SEC RISC-V Workshop 2020

Examination of applicability of RISC-V security specifications to low-end processors

Fumio Arakawa†, Akira Tsukamoto‡, Kuniyasu Suzaki §‡, Makoto Ikeda†

† Graduate School of Engineering, The University of Tokyo

‡ Cyber Physical Security Research Center (CPSEC), AIST

§ Technology Research Association of Secure IoT Edge application based on RISC-V Open architecture (TRASIO)

First International Workshop on Secure RISC-V Architecture Design Exploration (SECRISC-V'20)

SEC RISC-V Workshop 2020

Outline

- Introduction
- Overview of RISC-V
- Security Extension of RISC-V
  - Physical Memory Protection (PMP)
  - PMP implementation in Rocket Tile
- Area evaluation of RocketTile PMPs
- Comparison of PMP and TLB
- Conclusion

Introduction

- Trend of security enhancement even in low-end processors.
- Security of Large-scale system in the IoT era:
  - affected by the vulnerability of a low-end processor in a terminal device of the system.
- Low-end processors
  - Reducing area and power even by compromising on performance
  - It is important to reduce the cost for security.
- RISC-V (open instruction set architecture for general-purpose processors)
  - RISC-V is gaining attention and expected to spread to actual products.
  - We evaluated the area overhead of security extensions of RISC-V.

Overview of RISC-V

- A free and open ISA developed at UC Berkeley
  - Berkeley Architecture Research released processor cores of Rocket and BOOM.
- Maintained by RISC-V International (more than 500 members)
- Chisel: Original hardware construction Language
- Tilelink: Original standard for SoC interconnect
- Chipyard (Rocket-based SoC construction environment)
  - Tool chain that automatically translates codes from Chisel to Verilog
  - Easy to replace modules or to change various parameters
- Software
  - development environments: compilers, debuggers, simulators, etc.
  - runtime environments: boot loaders and OSes, etc.
- Many RISC-V ISA cores have been developed along with fulfilling its ecosystem.

What's Different about RISC-V?

- **Simple**
  - Far smaller than other commercial ISAs
- **Clean-slate design**
  - Clear separation between user and privileged ISA
  - Avoids µ-architecture or technology-dependent features
- A **modular** ISA
  - Small standard base ISA
  - Multiple standard extensions
- Designed for **extensibility/specialization**
  - Variable-length instruction encoding
  - Vast opcode space available for instruction-set extensions
- **Stable**
  - Base and standard extensions are frozen
  - Additions via optional extensions, not new versions

From:
RISC-V Foundation
Summit 2018

Security Extension of RISC-V

- Up to 3 levels of security mode
  - User (U) mode: For application
  - Supervisor (S) mode: For OS
  - Machine (M) mode: Highest privilege level
- Implementation types:
  - M mode only (simple embedded system)
  - M, U mode (secure embedded system)
  - All M, S, U modes (systems running Unix-like OS)
- Physical memory attribute (PMA) checker manages PMA of each area.
  - Check HW-specific attributes with HW
  - Check other attributes with M-mode SW
  - Inform attributes to U- and S-mode SW
- Physical memory protection (PMP) for security enhancement -- realized in M-mode
- Page-based virtual memory (VM) system is defined as S-mode function.
- Three types of VM: Sv32, Sv39, and Sv48 (32-, 39- and 48-bit address space)
- Page table walk (PTW) -- Defined to accelerate miss handling by HW

Physical Memory Protection (PMP)

SECRISC-V Workshop 2020

- PMP is managed by M-mode SW. -- A certain region privilege can be fixed by HW.
- PMP CSR: To control the PMP of read, write, and execute
  - Pair of 8-bit configuration CSR (pmpcfg) and XLEN (32/64) -bit address CSR (pmpaddr)
  - 16 entries at maximum.
  - Granularity is platform specific. -- Standard supports as small as 4 bytes.
  - Choose the entry with the minimum numbering if matching to multiple entries.
  - Each entry is accessible only in M-mode.
  - An entry with L-bit set cannot be updated, and is to be cleared by reset.

bit		description
7	L	Lock control (0: writable, 1: write is locked)
6-5		reserved
4-3	A	Address-matching mode 0: OFF Null region (disabled) 1: TOR Top of range 2: NA4 Naturally Aligned (NA) 4-byte region 3: NAPOT NA Power-of-two region, ≥8 bytes
2	X	Execution control (0: denied, 1: permitted)
1	W	Write control (0: denied, 1: permitted)
0	R	Read control (0: denied, 1: permitted)

TOR: An address A is matched when
(i-1)-th pmpaddr ≤ A < i-th pmpaddr for i-th entry,
0 ≤ A < 0-th pmpaddr for 0-th entry.

NA4: An NA address matches if it is in the 4-byte region of pmpaddr.

NAPOT: When bit j of pmpaddr is the first '0' from the LSB, an NA address matches if it is in the 2^{j+3}-byte region of the address made by clearing lower j bits of the pmpaddr to '0's.

Physical Memory Protection (PMP)

SECRISC-V Workshop 2020

- Access privilege is checked with R, W, X, and L and security mode.
- Access exception occurs for read, write, or execution if it is not permitted.

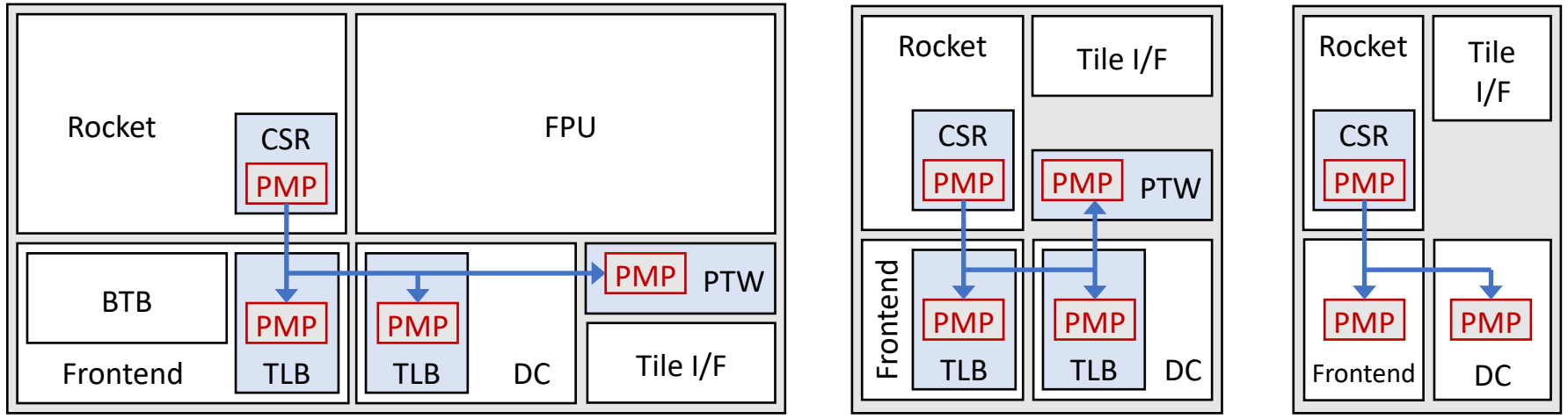
R	W	X	S-/U-mode	M-mode	
				L=1	L=0
0	0	0	– – –		r w x
0	0	1	– – x		r w x
0	1	0	reserved		reserved
0	1	1	reserved		reserved
1	0	0	r – –		r w x
1	0	1	r – x		r w x
1	1	0	r w –		r w x
1	1	1	r w x		r w x

r : read permission
w: write permission
x : execute permission
- : no permission

- Combination of R=0 and W=1 is not necessary, and reserved for future extension.
- Full access is permitted if L=0 in the M-mode.
- Full and no access is permitted if no entry matches in M- and S-/U-mode, respectively.

PMP implementation in RocketTile

- RocketTile: Rocket core conforming to Tilelink
- Chipyard: Supporting 5 configurations of Big, Mid and Small RV64 tiles, and Big and Tiny RV32 tiles -- Other types are configurable by setting parameters.
- PMP: Distributed and implemented in CSRs, TLBs and PTW
  8 entries (Half of the maximum), 30 bits of pmpaddr width
- Instruction/Data TLB: 32 entries each for Big cores, 4 entries each for the other cores
- VM: Enabled for Big and Mid cores
- MMU corresponds to TLB and PTW.



Difference between PMP and TLB (affecting to area)

	PMP	TLB
Mapping	Overlapped, choice by priority	Exclusive
Region size	4 Bytes or more, Arbitrary sizes	4 KiB or more, Sv32/39/48: 2/3/4 sizes, respectively
Entries	Defined as Architecture Max 16, Rocket: 8	Arbitrary numbers Big/Mid core: 32/4 each, respectively
Access control & # (place)	Physical address, bitwise mask, Full associative only, priority control, 3 (instruction/data/PTW)	Logical address, Set associative is selectable, 2 (instruction/data)
Output	Access permission	Physical address, Access permission, Page fault exception
Memory	Not used	Used for page tables
others	-	PTW HW

Area evaluation of RocketTile PMPs

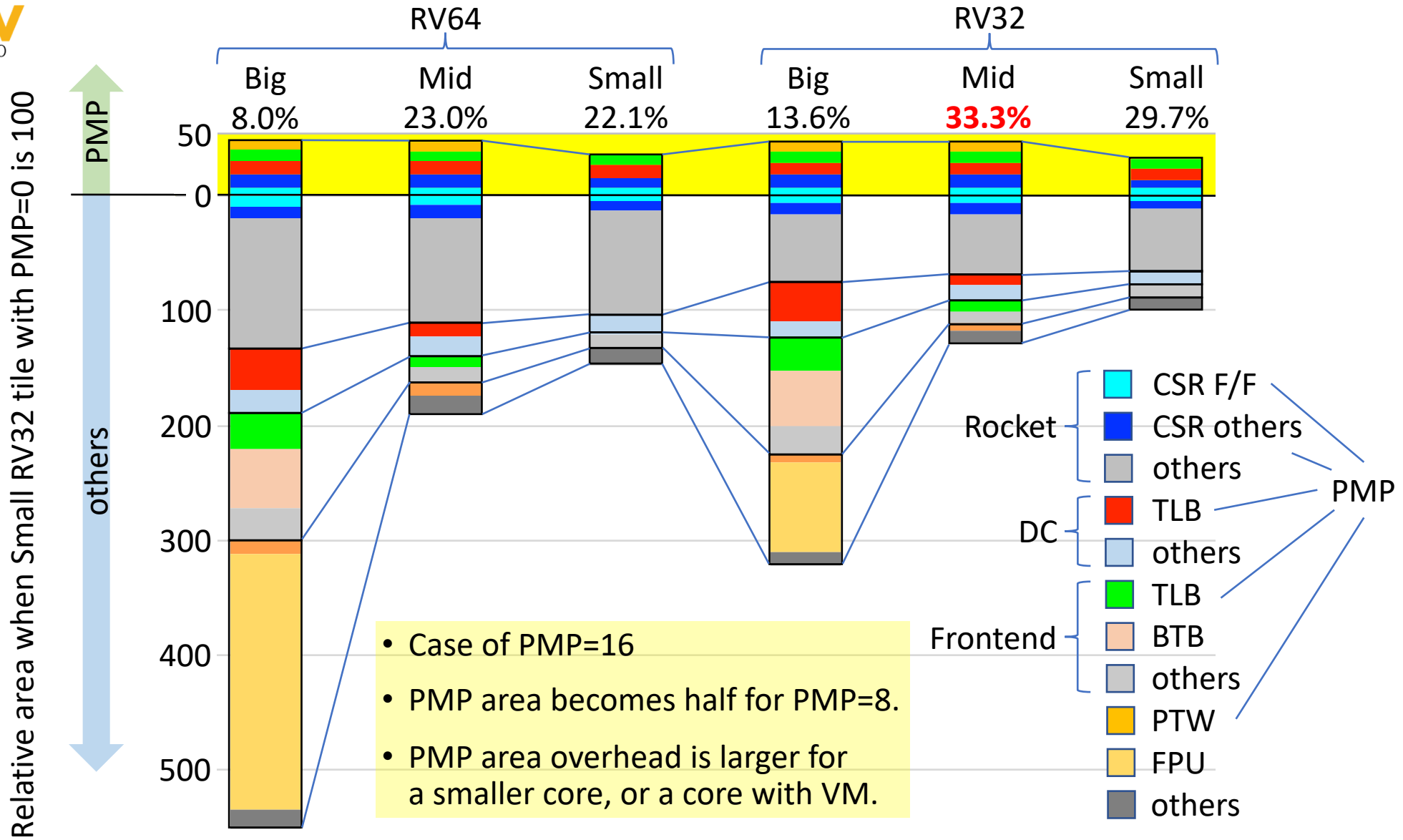
SEC RISC-V Workshop 2020

- Evaluated types of RocketTiles:
  - [RV64/32] x [Big/Mid/Small] => 2 x 3 = 6 types
  - PMP: 3 types of 0/8/16 entries
  - Total: 6 x 3 = 18 types

- with Chipyard
  - Prepare above by editing "Configs.scala"s of System & Subsystem of rocket-chip
  - Generate Verilog in verilator construction environments

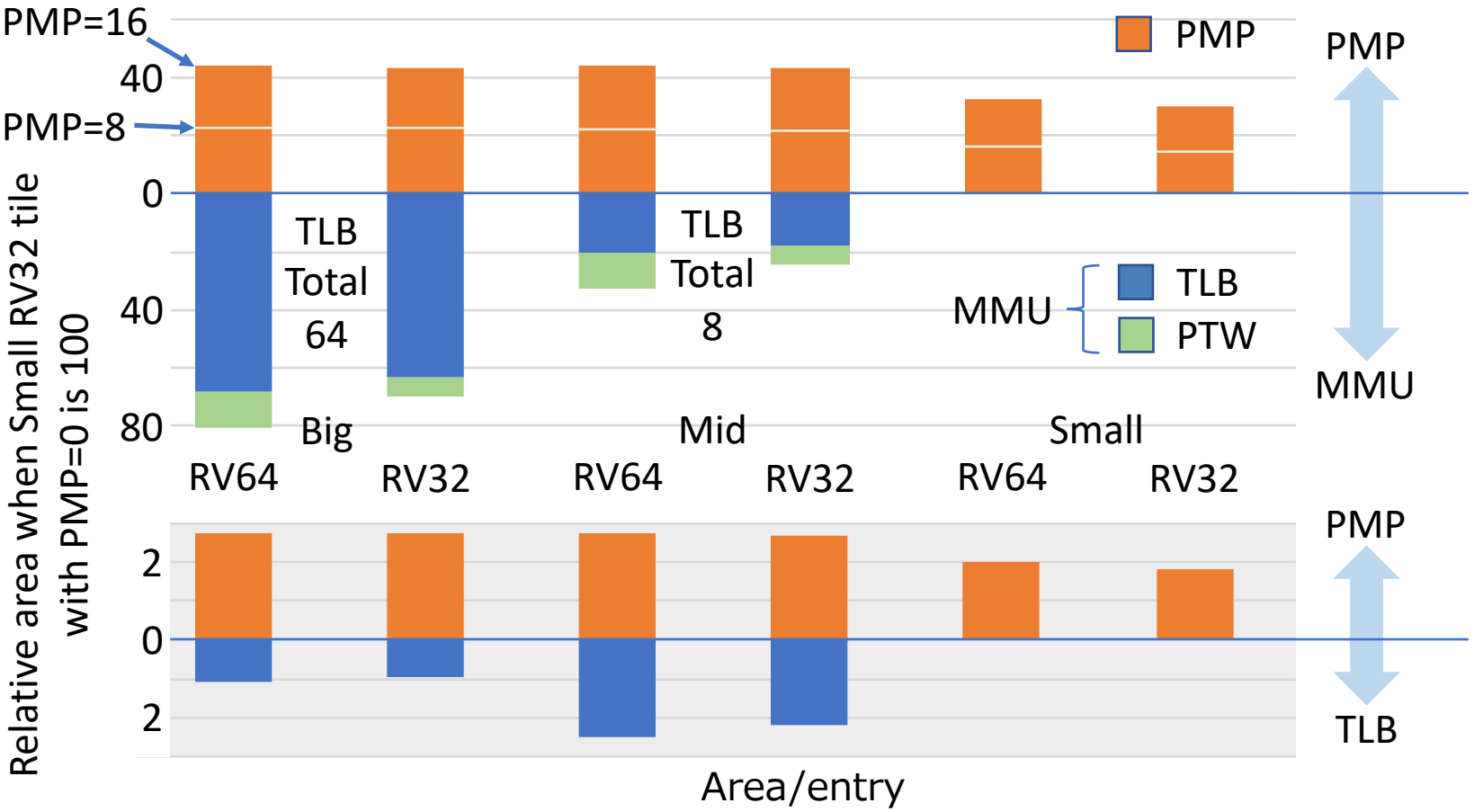
- with Synopsis Design Compiler
  - Specify RocketTile as top module
  - Synthesize gate-level verilog w/o hierarchical structure flattening
  - Get the module cell areas (not including RAM area)

Area evaluation of RocketTile PMPs



Comparison of PMP and TLB

- TLB: Set associative; The more sets, the higher efficiency; Efficient for Big core => 1.0/entry
- PMP: Low efficiency for full associative and bitwise mask; 2.8 & 2.0/entry for w/ & w/o VM



F/F Ratio of Each Module (%)

- F/F ratio of FPU, a computing intensive unit, is low, but that of PMP is lower.
- This means PMP has large processing logic per F/F.
- Conversely, TLB has small processing logic per F/F.

	RV64			RV32			
	Big	Mid	Small	Big	Mid	Small	Tiny
RocketTile	33.0	42.9	40.9	40.2	43.1	40.7	41.7
Rocket	33.5	38.9	38.4	36.2	37.5	36.4	36.8
CSR	44.0	44.3	41.9	43.1	42.8	41.0	41.0
Frontend	46.8	52.8	54.1	47.8	52.9	54.3	54.3
BTB	46.3			47.2			
TLB	**50.7**	**50.7**		**50.9**	**50.5**		
DC	44.2	40.3	34.1	46.1	43.3	37.3	39.2
TLB	**50.8**	**50.7**		**50.9**	**49.8**		
FPU	**20.5**			**27.4**			
PTW	**53.0**	**53.3**		**52.6**	**52.6**		
PMP 8	**15.8**	**15.9**	**21.9**	**15.5**	**16.1**	**24.0**	**22.4**
CSR	39.3	39.1	51.3	38.3	39.4	50.4	50.7
PMP 16	**15.6**	**15.8**	**21.2**	**15.8**	**16.0**	**23.2**	**22.4**
CSR	39.2	39.4	49.8	39.0	39.3	50.1	49.9

RocketTile without PMP

F/F ratio of TLB/PTW is very high.

F/F ratio: PMP (very low) < FPU (low)

SEC RISC-V Workshop 2020

Conclusion

- We worried an area overhead problem on a small processor caused by adding the PMP function for the security enhancement, and evaluated the influence on the Rocket tiles.
- Area overhead is a minimum of 4.0% for the Big RV64 tile with PMP=8,
  and a maximum of 33.3% for the Mid RV32 tile with PMP=16.
- There are smaller 2-/3-stage pipeline processors for the IoT, and the area overhead of the PMP function must be further serious.
- Area per entry of the PMP reaches to 2.8 when the VM is enabled, whereas that of the TLB is 1.0 to 2.5 depending on its configuration.
- PMP entries are 16 at maximum, but some say that it is not enough.
  A specification-level enhancement is necessary to increase the entries efficiently.
- Privileged Architecture of RISC-V is the phase of enhancing while updating Draft,
  and the specification is discussed in the Task Group.
- By joining to the discussion now, the specifications will fit to your field in the future.

Acknowledgment

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO) entitled "Research and Development of Fundamental Technology of Secure Open Architecture and its Application to Edge AI." The synthesis library and logic synthesis tool used in this study were provided for chip prototype service at System Design Laboratory of Graduate School of Engineering of the University of Tokyo.

Thank you for your attention!