国立大学法人
電気通信大学
The University of Electro-Communications

Pham Laboratory
Integrated circuit design laboratory

Secured Network-on-Chip Framework for RISC-V Computing Systems

Hoang Trong Thuc, Binh Kieu-Do-Nguyen, and Cong-Kha Pham

2024/06/06

Outline

1. Introduction
2. Many-core Architecture
3. Memory Bottleneck
4. Secured Architecture
5. Conclusion

Outline

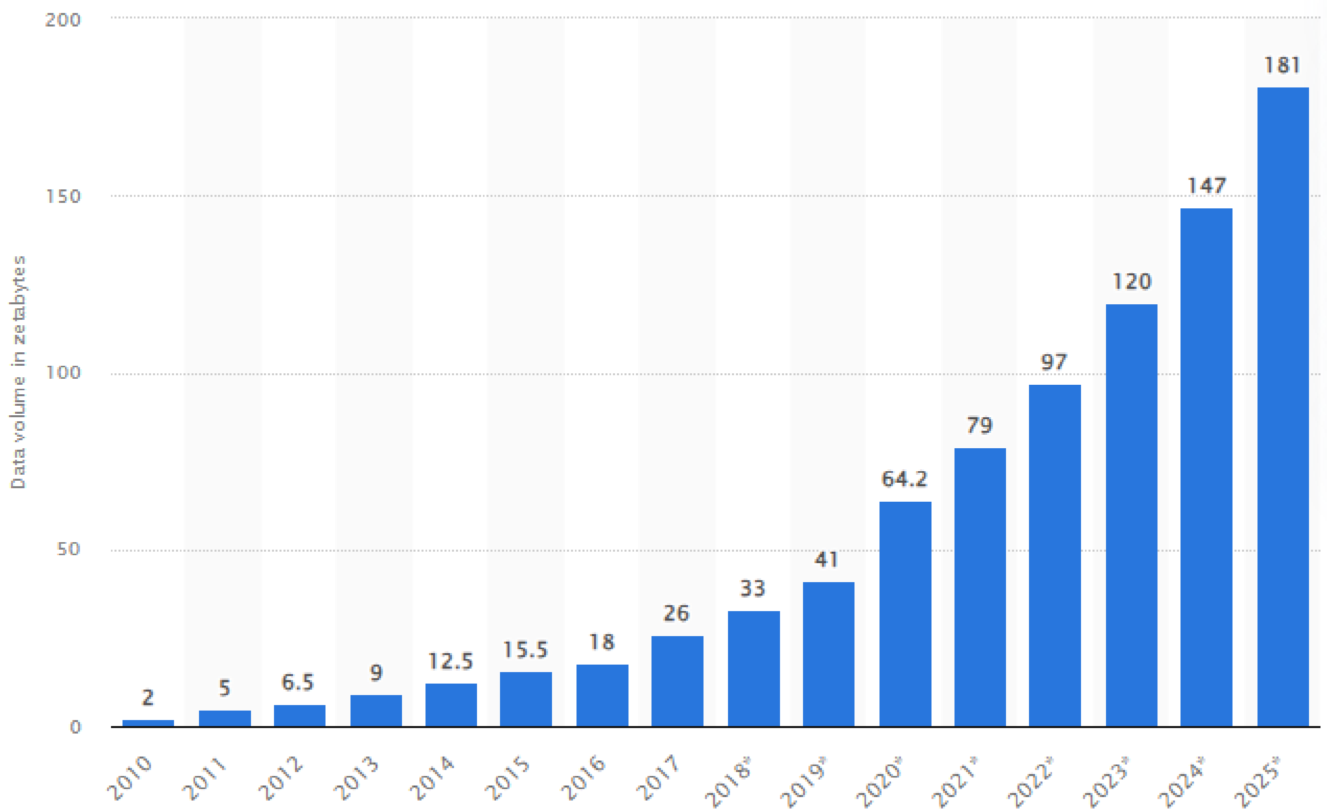
1. Introduction
2. Many-core Architecture
3. Memory Bottleneck
4. Secured Architecture
5. Conclusion

1. Introduction (1/5) Requirement of big data

Smart city, smart society:

- ➔ **Human-centered:** human well-being and aims to create a society where everyone can live a comfortable, healthy, and fulfilling life.
- ➔ **Data-driven:** leverages vast amounts of data to gain insights, make informed decisions, and optimize systems and processes.
- ➔ **Cyber-physical integration:** It emphasizes the seamless integration of cyberspace (the virtual world) and physical space (the real world) to create a hyper-connected society.
- ➔ **Sustainable and resilient:** It aims to build a society that is environmentally sustainable and resilient to various risks and challenges.

1. Introduction (2/5) Requirement of big data

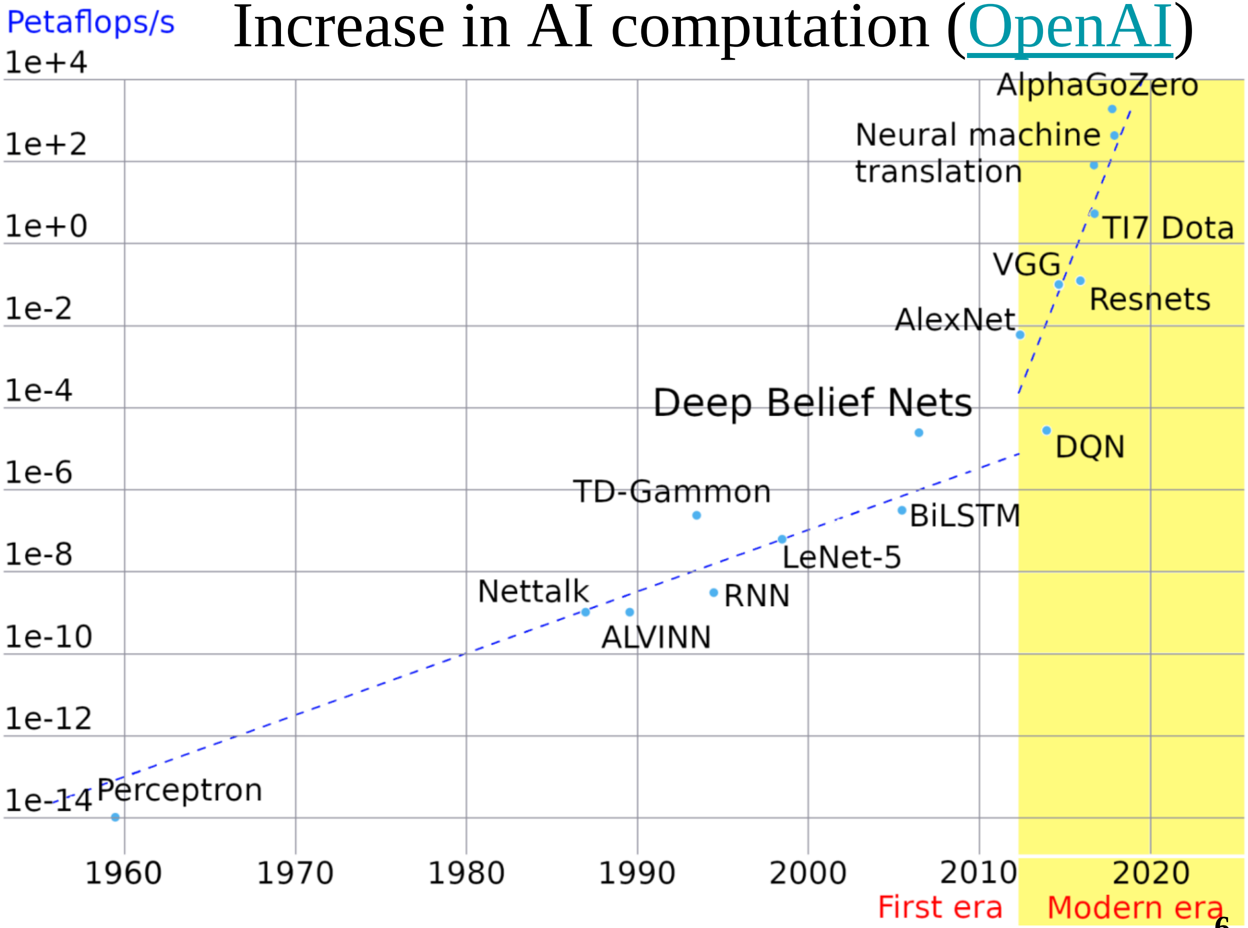


- ➔ Data volume increase rapidly over years.
- ➔ Achieve 181 Zetabytes in 2024.

1. **Introduction** (3/5) Requirement of Computation

- ➔ First era: Compute requirements doubling every 2 years → suitable with Moore's law.
- ➔ Modern era: Compute requirements doubling every 3.4 months.

Increase in AI computation (OpenAI)

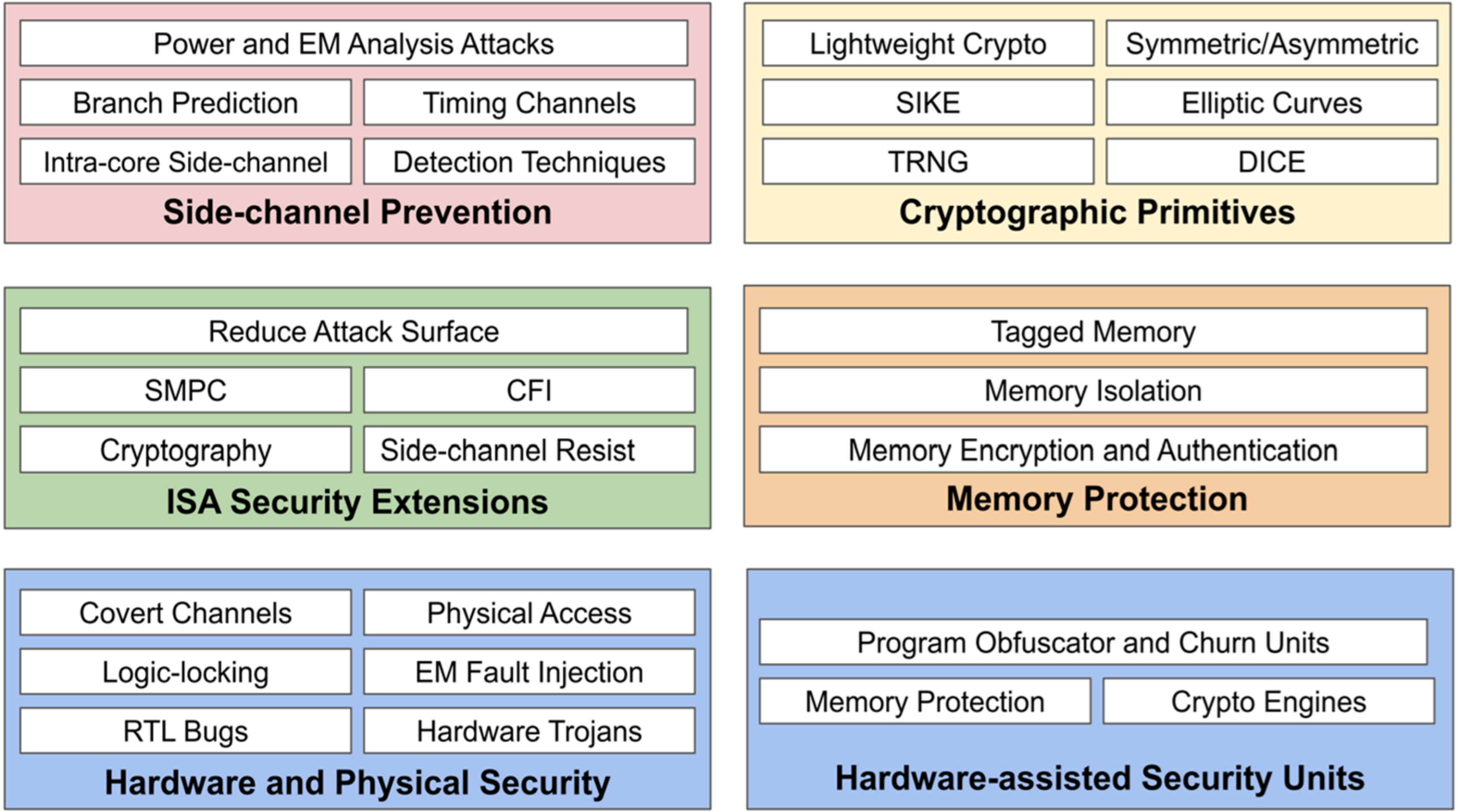


1. Introduction (4/5) Requirement of Power

Power efficiency is not merely a technical consideration but a **strategic necessity** for **H**igh-**P**erformance **C**omputers (**HPC**s).

- **Cost Reduction:** High performance computers consume massive amounts of energy for powering and cooling IT equipment. By improving power efficiency, data center operators can significantly reduce their electricity bills, leading to substantial cost savings over time.
- **Environmental Impact**: High performance computers are major consumers of electricity, often generated from fossil fuels, contributing to greenhouse gas emissions and climate change. Increased power efficiency directly translates to reduced carbon footprint and environmental impact.
- **Scalability:** As data demands grow, data centers need to scale their operations accordingly. Power-efficient designs allow for flexible scaling without exceeding power capacity limits or incurring excessive energy costs.

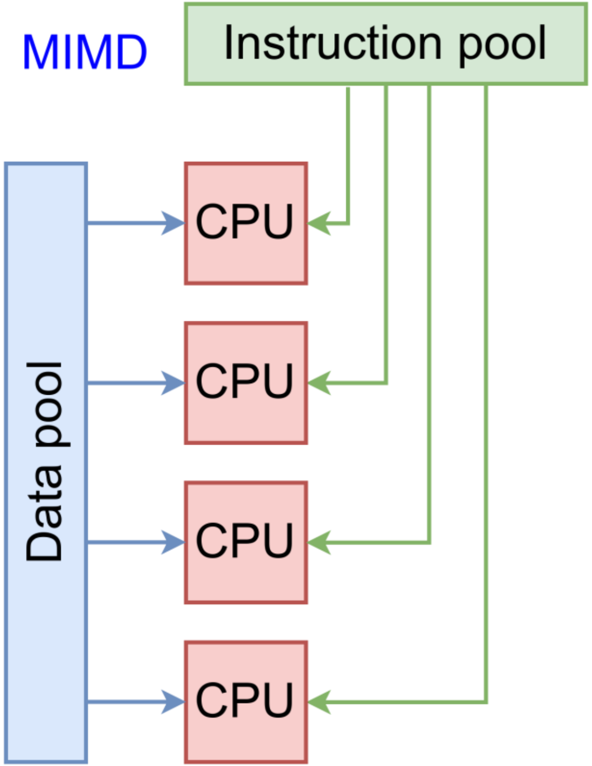
1. Introduction (5/5) Requirement of Security



Outline

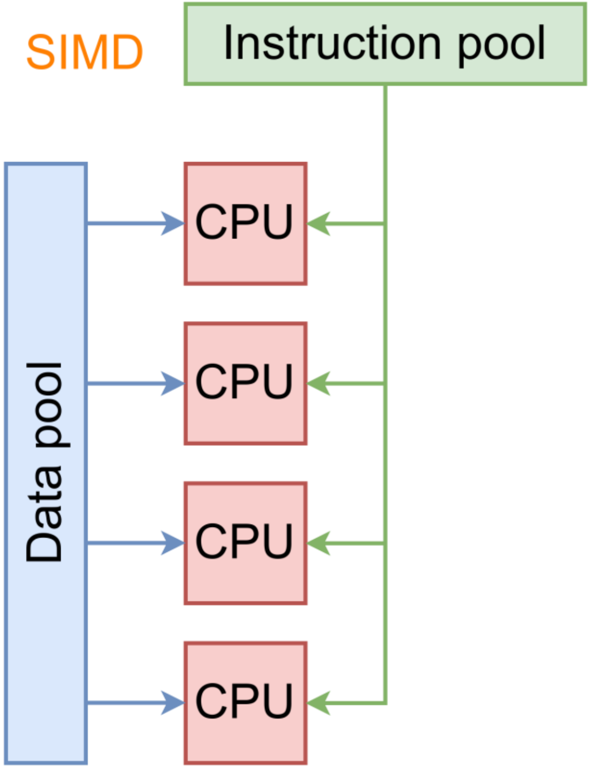
1. Introduction
2. Many-core Architecture
3. Memory Bottleneck
4. Secured Architecture
5. Conclusion

2. Many-core Architecture (1/7) MIMD vs. SIMD



Multiple-**I**nstruction
Multiple-**D**ata
Multi-core processor.
Each processor perform different instruction.

Single-**I**nstruction
Multiple-**D**ata
Many-core processor.
All processors perform same instruction.



2. Many-core Architecture (2/7) Multi- vs. Many-

Feature	Multi-Core Processor	Many-Core Processor
Core Count	2 to 8	Dozens to thousands
Flynn's model	MIMD	SIMD
Design	General-purpose, balanced parallel & serial	Highly parallel, optimized for throughput
Single-Thread Perf.	High	Lower
Applications	Desktop, laptop, server, general-purpose tasks	HPC, embedded systems, specialized accelerators

2. **Many-core Architecture** (3/7) Brick wall data

Integer Ops

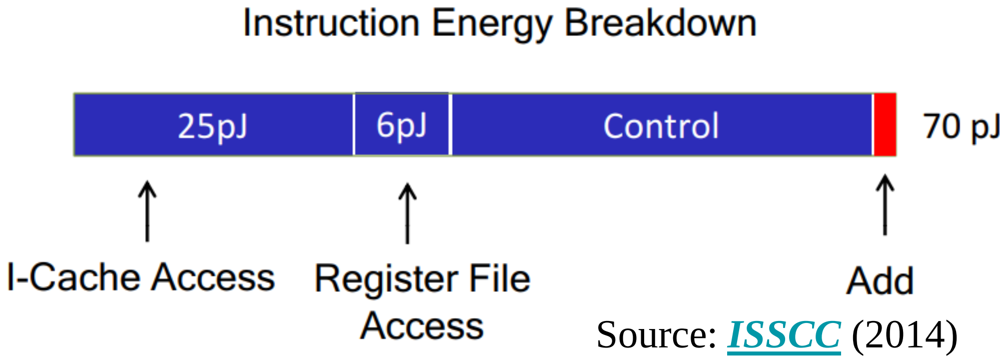
Add	Power(pJ)
8-bit	0.03
32-bit	0.10
Mult	**Power(pJ)**
8-bit	0.2
32-bit	3.1

Float Ops

FAdd	Power(pJ)
16-bit	0.4
32-bit	0.9
FMult	**Power(pJ)**
16-bit	1.1
32-bit	3.7

Memory Ops

Cache	Power(pJ)
8-KB	10
32-KB	20
1-MB	100
DRAM	1300-2600



Source: *ISSCC* (2014)

Data movement cause bottleneck:

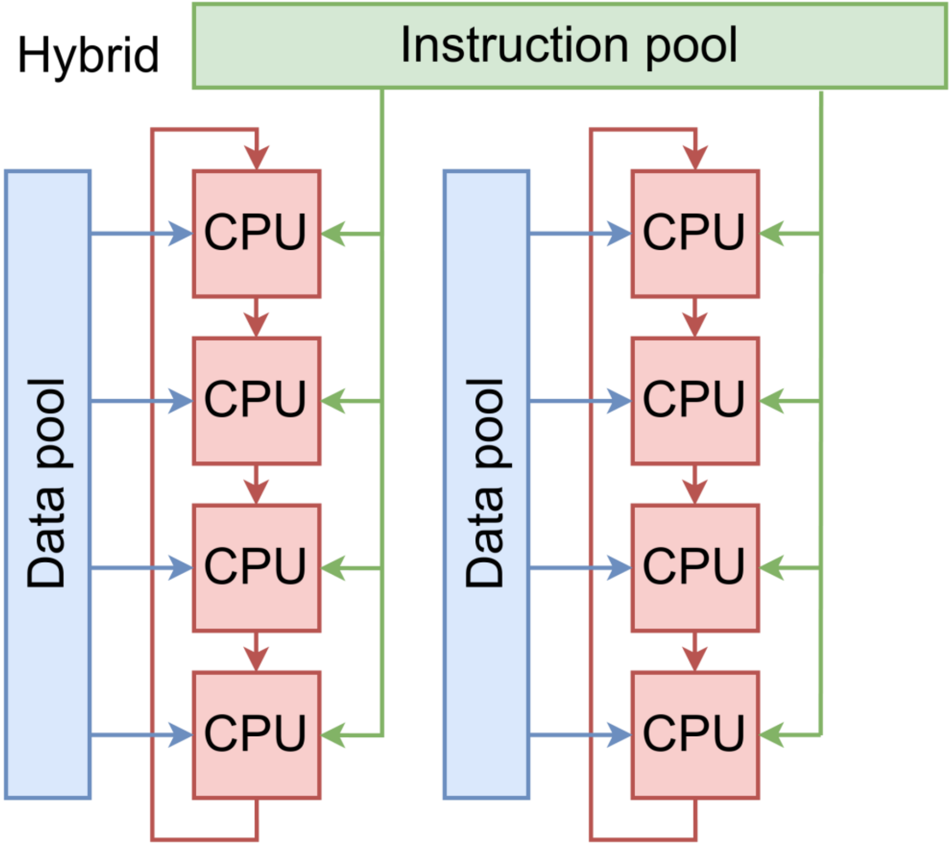
⇨Limit by memory bandwidth.

⇨Delay.

⇨Power consumption.

⇨Thermal issues.

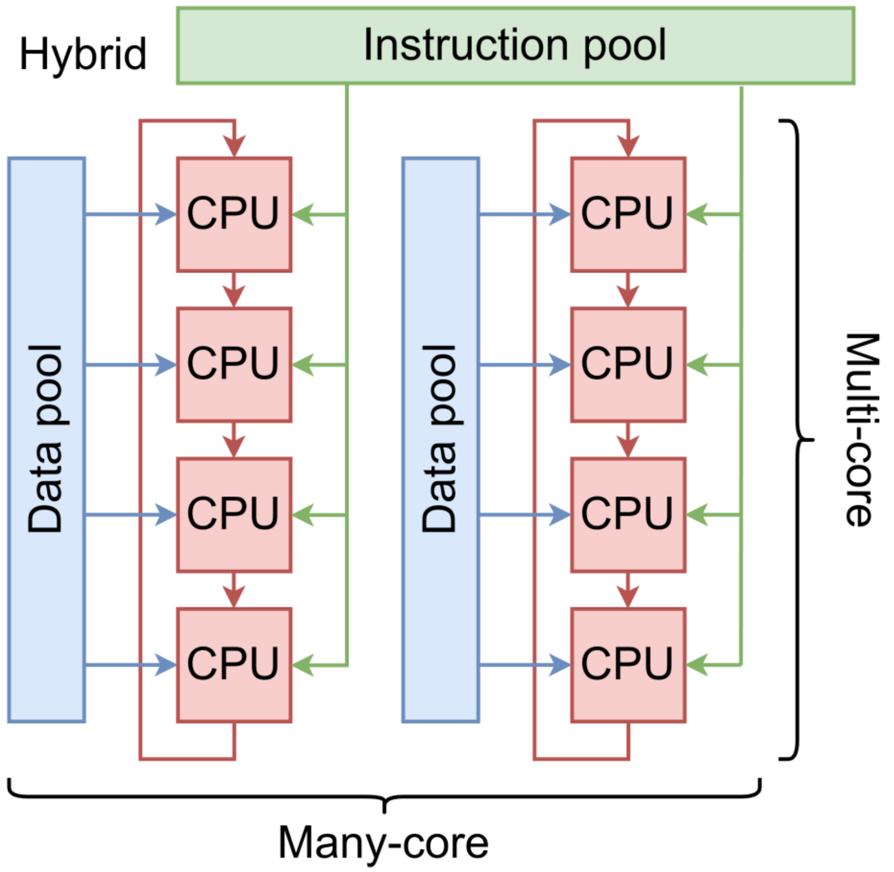
2. Many-core Architecture (4/7) Data-centric design



Data-centric

- ➔ Single program is applied for multiple group of cores.
- ➔ Multiple threads are applied for multiple group of cores.
- ➔ Data could be streamed between adjacent cores.
- ➔ Data are kept in pool as long as possible.

2. Many-core Architecture (5/7) Hybrid architecture

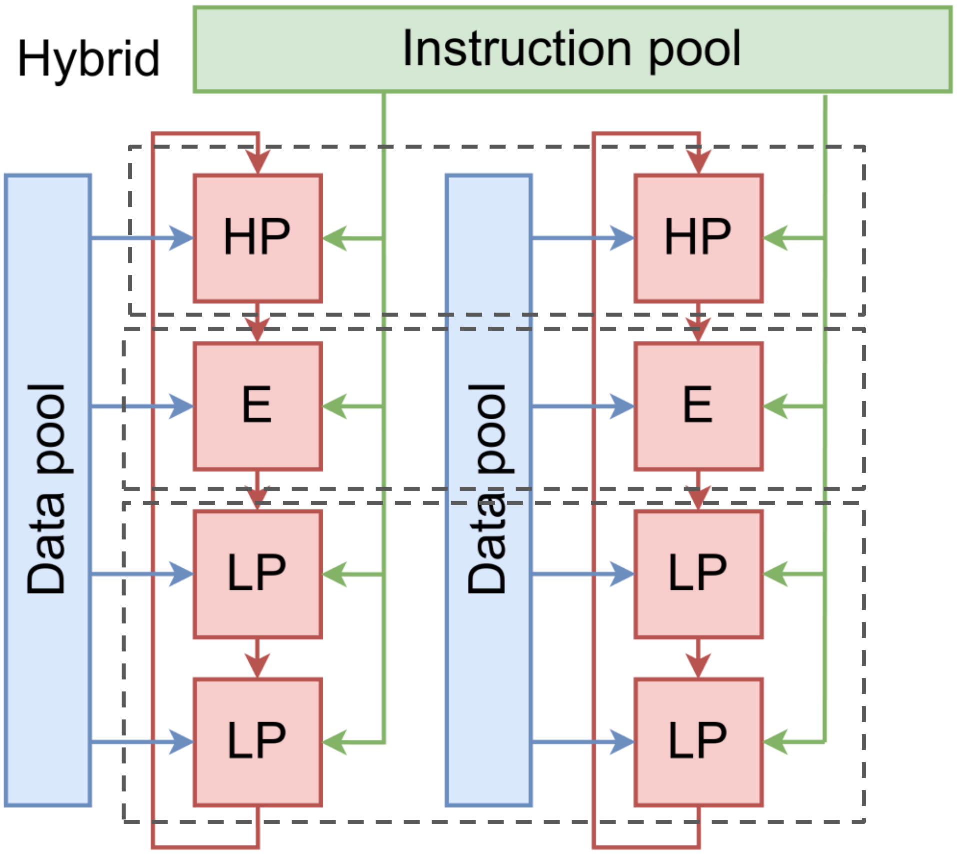


Multi-core: **M**ultiple **I**nstruction **M**ultiple **D**ata.

Many-core: **S**ingle **I**nstruction **M**ultiple **D**ata

Hybrid: combines both **SIMD** and **MIMD**.

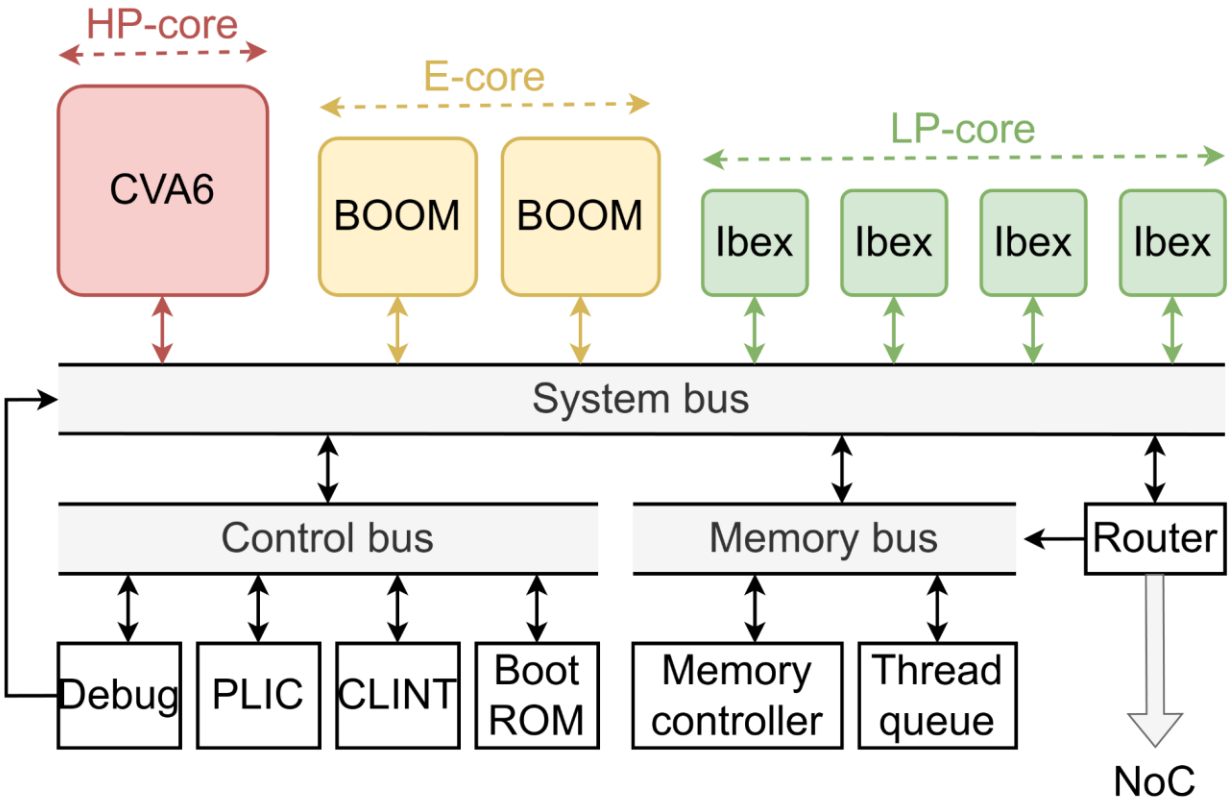
2. Many-core Architecture (6/7) Hybrid architecture



Asymmetric architecture

- ➔ High-performance core (**HP-core**) targets low delay for single-thread.
- ➔ Efficient-core (**E-core**) maximizes multi-thread performance.
- ➔ Low-power-core (**LP-core**) minimizes power consumption for hybrid tasks (single-thread/multi-thread).

2. Many-core Architecture (7/7) Asymmetric



Parallel level: up to 7 threads, 21 instructions.

Processor system:
- ➔ HP-core: CVA6
- ➔ E-core: BOOM
- ➔ LP-core: Ibex

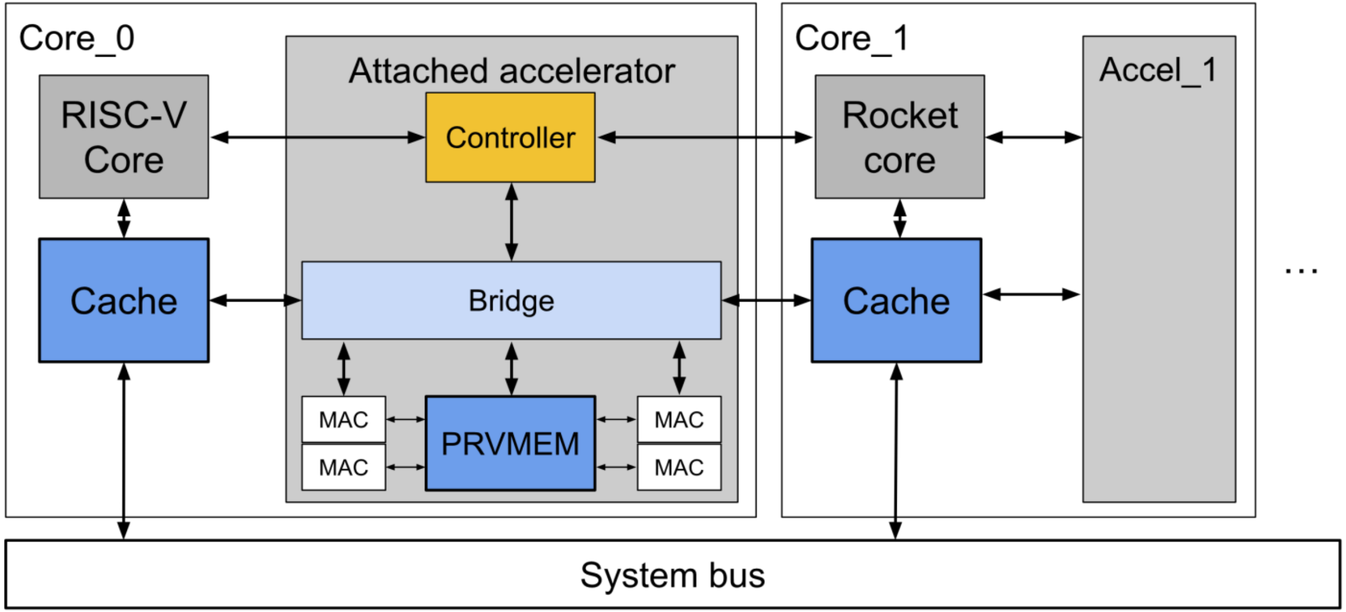
Bus system:
- ➔ System bus
- ➔ Control bus
- ➔ Memory bus

Router:
Connect the system with Network-on-Chip (NoC).

Outline

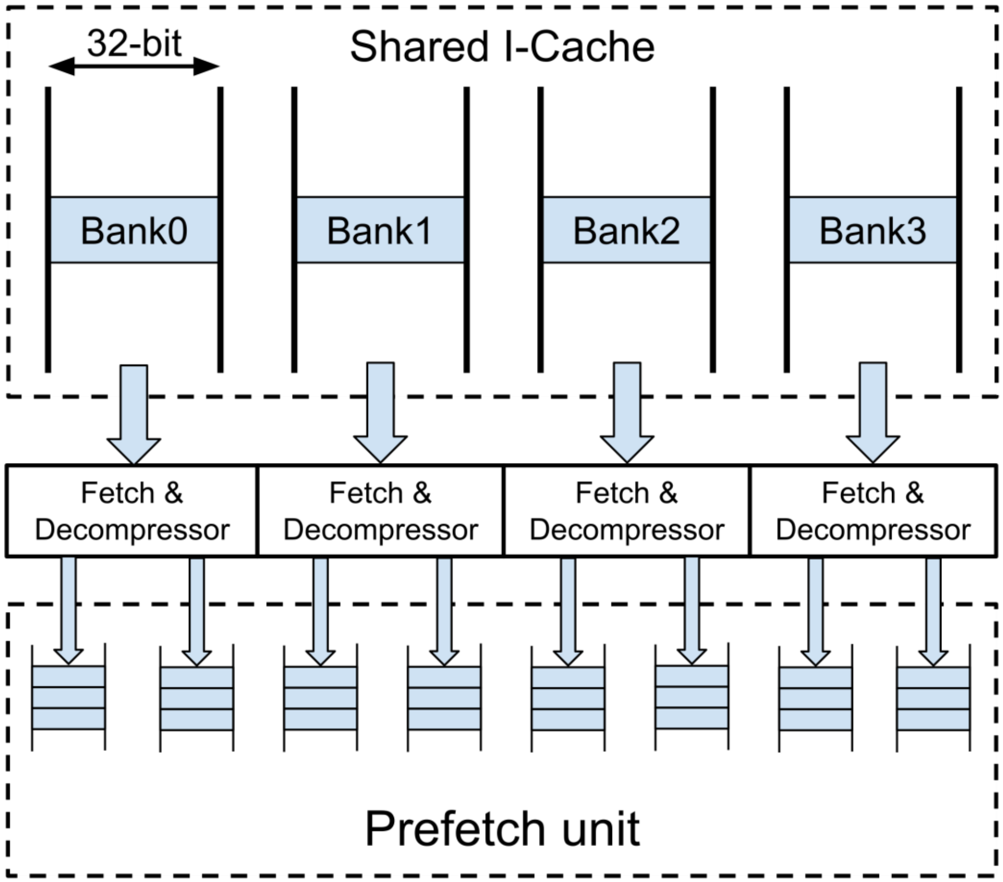
1. Introduction
2. Many-core Architecture
3. Memory Bottleneck
4. Secured Architecture
5. Conclusion

3. Memory Bottleneck (1/5) Coupled data tunnel



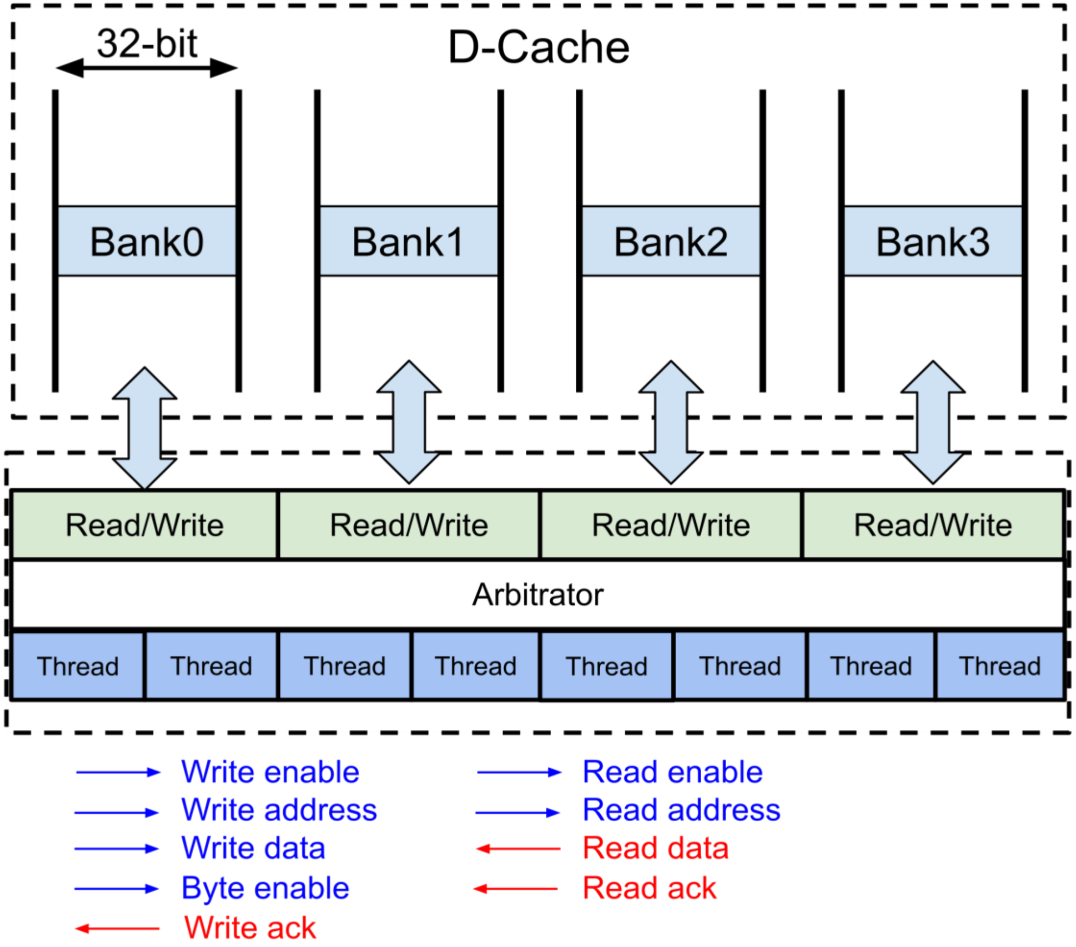
- ➔ Data stream can be exchanged between two consecutive core through a tightly-coupled bridge.
- ➔ Tightly-coupled accelerator (e.g. MAC) can be integrated with private memory (PRVMEM).

3. Memory Bottleneck (2/5) Instruction cache



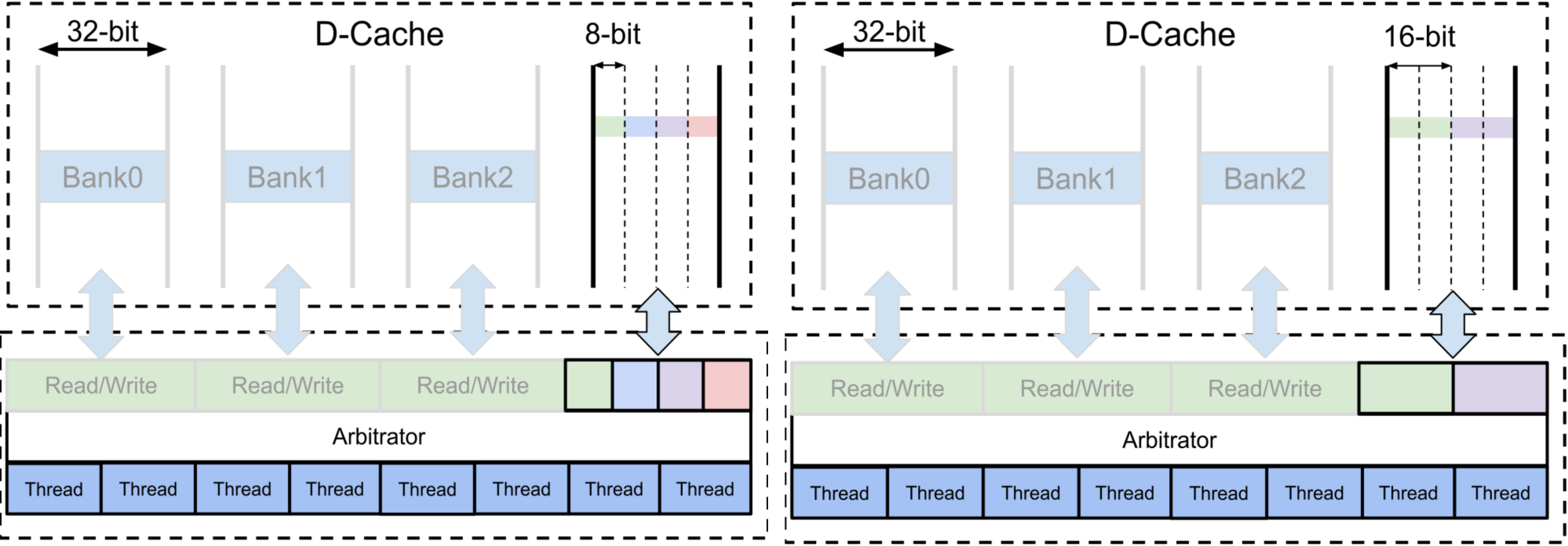
- ➔ Multiple banks increased the cache bandwidth.
- ➔ Compress extension allows multiple instructions could be fetched in single cycle.
- ➔ Prefetch unit hides the latency of memory accesses, ensuring that the processor has a steady stream of instructions. It also supports for advanced branch prediction strategy.

3. **Memory Bottleneck** (3/5) Data cache



- ➔ Read/Write (32/64-bit).
- ➔ Arbitrator (channel & thread arbitrators) arbitrates the request from threads.
- ➔ Handshake by ENABLE and ACK signals.
- ➔ Support Word/Half-word/Byte access.

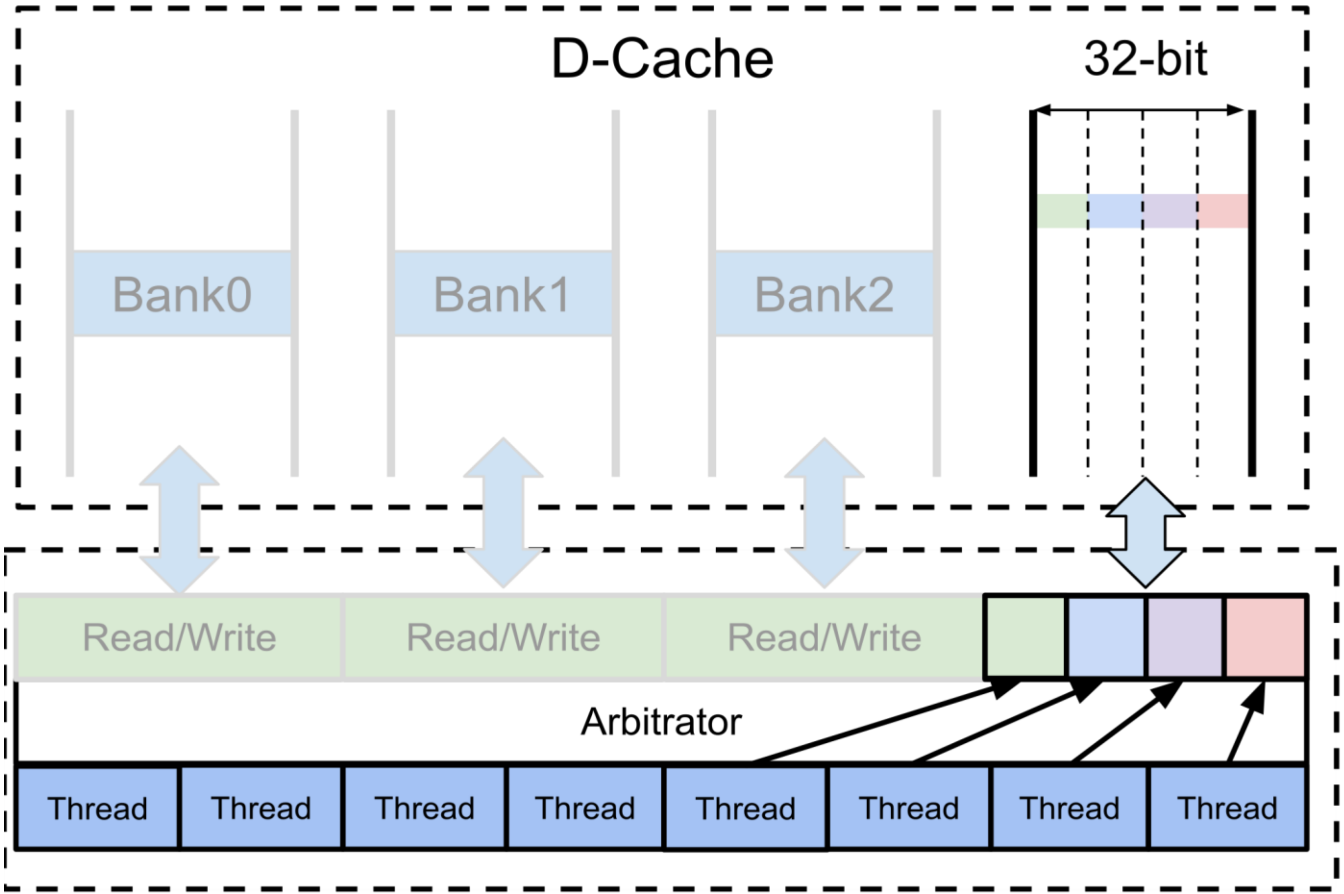
3. Memory Bottleneck (4/5) Data cache



Pseudo-channel

- ➔ Leverages the bandwidth of Byte-banks with multiple threads.
- ➔ Support 8/16/32-bit access.

3. Memory Bottleneck (5/5) Data cache

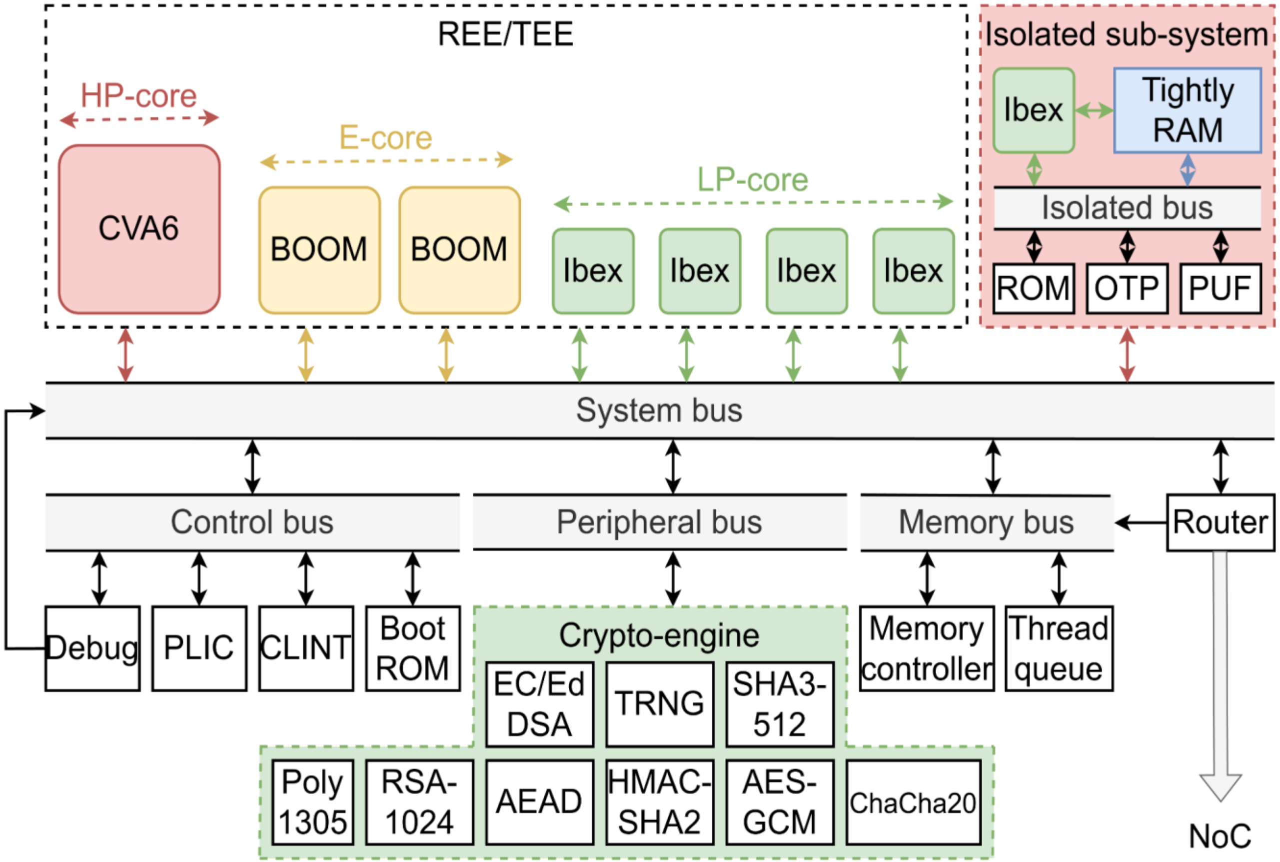


Multiple threads could When threads write bytes to same word address, written **bytes from different thread are composed in a single channel**, an write to the memory in the same time.

Outline

1. Introduction
2. Many-core Architecture
3. Memory Bottleneck
4. Secured Architecture
5. Conclusion

4. Secured Architecture (1/5) System



Isolated sub-system:

➔ Secured boot process for HP-Core (main core).

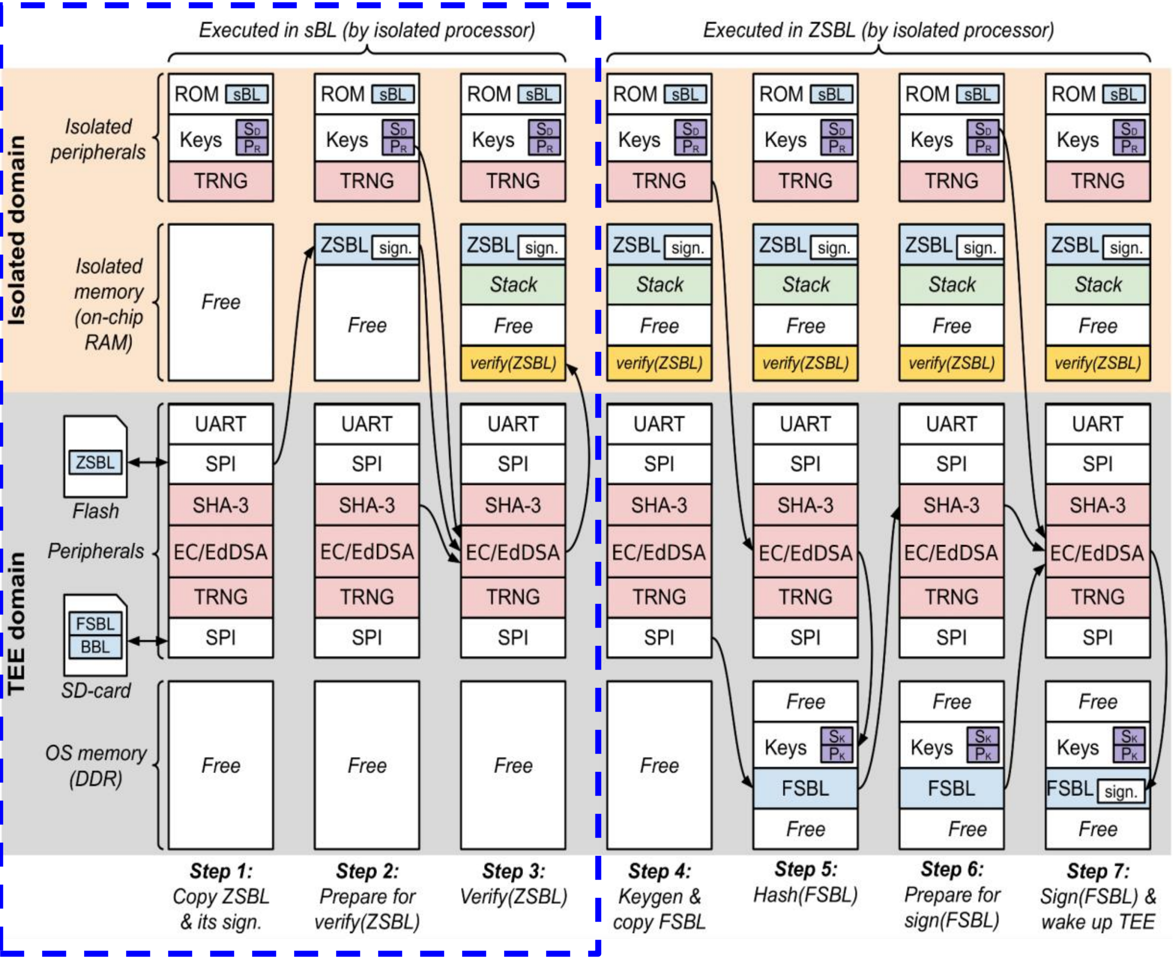
Crypto-engines:

➔ Encrypt/decrypt data.

➔ Sign/verify program.

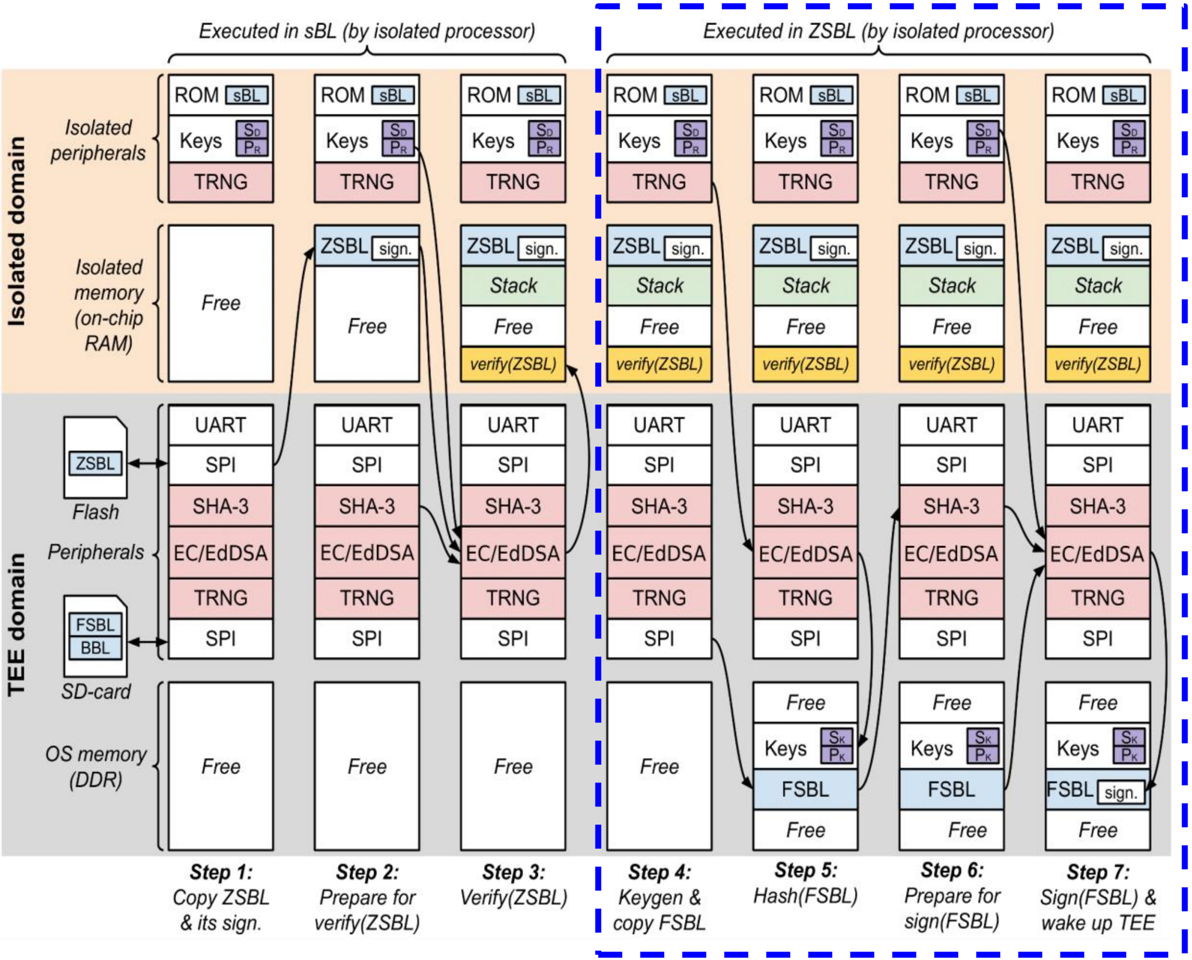
➔ Authenticated debug unit.

4. Secured Architecture (2/5) Secured boot flow



- ➔ The Zero Stage Bootloader (ZSBL) and its signature are copied into the isolated memory.
- ➔ System prepares for the verification of the ZSBL.
- ➔ The authenticity and integrity of the ZSBL are verified using cryptographic techniques and the stored signature.

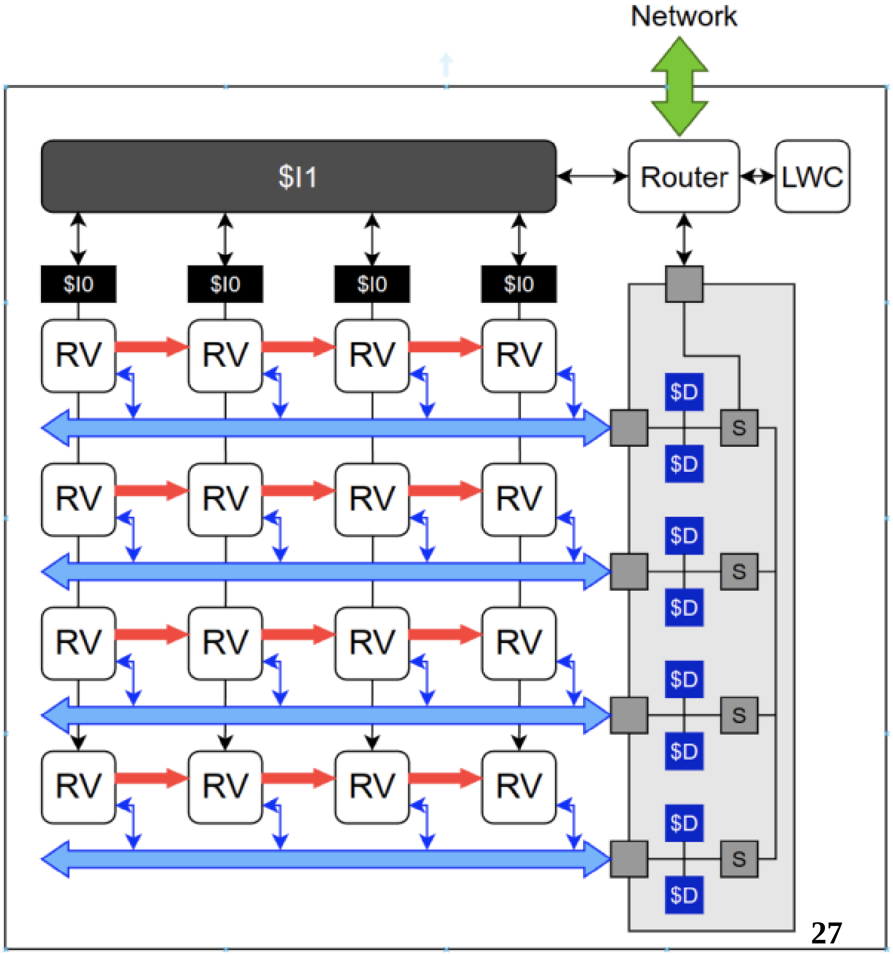
4. Secured Architecture (3/5) Secured boot flow



- ➔ The system generates the required cryptographic keys and copies the First Stage Bootloader (FSBL) into the isolated memory.
- ➔ A hash value is calculated for the FSBL to ensure its integrity.
- ➔ The system prepares for the signing of the FSBL using the generated keys.
- ➔ The FSBL is signed using the cryptographic keys, and the TEE domain is activated to continue the boot process.

4. Secured Architecture (4/5) Secured NoC

- ➔ Crypto-engines sign and verify for income/outcome package.
- ➔ Connect with global network through global router.
- ➔ The cores that share the common bus and are connected with each other by by shared bus and the tightly-coupled bridge (stream).
  - ◆ Bus is effective for small number of cores in a line.
  - ◆ Stream channel reduce the usage of bus by keeping data in process stream as long as possible.



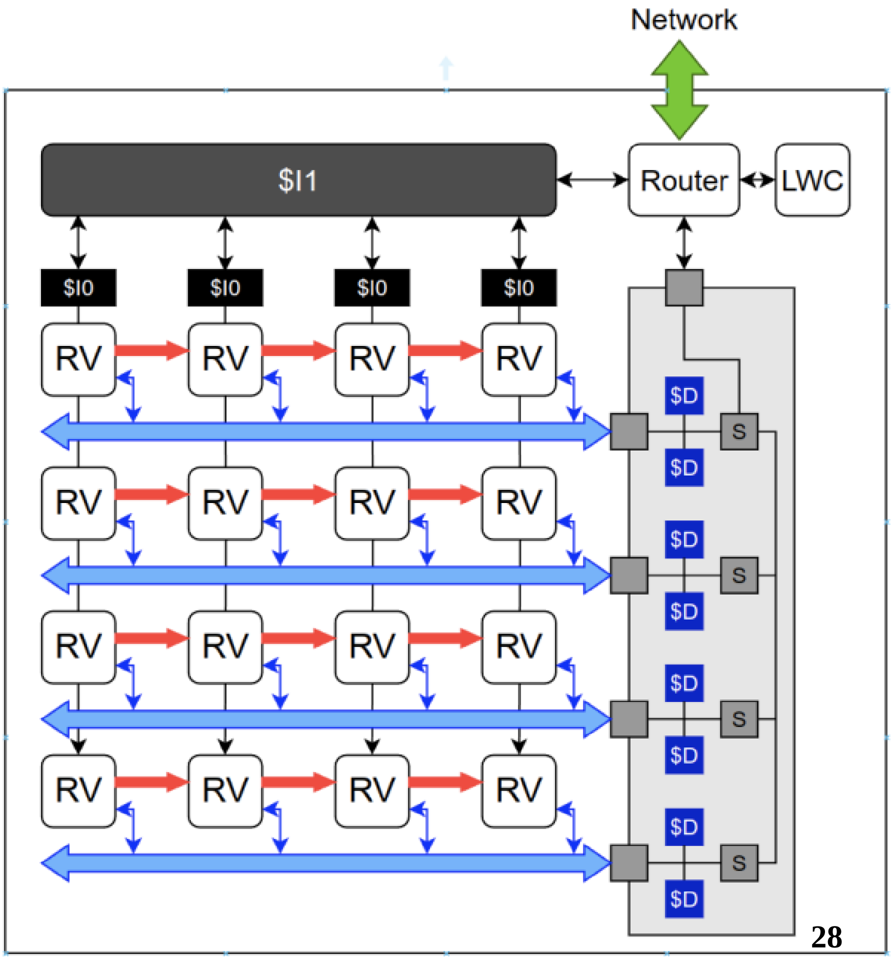
4. Secured Architecture (5/5) Secured NoC

The local network is organized as two level hierarchical architecture:

- ➔ **1st level:** multi-cores are connected with each other by a secured bus system.
- ➔ **2nd level**: secured router nodes are connected with each other through a crossbar.

The security of the network is ensure by the crypto-engines.

- ➔ Signed/verify for income/outcome package.
- ➔ Secure router nodes take responsible for the integrity of data of a line and prevent flooding attacks from this line.



Outline

1. Introduction
2. Many-core Architecture
3. Memory Bottleneck
4. Secured Architecture
5. Conclusion

5. Conclusion (1/1)

From this work, we address the challenges of modern computing:

➔ **Asymmetric cores:** promote the power efficiency as well as parallelism.

➔ **Data-centric techniques:** reduce power and latency of data exchange. The cache techniques promotes the bandwidth of cache for multi-processors, multi-threads system.

➔ **Robust security measures:** ensures data integrity and protect the chip over software and hardware treats.

THANK YOU

2024/06/06