



RISC-V自作プロセッサプロジェクト 「PRO^トO-V」始動



SUSSIKITIGA18

目次

1. これまでのあらすじ
2. PROTO-Vとは？
3. 目玉機能「Keystone Enclave」について

これまでの
あらすじ



昨年までの話

- 昨年の秋ごろ、友人と自作CPUを始める
 - 「OpenMPW^{テープアウト}に応募して、作ったCPUを無料で製造してみないか？」
- OpenMPWとは?
 - 誰でも無料でオリジナル半導体チップが作れる！
 - 実際にチップが製造されて手元に届く！
- 応募するために完成を目指すぞ～💪
-



昨年までの話

- 昨年の秋ごろ、友人と自作CPUを始める
 - 「OpenMPW^{テープアウト}に応募して、作ったCPUを無料で製造してみないか？」
- OpenMPWとは?
 - 誰でも無料でオリジナル半導体チップが作れる！
 - 実際にチップが製造されて手元に届く！
- 応募するために完成を目指すぞ～💪
 - 締切に間に合わず、あえなく空中分解… 😭



今年の話

「チーム再結
成！！！リ
ベンジじゃい



今年の話

「PROTO-V
」プロジェクト
始動！！！



「PROTO-V」 とは？



「PROTO-V」とは？

- オレたちの自作パイプラインプロセッサ
- 「プロトタイプ」と「RISC-V」のもじり
- RV32I,ZICSR,ZIFENCEI,ZICNTRを実装
- GPIO、UART、SPI通信ができる
- 目玉機能は「Keystone Enclave」

動作 デモ

開発スケジュール

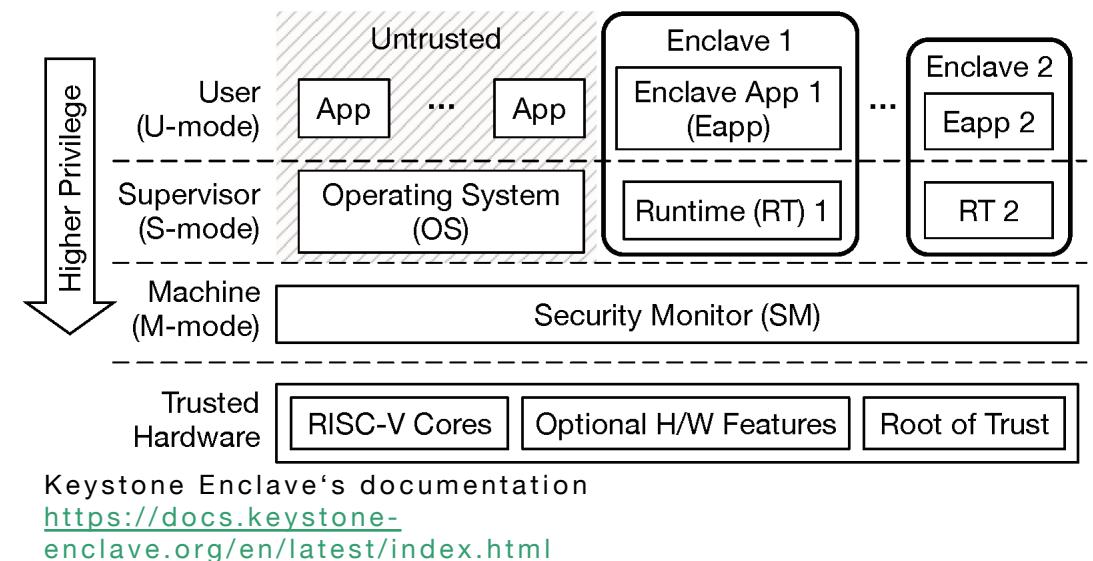
- ・今年度の三月

KEYSTONE ENCLAVE とは？



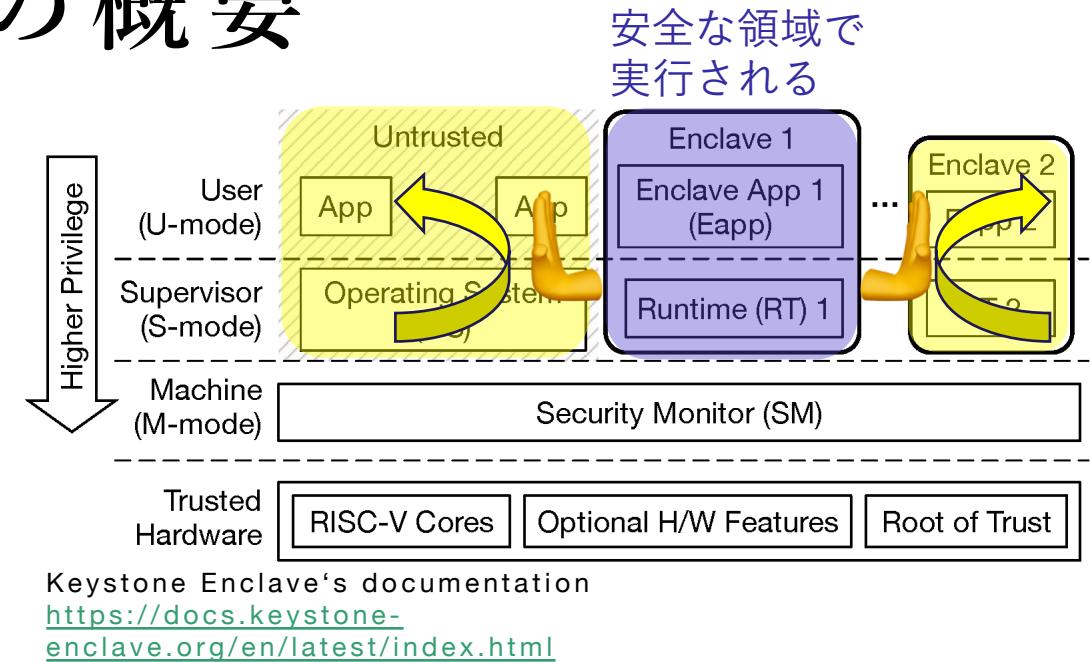
Keystone Enclaveの概要

- RISC-V向けのTEE(**Trusted execution environment**)を提供するフレームワーク。
- TEEとは、OSから隔離された領域(Enclave)を作つて安全に処理を行う秘密計算技術
- RISC-VのPMP(**Physical Memory Protection**)を使う
- PMPは、RISC-Vの**3つの特権モード**に対し物理メモリへのアクセス制御を行う機能



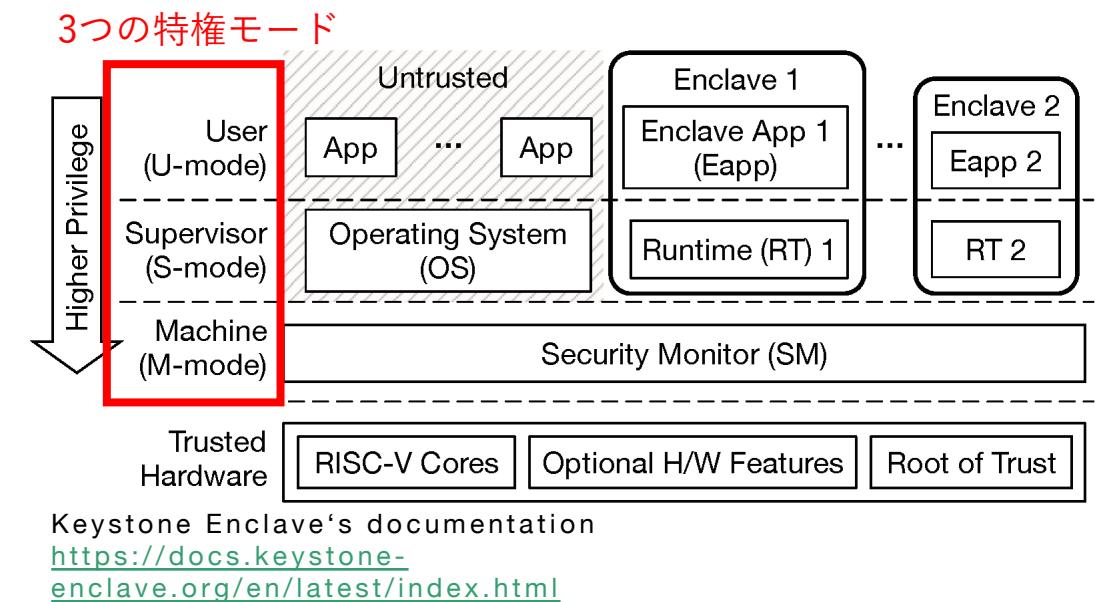
Keystone Enclaveの概要

- RISC-V向けのTEE(**Trusted execution environment**)を提供するフレームワーク。
- TEEとは、OSから隔離された領域(Enclave)を作つて安全に処理を行う秘密計算技術
- RISC-VのPMP(**Physical Memory Protection**)を使う
- PMPは、RISC-Vの**3つの特権モード**に対し物理メモリへのアクセス制御を行う機能



Keystone Enclaveの概要

- RISC-V向けのTEE(**Trusted execution environment**)を提供するフレームワーク。
- TEEとは、OSから隔離された領域(Enclave)を作つて安全に処理を行う秘密計算技術
- RISC-VのPMP(**Physical Memory Protection**)を使う
- PMPは、RISC-Vの**3つの特権モード**に対し物理メモリへのアクセス制御を行う機能



RISC-Vの3つの特権モード

1. U-mode(User mode):
ユーザアプリなど
2. S-mode(Supervisor mode):
OS,ブートローダなど
3. M-mode(Machine mode):
ファームウェア

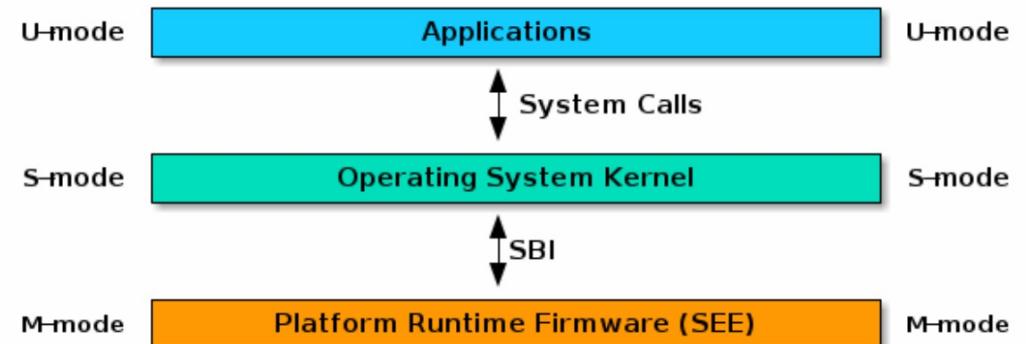


Figure 1. RISC-V System without H-extension

riscv-privileged-v1.10.pdf <https://riscv.org/wp-content/uploads/2017/05/riscv-privileged-v1.10.pdf>

PMP(Physical Memory Protection)

- PMPは、RISC-Vの3つの特権モードに対して物理メモリへのアクセス制御を行う機能
- S/Uモードによる物理メモリアクセスとMモードによるアクセスを分離
- 制御に使用するCSR(Control and Status Register)は二種類
 - pmpcfg : PMPの設定を行う
 - pmpaddr : アクセス制御する領域を指定

PMP entry とレジスタ構成

- PMP entryを2つのCSRで定める
- アーキテクチャによってEntryの最大数は決まっている
- 例えばRV32なら、4つのCSR(pmpcfg0~3)を使って16個のentryを設定(pmp0cfg~pmp15cfg)
- pmp(i)cfg(pmp0cfg~pmp15cfg)
 - 8bit構成
 - L: Lock bit.
 - A: Address matchingの方法を指定
 - R/W/X: Read,Write,Executeの許可
- pmpaddr(pmpaddr0~pmpaddr15)
 - 32bit構成
 - 物理メモリアドレスを指定

pmpcfg(i)

31	24 23	16 15	8 7	0	
	pmp3cfg	pmp2cfg	pmp1cfg	pmp0cfg	pmpcfg0
8		8	8	8	
31	24 23	16 15	8 7	0	
	pmp7cfg	pmp6cfg	pmp5cfg	pmp4cfg	pmpcfg1
8		8	8	8	
31	24 23	16 15	8 7	0	
	pmp11cfg	pmp10cfg	pmp9cfg	pmp8cfg	pmpcfg2
8		8	8	8	
31	24 23	16 15	8 7	0	
	pmp15cfg	pmp14cfg	pmp13cfg	pmp12cfg	pmpcfg3
8		8	8	8	

Figure 3.23: RV32 PMP configuration CSR layout.

pmp(i)cfg

7	6	5	4	3	2	1	0
L (WARL)	WIRI	A (WARL)	X (WARL)	W (WARL)	R (WARL)		
1	2	2	1	1	1	1	1

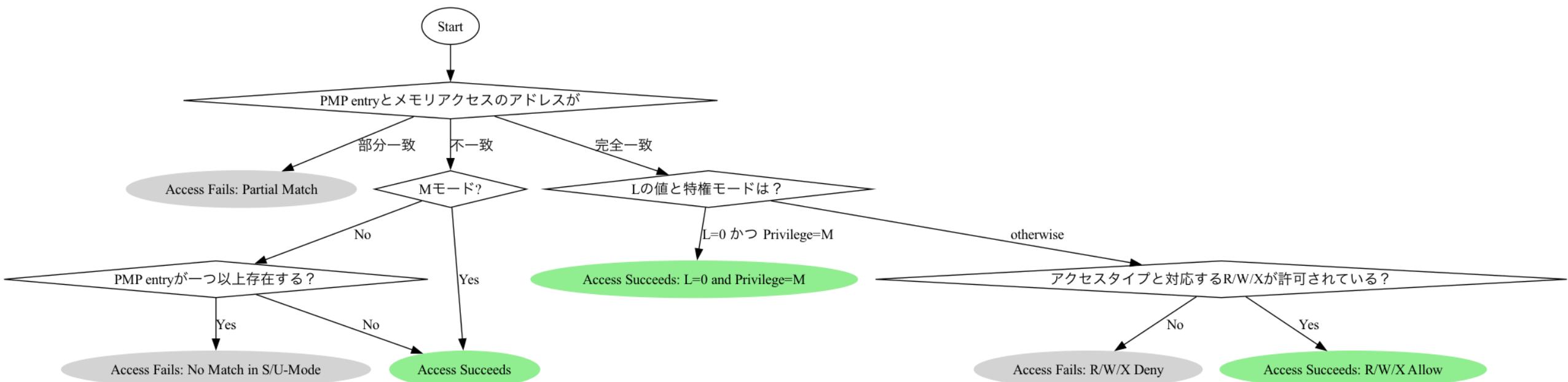
Figure 3.27: PMP configuration register format.

pmpaddr

31	address[33:2] (WARL)	0
32		

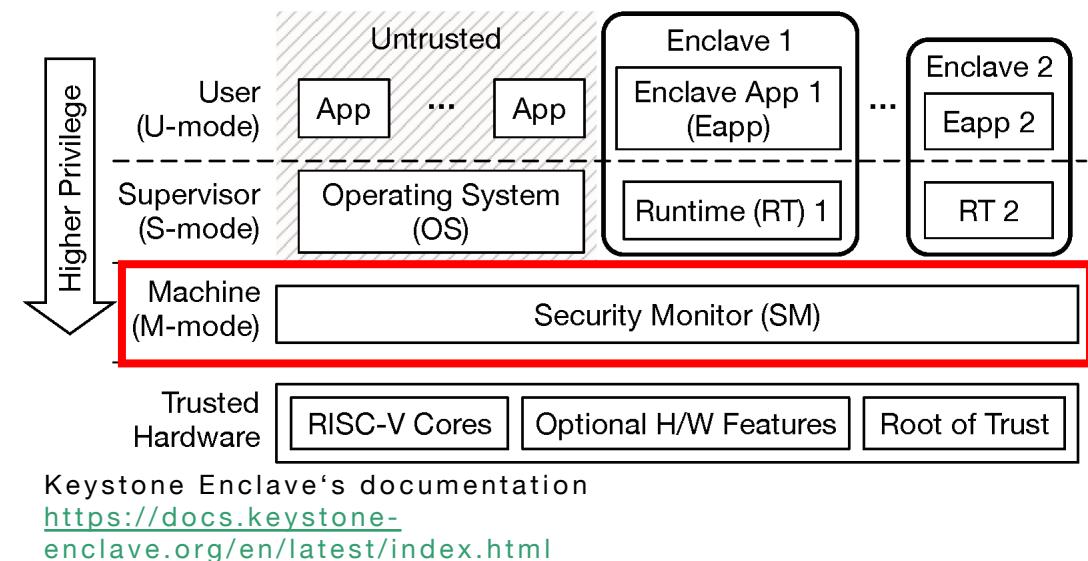
Figure 3.25: PMP address register format, RV32.

PMPによる物理メモリへのアクセス制御



SM (Security Monitor)

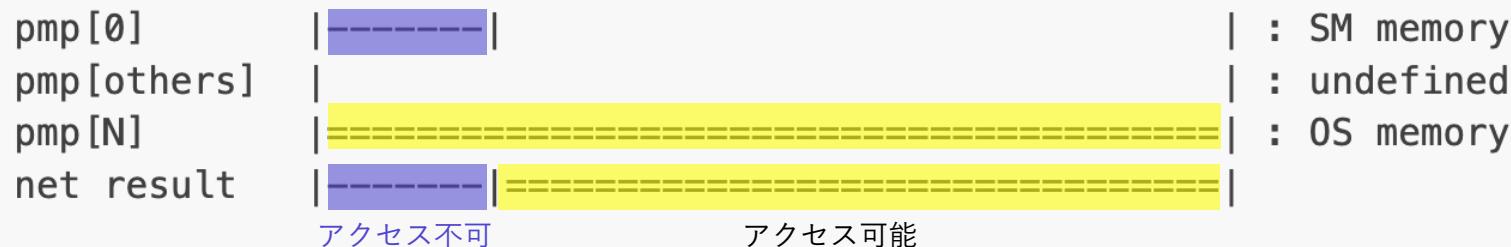
- Mモードで実行されるソフトウェア
- PMPを用いたメモリ分離を行い、Keystoneにおける安全な処理を実現
- Enclaveの作成、実行、破棄の際にPMP権限の変更を行う
- SMは少なくとも2つのPMP entryを使用
 - SM memory
 - Untrusted memory(OS memory)
- それ以外のPMP Entryは、各Enclaveに対し専用で割り当てられる



SM(Security Monitor)

- Security Monitorは、以下の2つのPMP entryを自身に使用
 - SM memory
 - Untrusted memory(OS memory)

-: inaccessible (NO_PERM), =: accessible (ALL_PERM)



SM(Security Monitor)

- SMは作成したEnclaveに対しPMP entryを割り当てる

-: inaccessible (NO_PERM), =: accessible (ALL_PERM)



SM (Security Monitor)

- SMがEnclaveを実行する際は、以下の二つのアクセス権限が反転
 - enclave memory : アクセス不可→アクセス可へ
 - OS memory : アクセス可→アクセス不可へ
- OS memory空間上にEnclaveと通信を行うバッファ(Untrusted shared buffer)を割り当て可能

-: inaccessible (NO_PERM), =: accessible (ALL_PERM)



まとめ

- ・俺たちは自作CPUにKeystoneを実装したいよ
- ・KeystoneはOSから隔離された領域（Enclave）を作つて安全に処理を行うRISC-Vの秘密計算技術だよ
- ・RISC-VにはU,S,Mモードが存在し、それらに対しメモリアクセス制御を行うPMPという機能があるよ
- ・Keystoneでは、Mモードで動作するSecurity MonitorがPMP entryの設定を書き換えて、Enclave appを安全に実行するよ

参考文献

- riscv-privileged-v1.10.pdf <https://riscv.org/wp-content/uploads/2017/05/riscv-privileged-v1.10.pdf>
- RISC-V Supervisor Binary Interface Specification
<https://www.scs.stanford.edu/~zyedidia/docs/riscv/riscv-sbi.pdf>
- Keystone Enclave's documentation <https://docs.keystone-enclave.org/en/latest/index.html>
- Make Your Own Chips for Free https://efabless.com/open_shuttle_program