RISC-V TEE I/Oまわりの状況

早稲田大学 木村啓二

自己紹介

- マルチコアアーキテクチャ・自動並列化コンパイラの研究に従事
  - OSCAR自動並列化コンパイラ
  - 各種サーバマルチコア，組み込みマルチコアでのアプリケーション並列性能評価
- 最近はセキュアなコンピュータシステムの研究も
  - 不揮発性メインメモリのメモリプロテクション
  - Trusted Execution Environment (TEE)
  - 今回はこのあたりの話

本日の内容

- RISC-V TEEのI/O周りの状況を紹介
  - まずはRISC-V PMPとKeystoneの復習
  - SiFive WorldGuard
  - IOPMP
  - IOMMU
  - 及び関連の話題
- メモリ隔離（保護）と割り込みの観点から

Trusted Execution Environment (TEE)に関して極めて簡単におさらい

- 信頼可能な（隔離された）プログラム実行環境
  - ごく限られた信頼できるもののみに依存してプログラムを実行する
  - OSも信用しない
- 要素技術
  - 隔離されたメモリ空間
    - OSが管理する通常の仮想記憶では不足
    - **今回はこれをI/Oの観点からみていく**
  - メモリの完全性保証
  - Attestation

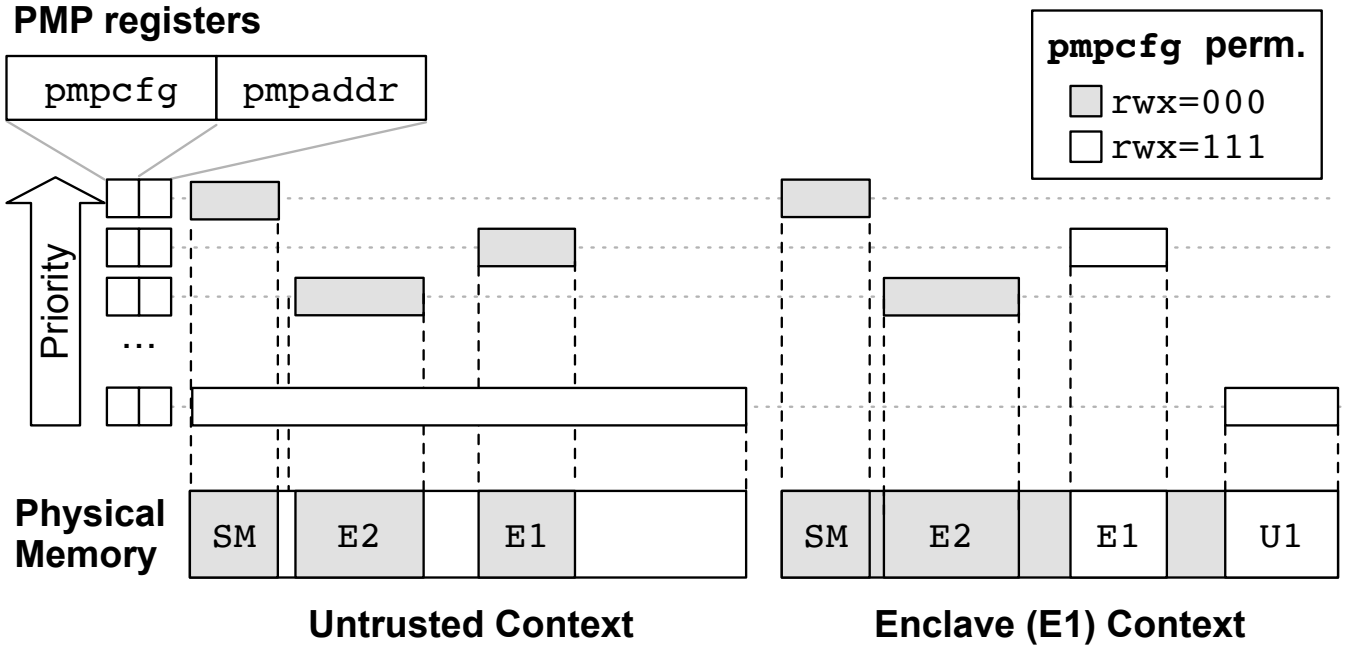
RISC-Vの物理メモリ保護関連の復習
特権モード

- 3つの特権モード
  - U-mode (user)
    - ユーザアプリが動作するモード
  - S-mode (supervisor)
    - OSが動作するモード
  - M-mode (machine)
    - ファームウェアやハイパーバイザが動作するモード
      - RISC-V Linuxはここで動作するOpenSBIを利用する
    - TEEを利用する場合はここでSecurity Monitor (SM)が動作する

RISC-Vの物理メモリ保護関連の復習 Physical Memory Protection (PMP)

- RISC-Vのメモリ隔離機構
- 指定した物理アドレス領域のU/Sモードに対するアクセス権限を設定する
  - 読み(r)・書き(w)・実行(x)
  - サイズは2^n (NAPOT)か任意のサイズの領域 (TOP)か選べる
  - 二つ一組のレジスタで領域を指定する（pmpcfg, pmpaddr)
    - 命令セット仕様書では64組（もともとは16組）まで持つことが出来る
- **各コアが持つ**
  - コンテクストスイッチ時にIPIでコア間の一貫性を維持する必要がある

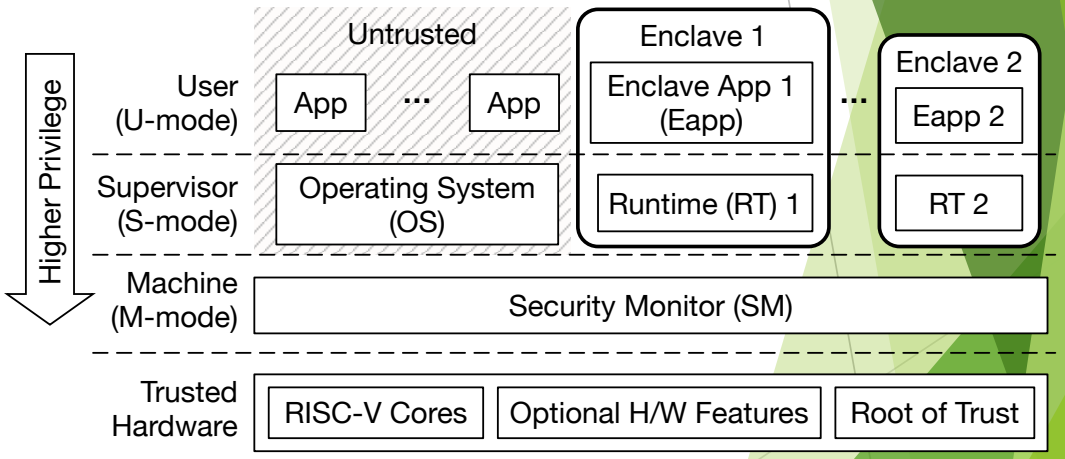
RISC-Vの物理メモリ保護関連の復習
PMPによる物理メモリ保護の様子



[Keystone]より引用

RISC-Vの物理メモリ保護関連の復習 Keystone

- 隔離された実行環境をEnclaveと呼ぶ
  - Enclave内部で動作するアプリケーションをEappと呼ぶ
- Enclaveが利用するメモリ領域にはOSも手出しできない
  - OSの代わりにRuntime (RT)がEappの実行をサポートする
- Eapp起動時は、そのEnclaveが利用するメモリ領域のPMP設定をSMで行う
  - Eappのコンテクストスイッチの際もSMでPMPの設定変更を行う



[Keystone]より引用

現状、何が問題か？

- I/Oは考慮されていない
- PMPはCPUコアの持つ機能
  - コアから物理メモリへのロード・ストアリクエストの可不可はコア内部のPMPによってチェックされる
  - メモリ側はノーチェック
  - DMA等のCPUコア以外のバスマスタはメモリアクセス し放題
  - I/O側へのアクセスもノーチェック
- 割り込みもEappは受け取れない
  - 現在の実装では、割り込みは一度SMで受け取ってからホストOSに渡される
  - 受け取れたとしてもEnclaveのコンテクストスイッチは重い

（RISC-Vではないけれど）事例紹介
GPU内部データの盗聴

LeftoverLocals: Listening to LLM responses through leaked GPU local memory

POST JANUARY 16, 2024 2 COMMENTS

By Tyler Sorensen and Heidy Khlaaf

We are disclosing LeftoverLocals: a vulnerability that allows recovery of data from GPU local memory created by another process on Apple, Qualcomm, AMD, and Imagination GPUs. LeftoverLocals impacts the security posture of GPU applications as a whole, with particular significance to LLMs and ML models run on impacted GPU platforms. By recovering local memory—an optimized GPU memory region—we were able to build a PoC where an attacker can listen into another user's interactive LLM session (e.g., llama.cpp) across process or container boundaries, as shown below:

https://blog.trailofbits.com/2024/01/16/leftoverlocals-listening-to-llm-responses-through-leaked-gpu-local-memory/

RISC-V界隈の動き
security@lists.riscv.orgより

Physical Memory Protection schemes (IOMMU/IOPMP/CVM/WorldGuard etc)

Krste Asanovic

[Mark - Not sure of how to get this to all the right lists]

Physical-Memory Isolation Techniques

Let me try and clarify the various physical-memory isolation mechanisms folks are discussing here.

The exec summary of the following is that I believe there is a small coherent set of complementary non-overlapping mechanisms that can be composed to support all the required capabilities. We should structure the TGs around the mechanisms rather than the use cases, as each use case needs more than one mechanism and we don't want duplication of similar mechanisms. I avoid explicitly talking about security as the real abstractions here are software contexts and privilege modes - security software architectures are built on top of these. I wrote this as a summary rather than trying to respond to all the email threads.

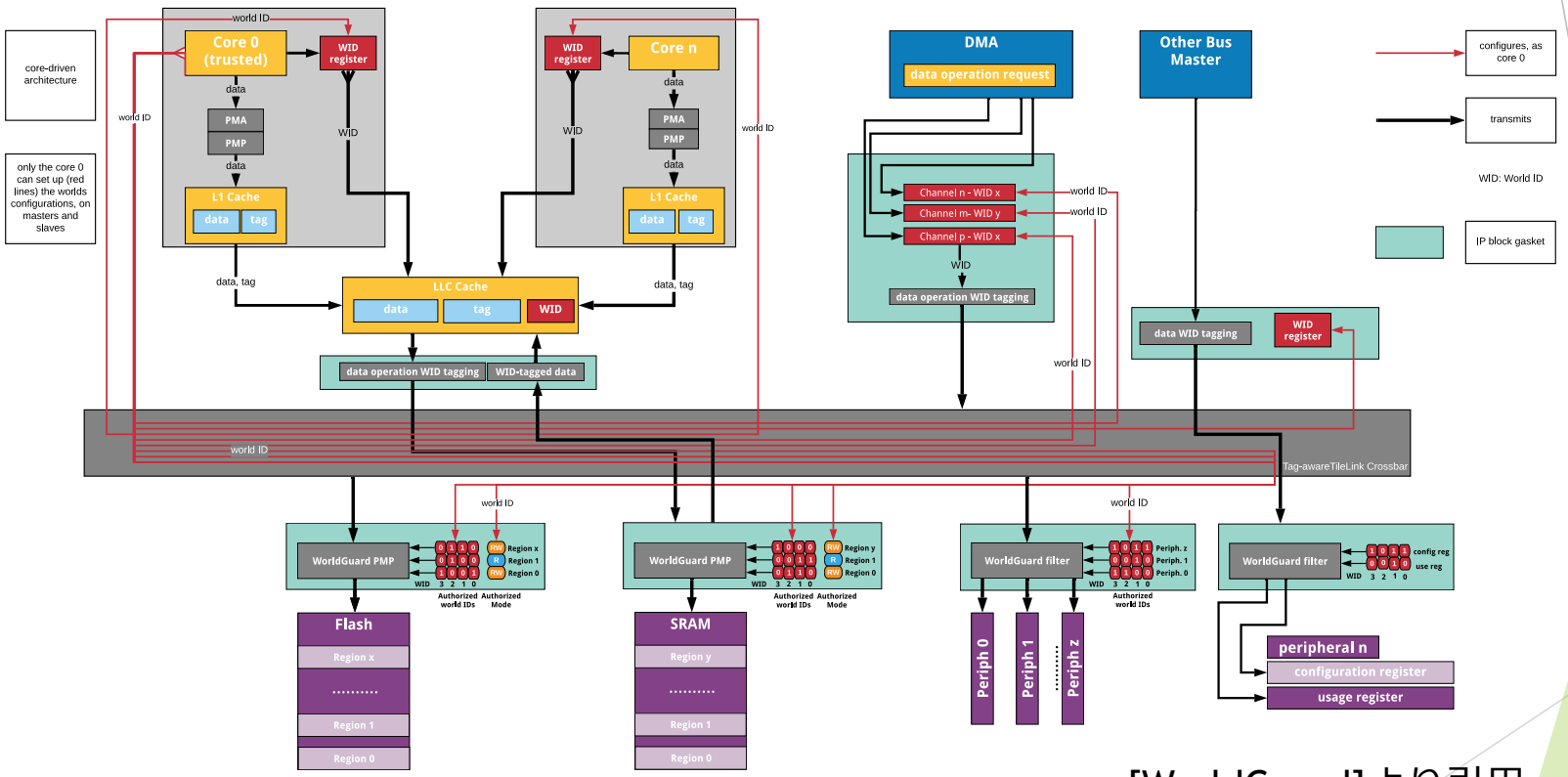
本日の本題

- I/Oも含めてTEEを構築するような仕組みは？
- 以下のような提案がある
  - SiFive WorldGuard
  - IOPMP
  - IOMMU
- 先のProf. Asanovicの文面に倣い、バスリクエストの要求元をsource、宛先をtargetとしてみる
  - 古来よりmaster/slaveなどと言っていますが…

SiFive WorldGuard

- SiFive策定による仕様
- 基本的なアイデア
  - トランザクションに付与したタグの一致・不一致で処理の可不可を判別する
- タグの値がWorldを示す
  - 各Worldはそれぞれ隔離された空間を表現する
- Source/Targetにそれぞれが所属するWorldのタグを事前に設定しておく
- Souceはリクエスト発行時にタグを付与する
- Targetはリクエストに付与されたタグと自分のタグから対応するPMPをチェックし、そのトランザクションを処理するかどうかを決定する
- ARM TrustZoneを拡張したものと考えられる
  - リクエストにNSビットを付与し、secure/non-secureを区別する

WorldGuardを持つSoCのブロック図例



[WorldGuard]より引用

WorldGuard
Core-driven

- シンプルなモデル、ハードウェアの修正も最小限
- コアは固定的にWorldと結びつけられる
  - U/S/M各モード問わない
- L2/L3キャッシュのタグアレイはWorldのタグも保持する
  - あるWorldがアクセスしたラインは他のWorldはアクセス不能
  - L1キャッシュは修正無し

WorldGuard
Process-driven

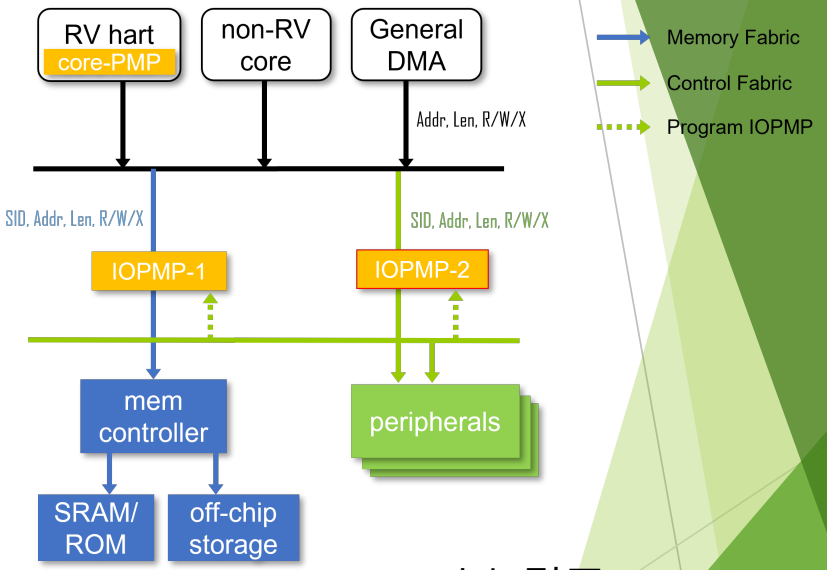
- プロセスがWorldと結びつけられる
  - 特権モード毎にも別々のWorldタグを持つ
- 各コアは自分が所属しうるWorldのタグリストを割り当てられる
  - 割り当てはシステム中の特権コアにより行われる
  - MモードはSモードのWorldタグを、SモードはUモードのWorldタグを、それぞれリストの中から設定可能
- **L1キャッシュのタグアレイもWorldタグを保持する**
  - 一つのコアが複数のWorldに所属しうるから

WorldGuard
割り込みの扱い

- Core-Local Interrupt (CLINT)
  - タイマ割り込みに関連するレジスタ（mtimecmp）が保護される
- Platform-Level Interrupt Controller (PLIC)
  - Process-drivenが前提
    - 各特権モードは別々のWorldに所属する
  - SモードのOSで処理される割り込みハンドラのアドレステーブルはMモードにてPMPを使ってロックされる
  - PLICからの割り込みは一度Mモードで受け付ける
  - 割り込みの発生元に応じた割り込みハンドラを選択してSモードに遷移する
  - 複数のEappが動いていて、それぞれ割り込みを受け付けたい場合はどうする？

IOPMP

- 2023年12月時点でVersion1.0.0-draft5
- Sourceとバスの間に接続するモジュール
- 各sourceはid (SID)を持つ
- IOPMPは内部にRISC-V PMPと同様のアドレス・コンフィグの配列(IOPMP ARRAYを持つ)
  - SIDとIOPMPでリクエストをバスに流すかどうか決める



[IOPMP]より引用

IOPMP
どうかな、と思うところ

- 仮想記憶は考慮されず
- 割り込みに関する記述もない
- 設定変更はどのコアのどのモードからも可能なように見える
  - 設定変更の手順として，一度全てのトランザクションをストールしてから変更する，といったことは書いてある
  - ブート時にPMPで保護してSMだけが操作できるようにすれば良いか
- ベアメタルに近い状況では良いのかも

IOPMP
オープンソース実装：Protego

ETH zürich ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

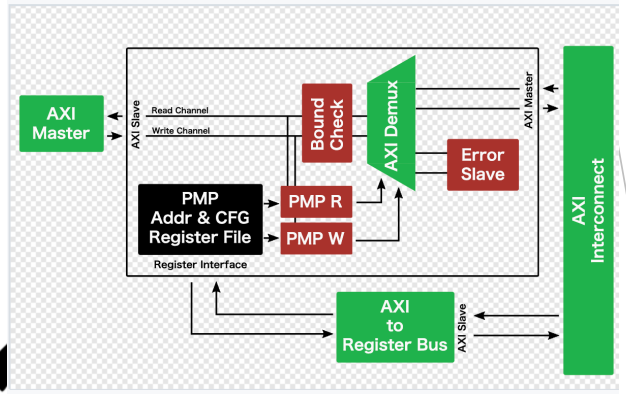
Protego: A Low-Overhead Open-Source I/O Physical Memory Protection Unit for RISC-V

SSH-SoC 2023, July 9th, 2023

Integrated Systems Laboratory (ETH Zürich)

Nils Wistoff	nwistoff@iis.ee.ethz.ch
Andreas Kuster	mail@andreaskuster.ch
Michael Rogenmoser	michaero@iis.ee.ethz.ch
Robert Balas	balasr@iis.ee.ethz.ch
Moritz Schneider	moritz.schneider@inf.ethz.ch
Luca Benini	lbenini@iis.ee.ethz.ch

PULP Platform
Open Source Hardware, the way it should be!



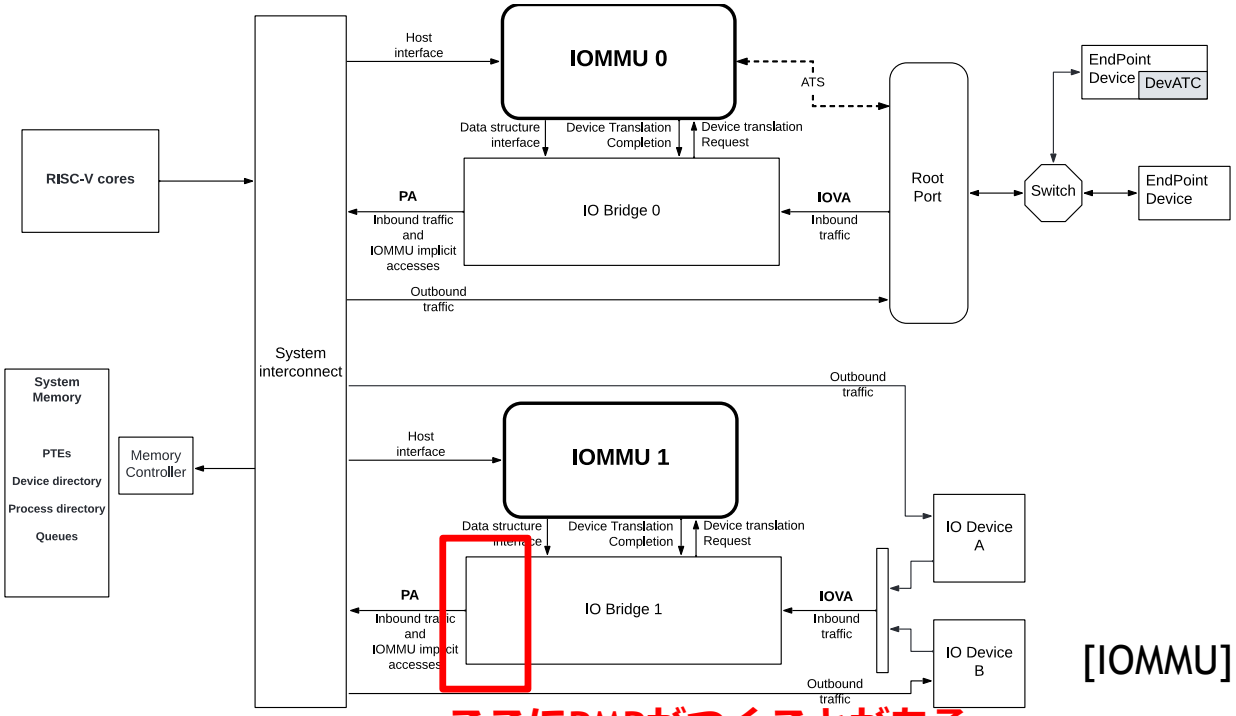
@pulp
pulp-pl
youtube.com/pulp

githubのURL

IOMMU

- 2023年6月Version 1.0がRatified
- I/Oデバイス（主にDMA）用のMMU
- Source側のデバイスが物理アドレスではなく論理アドレスで主記憶アクセスを行う
  - 例：GPU（のDMA）がホストメモリからデバイスメモリにデータを転送するとき、DMAはホストメモリに論理アドレスでアクセスする
- IOMMUが使うページテーブルがI/Oデバイス毎に設定でき、それらがOSから適切に保護できればI/Oもメモリ隔離できる
- IOMMU自体はRISC-V特有ではない
  - 仮想化技術で活用される
    - ゲストの物理アドレスをホストの物理アドレスに変換する
    - おそらく第3回（次回）詳しく説明していただけるのではないかと

IOMMU
IOMMUを持つRISC-V SoCのブロック図の例

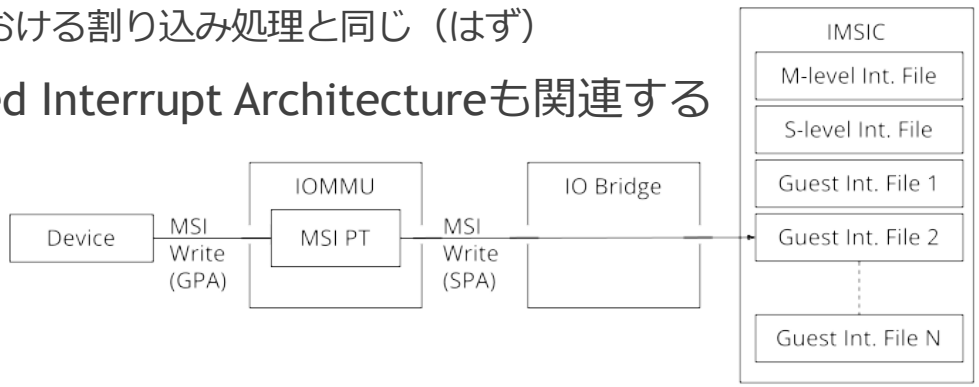


[IOMMU]より引用

ここにPMPがつくことがある

IOMMU
割り込み

- MSI（割り込み通知はメモリ書き込みトランザクションで処理）
  - 割り込みに応じたメモリ上の割り込みファイルに書き込まれる
  - IOMMUのアドレス変換により実際の書き込み先を操作できる
    - 割り込みとEnclaveを紐付けておけば、特定のEnclaveで割り込みを受け取れる（はず）
    - 仮想化における割り込み処理と同じ（はず）
- RISC-V Advanced Interrupt Architectureも関連する [INTERRUPT]



ここから先は
コントローラで
CPUコアに通知する

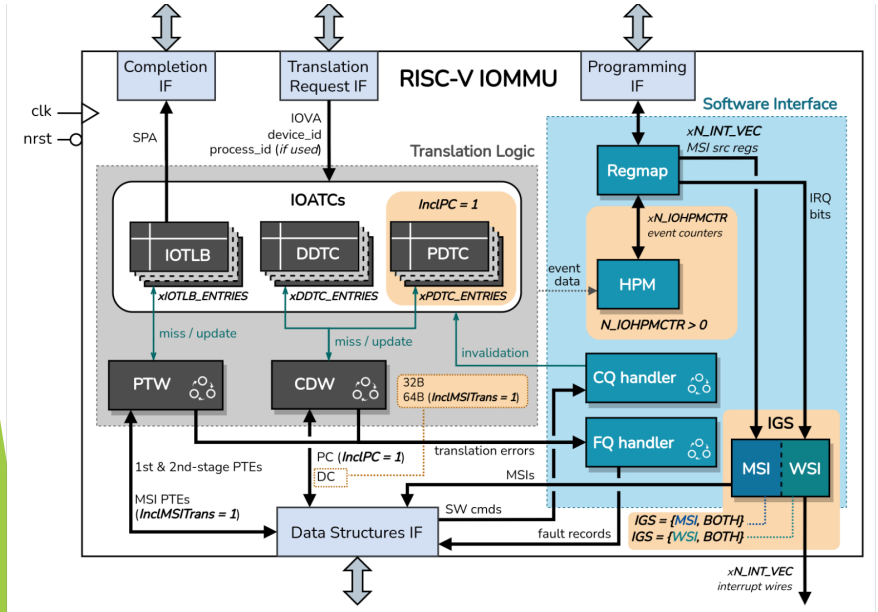
[IOMMU]より引用

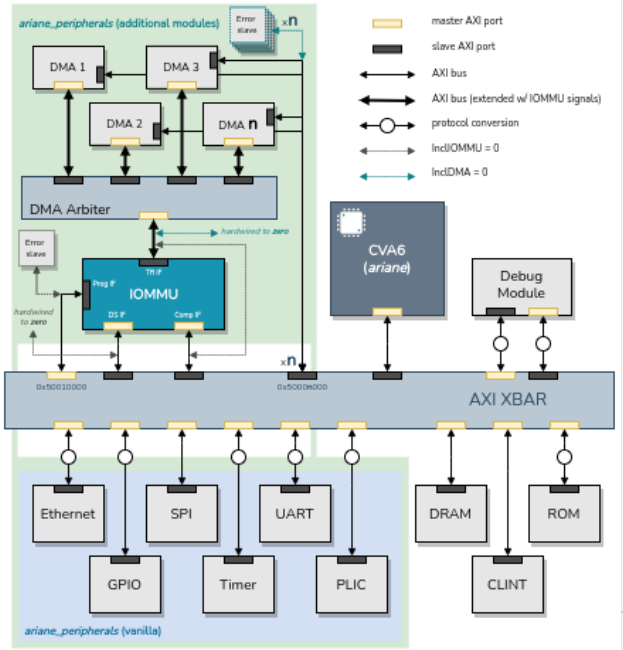
IOMMU
設定更新

- 設定更新に関しては特に記述無し
- こちらもブート時にPMPで保護してSMだけが操作できるようにすれば良いか

IOMMU
オープンソースのHW実装（Zero-Day Labs）

- https://github.com/zero-day-labs/riscv-iommu
- これとは別にQEMUのパッチも出ている



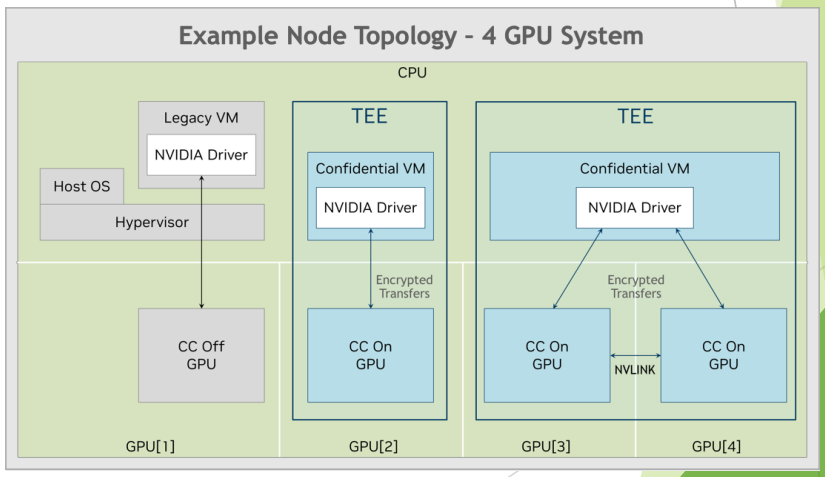


Intel SGXのI/Oは？ 他のCPUは？

- Intel SGX
  - I/Oは扱わない
  - 隔離メモリ領域であるEPC (Enclave Page Cache)はDMA禁止
  - ファイルとかデータ出力したい場合はSealing（暗号化）してホスト側に渡す
- ARM TrustZone
  - Secure World側にI/Oデバイスを配置できればその範囲内で利用可能（のはず）
  - デバイスドライバとかSecure Worldに配置したくなければIntel SGXと同じ扱い
- AMD SEV
  - 仮想化技術ベースなので、そのやり方でI/Oも扱う（はず）
  - AMD SEV-TIO (Trusted IO)という提案もある（white paperが2023年3月に発行）
    - I/Oチャネルの暗号化ベースの隔離技術
  - 最近提案されたIntel TDXもSEV/SEV-TIOと同じような仕組み

（RISC-Vではないけれど）GPUのTEE
NVIDIA H100

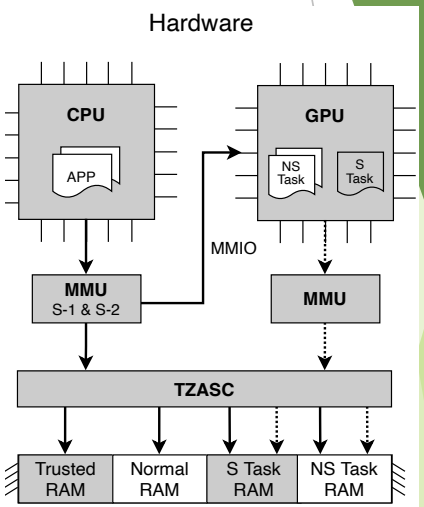
- ホスト・デバイス間の暗号化も含めた安全な通信路の確立
  - DH鍵交換使ったり
- GPUそのものは、隔離したいカーネル実行時にはそれ専用にロックされる
- Gravitonかな？
  - S. Volos, et al., “Graviton: Trusted execution environments on GPUs”, OSDI 2018.



[H100CONF]より引用

（RISC-Vではないけれど）論文紹介
StrongBox

- Y. Deng, et al., “StrongBox: A GPU TEE on Arm Endpoints”, Proc. of CCS’22, November, 2022.
- 統合型GPU (GPGPU)用のTEE
- ARM Cortex-Aシリーズを前提とする（HW無改造）
  - TrustZoneによるメモリの隔離
  - 2段階アドレス変換の利用
    - 2段階目の変換でページフォルトを起こして、アクセス権限のチェックを行う
- なるべく既存のデバイスドライバを利用する
  - GPU kernel用のデータはsecure側で暗号化して normal側のドライバに渡す



まとめ

- RISC-V TEEのI/Oまわりの状況を紹介
  - SiFIVE WorldGuard, IOPMP, IOMMU
  - IOMMUに関してはQEMUのパッチも提供されている
- RISC-V以外の話題も少しだけ紹介
- 個人的な興味
  - （実用レベルの）ハードウェア実装はどうなるか？
    - IOMMUをサポートするPCIeのIPは誰か作るか？
  - TEE側で割り込みを低コストで扱えるか？
  - デバイスドライバは簡単な拡張ですむのか？

References

- [Keystone] Dayeol Lee, et al., “Keystone: An Open Framework for Architecting Trusted Execution Environments”, EuroSys’20, April, 2020
- [WorldGuard] SiFive, “SiFive WorldGuard White Paper Version 1.2”, 2020
- [IOPMP] RISC-V IOPMP Task Group, “RISC-V IOPMP Architecture Specification”, Version 1.0.0-draft5, December, 2023
- [IOMMU] RISC-V IOMMU Task Group, “RISC-V IOMMU Architecture Specification”, Version 1.0, 06/2023
- [INTERRUPT] John Hauser, “The RISC-V Advanced Interrupt Architecture”, Version 1.0, June, 2023
- [H100CONF] NVIDIA, “Confidential Compute on NVIDIA Hopper H100”, WP-11459-001, July, 2023