mW]	Efficiency [GOPS/W]
TOTAL	**20,537**	**9,776**	**2,100.8**
ARM CA53	256	1,286	199.1
DSP	1,024	1,377	743.6
DNN	1,597	1,593	1,002.8
ISP	8,880	1,579	5,623.8
STMAT	3,480	419	8,305.5
AKAZE	2,681	726	3,692.6
MATCH	1,175	254	4,627.6
PYRAM	396	130	3,046.2
AFFINE	204	281	726.0
HOX	842	1,696	496.7
ARM CR4	1	435	2.8

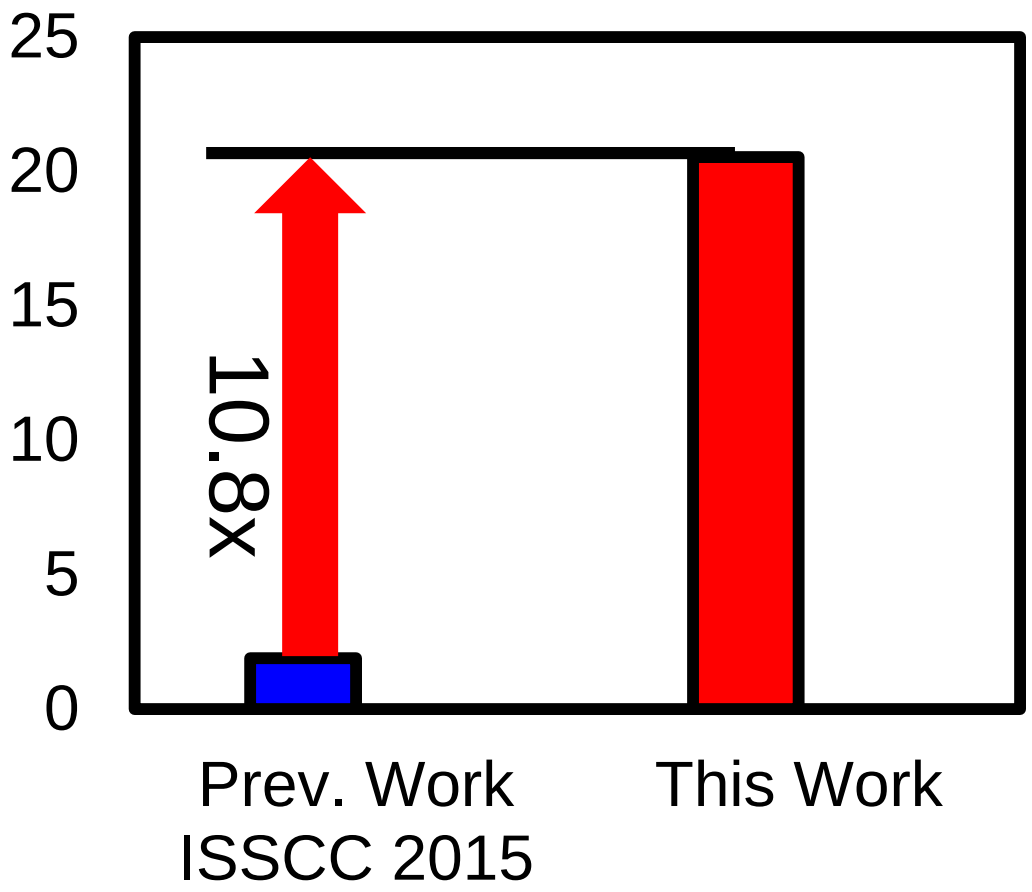
Condition

- Vdd=0.8V
- Process=center
- Temp=25°C

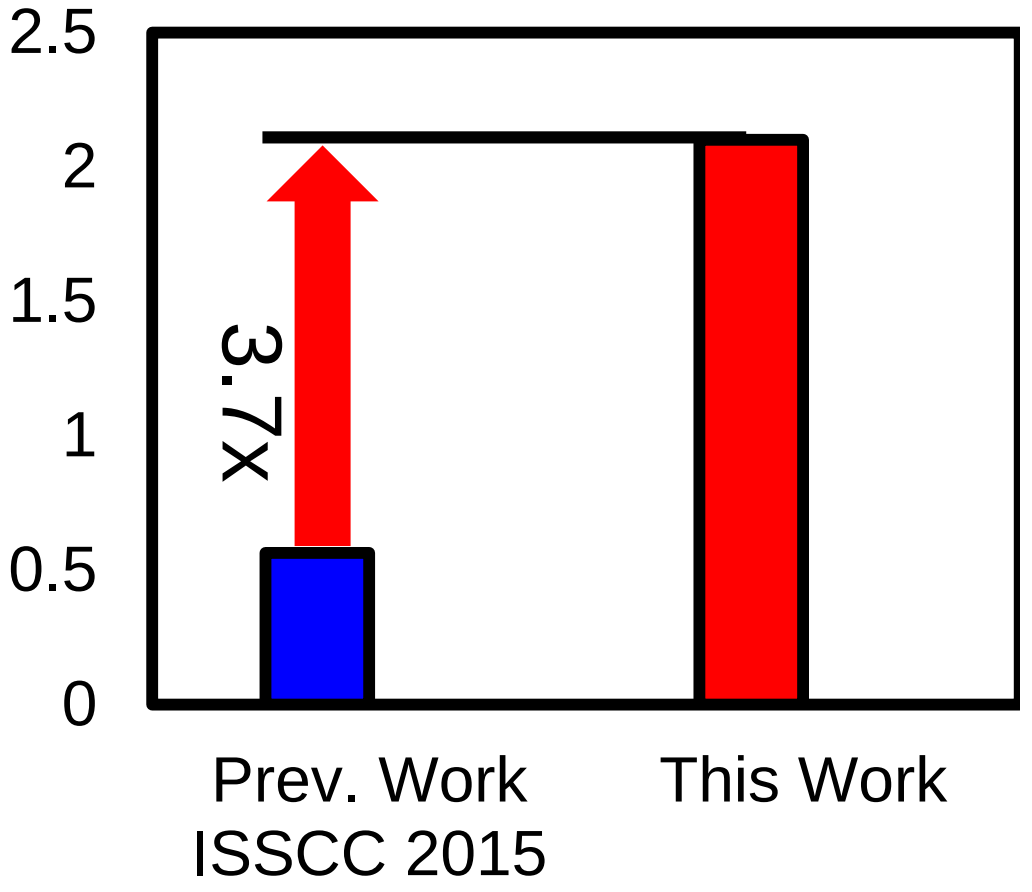
Note: 1 operation of OPS means 1 arithmetic or logical operation

e.g. 1 MAC operation is 2 operations (MUL + ADD)

Performance Comparison : Total Performance

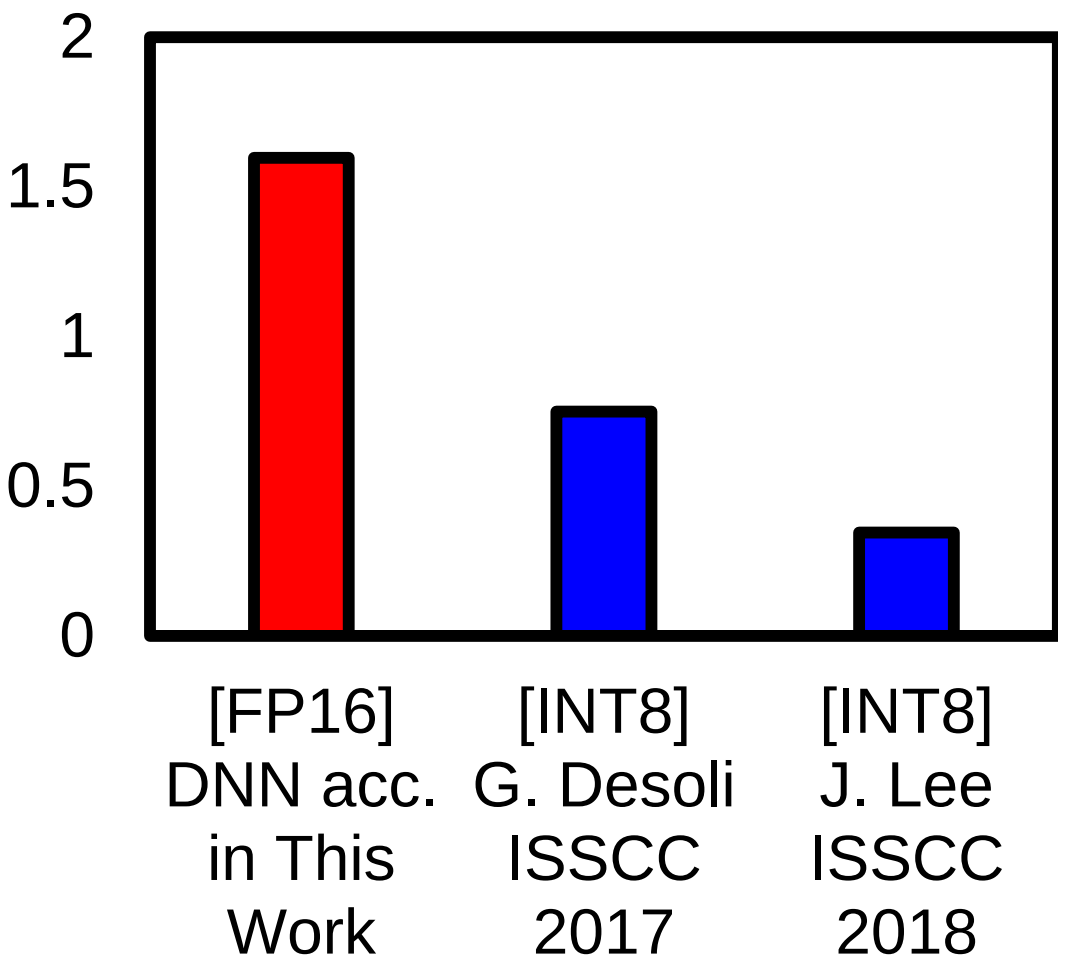


Performance[TOPS]

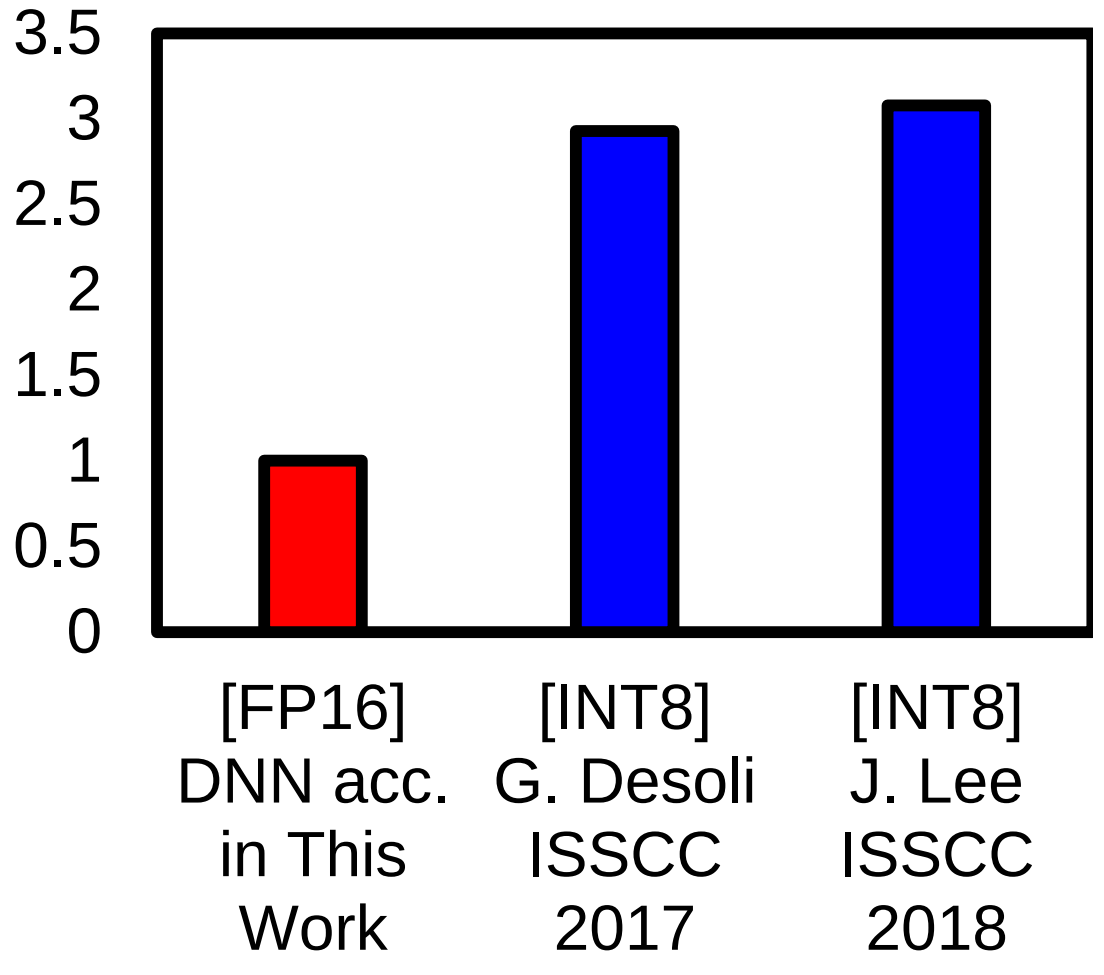


Efficiency [TOPS/W]

Performance Comparison: DNN Performance



Performance [TOPS]



Efficiency (Right Axis) [TOPS/W]

Outline

1 Background

2 Architecture of the SoC

3 **Functional Safety**

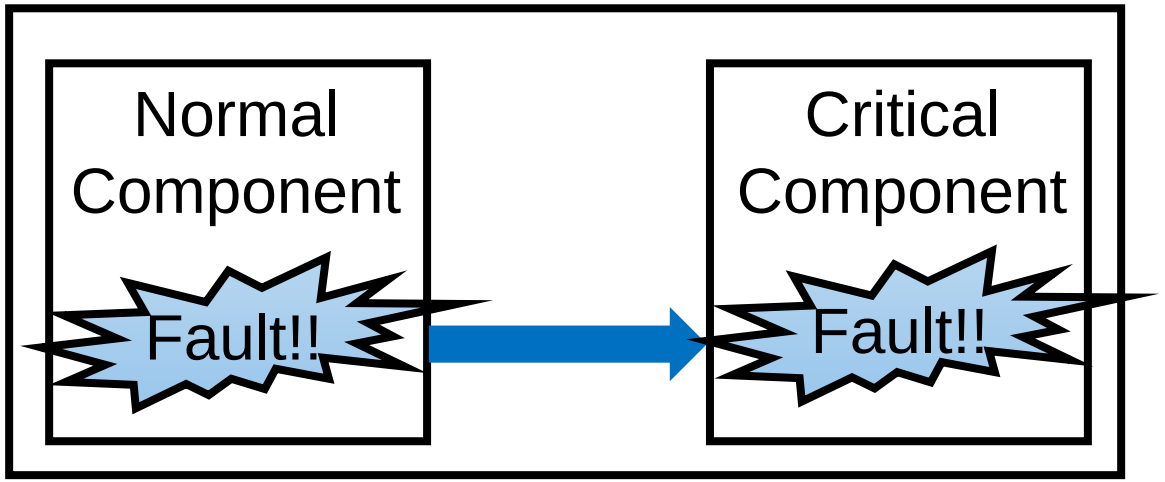
4 Implementation Results

5 Conclusion

Requirement #2 : Functional Safety

- SoCs for ADAS require high safety to avoid serious accidents.

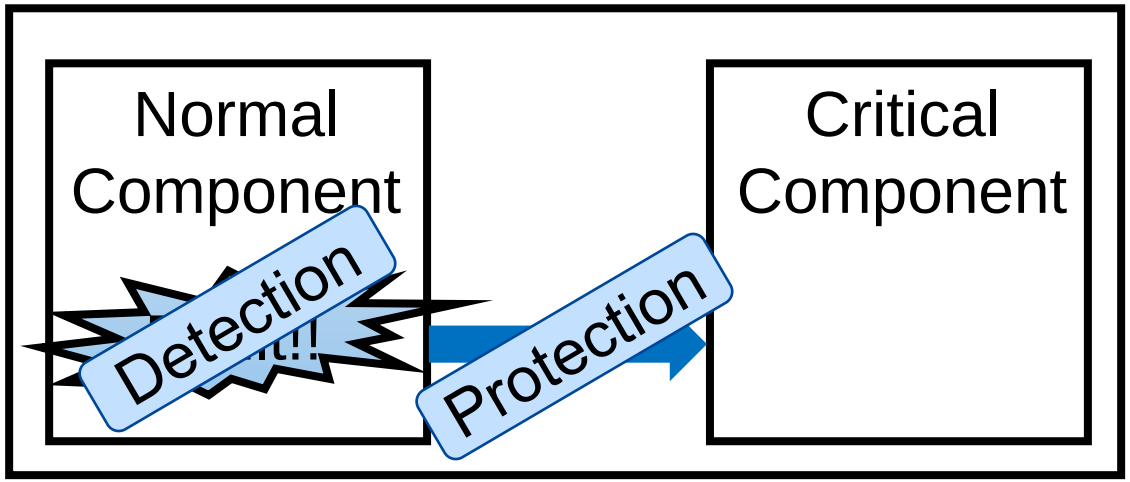
- When a component in SoC is faulty, the SoC has to …
  - Detect the fault.
  - Continue to operate until reaching safe state.
    - Especially, the critical components like the central control system



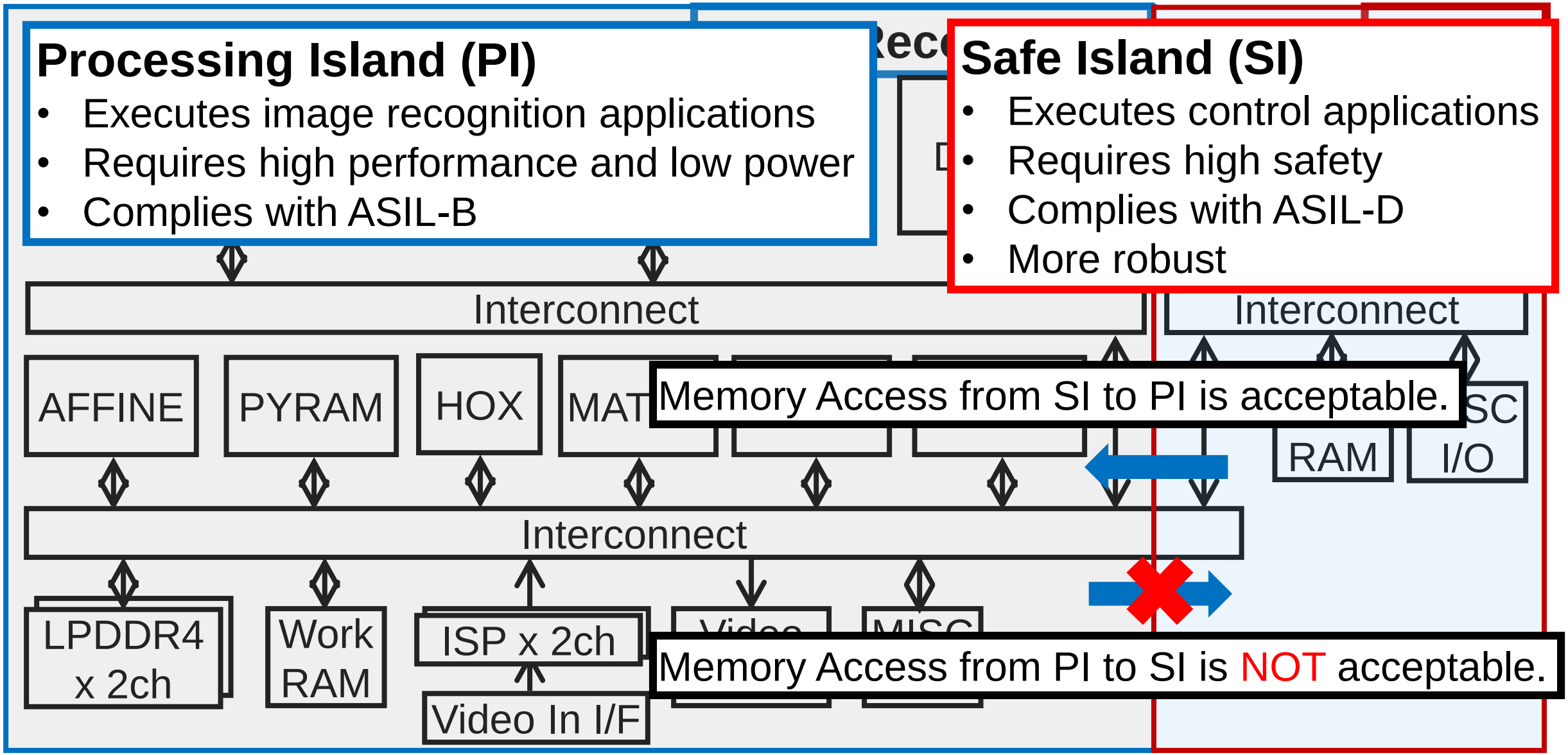
Approaches for Functional Safety

- Goal of functional safety
  - Reduce the risk of accidents to acceptable levels

- **Safety mechanisms** (SMs) are needed to reduce the risk
  - Partitioning critical regions to prevent fault propagation
  - Diagnostic features to detect faults



Critical Region Partitioning



Safety Mechanisms for Fault Detection

Processing Island (PI)
- Executes image recognition applications
- Requires high performance and low power
- Complies with ASIL-B

Safe Island (SI)
- Executes control applications
- Requires high safety
- Complies with ASIL-D
- More robust

Target of Fault Detection	PI complies with ASIL-B	SI complies with ASIL-D
Random Logic	Runtime BIST*	Duplicated Logic
Memory Logic	Parity / ECC	ECC
Memory Access	MPU	Duplicated MPU
Clock, Voltage, etc	Monitor	Duplicated Monitor
Bus Access	ECC with bus payload	ECC with bus payload

*Built-In Self Test (BIST)

Control of ISP including Runtime BIST

- Processor controls HWAs including runtime BIST operation
- However, ISP is difficult to be controlled with processor
  - **External imager controls**, not processor, start timing

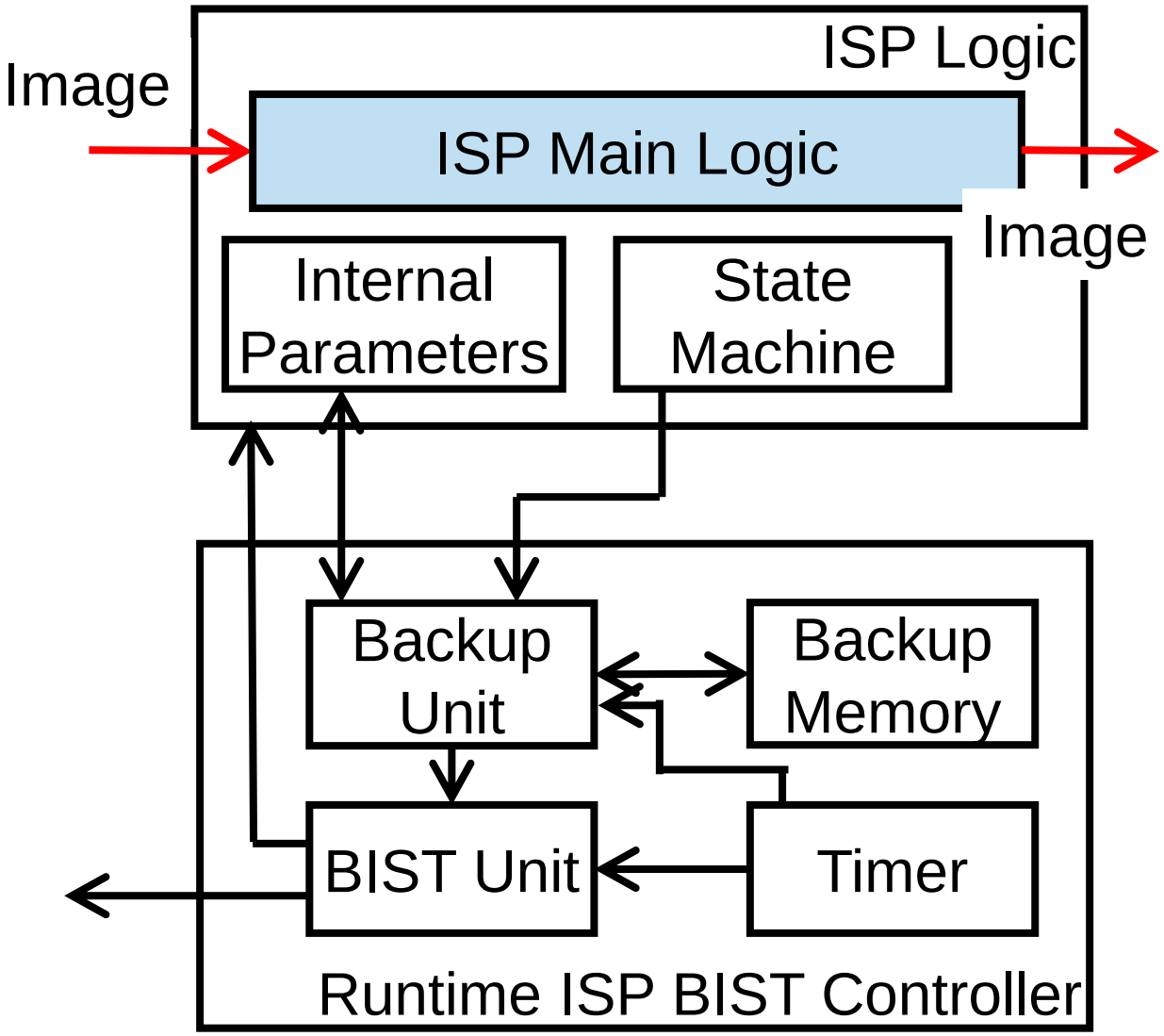
- Requirements
  - **Only during V blanking period,** BIST has to be running

➡ Dedicated runtime ISP BIST controller

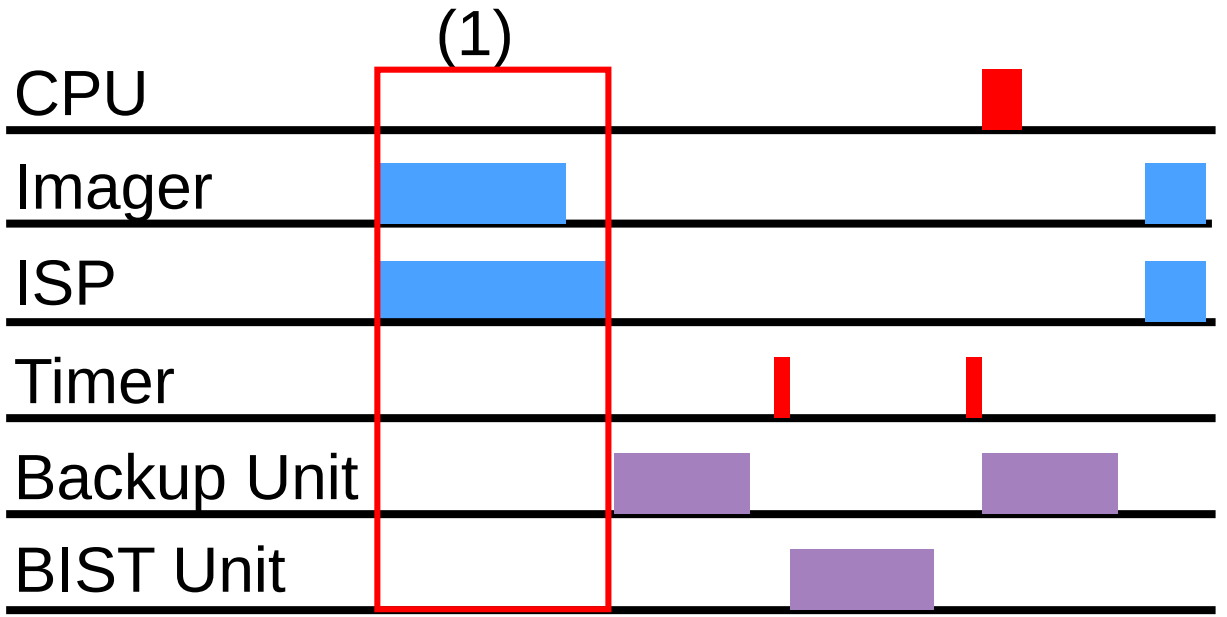
  - All parameters have to **be re-configured** after BIST

➡ Backing up and restoring before and after BIST

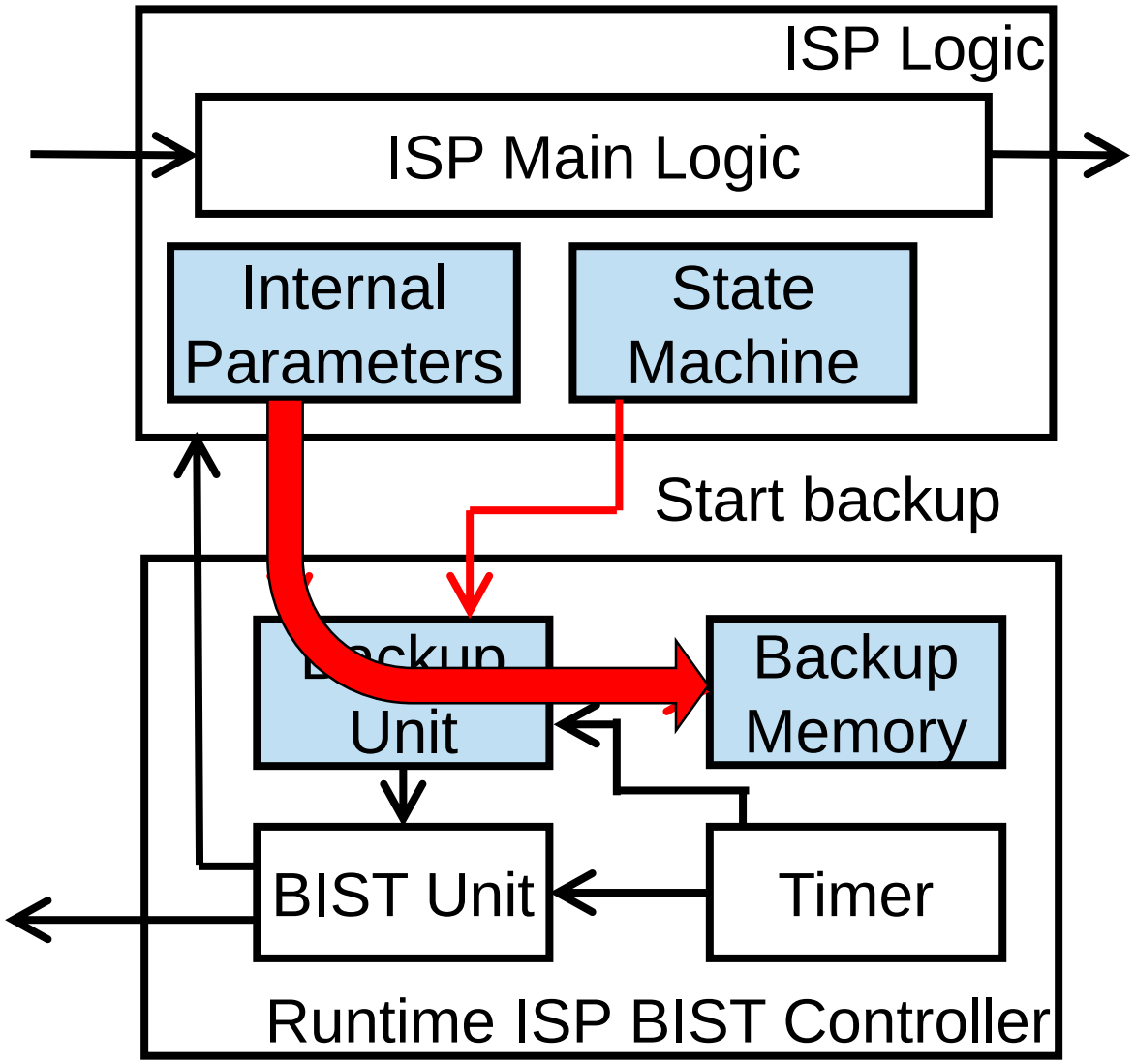
Control of ISP including Runtime BIST



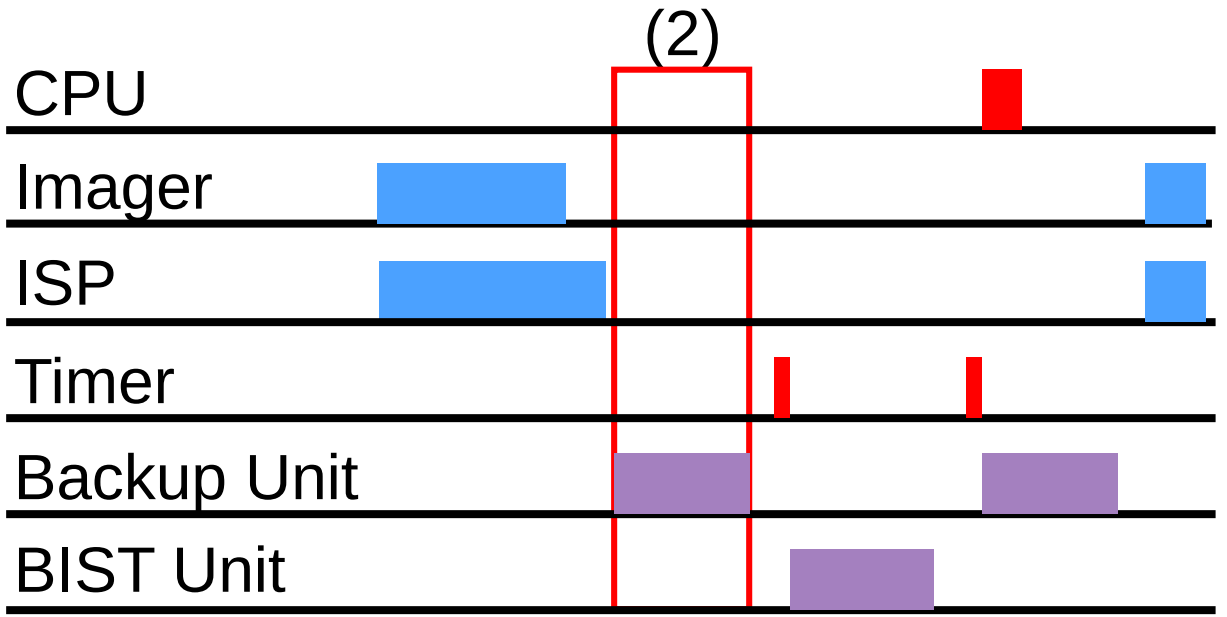
1. ISP receives new frame & starts its operation



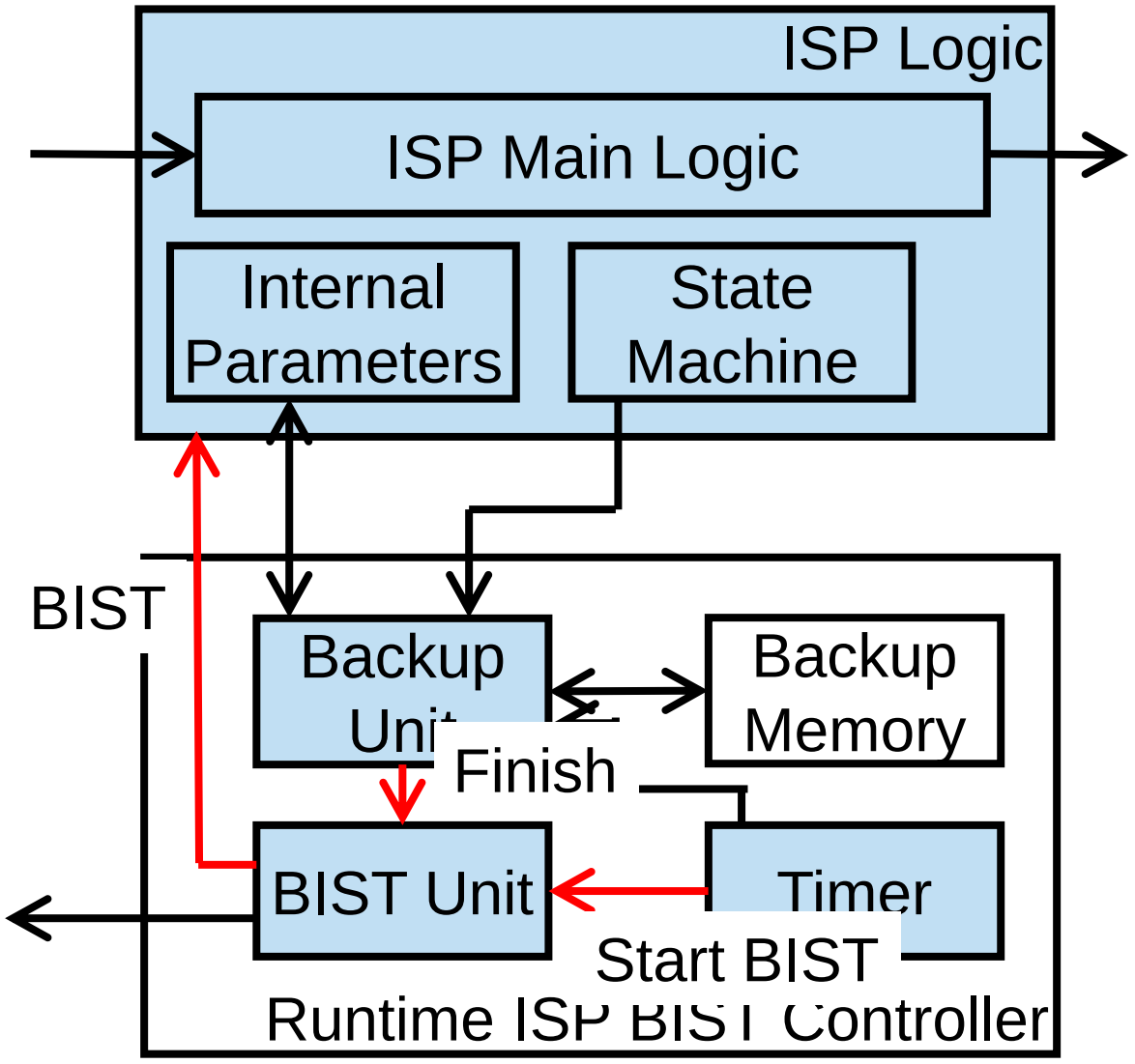
Control of ISP including Runtime BIST



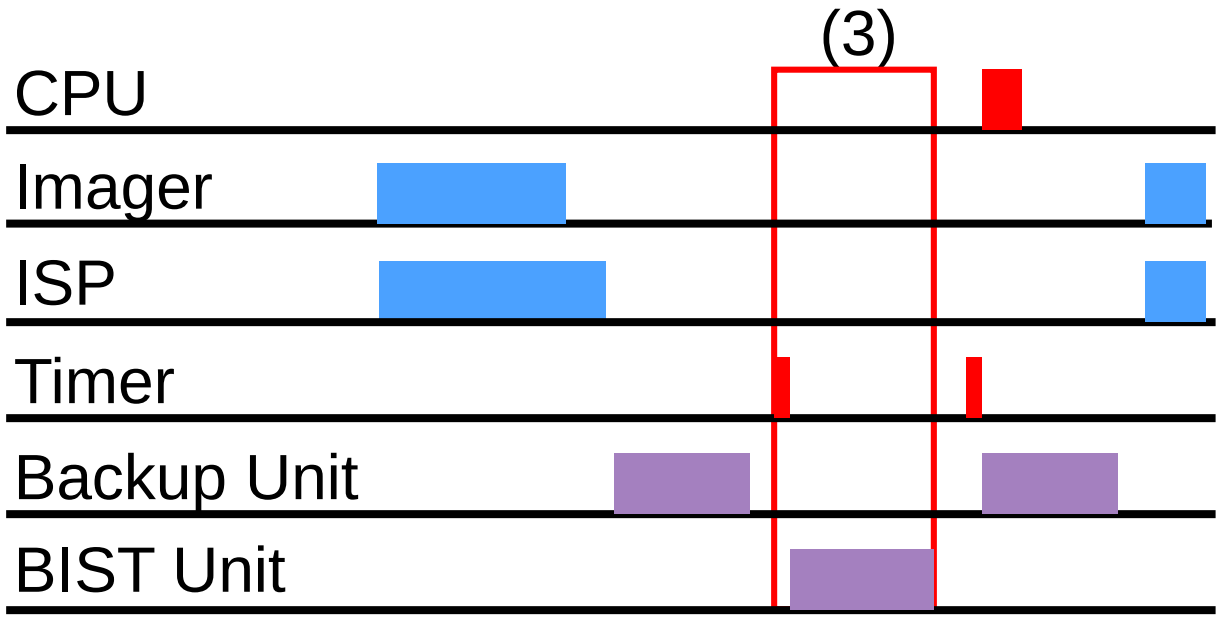
1. ISP receives new frame & starts its operation
2. Backup unit backs up all parameters



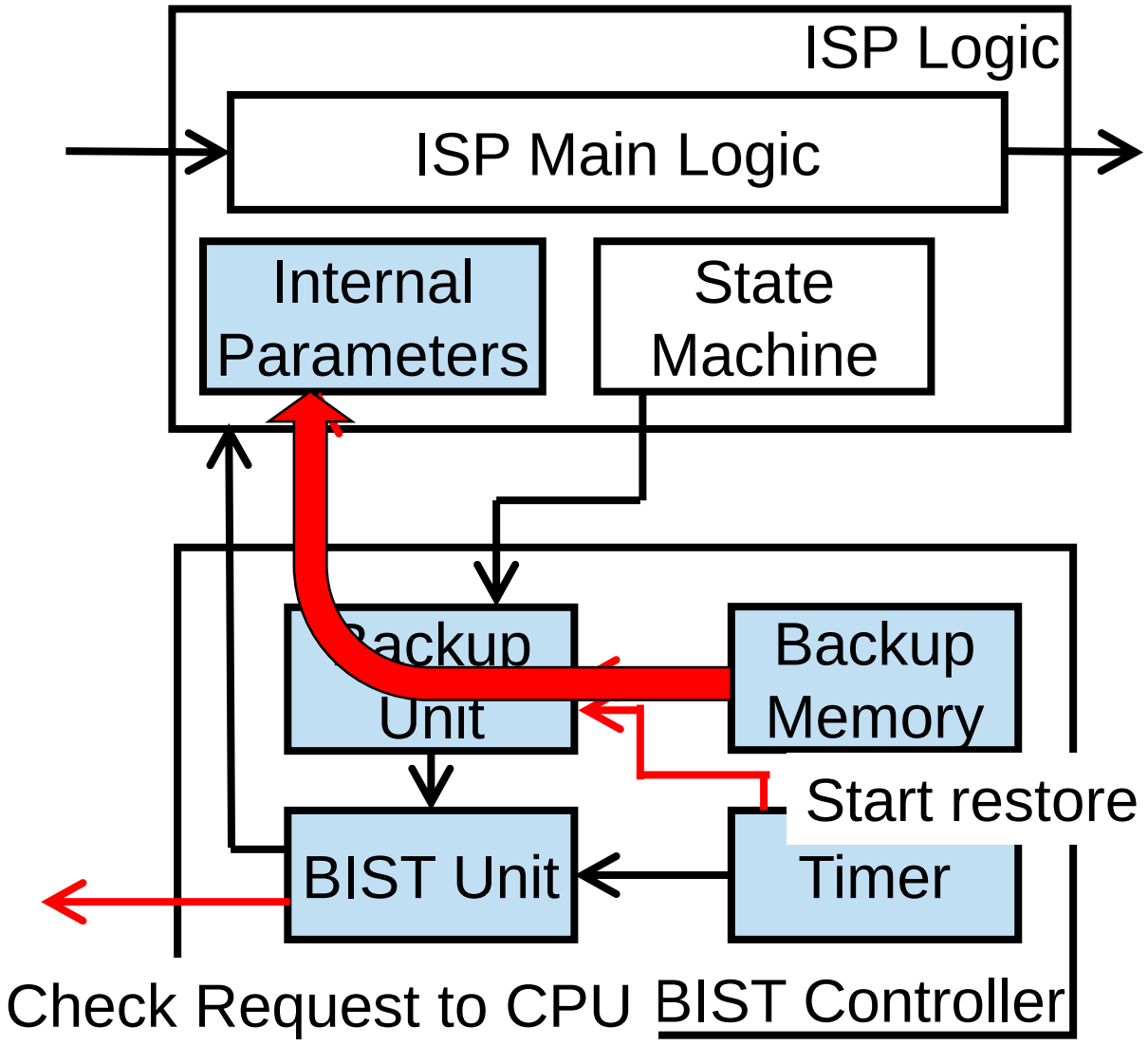
Control of ISP including Runtime BIST



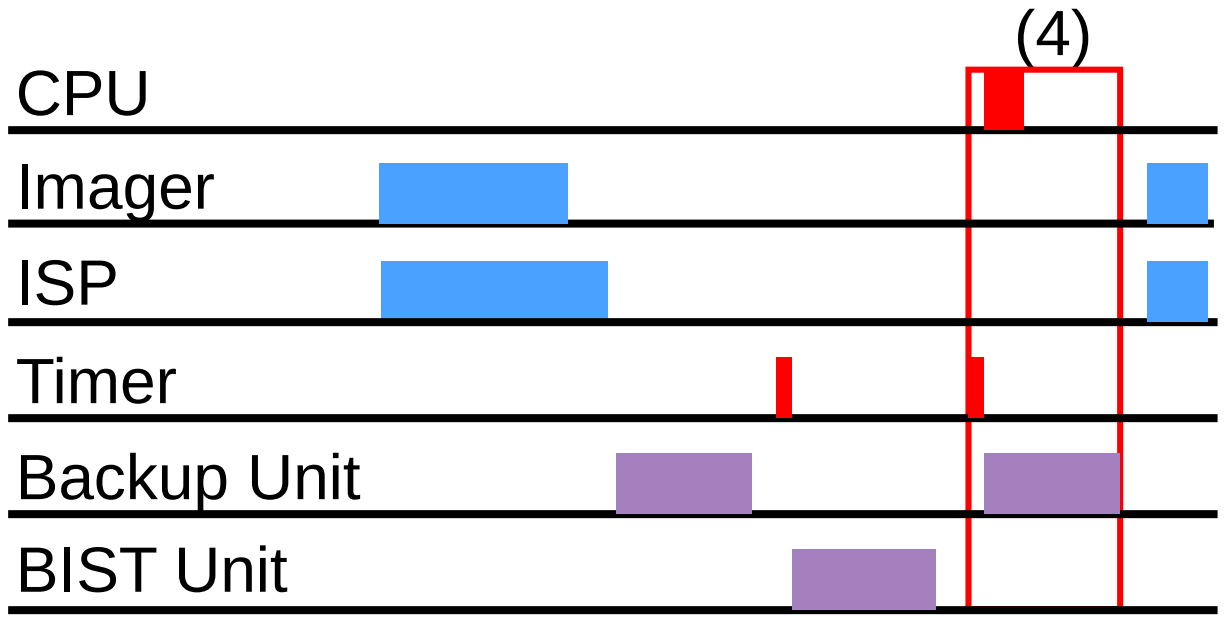
1. ISP receives new frame & starts its operation
2. Backup unit backs up all parameters
3. BIST operation starts if backup ends before Timer indicates



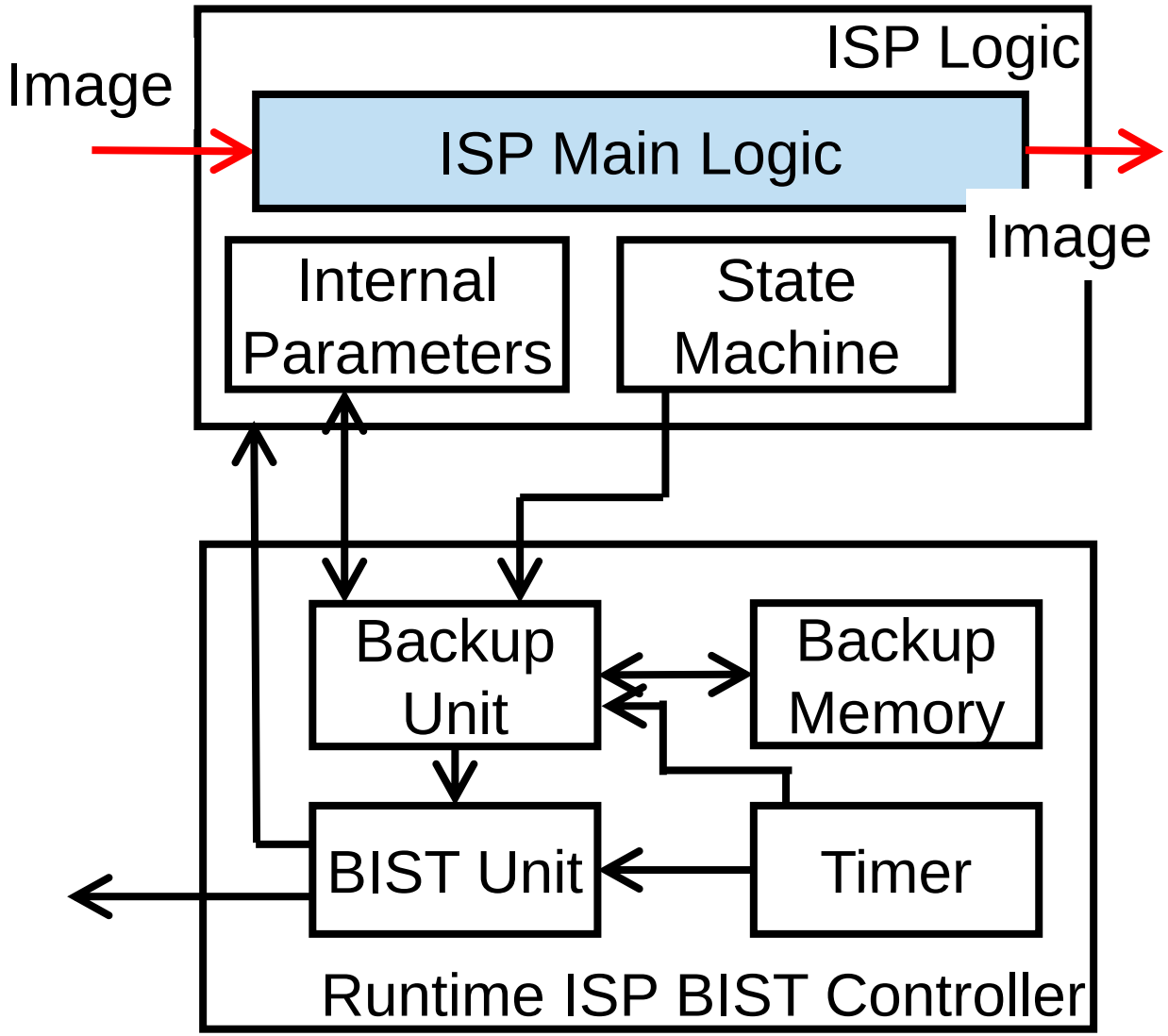
Control of ISP including Runtime BIST



1. ISP receives new frame & starts its operation
2. Backup unit backs up all parameters
3. BIST operation starts if backup ends before Timer indicates
4. CPU checks BIST result & Backup unit restores all parameters



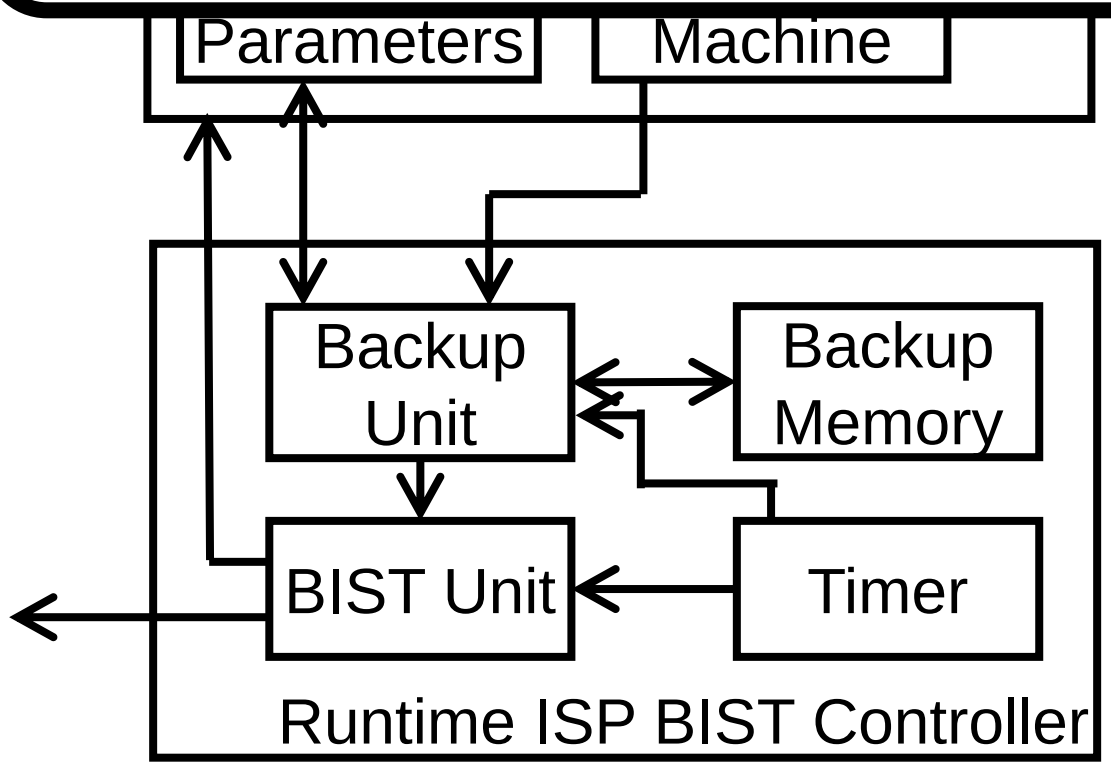
Control of ISP including Runtime BIST



1. ISP receives new frame & starts its operation
2. Backup unit backs up all parameters
3. BIST operation starts if backup ends before Timer indicates
4. CPU checks BIST result & Backup unit restores all parameters
5. Continue ISP operation for next frame (5)

Control of ISP including Runtime BIST

Runtime ISP BIST Controller can perform backup, BIST, and restore during V blanking period



1. ISP receives new frame &
4. CPU checks BIST result &
   Backup unit restores all parameters
5. Continue ISP operation for next frame

Outline

1 Background

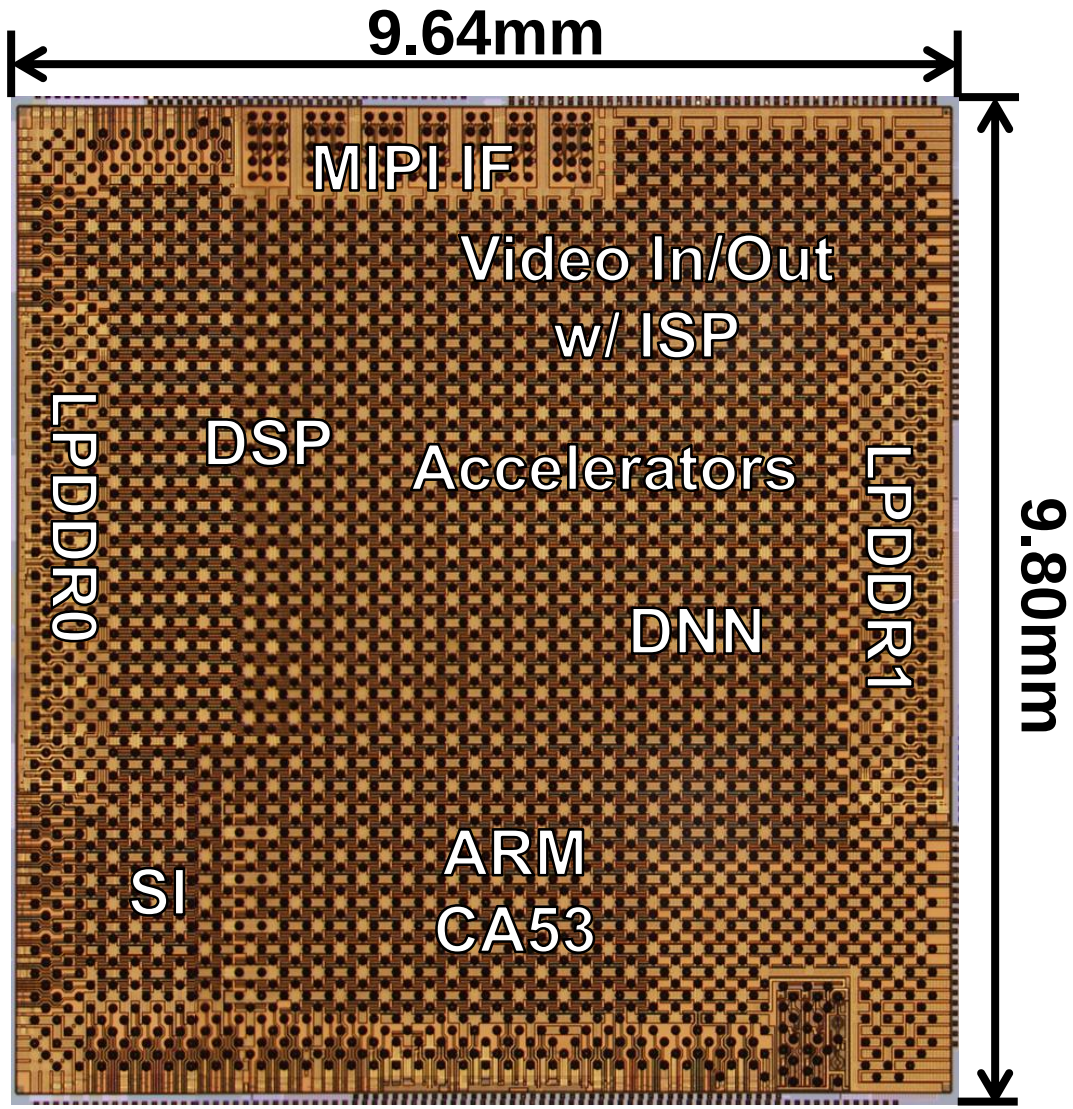
2 Architecture of the SoC

3 Functional Safety

4 **Implementation Results**

5 Conclusion

Chip Micrograph



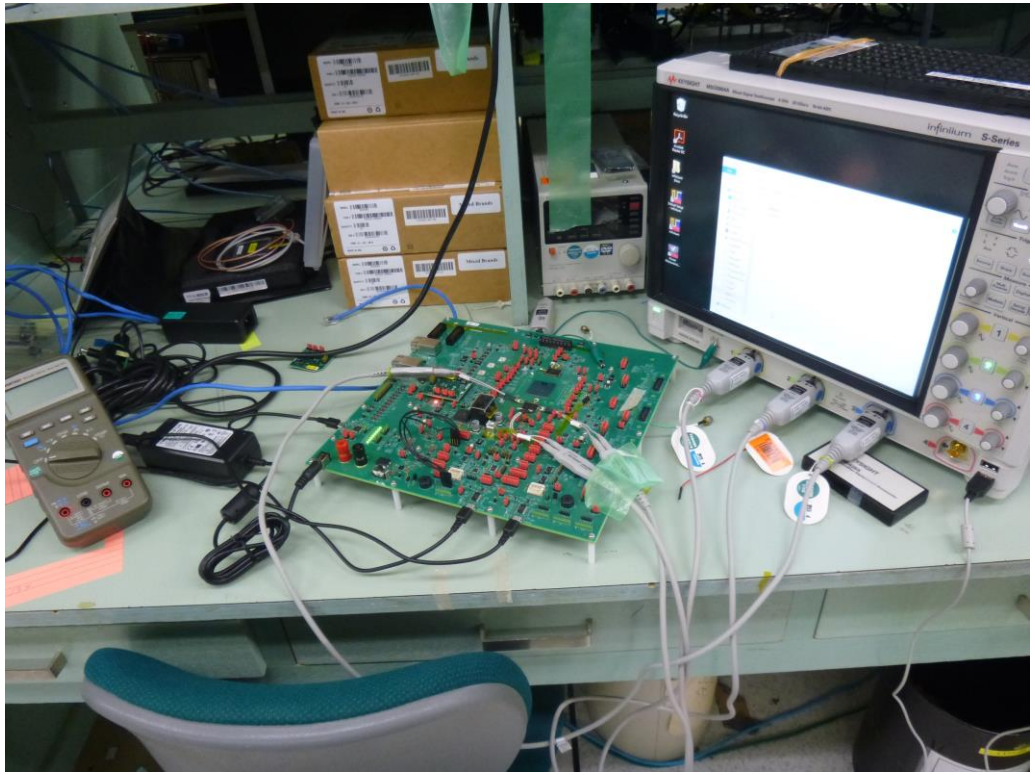
Process	16nm
Chip Size	94.52mm^2
On-Chip SRAM	142.9Mbit

Supply Voltage	
SI Core	0.8V
SI I/O	1.8～3.3V
PI Core	0.8V
PI I/O	1.8～3.3V

Power Consumption on Evaluation Board

- The power consumption of PI is 2.73W
- Executing 5 applications using one video stream
  - Pedestrian detection
  - Vehicle detection
  - Traffic signal recognition
  - Lane detection
  - Head light detection
- Running Resources
  - 2 cores of CA53
  - 4 DSPs
  - 6 types of accelerators: DNN, HOX, AFFINE, AKAZE, MATCH, PYRAM

Power Evaluation Environment



Demonstration of DNN Application

This Dataset is “Cityscapes Dataset”
Reference: M. Cordts, et.al, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Outline

1 Background

2 Architecture of the SoC

3 Functional Safety

4 Implementation Results

5 **Conclusion**

Conclusion

- **We implemented the SoC for ADAS applications** to realize high performance with low power consumption and high safety

- Heterogeneous architecture achieves over 20 TOPS and 2TOPS/W
  - 8 processors and 4 DSPs
  - 8 types of hardware accelerators such as DNN and ISP
  - Power consumption of PI is 2.73 W by running use case on the evaluation board

- Safety mechanisms are introduced for high safety
  - e.g. partitioning (PI and SI) and runtime BIST

TOSHIBA

Thank you for your attention!!