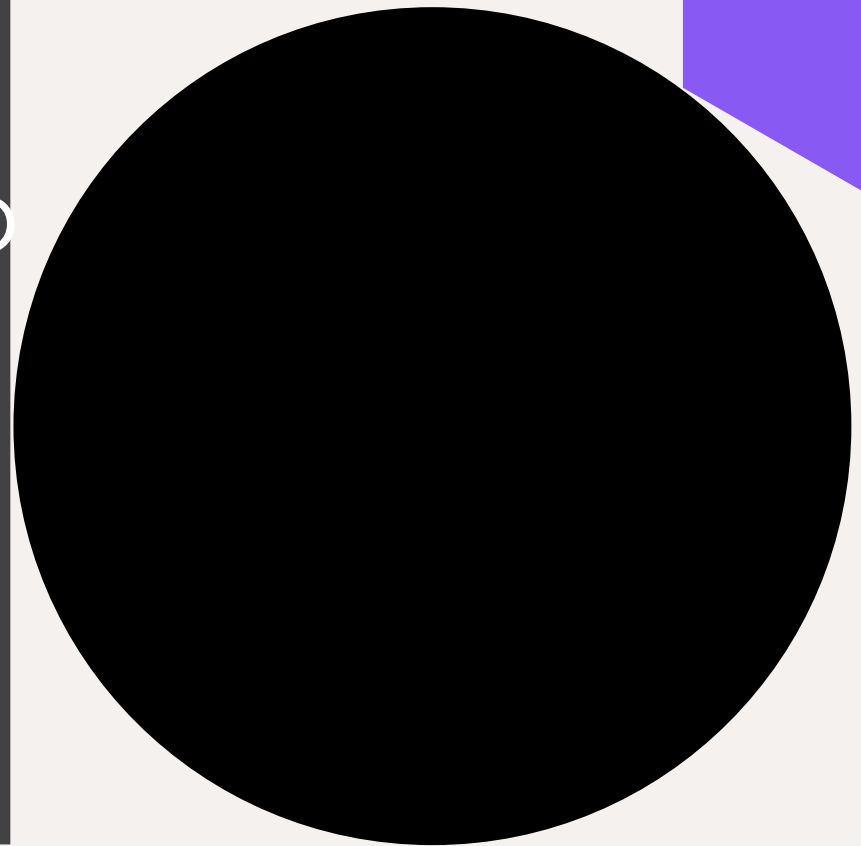
→ CHERIでメモリ安全性を確保するRISC-Vソリューション と カスタム・コンピュートの薦め

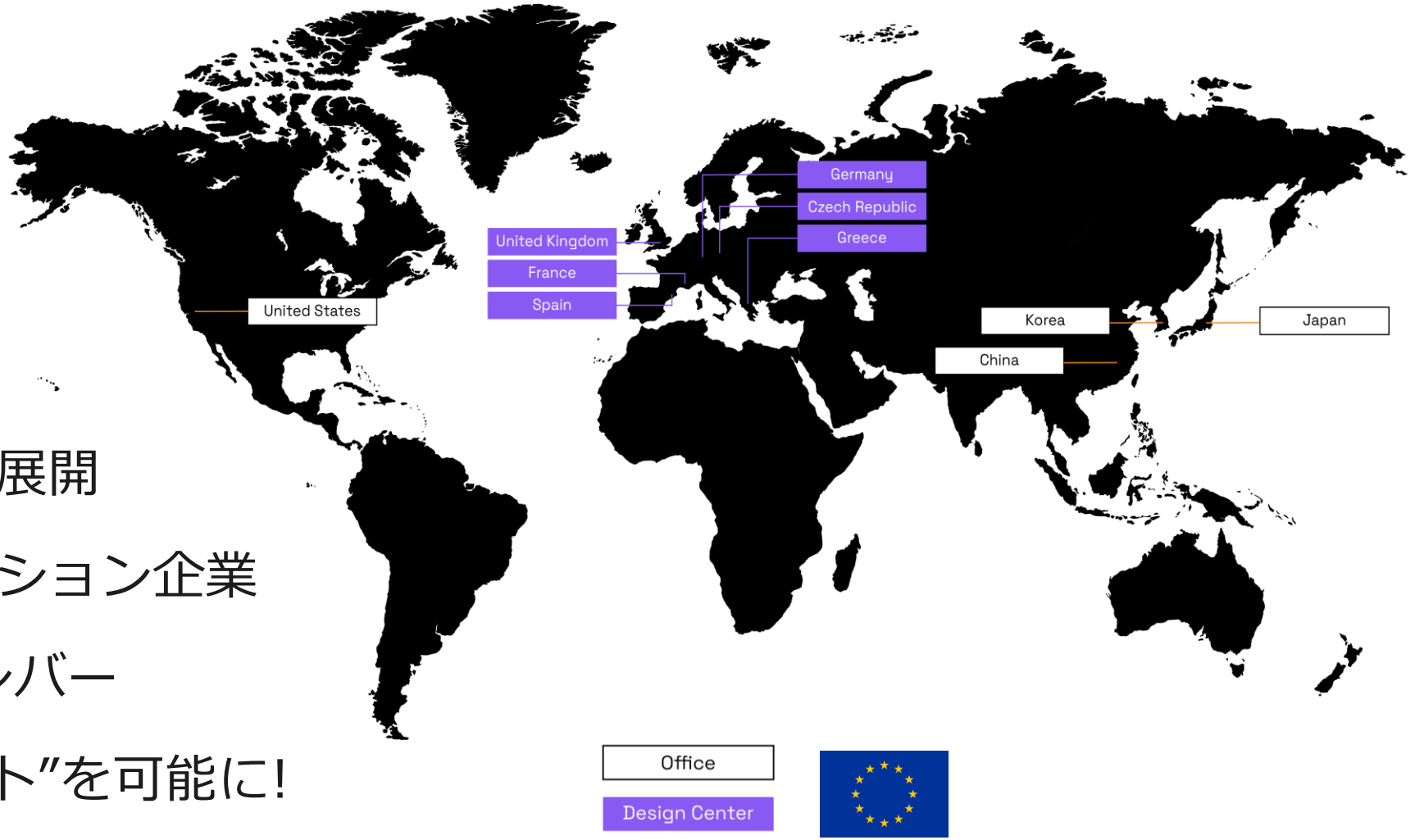
明石 貴昭, Country Manager – Japan, Codasip GmbH

2024年1月16日(火) @東京大学伊藤国際学術研究センター



→ コダシップ概要

- 2014年設立
- 独ミュンヘン本社
- 約250名の社員
- ヨーロッパで開発
- グローバルなオフィス展開
- プロセッサ・ソリューション企業
- RISC-V® 創設メンバー
- “カスタム コンピュート”を可能に!



→ コダシップ製品

Codasip

Codasip Studio

- プロセッサ開発ツール
- 高速なアーキテクチャ探索
  - 命令精度モデリング
  - サイクル精度モデリング
  - シミュレーション
  - プロファイリング
  - コ・シミュレーション
- HDK(RTL他)とSDK(コンパイラ他)を自動生成

Codasip RISC-V Processors

- そのまま使える高品質なRISC-V IP
- プロセッサ専用記述言語で開発
- RTLも有
- ユーザがカスタマイズ可能
- RISC-V標準準拠
- 業界トップレベルの検証品質

Codasip

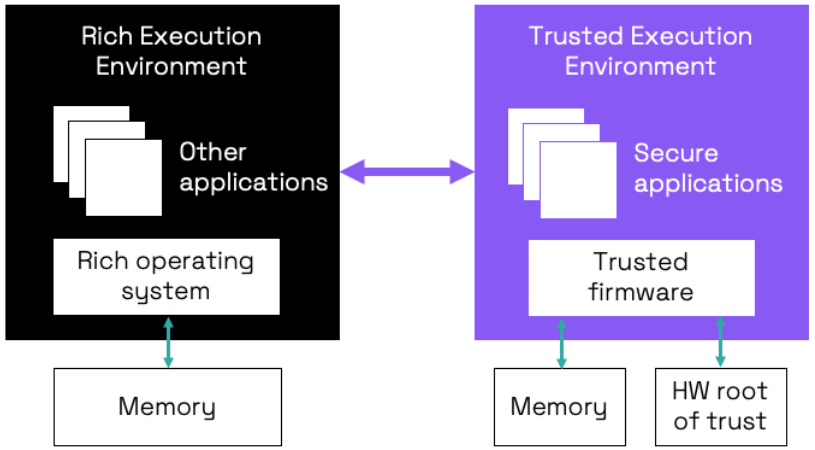
↘

CHERI
(Capability Hardware Enhanced RISC Instructions)
テクノロジ

→ セキュリティ技術...

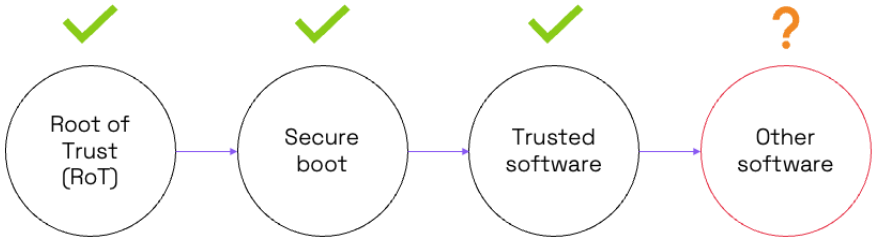
Codasip

隔離実行環境(TEE)



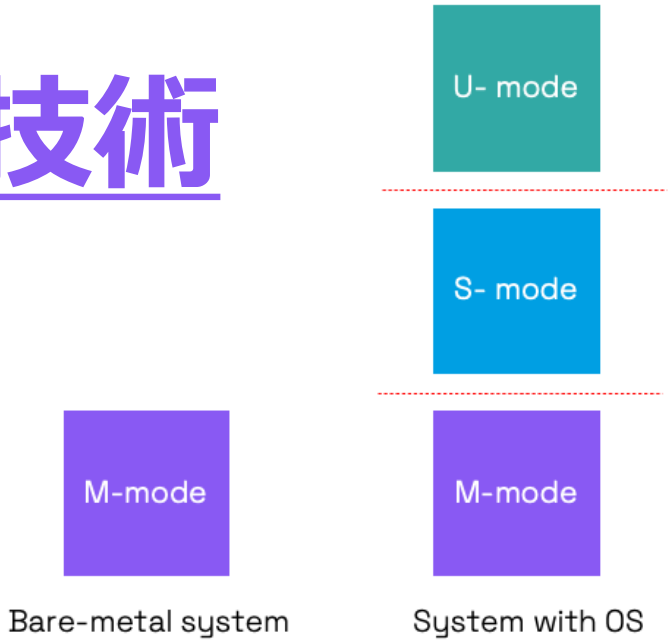
サイドチャネル攻撃対策

メモリ安全技術



信頼の起点(RoT)/チェーン

各種暗号化

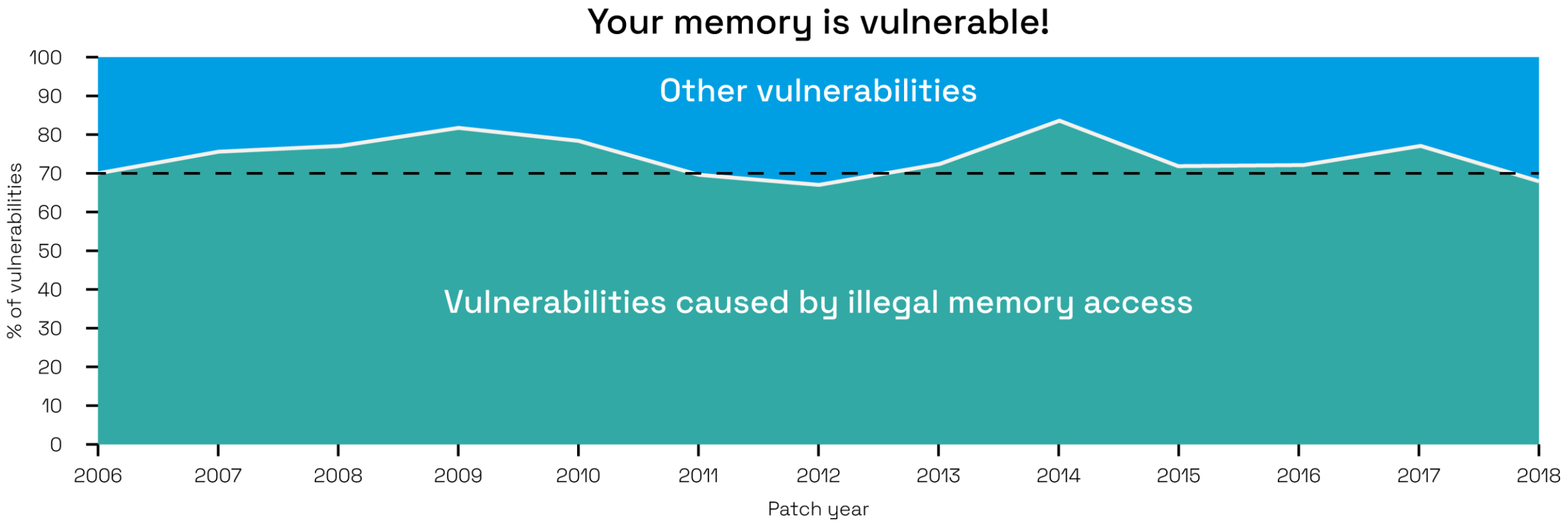


特権モード

RISC-V PMP(Physical Memory Protection)

Codasip

脆弱性の70%はメモリ起因



Source: "Trends, challenges and shifts in vulnerability mitigation", Matt Miller (MSRC), BlueHat IL 2019.

→ 脆弱性 (Vulnerability)

Codasip

- コンピューターシステムやソフトウェアにおいて、悪意を持った攻撃や不正アクセスなどに対して脆弱である状態や欠陥を指します。これは、攻撃者がシステムに侵入したり、不正なアクションを起こすための入り口となる可能性があるものです。脆弱性が存在すると、それを悪用する攻撃が行われる可能性が高まります。
- 脆弱性はさまざまな形で現れ、ソフトウェアの設計、プログラミングミス、セキュリティ対策の不備などが原因となります。セキュリティの専門家は、脆弱性を特定し、修復するための対策を講じ、システムやソフトウェアの安全性を確保することが求められます。

→ CVE (Common Vulnerabilities and Exposures)

Codasip

- セキュリティ脆弱性に対して一意の識別子を提供するための標準的な体系。CVE番号は、脆弱性が特定の製品やソフトウェアに関連している場合でも、異なる組織やベンダが同じ脆弱性に言及する際に利用されます。CVEの目的は、異なる組織や製品間で脆弱性情報を共有し、効果的なセキュリティ対策を促進することです。
  - https://www.cve.org/
  - https://cve.mitre.org/
  - https://www.cvedetails.com/
  - 共通脆弱性識別子CVE概説
    - https://www.ipa.go.jp/security/vuln/scap/cve.html

- NVE (NATIONAL VULNERABILITY DATABASE) by NIST (National Institute of Standards and Technology)
- JVN (Japan Vulnerability Notes) by JPCERT/CC (JPCERT コーディネーションセンター)
  - https://jvn.jp/
  - https://jvndb.jvn.jp/

→ Android 5.0 – 14.0 CVE 累計 2024/01/16 時点

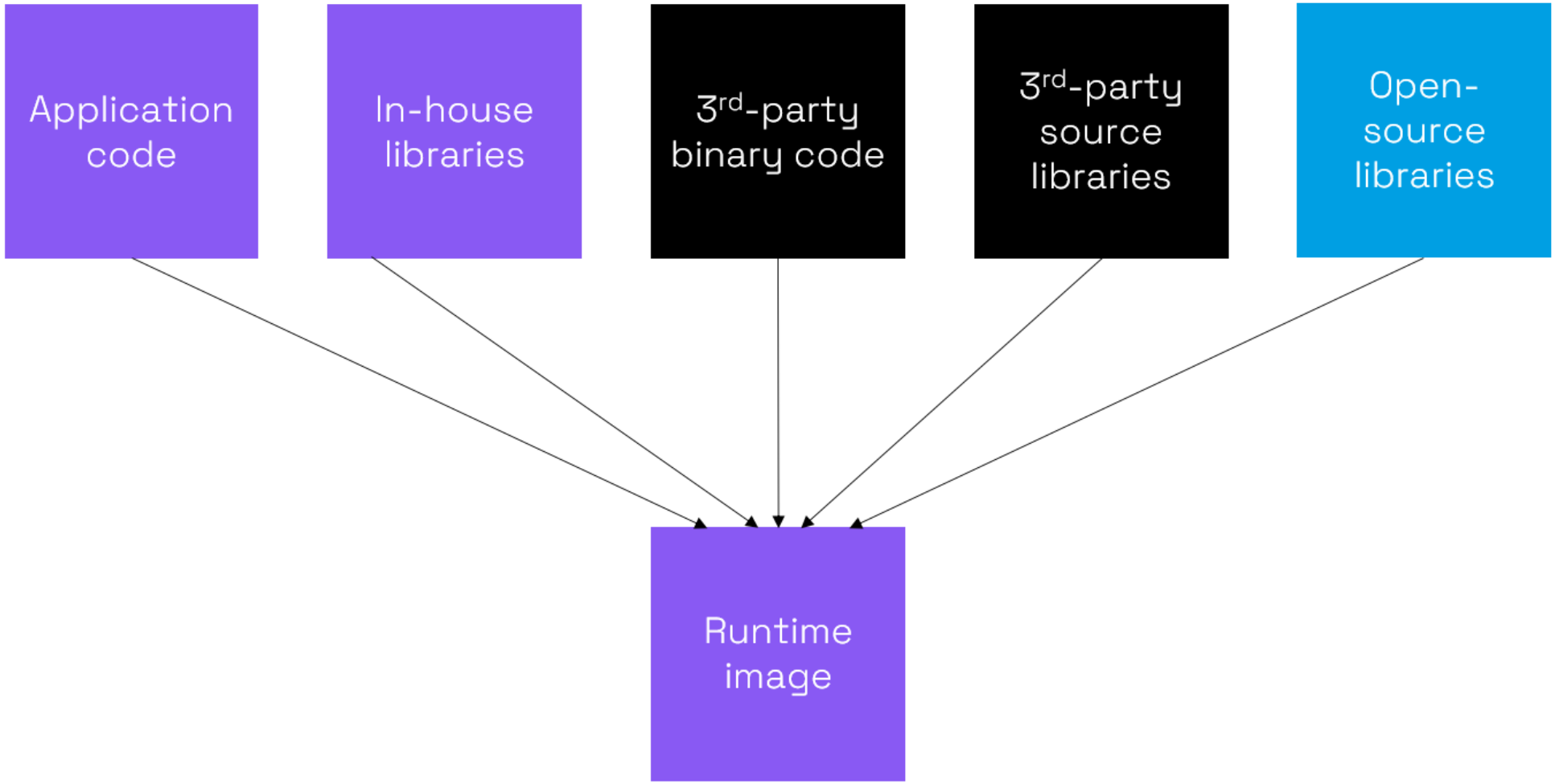
Codasip

Year	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	File Inclusion	CSRF	XXE	SSRF	Open Redirect	Input Validation
2015	48	41	0	0	0	0	0	0	0	0	6
2016	175	84	0	0	0	0	0	0	0	0	96
2017	189	108	0	0	0	0	0	0	0	0	61
2018	41	141	5	0	6	0	0	2	0	0	40
2019	56	257	12	0	1	0	0	0	0	0	41
2020	169	338	18	0	10	0	0	0	0	0	89
2021	147	341	4	0	12	0	0	0	0	0	122
2022	196	560	7	0	23	0	0	0	0	0	180
2023	247	634	6	0	10	0	0	0	0	0	130
2024	6	19	0	0	0	0	0	0	0	0	0
Total	1274	2523	52	0	62	0	0	2	0	0	765
	27%	54%	1%	0%	1%	0%	0%	0%	0%	0%	16%

Source: https://www.cvedetails.com/version-list/1224/19997/1/Google-Android.html を集計

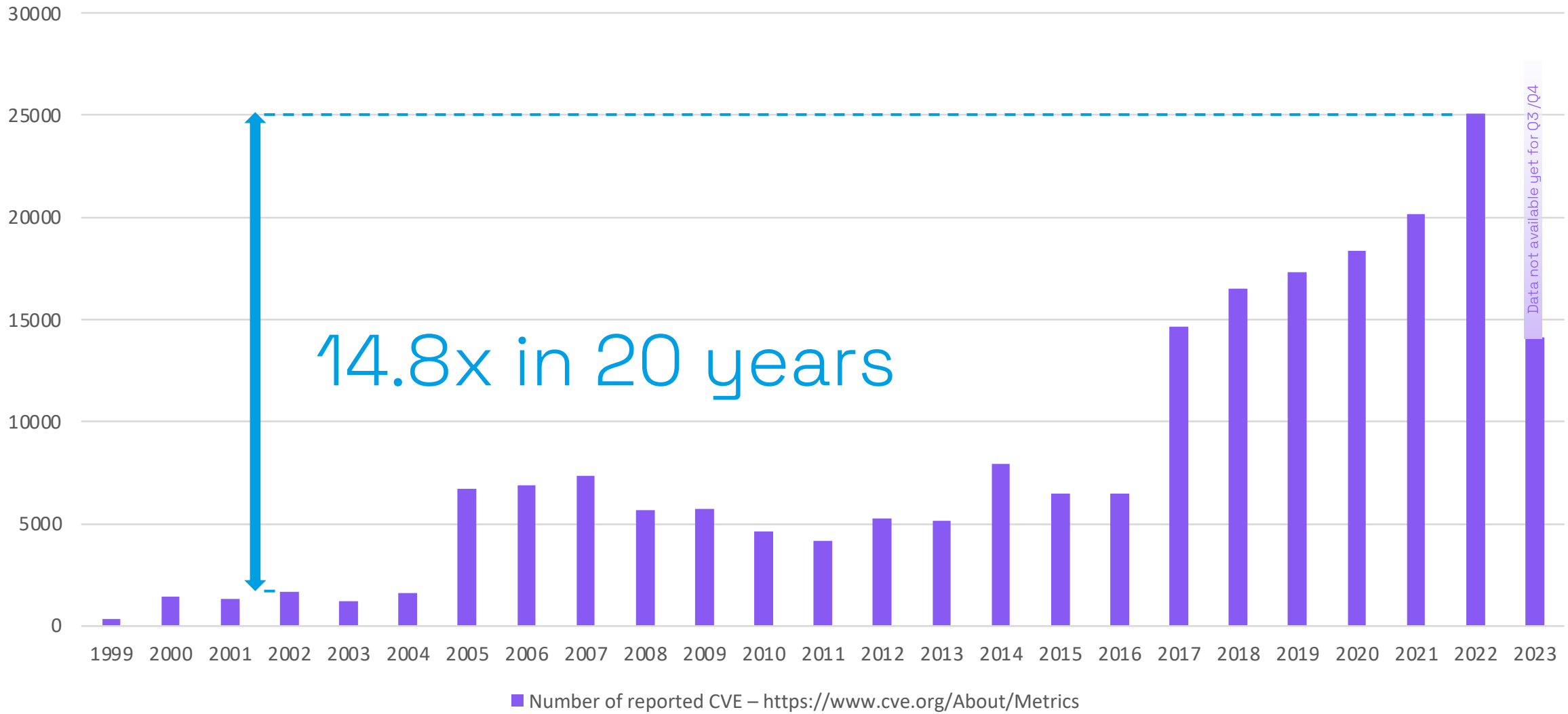
→ ますます複雑化するソフトウェア

Codasip

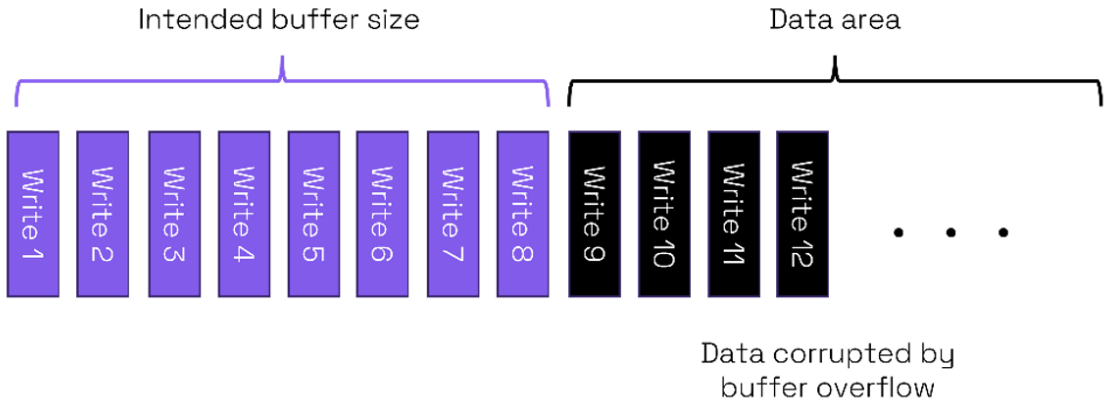


→ 脆弱性は日々増加している

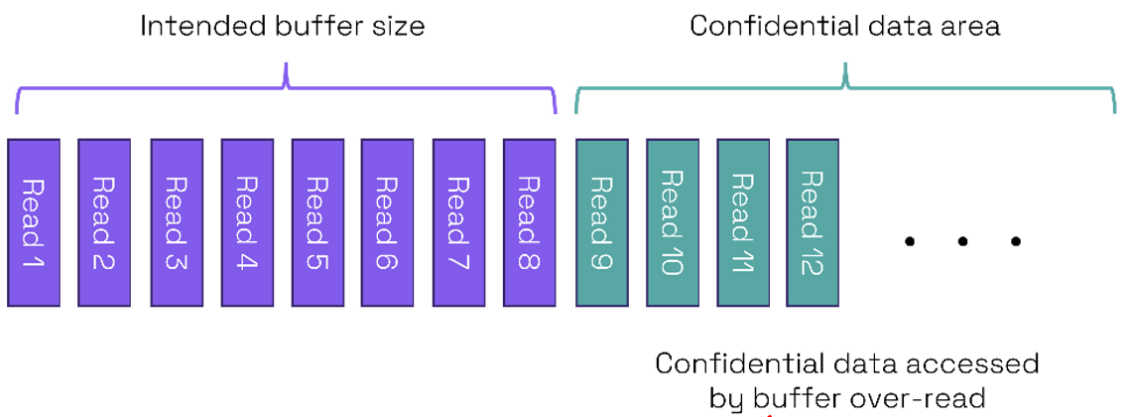
Codasip



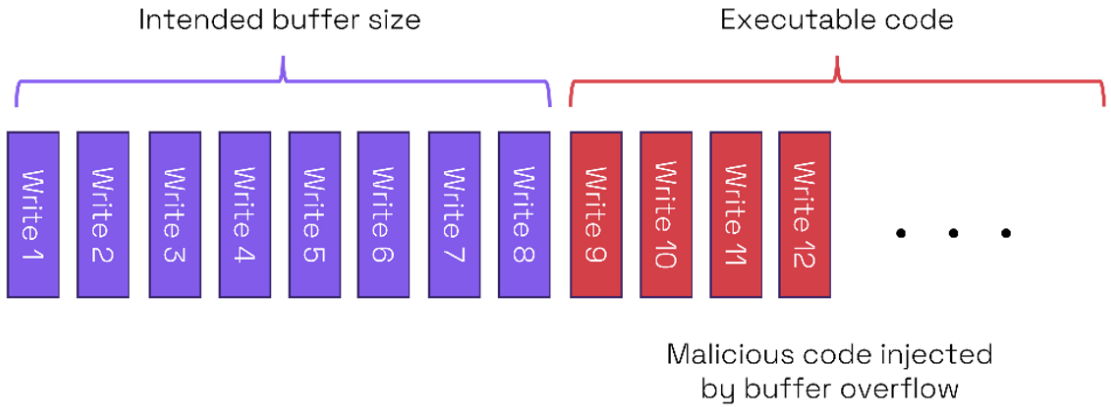
→ バッファ オーバーフロー…



オーバーフロー



オーバーリード



悪意コード実行

過去のサイバー攻撃:

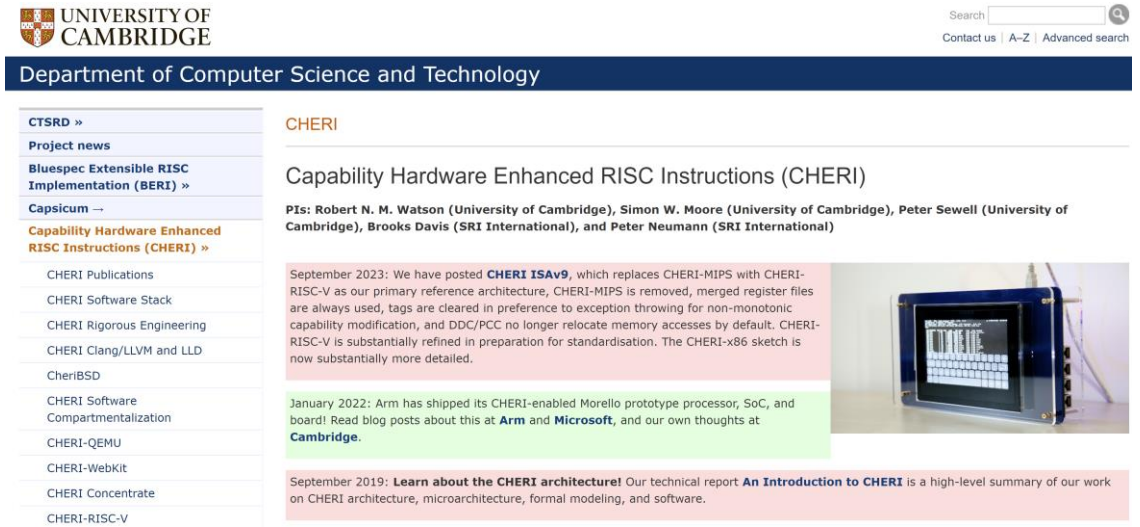
- 1988: Morris Worm – sendmail, rsh/rexec
- 2003: SQL Slammer – MS SQL server
- 2014: Heartbleed - OpenSSL
- 2018: Adobe Flash Player – Win, Mac, Linux and ChromeOS
- 2019: WhatsApp VOIP - smartphones

Source: Comparitech

→ CHERI(Capability Hardware Enhanced RISC Instructions)

Codasip

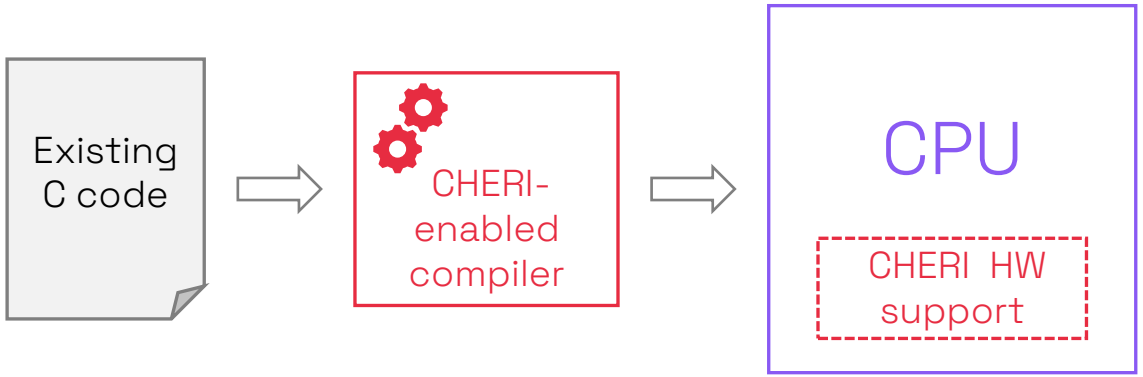
- 英ケンブリッジ大学で開発されたセキュリティ強化型の命令セットアーキテクチャ
  - https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/
  - ハードウェア上で実装された能力(capability)に基づくメモリ保護を提供し、セキュリティの向上を目指す
  - **最も脆弱なC/C++ソースの救済**
  - CHERIプロジェクトは、DARPAが資金提供している研究プロジェクトの一つ



→ CHERIのコンセプト
ハードウェアによる保護

Codasip

- CHERIはCHERI-ISAに適応したプロセッサを必要とする
  - 多くの種類のコアに適用可能
- ハードウェアによるセキュリティ
  - ソフトウェアによるバイパス不可能
  - 実証済み
- **既存コードの再利用**
  - **必要なコード変更はゼロか最小限**
  - **どの部分を保護するかを選択する**

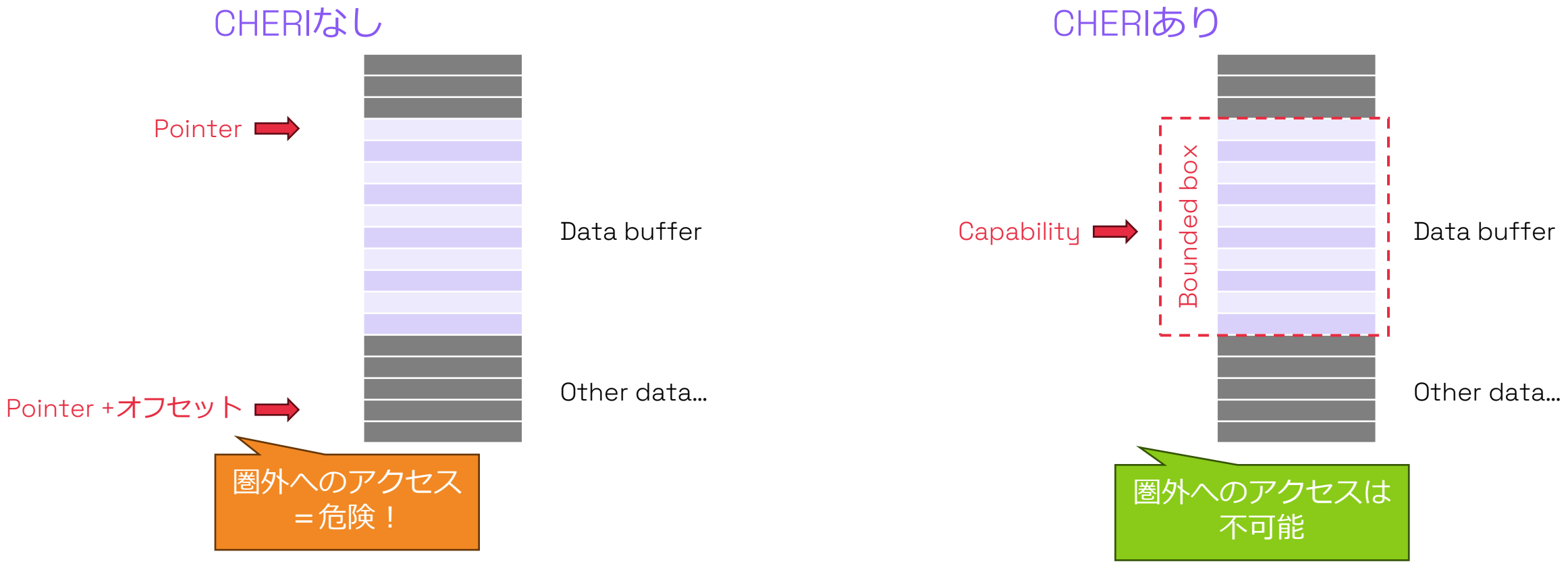


- CHERIのメリット
  - 危険なコードは拒否
  - クリティカルなコード用にCHERIコンパートメントを作成する
  - Arm、~~MIPS~~、x86など、さまざまなアーキテクチャに対応

→ CHERIのコンセプト
メモリ空間の安全性

Codasip

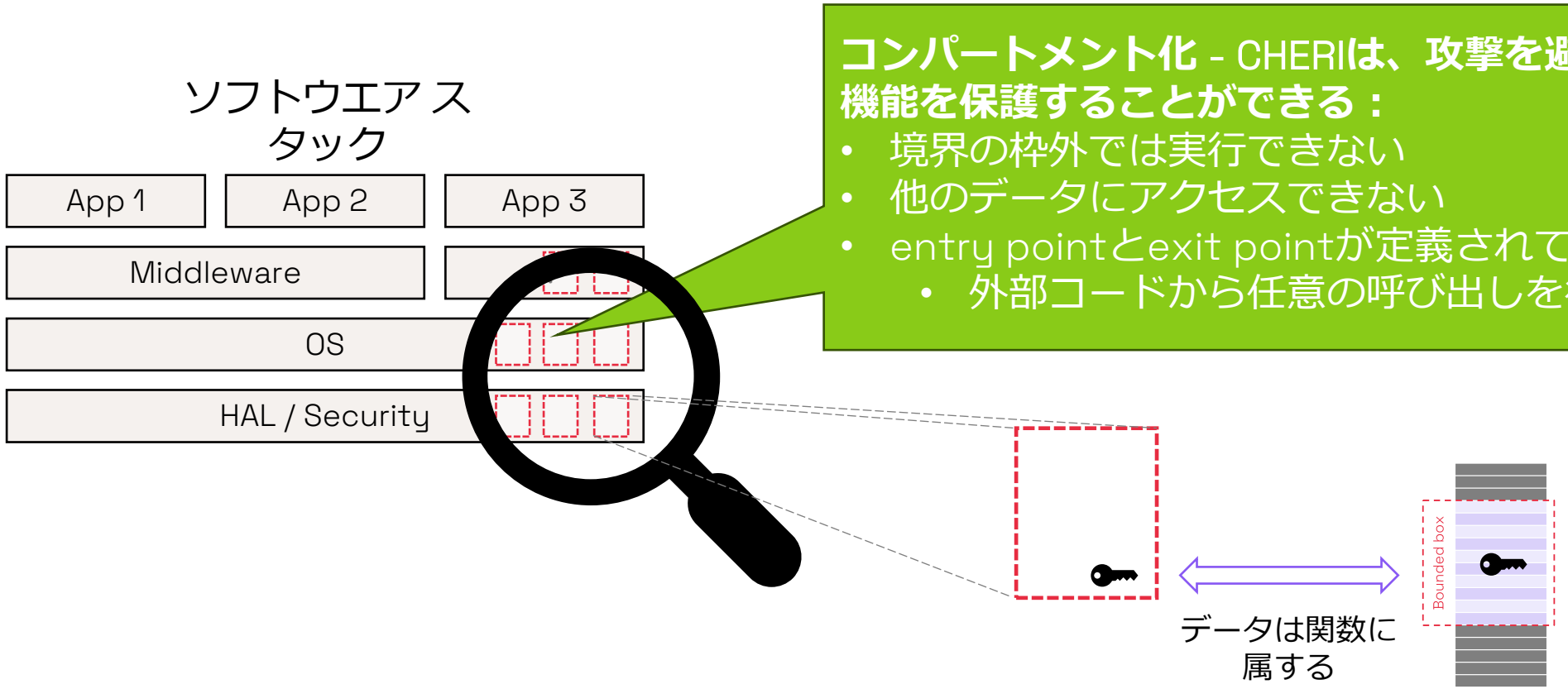
- PointerをCapabilityに置き換える - ハードウェア制御によって



→ CHERIのコンセプト
コンパートメント化

Codasip

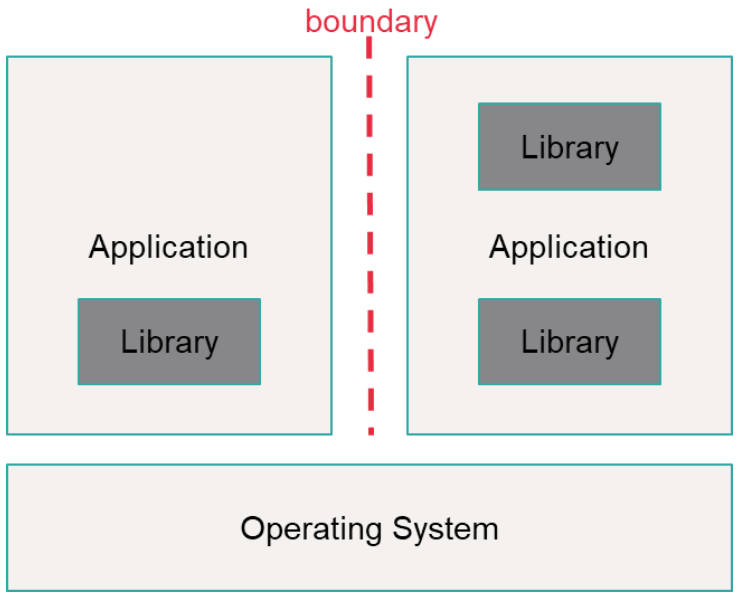
- Capabilityは、特定された機能／実行コンテキストに属する



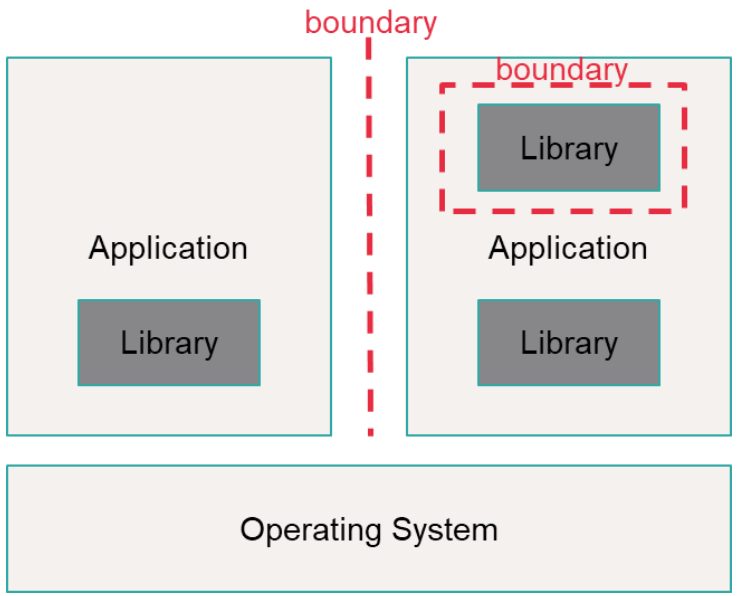
→ コンパートメント化

Codasip

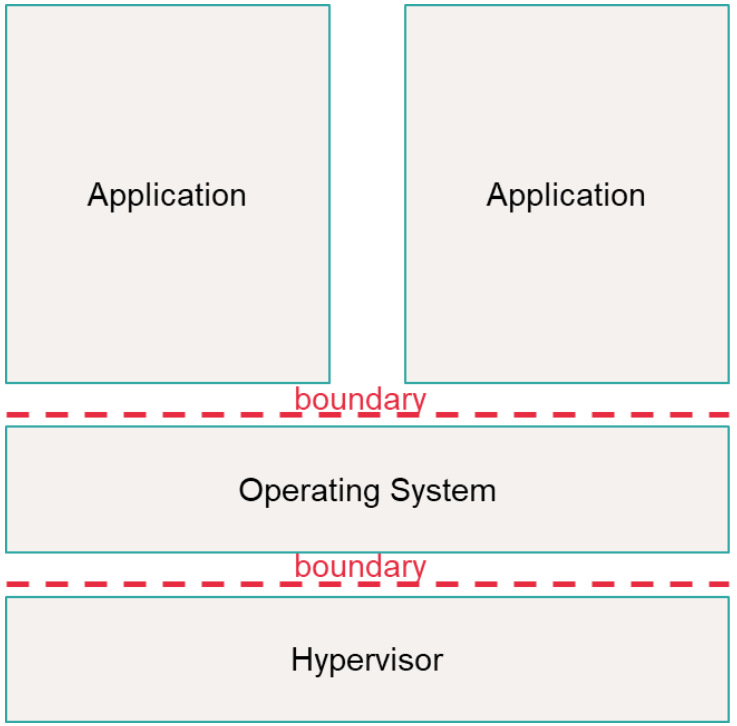
垂直



垂直 + 特定部分



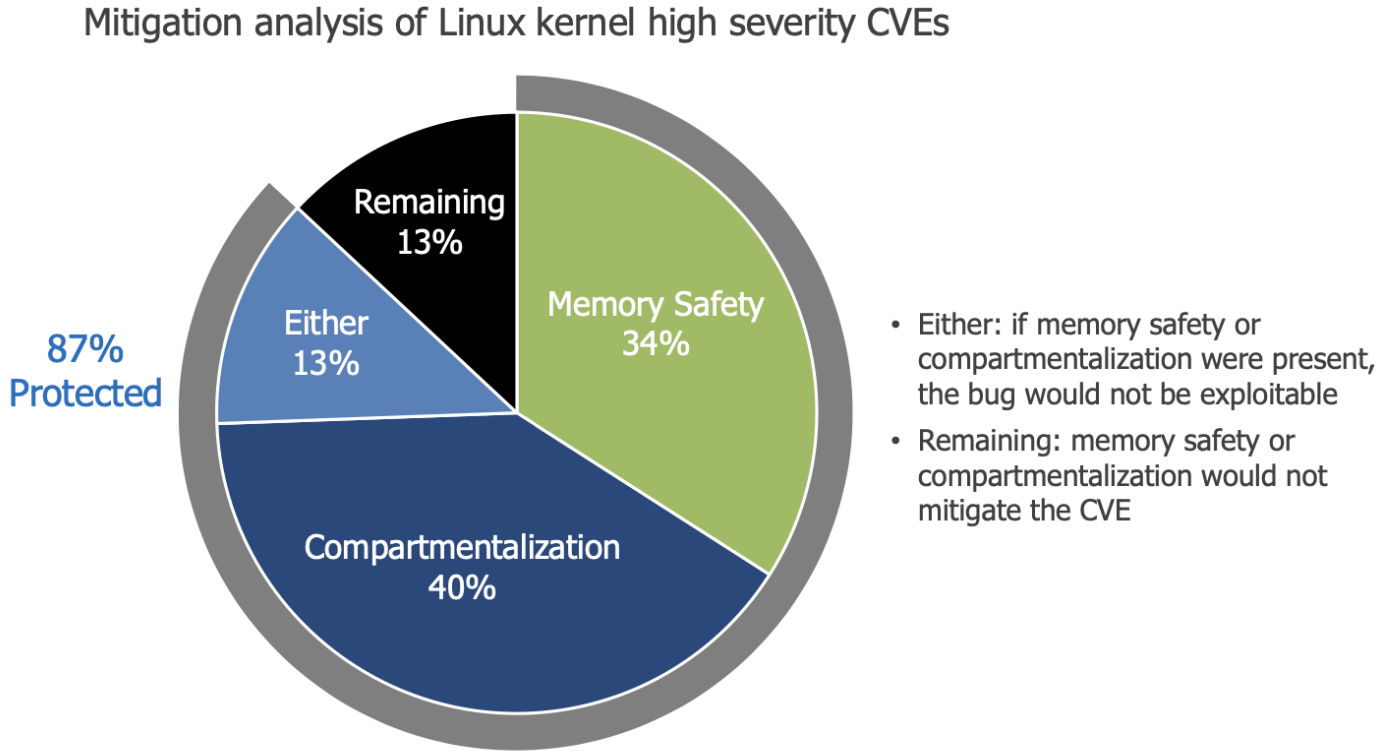
水平



→ Linuxカーネルの深刻なCVEの87%に有効

Codasip

Memory safety and compartmentalization would stop most campaigns



CVE: Common Vulnerabilities and Exposures

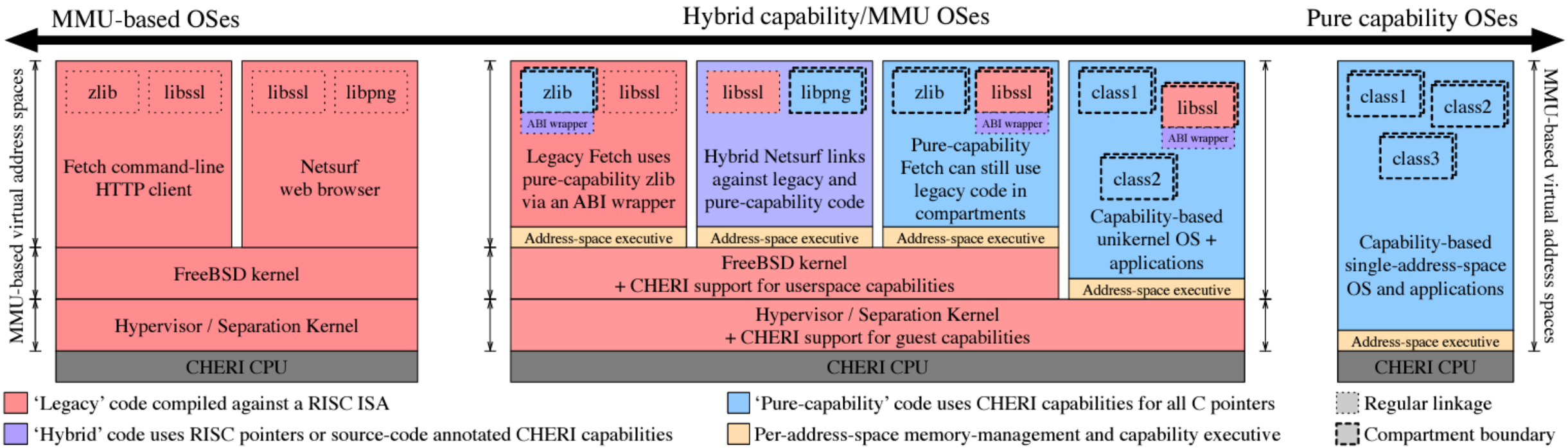
McKee, Derrick, et al. "Preventing kernel hacks with HAKC." Proceedings 2022 Network and Distributed System Security Symposium. NDSS. Vol. 22. 2022.

Distribution Statement A: Approved for public release. Distribution is unlimited.

5

→ ケンブリッジ大学CHERIペーパーより コンパートメント化

- CHERIは、ハイブリッドを含む様々なアーキテクチャをサポート



Source: UoCambridge

→ CHERI-RISC-V ISA サンプル

Codasip

B.3. ASSEMBLY PROGRAMMING 425

Integer Instruction	Capability Instruction
`l{b|h|w|d}[u] rd, offset(rs1)`	`cl{b|h|w|d}[u] rd, offset(cs1)`
`lc cd, offset(rs1)`	`clc cd, offset(cs1)`
`s{b|h|w|d} rs2, offset(rs1)`	`cs{b|h|w|d} rs2, offset(cs1)`
`sc rs2, offset(rs1)`	`csc cs2, offset(cs1)`
`fl{h|w|d|q} fd, offset(rs1)`	`cfl{h|w|d|q} fd, offset(cs1)`
`fs{h|w|d|q} fs2, offset(rs1)`	`cfs{h|w|d|q} fs2, offset(cs1)`
`lr.{b|h|w|d} rd, (rs1)`	`clr.{b|h|w|d} rd, (cs1)`
`lr.c cd, (rs1)`	`clr.c cd, (cs1)`
`sc.{b|h|w|d} rd, rs2, (rs1)`	`csc.{b|h|w|d} rd, rs2, (cs1)`
`sc.c cd, cs2, (rs1)`	`csc.c cd, cs2, (cs1)`
`amo<op>.{w|d}[.order] rd, rs2, (rs1)`	`camo<op>.{w|d}[.order] rd, rs2, (cs1)`
`amo<op>.c[.order] cd, cs2, (rs1)`	`camo<op>.c[.order] cd, cs2, (cs1)`
`auipc rd, offset`	`auipcc cd, offset`

Table B.2: Uncompressed Instructions Dependent on Encoding Mode

Source: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-987.pdf

→ メモリ安全性 技術特徴

Codasip

方法	HW | SW	統計的 | 確定的	粒度	境界保護の有効性
MMU	HW	確定的	粗い	非常に限定的
MPU	HW	確定的	粗い	非常に限定的
スタック カナリア	SW	統計的	粗い	限定的
Memoryタグ/MTE	HW	統計的	細かい	適度
Firebloom	SW	確定的	細かい	網羅的
CHERI	**HW**	**確定的**	**細かい**	**網羅的**

Codasip

↘

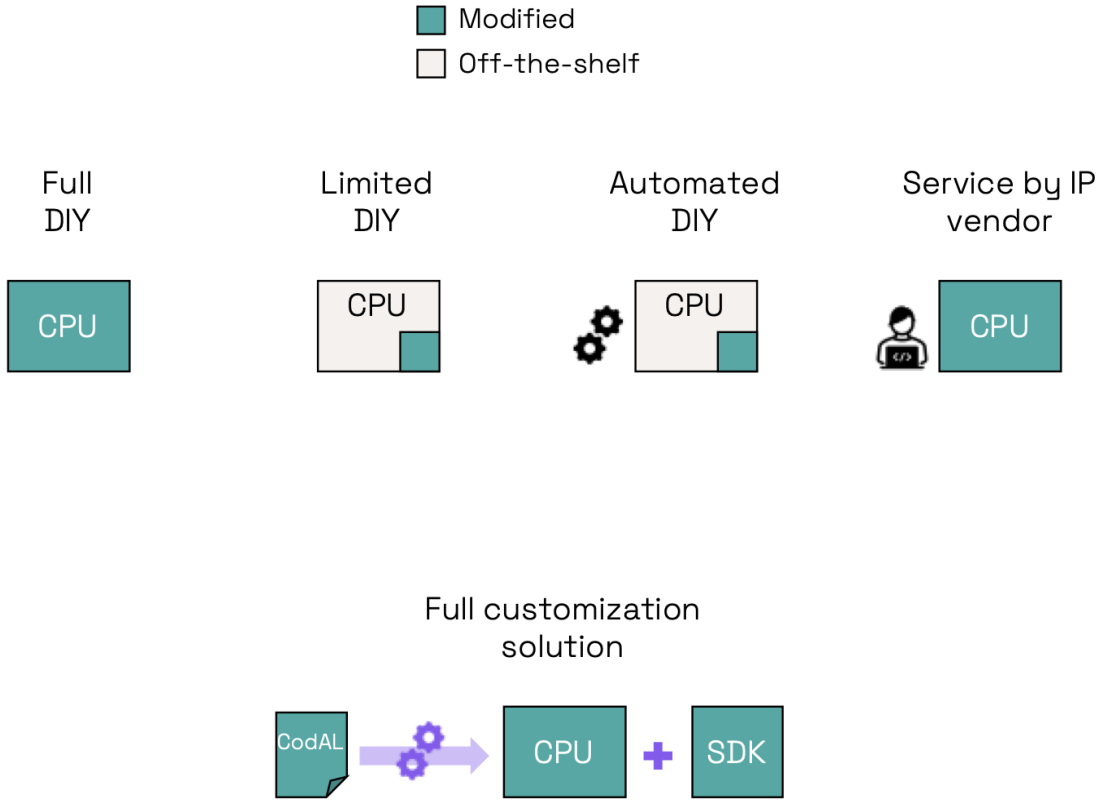
カスタム・コンピュートの薦め

Codasip

RISC-Vの時代が訪れました

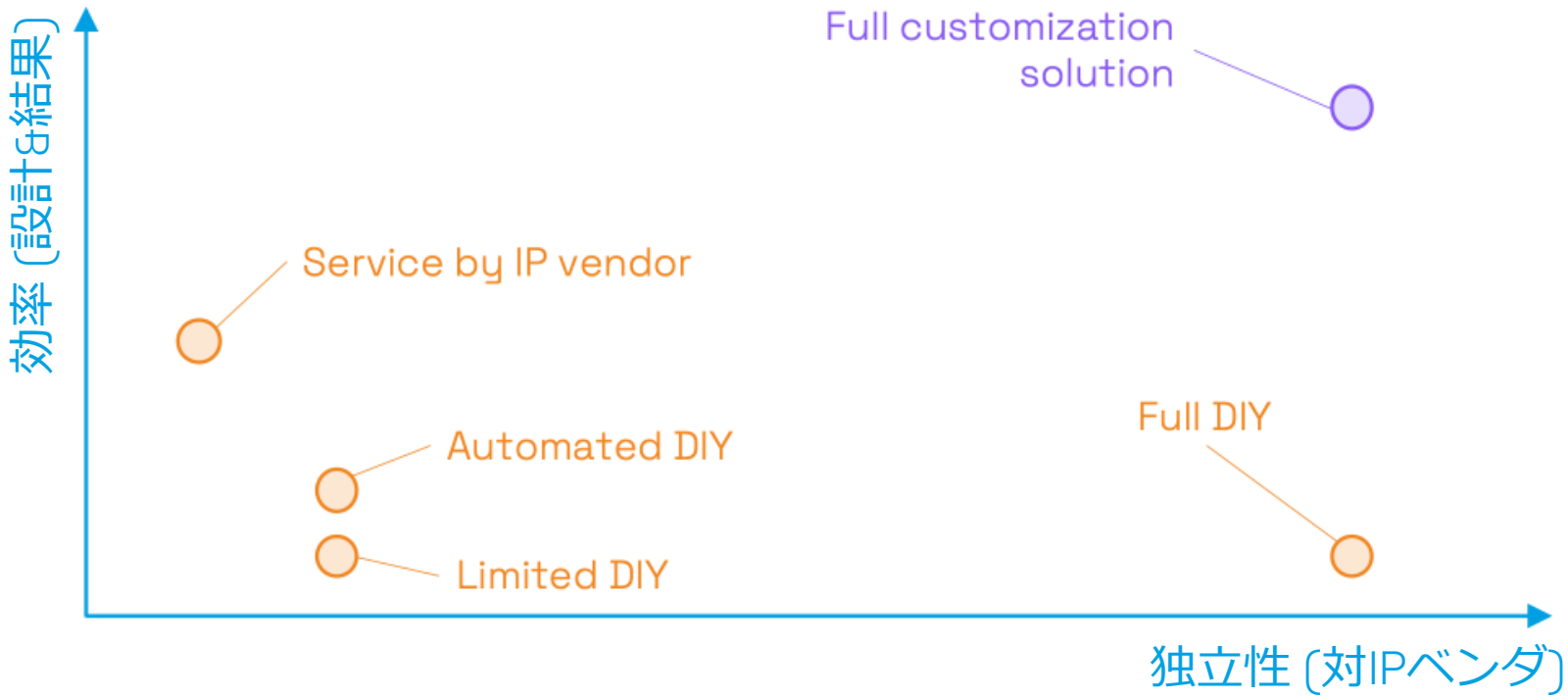
貴方はプロセッサIPを購入しますか？
DIYしますか？
　　ZEROから？
　　差別化ポイントだけ？

→ DIY or Buy??



- **完全な**DIY – ゼロからRTLコーディング
- **部分的**DIY – 固定CPUベンダ固有のインターフェイスに手書きの RTLを接続
- **自動化**DIY – いくつかの自動化ツールを活用
- IP**ベンダによるサービス** – IP ベンダが顧客用にIPをカスタマイズ
- **完全なカスタマイズ ソリューション** – カスタム コンピュートを自動化できる統合されたツールとベースIP

→ それぞれの効率と独立性 (対ベンダ)

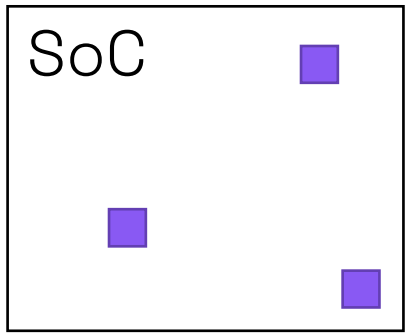


- **完全な**DIY – ベンダから完全独立、作業効率は高くない. 特に検証が課題
- **部分的**DIY – ベンダの制約に依存
- **自動化**DIY – 複数ツールによる無駄、不整合
- IP**ベンダによるサービス** – 大胆な情報開示が必要
- **完全なカスタマイズ ソリューション** – 自らツールを用いて最速でDIY可能

→ カスタム・コンピュート コアはどこにもフィット

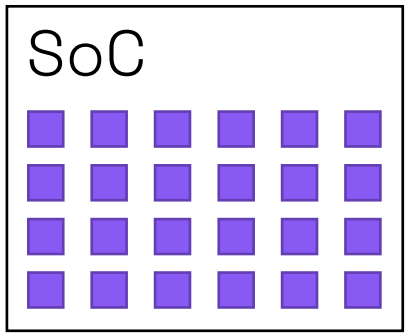
Codasip

組み込みコア最適化



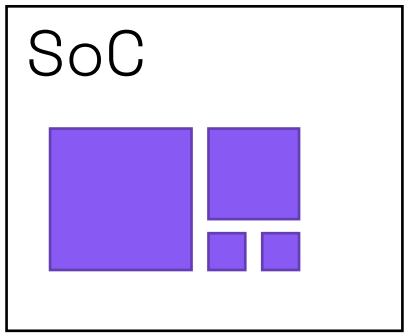
- システム管理
- 電源管理
- セキュリティ
- …

スペシャル コア



- AI
- Systolicエンジン
- コンピュート イン メモリ
- …

ヘテロジニアス・コア



- メインCPU最適化
- Vectorエンジン
- DSP
- 非対称コア

→ カスタム・コンピュートのメリット

Codasip

パフォーマンス

- 性能を押し上げる
- 古い製造プロセスでも高速化

効率化

- 低消費電力化
- エリア削減

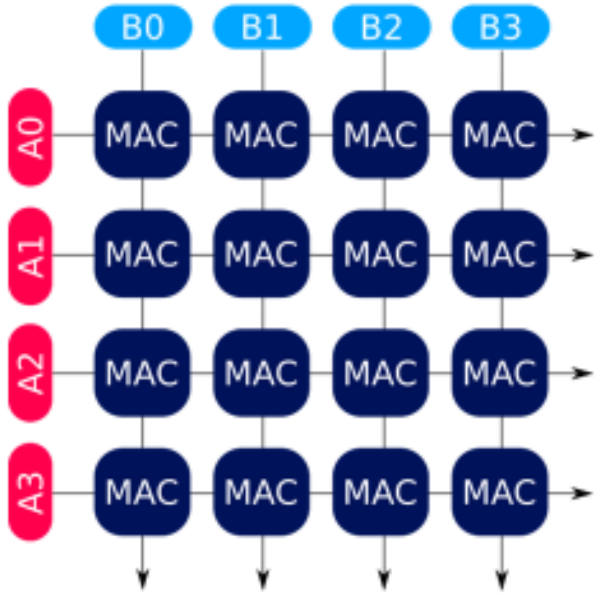
差別化

- 競合に差をつける
- コピー製品対策

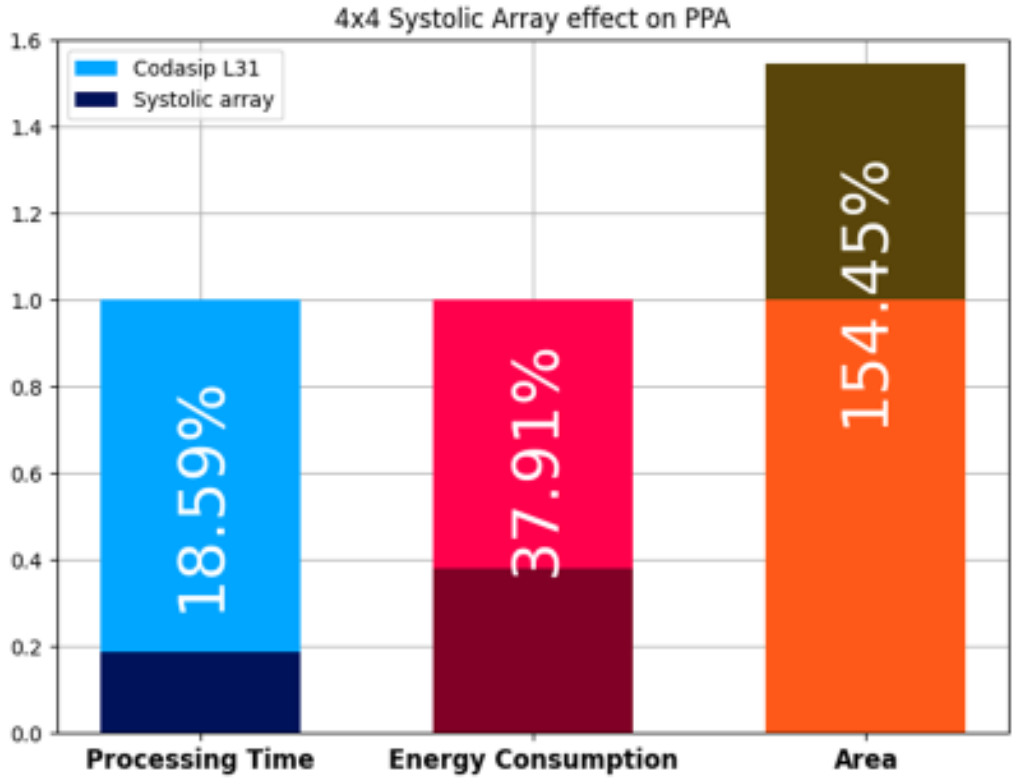
→ ケーススタディ: 4x4 Systolic array

Codasip

4x4 Systolic array of MACs

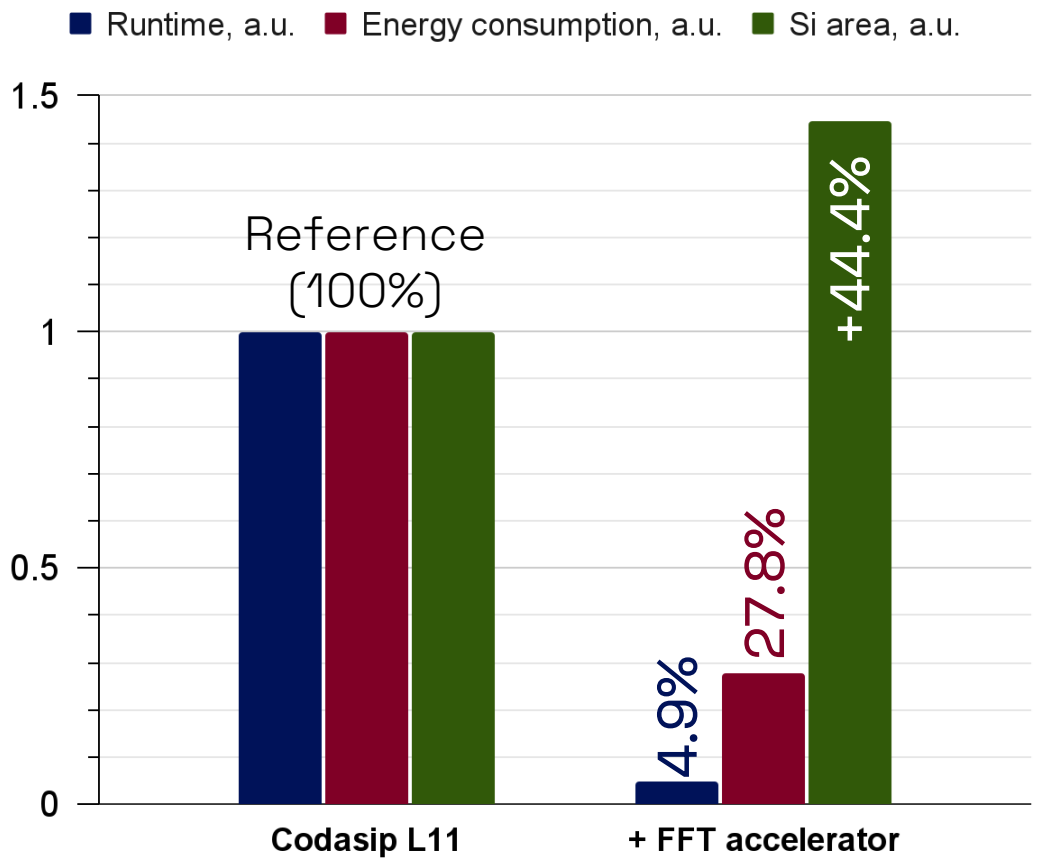


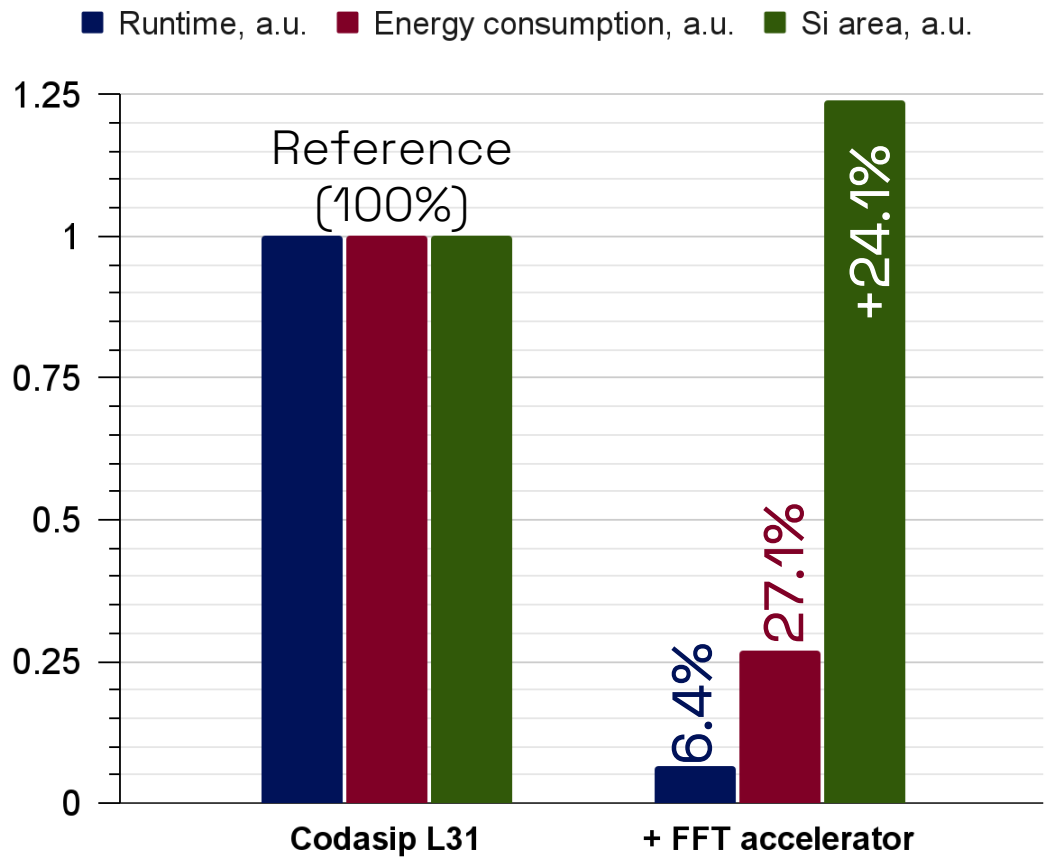

```
case OPC_SYS_ARR_CALC:
   for (i=0; i<SYS_LENGTH; i++) {
      S0[i] = (int32)(S0[i]) + (int32)(A[i]) * (int32)(B[0]);
      S1[i] = (int32)(S1[i]) + (int32)(A[i]) * (int32)(B[1]);
      S2[i] = (int32)(S2[i]) + (int32)(A[i]) * (int32)(B[2]);
      S3[i] = (int32)(S3[i]) + (int32)(A[i]) * (int32)(B[3]);
   } break;
```



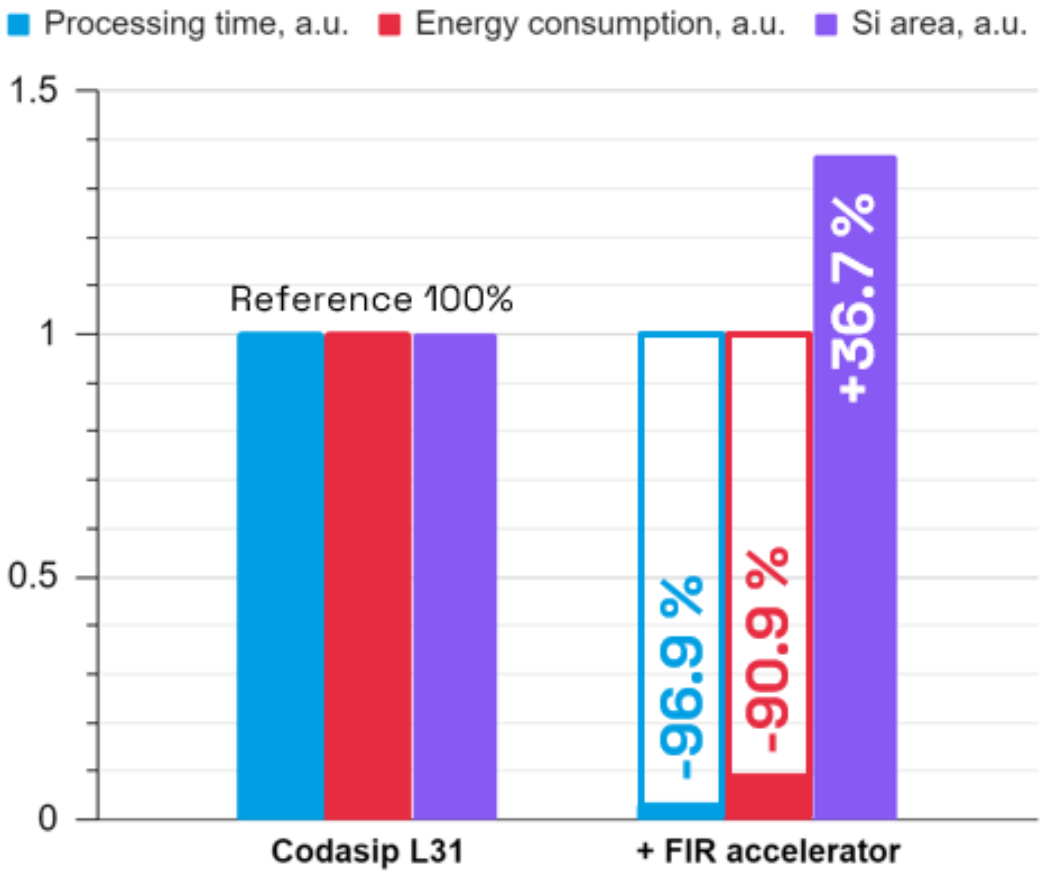
accellera SYSTEMS INITIATIVE 2023/06/22 2023 DVCON JAPAN

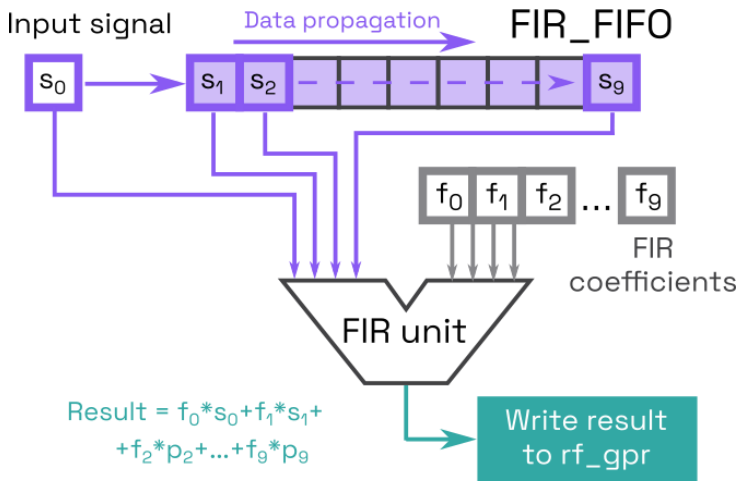
→ ケーススタディ: FFTアクセラレータ





→ FIRフィルタ アクセラレータ





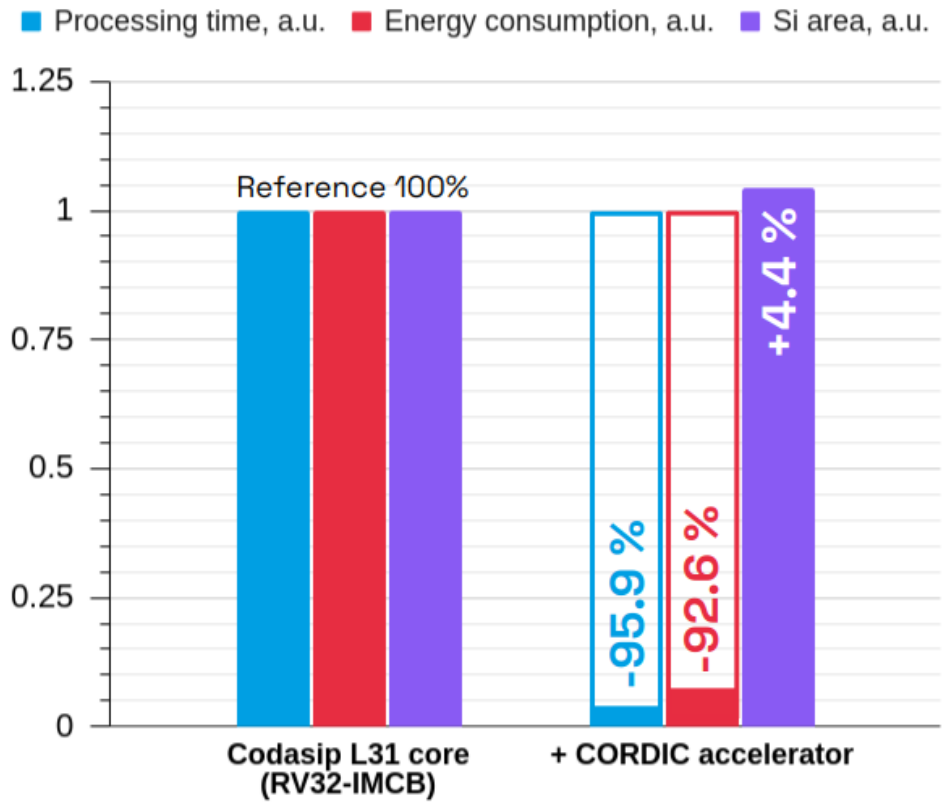
- 実行時間 96% 削減
- 消費電力 90% 削減
- 面積 +36% 増加
- ~~最~~高周波数に影響無

→ ケーススタディ:CORDICアクセラレータ

Codasip

- 既製L31にCORDICアクセラレータを追加カスタマイズした場合のPPA結果

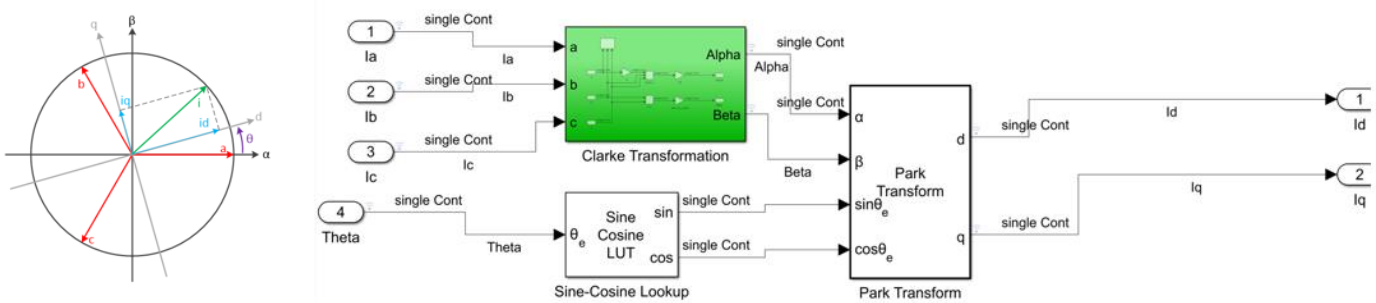
- パフォーマンス 24倍+ 向上
- 消費電力 1/13.5に削減
- 面積 4.4% 増加



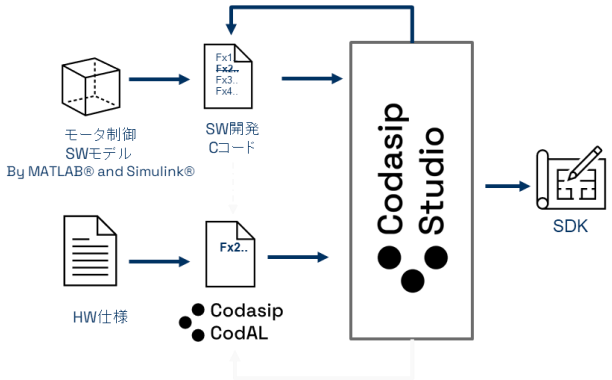
→ カスタム・コンピュート: 自動車向けユースケース

- **目的: DQ0変換のアプリケーション(Cコード)実行をカスタム命令を用いて最適化する方法を示し、標準RISC-Vコアの効率と性能を向上させる**

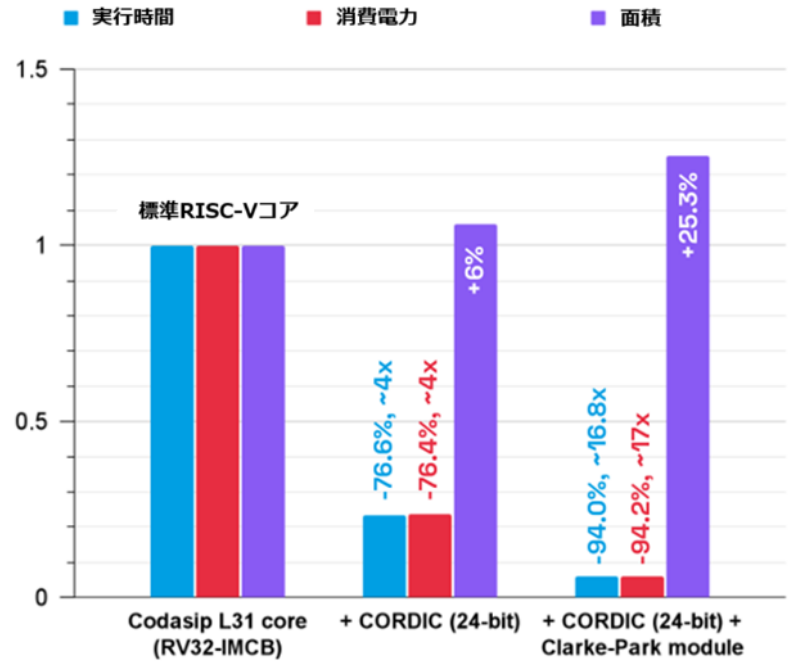
直接直交ゼロ変換



カスタマイズの手順



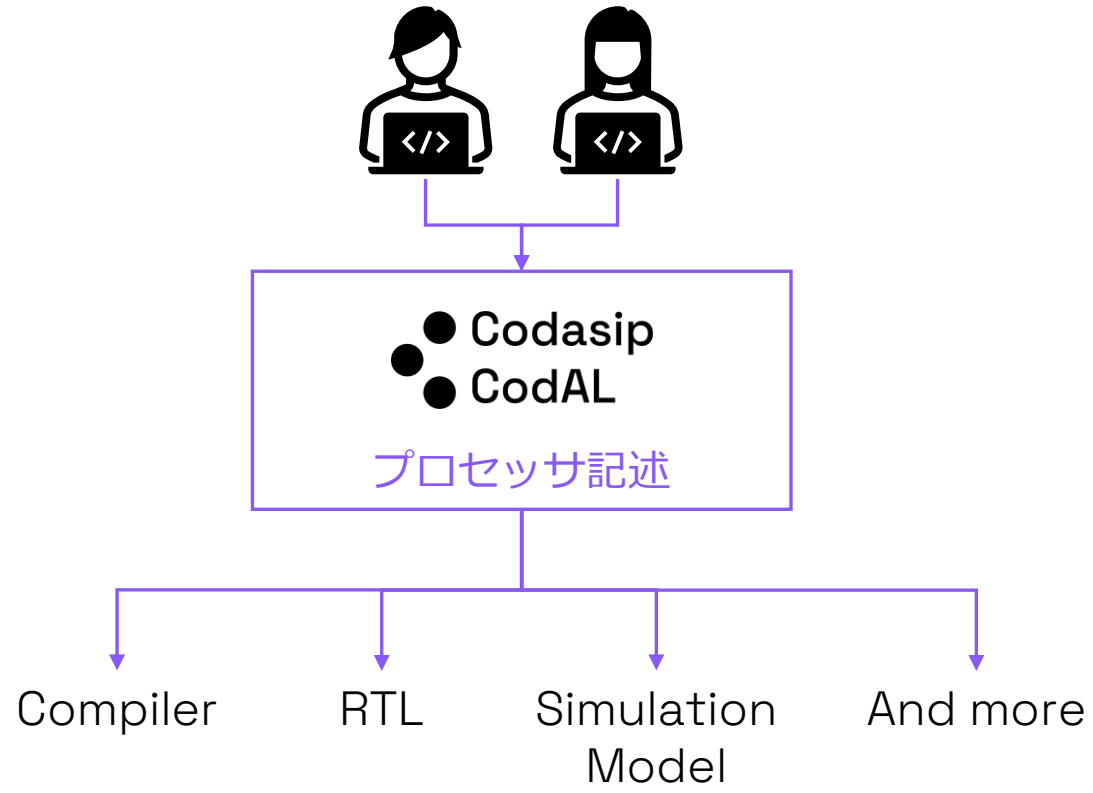
最適化の結果



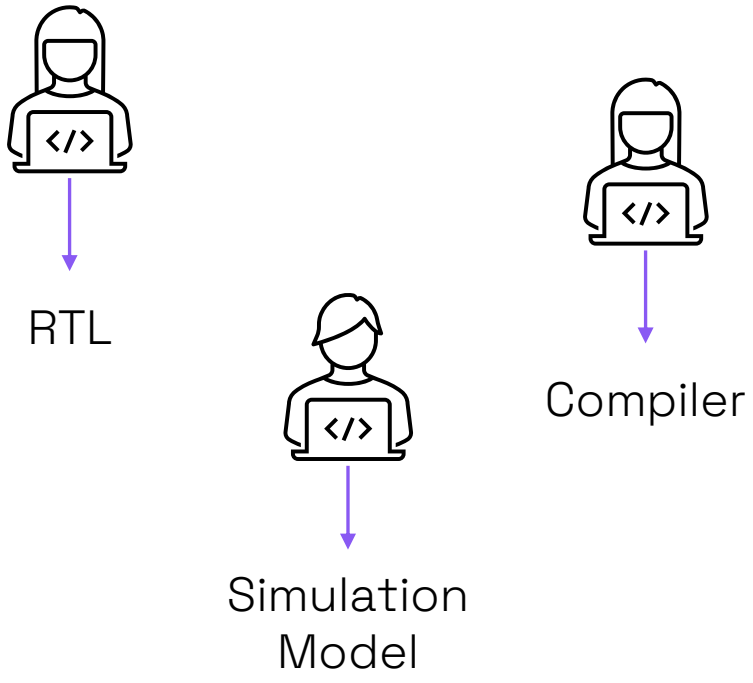
→ 真の単一ソースから様々なアウトプット

Codasip

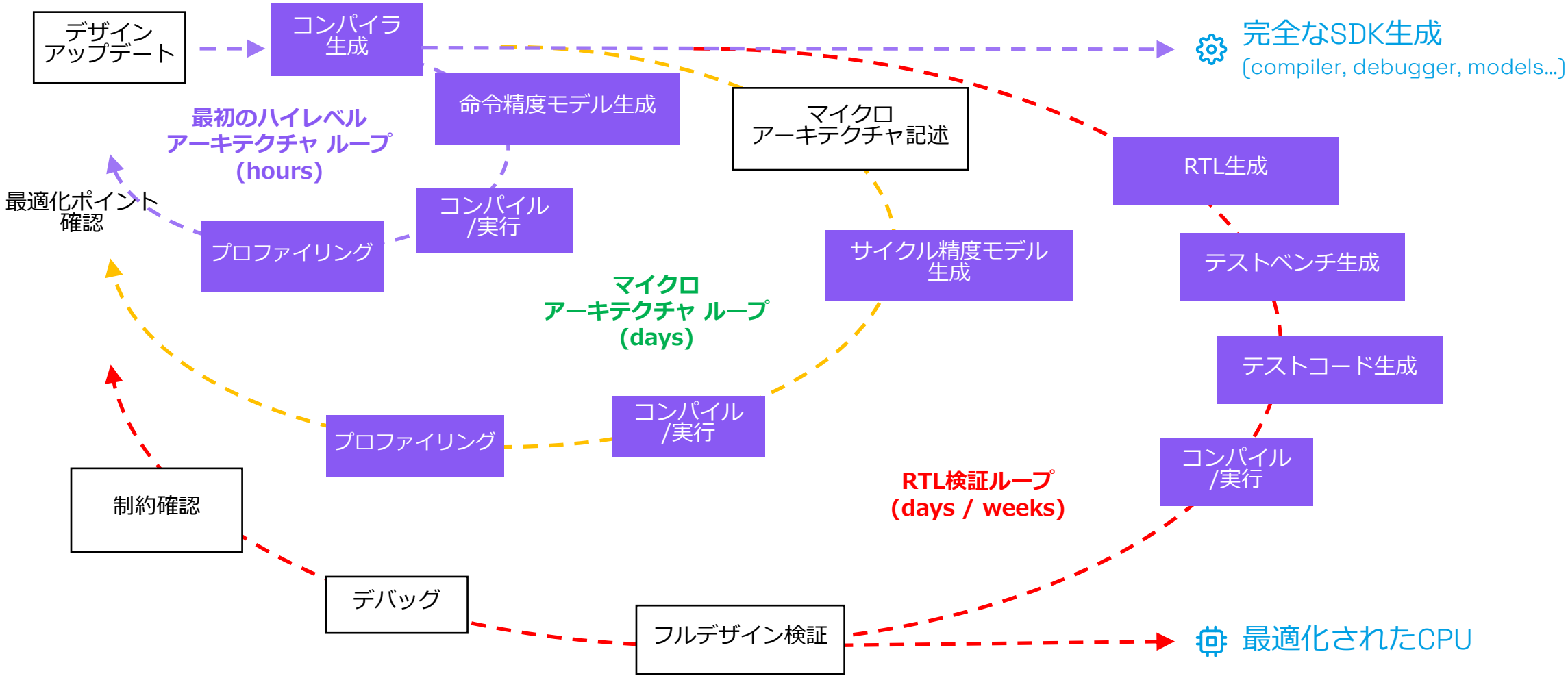
With Codasip Studio



Without Codasip Studio



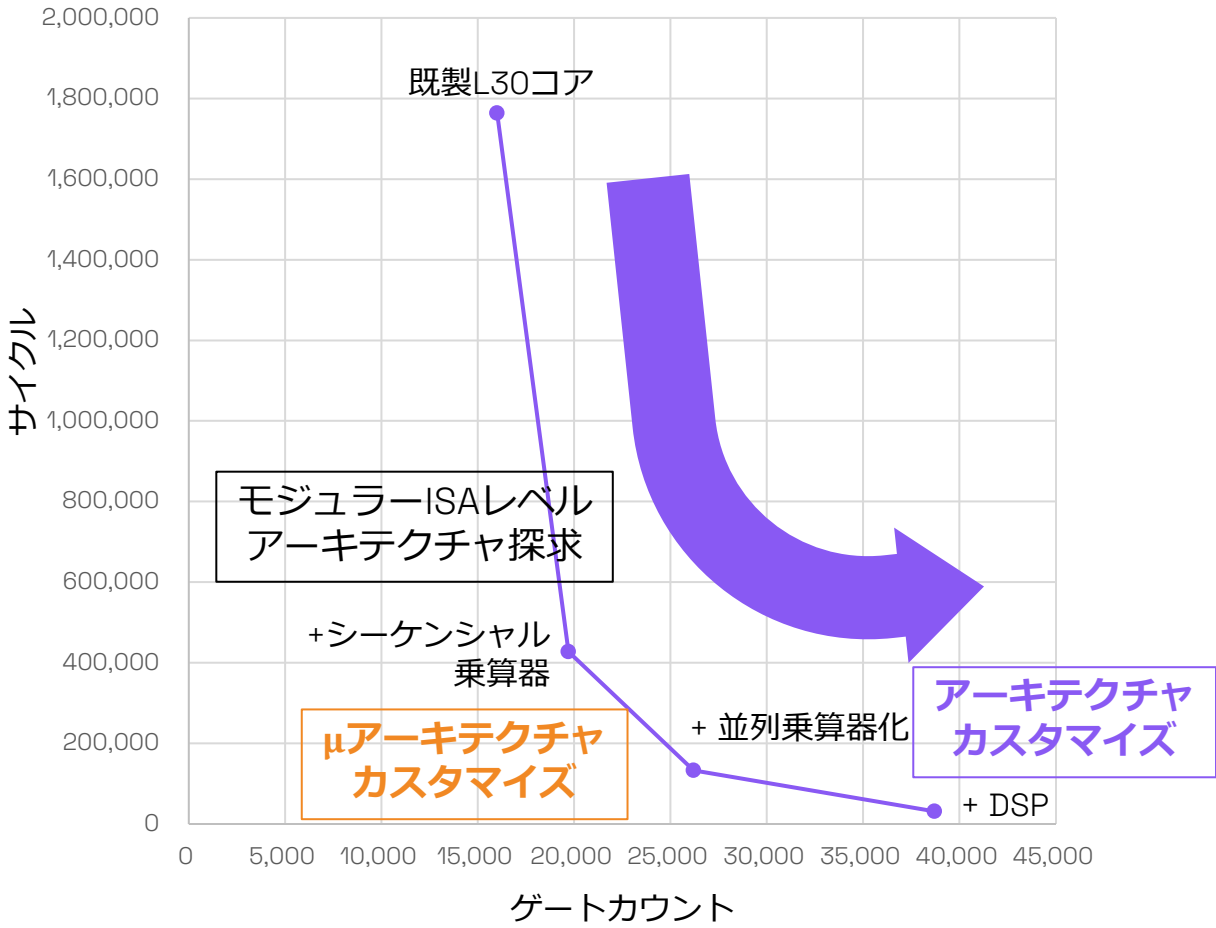
→ 最短で最適な結果を得る開発フロー
自動化された反復的デザインプロセス



→ カスタム命令を追加するメリット

Codasip

- スループット 56.24倍 向上
- コードサイズ 3.62倍 圧縮
- ゲートカウント 2.43倍 増加
  - プロセッサコアではなく、命令メモリが領域で支配的であることに注意
  - いずれもベースのL30コアとの比較
- 古い製造プロセスノードでシリコン化することで大幅なコスト削減を実現



シンプルに、速く、正確に、デザインスペースの探求

... 数日で完了

→ ケーススタディ：SecureRF社（現Veridify社）

	HWアクセラレータ無	HWアクセラレータ有	改善結果
検証時間	10.7 ms	3.8 ms	2.8x 高速化
署名の検証時間/秒（50MHz）	93	263	
コードサイズ	4,504 bytes	3,052 bytes	32% 小型化

命題は、Codasip RISC-Vプロセッサで実行されている
WalnutDSA（デジタル署名アルゴリズム）の高速化

結果は、わずか2%の追加HWリソースで、実行時間は3倍改善

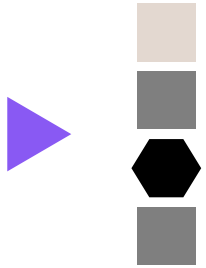
わずかな努力で、大きな改善が可能

→ Codasip StudioはVILWもスーパースカラもサポート

Codasip

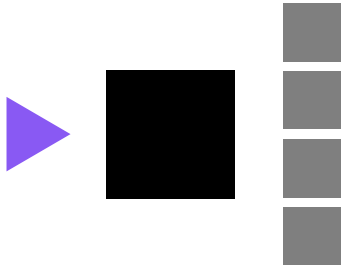
VLIWアーキテクチャ

- コンパイラ**が**コアの命令をスケジューリング
  - 処理要素は異なることがある
- VLIWに特化したコンパイラ
  - すべてのアプリケーションを再コンパイルする必要有り



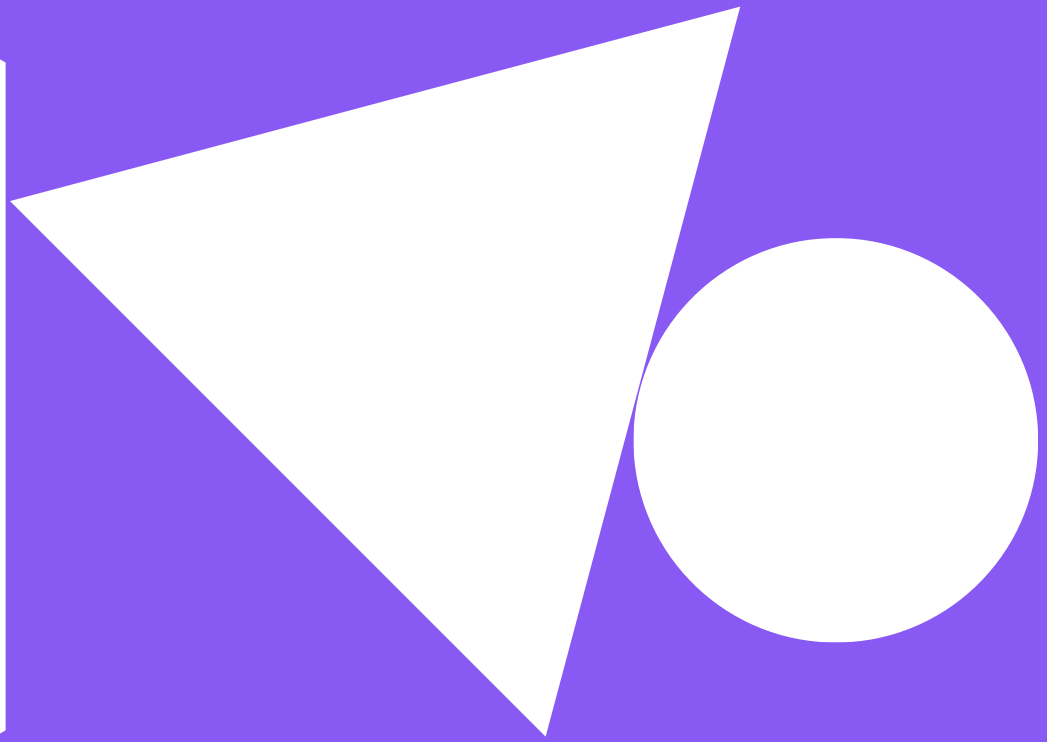
スーパースカラ アーキテクチャ

- プロセッサ ハードウェア コア**が**命令をスケジューリング
  - 一般的に、すべてのコアは同一である
- 汎用コンパイラ
  - ISA互換のバイナリであれば実行可能



Codasip

RISC-V Processors



→ 既製プロセッサのポートフォリオ

Codasip

Codasipのカスタム コンピュート技術
ひとつのベースから無限の可能性

組み込みプロセッサ		アプリケーション プロセッサ	
32-bit ベアメタル、RTOSアプリケーション		64-bit アプリケーション キャッシュ・コヒーレント・システム(最大4コア)	
L11	**L31**	**A70**	**A730**
RV32EMC アーキテクチャ 16 レジスタ シーケンシャル乗算器 Machine モード	RV32IMCB アーキテクチャ 32 レジスタ 並列乗算器 Userモード FPU 分岐予測器	RV64GC アーキテクチャ シングル イシュー FPU AXI-4 64インターフェース	RVA22アーキテクチャ デュアル イシュー FPU AXI-5 128インターフェース

コンフィギュレーションとカスタマイズによる最適化

Codasip

L11

電力効率と面積効率の良い**3段パイプライン**

圧縮命令で最小のコードサイズを実現

16レジスタ内蔵命令セット (RV32E)

最小面積の**シーケンシャル乗算器**

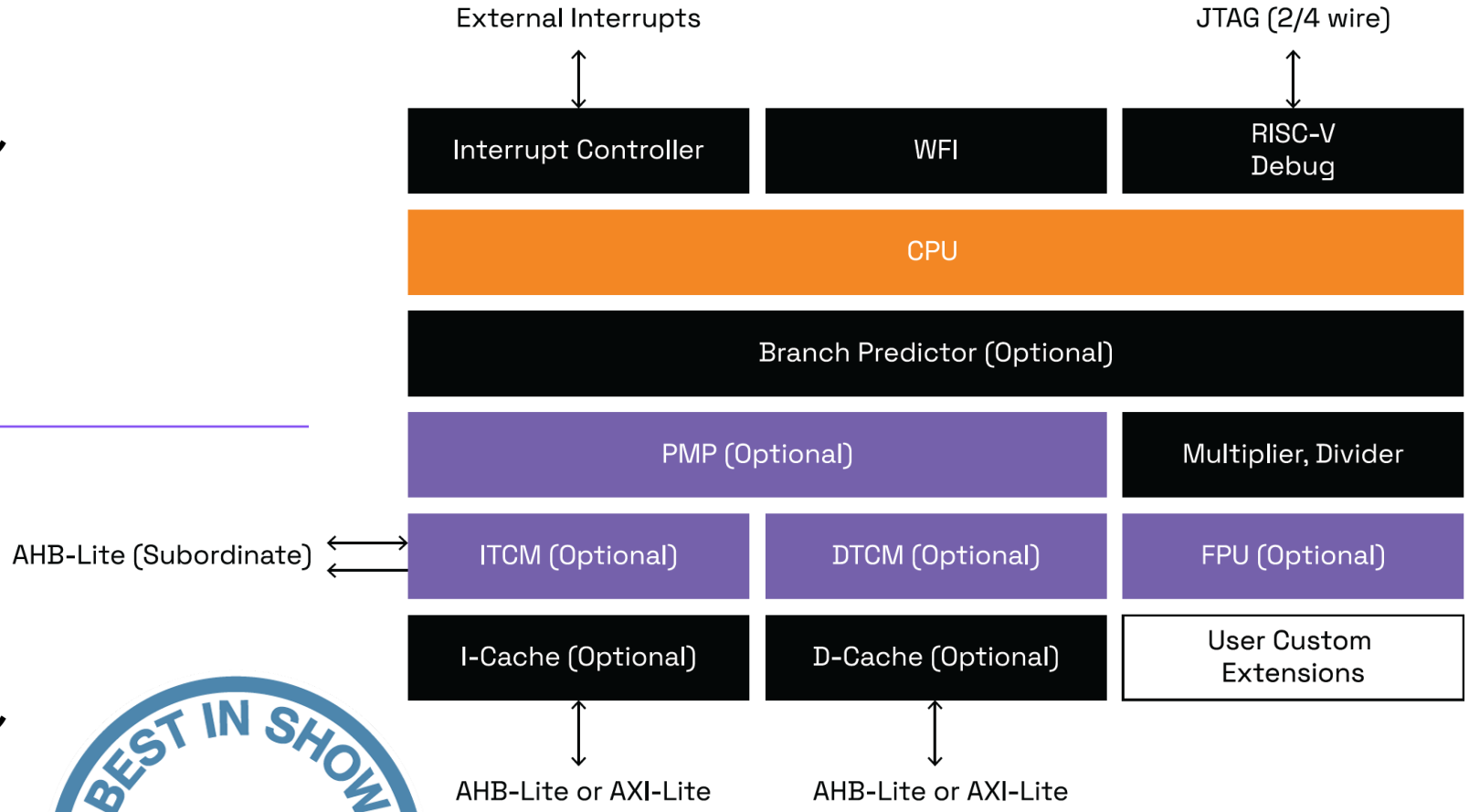
L31

電力効率と面積効率の良い**3段パイプライン**

圧縮命令で最小のコードサイズを実現

標準的32レジスタ内蔵命令セット (RV32I)

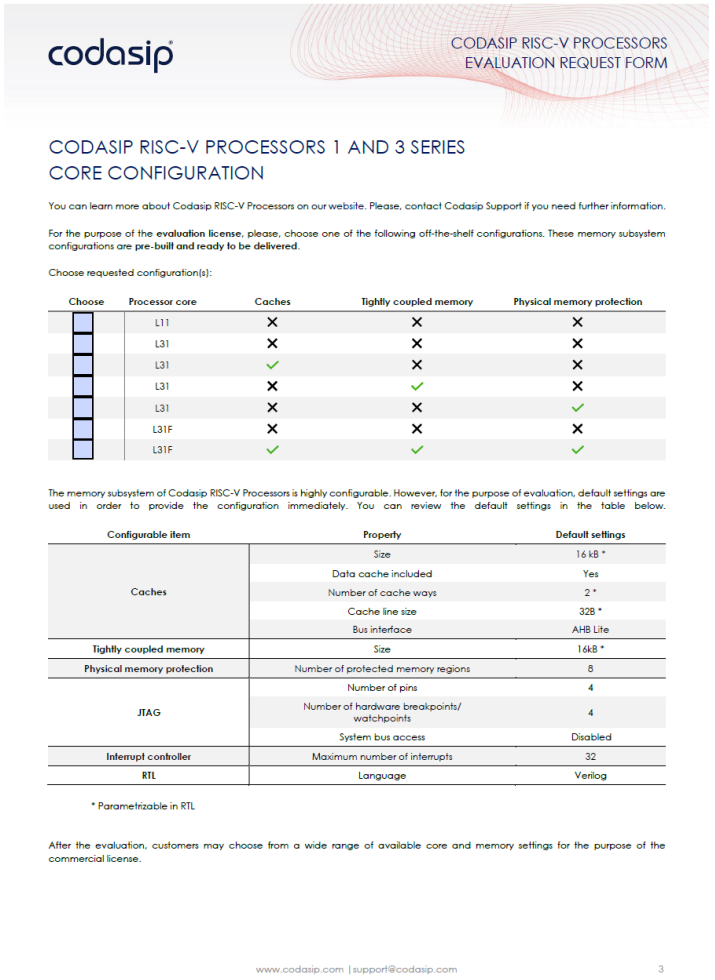
高性能な**並列乗算器**



→ 幅広いコンフィギュレーション

- 3シリーズには全てのメモリオプション有

- コンフィギュラブル L1 キャッシュ
  - サイズ, キャッシュウェイ数,
    キャッシュラインのサイズ
- 密結合メモリ
  - 最大2MBまでカスタマイズ可能
- RISC-V PMP (Physical Memory Protection)
  - 最大16個



codasip

CODASIP RISC-V PROCESSORS
EVALUATION REQUEST FORM

CODASIP RISC-V PROCESSORS 1 AND 3 SERIES
CORE CONFIGURATION

You can learn more about Codasip RISC-V Processors on our website. Please, contact Codasip Support if you need further information.

For the purpose of the **evaluation license**, please, choose one of the following off-the-shelf configurations. These memory subsystem configurations are **pre-built and ready to be delivered**.

Choose requested configuration(s):

Choose	Processor core	Caches	Tightly coupled memory	Physical memory protection
☐	L11	✕	✕	✕
☐	L31	✕	✕	✕
☐	L31	✓	✕	✕
☐	L31	✕	✓	✕
☐	L31	✕	✕	✓
☐	L31F	✕	✕	✕
☐	L31F	✓	✓	✓

The memory subsystem of Codasip RISC-V Processors is highly configurable. However, for the purpose of evaluation, default settings are used in order to provide the configuration immediately. You can review the default settings in the table below.

Configurable item	Property	Default settings
Caches	Size	16 kB *
	Data cache included	Yes
	Number of cache ways	2 *
	Cache line size	32B *
	Bus interface	AHB Lite
Tightly coupled memory	Size	16kB *
Physical memory protection	Number of protected memory regions	8
JTAG	Number of pins	4
	Number of hardware breakpoints/ watchpoints	4
	System bus access	Disabled
Interrupt controller	Maximum number of interrupts	32
RTL	Language	Verilog

* Parametrizable in RTL

After the evaluation, customers may choose from a wide range of available core and memory settings for the purpose of the commercial license.

www.codasip.com | support@codasip.com 3

Example config, some items subject to change

→ A70

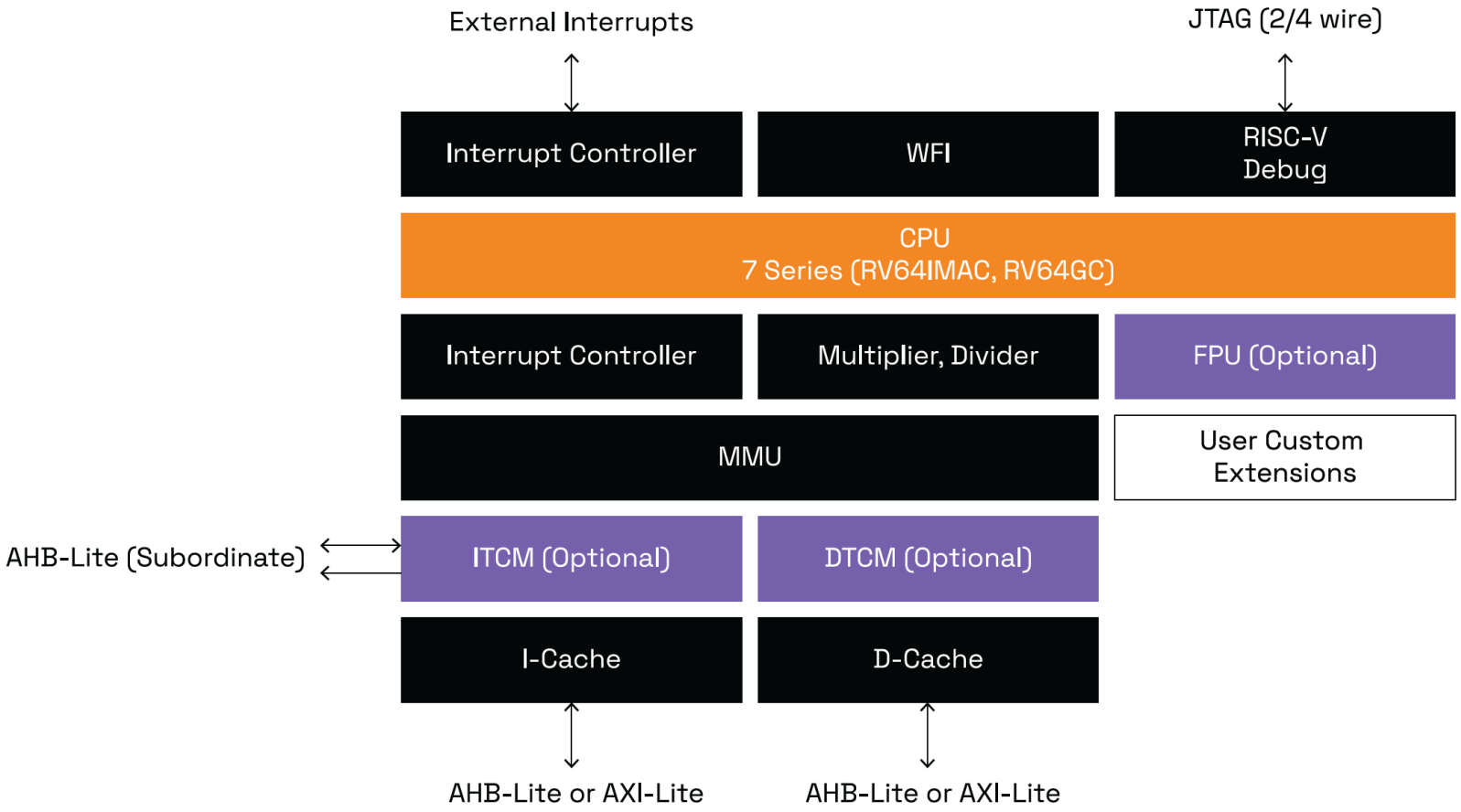
Codasip

複雑な7**段パイプライン**

圧縮命令で最小のコードサイズを実現

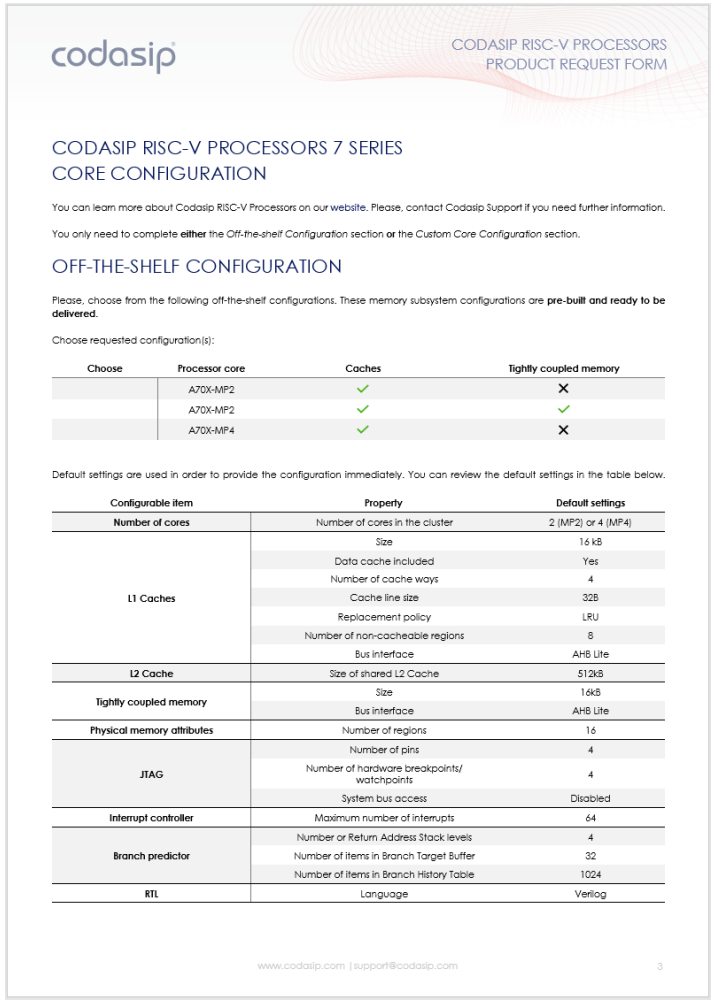
動的な**分岐予測**

2/4コア有



→ 幅広いコンフィギュレーション

- A70コア・コンフィギュレーション
  - Local memories (L1 cache)
  - Branch predictor and more
- コア数
  - Up to 4 identical cores
- コンフィギュレーション可能なL2キャッシュ
  - Up to 8MB
  - 64-bit AXI interface
- コンフィギュラブルな割り込み制御
  - Up to 1000 interrupts



codasip

CODASIP RISC-V PROCESSORS
PRODUCT REQUEST FORM

CODASIP RISC-V PROCESSORS 7 SERIES
CORE CONFIGURATION

You can learn more about Codasip RISC-V Processors on our website. Please, contact Codasip Support if you need further information.

You only need to complete **either** the *Off-the-shelf Configuration* section **or** the *Custom Core Configuration* section.

OFF-THE-SHELF CONFIGURATION

Please, choose from the following off-the-shelf configurations. These memory subsystem configurations are **pre-built and ready to be delivered**.

Choose requested configuration(s):

Choose	Processor core	Caches	Tightly coupled memory
	A70X-MP2	✓	✗
	A70X-MP2	✓	✓
	A70X-MP4	✓	✗

Default settings are used in order to provide the configuration immediately. You can review the default settings in the table below.

Configurable item	Property	Default settings
Number of cores	Number of cores in the cluster	2 (MP2) or 4 (MP4)
L1 Caches	Size	16 kB
	Data cache included	Yes
	Number of cache ways	4
	Cache line size	32B
	Replacement policy	LRU
	Number of non-cacheable regions	8
	Bus interface	AHB Lite
L2 Cache	Size of shared L2 Cache	512kB
Tightly coupled memory	Size	16kB
	Bus interface	AHB Lite
Physical memory attributes	Number of regions	16
JTAG	Number of pins	4
	Number of hardware breakpoints/ watchpoints	4
	System bus access	Disabled
Interrupt controller	Maximum number of interrupts	64
Branch predictor	Number or Return Address Stack levels	4
	Number of items in Branch Target Buffer	32
	Number of items in Branch History Table	1024
RTL	Language	Verilog

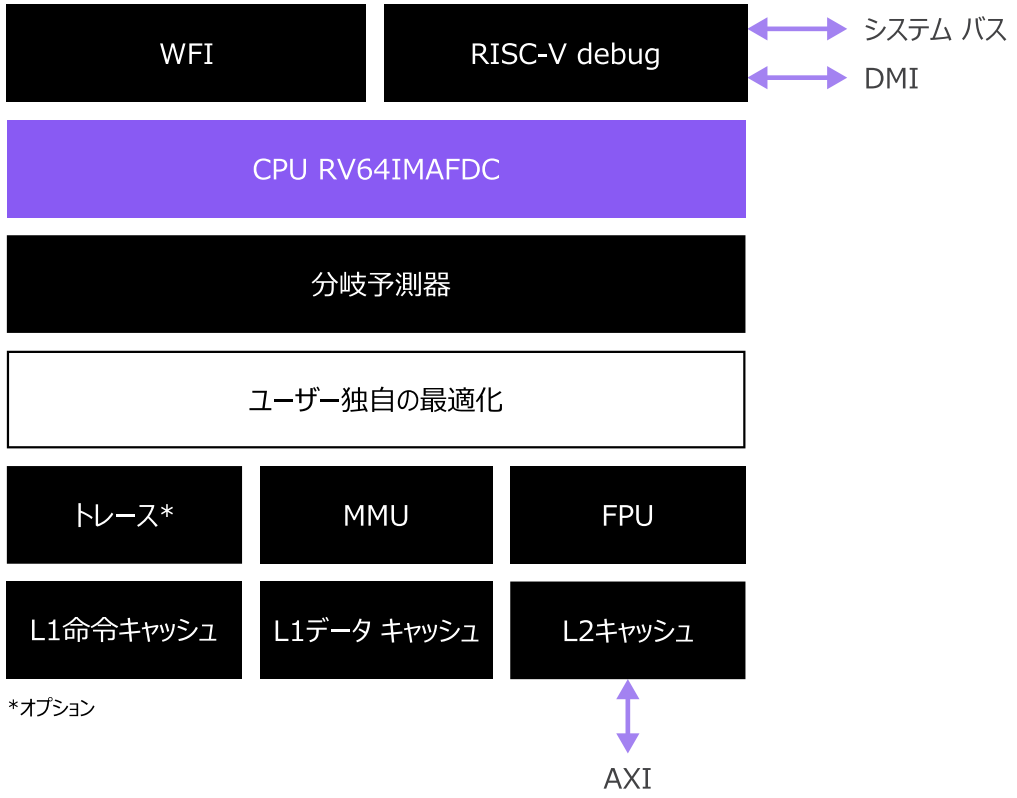
www.codasip.com | support@codasip.com 3

→ A730

64ビットRISC-Vアプリケーション プロセッサ

- **7段パイプライン**
- Linuxのようなリッチなオペレーティングシステムを
  サポートするためのMMU(メモリ管理ユニット)を搭載
- 柔軟性が高く、消費電力に制約のあるデバイスで複雑な
  計算タスクを実行する必要のあるアプリケーションが対象
- デュアル イシュー
- 高速で近接する
  キャッシュ・コヒーレント・メモリ・システム
- 前世代の2倍のパフォーマンス

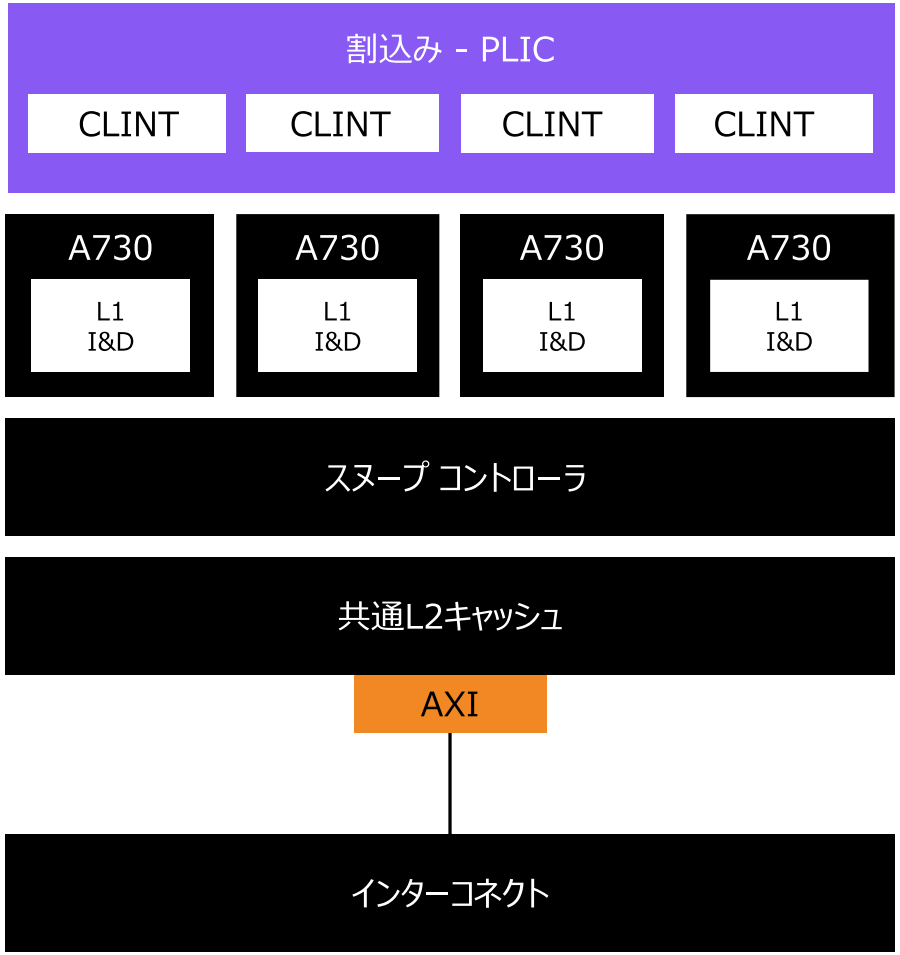
Codasip A730



A730 マルチコア

このミッドレンジ プロセッサは、シングルコア構成と、クラスタ内で最大4コアを使用できるマルチコア構成有り

Codasip A730マルチコア



→ A730仕様

Codasip

コア

- In-order, 7-stage, dual-issue pipeline
- RV64IMAFDC ISA
- RVA22 Compliant
- 1 to 4 homogeneous cores in a cluster
- Zcb for code-size improvement

分岐予測

- Increased single-thread performance
- Configurable with area performance options

割込み

- WFI and NMI
- PLIC

L1キャッシュ

- Private to the core
- Instruction and data
- 16, 32, or 64KB
- 4 ways

メモリ プロテクション

- Physical memory attributes with 8, 16, 32 regions
- MMU supporting Sv39 memory virtualization
- 16, 32, 64 instruction micro-TLB entries
- 16, 32, 64 data micro-TLB entries

デバッグ

- Standard RISC-V debug
- 4 debug triggers
- Debug Memory Interface (DMI) as an APB interface that can be used to interface with external debuggers
- System bus access

パフォーマンス

- Performance monitors

L2キャッシュ

- Shared in multicore configurations
- 512KB to 2MB
- 4,8,16 associativity

トレース

- RISC-V standard E-trace interface
- Instruction and data trace
- Optional

インターフェース

- AXI-5 with atomic support
- 128 bits

消費電力

- Low-power design
- Always-on power domain
- Granular architectural clock gating

→ ユニークなIPライセンス モデル

Codasip

License / Rights	Other vendors		Codasip	
	Fixed IP	Architectural ("VIP" customers only)	Fixed IP	Modifiable IP
Integrate CPU IP in SoC	●	No IP provided	●	●
Embed CPU IP in larger IP			○	●
Re-design CPU IP		●		●
Produce SoC with CPU IP	●	●	●	●
Modify existing CPU IP		No IP provided		●
Get full CPU IP source code & doc				●
Get re-verification suite for CPU IP				●
Cost	$	$$$$$	$	$$

● Standard
○ Option

→ 今後のプロセッサ コア ロードマップ

Codasip

組込みコア

High-performance embedded processors

Digital signal processing

Functional safety (ASIL D certification)

Security (CHERI)

アプリケーション コア

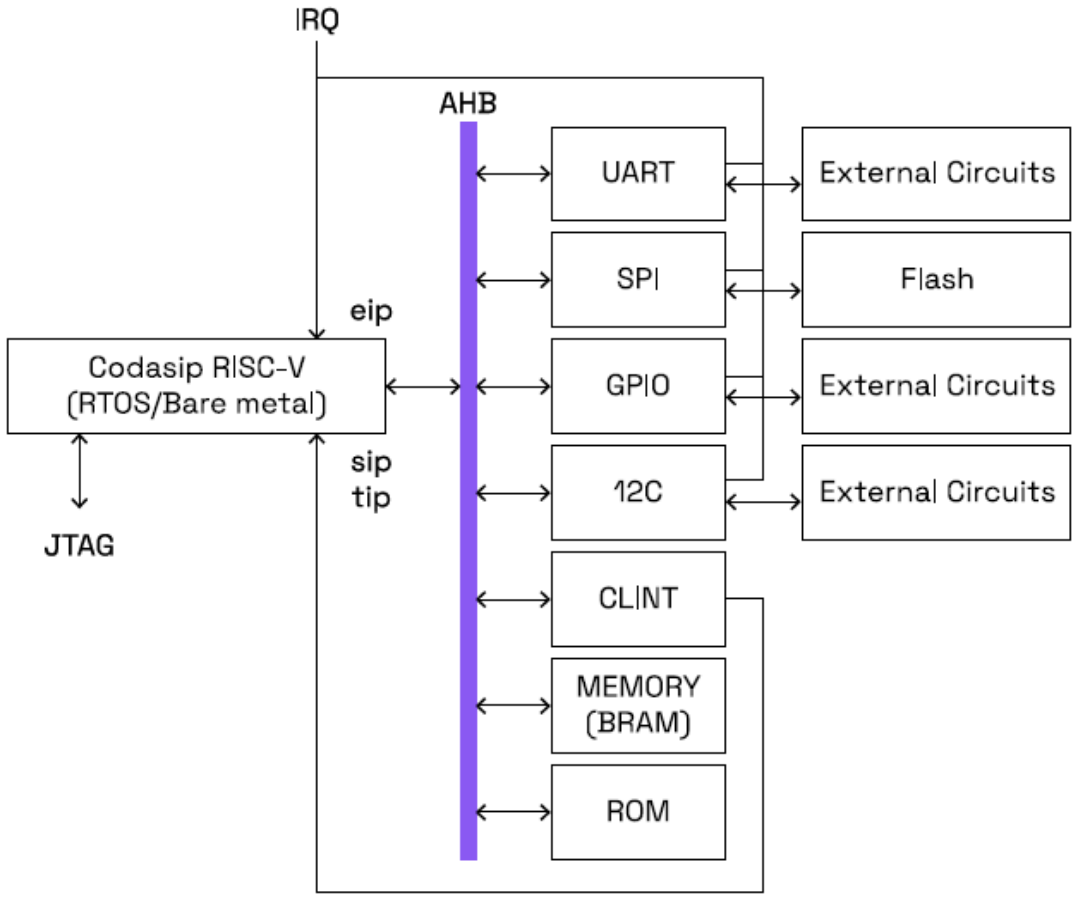
Security mechanism

High-performance processors

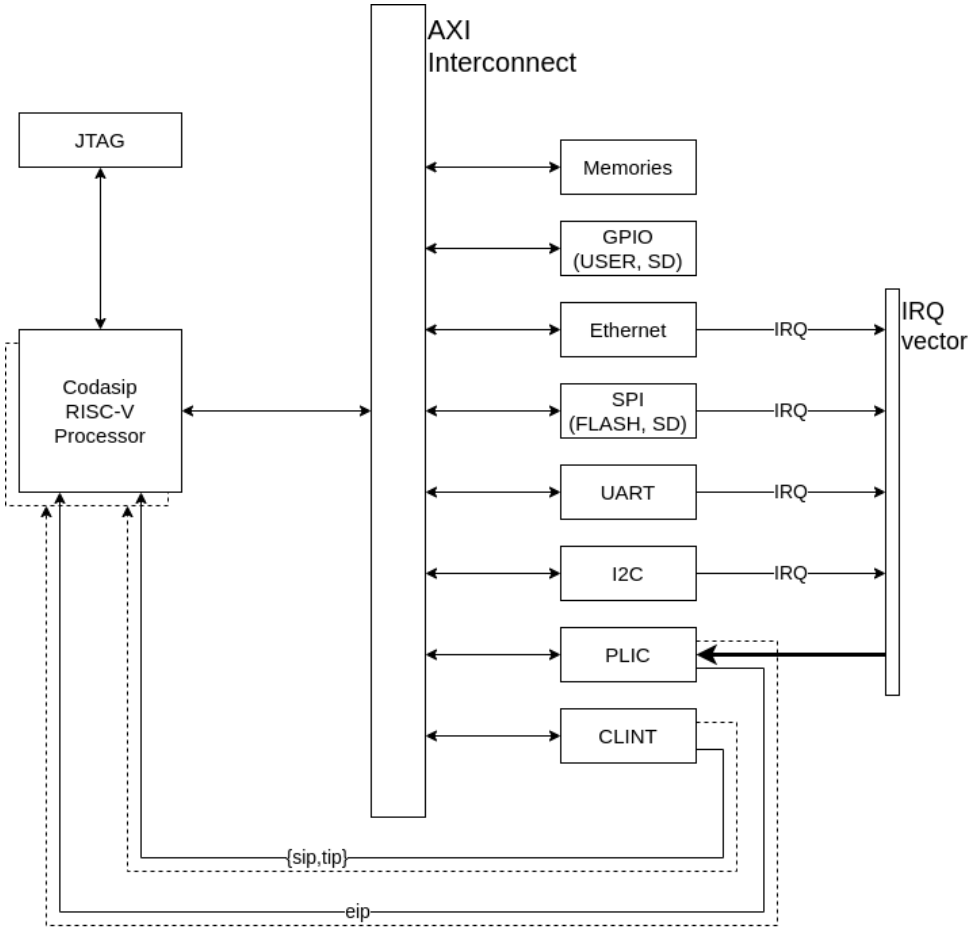
Vector Compute

→ FPGAベースの評価プラットフォーム

Codasip



Codasip evaluation platform for embedded cores



Codasip evaluation platform for application cores

Codasip

→

Thank you!

12:00-17:00 **ホワイエB**ブース

製品紹介・展示

16:15-16:35 ギャラリー1

FPGAを用いた**CHERIデモ**

お問い合せは、contact_japan@codasip.com