

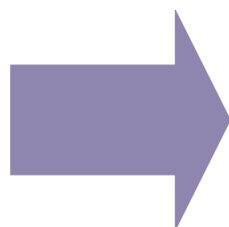
AXEのカスタムLSI設計サービス、 ROS2通信ハードウェアIP提供

2024/DEC
AXE, Inc.

もう、CPUは(簡単には)速くならないよ

- ついに、微細化 限界
- 微細化 による
 - 高集積
 - 高クロック周波数
 - コア数 増加

は終了



- 専用 ロジック回路による
 - 高速化
 - 省 消費電力
- を行うしか

背景:
カスタムLSI
設計&製造の
民主化

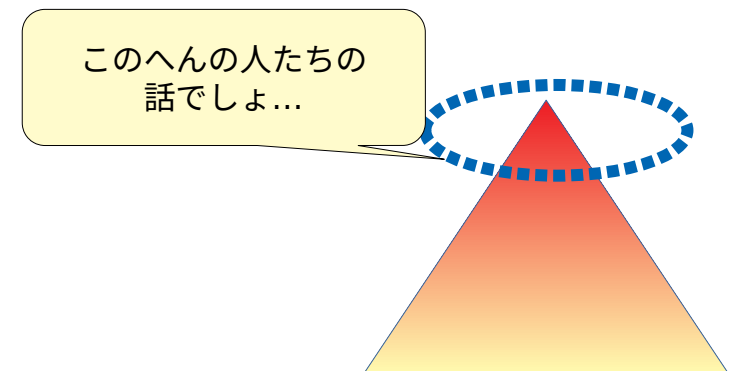
日本の半導体 産業 復興!

- ・ 国内8社が半導体製造会社「Rapidus」設立
- ・ 経産省キモ入り 「10年の遅れ」取り戻す
- ・ キオクシア、ソニーグループ、ソフトバンク、デンソー、トヨタ自動車、NEC、NTTがそれぞれ10億円、三菱UFJ銀行が3億円を出資した。



<https://www3.nhk.or.jp/news/html/20221111/k10013887921000.html>

- ・ 半導体 工場は、かろうじて 最新のものがある
- ・ でも、「お高いんでしょう〜?」
- ・ 技術者 不足
 - ・ 半導体 設計 技術者
 - ・ 論理回路 設計 技術者



技術者 不足をStop!日本の半導体 産業 復興!

・OSSの開発ツールで、LSI 開発

- 無料ツールの使い手が増える → 技術者不足 解消!

- 半導体 設計 技術者

- 論理回路 設計 技術者

・半導体 工場は、「お高くない」ものもある

- 65nm とか、安くて かなりいい

・レガシーファブの活性化



みんなのLSI
俺のASIC

お金持ちだけの
LSI

LSI開発の民主化だっ!!!

産総研 直下のAIST Solutionsも、 ロングテール半導体開発を推進

Open-Source は、ロングテール 半導体開発の夢を見るか？

- Do Open-Source Dream of Long-Tail
Semiconductors? -

11.17(金) 15:30-16:10 | 展示会場内 Room E

AIST Solutions
半導体事業 プロデューサー 岡村 淳一



2023/12/06

ONLY for 2023 Edge-Tech Conference

AISol の半導体事業の目標

国内の半導体アセット（チップ製造能力）を
本プロジェクトのプラットフォームに再整備することで
専用半導体の設計の参入障壁を下げ、
国内産業が専用半導体にて国際競争を勝ち抜く環境を提供
する。



Open Source Silicon for Japan



2023/12/06

ONLY for 2023 Edge-Tech Conference

Googleがカスタム半導体の民主化・自由化を推進

- Googleと半導体ファウンドリの「SkyWater」が協力し、業界初となるオープンソースのPDKを公開
 - Skywaterは2017年に米Cypress Semiconductorからスピノフしたファウンドリ企業
- PDK プロセス設計キット
- ある特定の半導体プロセスで回路設計を行う際に必要な設計情報
- **トランジスタ配置の制約条件などが書かれている**
- 半導体の設計者は、半導体製造のファウンドリから「Process Design Kit(PDK)」と呼ばれる開発キットを購入
- 半導体ファウンドリが提供するPDKは高価 → それが**無料OSSに!**
- SkyWaterの130nmプロセス「SKY130」で半導体チップの製造を行うための設計を無料で行うことが可能
- GitHub - google/skywater-pdk: Open source process design kit for usage with SkyWater Technology Foundry's 130nm node.

<https://github.com/google/skywater-pdk>

US DARPAによる OpenROADへの投資

川崎俊平氏の資料より引用

https://riscv.or.jp/wp-content/uploads/RVDs2022Spring_SHC-2022-06-02_c.pdf



DARPAによるOpenROADへの\$200Mの投資

- オープンソース EDAソフトウェア
 - 数十年前から存在。最先端ノード製造工程の複雑で商業ツールに遅れた
 - EDAツールのコスト: 3つのEDA会社が25%を研究開発に投資
 - Nvidia、Apple、Qualcommなどエンタープライズ顧客
 - \$10のIoTチップを作る小さなチーム
- 20の既存ツール → 1アプリにバンドル → エンドツーエンドのフロー
 - Qflow: Verilogソース → 物理レイアウト
 - SPICE: 回路シミュレーションプログラム
 - Magic: VLSI回路レイアウトの作成と変更
 - Mead-Conway設計手法で、単純ルールで、基本セルを階層設計
 - VIS: Verification Interacting with Synthesis

国内もLSI開発の民主化 推進

日本も、国の金で 施設を用意

ふくおかIST(公益財団法人 福岡県産業・科学技術振興財団)

福岡システムLSI総合開発センター

「システム L S I 設計試作センター」

- http://www.ist.or.jp/lsi/pg04_02.html
- ベンチャー企業が半導体の設計ツールを無料(安価)で利用できる
- LSI設計、少量 試作できる
 - 50～100万円 あれば、LSIの少量生産ができる仕組みがある

小規模EDA開発 日本でも流行

福岡システムLSI総合開発センター

「システムLSI設計試作センター」

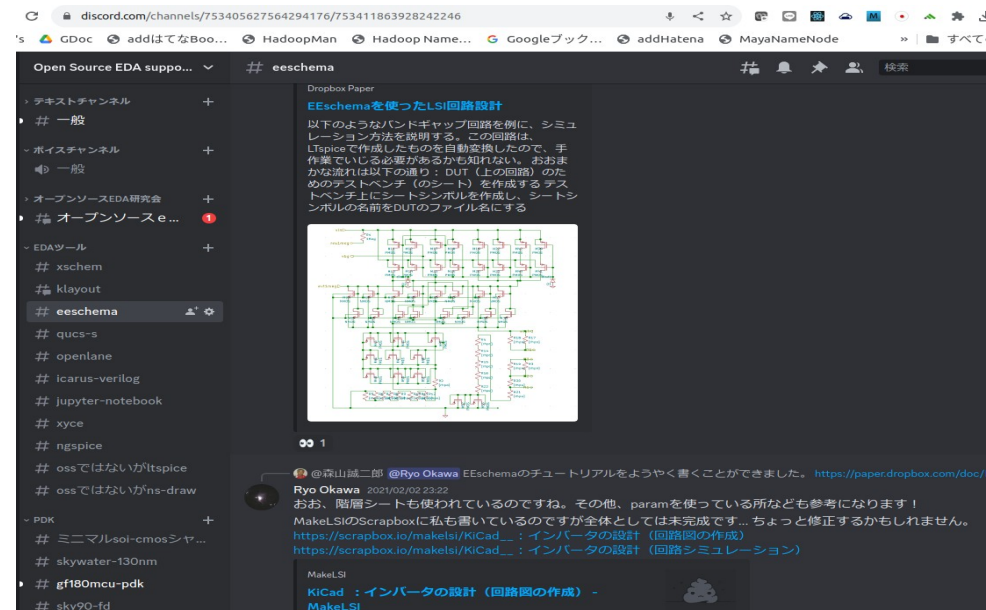
の設計ツール一覧

EDA機能		製品名
ハイレベル設計	クレベル合成	* CyberWorkBench ※NECの商品
フロントエンド設計	論理シミュレータ	* Incisive Enterprise Simulator L
	回路図エントリ	* Schematic Editor
		* ASCA
		* ASCA Basic
	シミュレーションIF	* Virtuoso ADE
		* ASCA Sim.faceA
	総合回路設計	* C ³
	Composer IFオプション	* Composer IF
	Verilog Interfaceオプション	* Verilog Interface
	SPICE Interfaceオプション	* Analog HSPICE IF
	アナログ回路シミュレータ	* Spectre circuit Sim
		* Msim
	汎用回路波形解析	* SimVision
レイアウト	レイアウトエディタ	* Virtuoso LE
		* ISMO
	Cadence Linkオプション	* Cadence Link (DF II Upgrade)
レイアウト検証 その他	DRC	* Calibre DRC
	LVS	* Calibre LVS
	IFオプション	* Calibre RVE
	DRC/ERC	* iDRC/ERC
	Caliber IFオプション	* Calibre IF
	寄生パラメータ抽出	* Calibre xRC

Open Source EDA Supporters (discord)

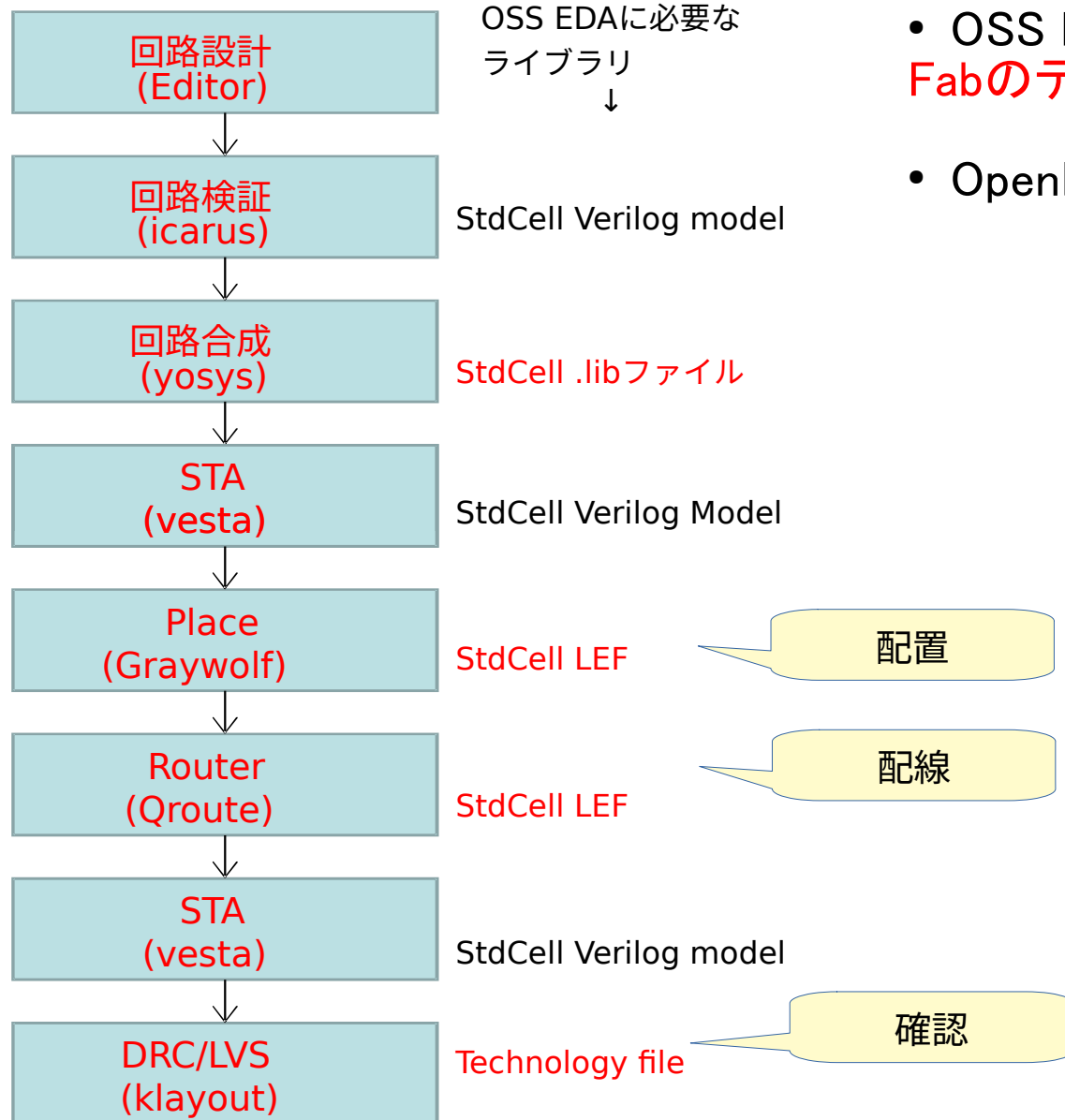
<https://discord.com/channels/753405627564294176/932209975558832128>

- OSS EDAをサポートする人たちの集まり
 - 親切で優しい
- ふくおかIST のOSS EDAサーバのユーザ会
 - サーバを維持するモチベーション
- オレオレ Open PDKを開発したり: 森山氏
- オレオレ Standard Cellライブラリを開発できるツールの開発者が居たり: 西澤氏@早稲田大
- オープンソースEDAフォーラム @福岡
 - 2023/DEC/08
 - OSSコンソーシアムEDA部会が共催
 - JASA OSS活用WG協力
- OSC福岡にも出展
 - OSSコンソーシアム&協力: JASA OSS-WG体制
 - 2023/DEC/09



OSSによるLSI開発のEDAフロー

ロジック部開発フロー



- OSS EDAに必要なライブラリは
Fabのデータから変換が必要

- OpenRAM、PLL、アナログは別なフローになる

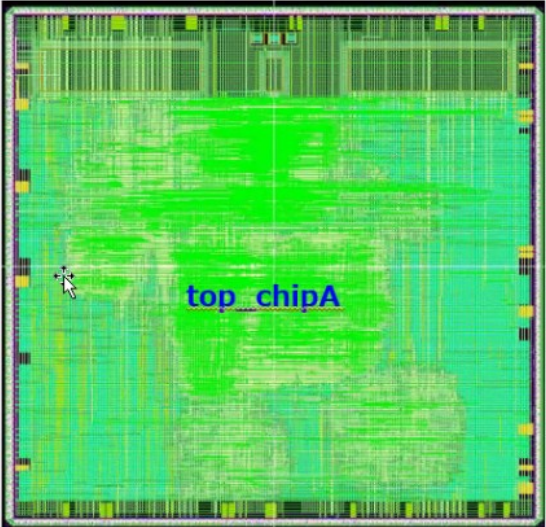
AXEの完全オリジナルLSI

”Laxer AX1001”概要仕様

TSMC 90nmで製造,920万トランジスタ
LQFP 100pin
27/SEP/2024 京都で1stシリコン動作

俺SoC

機能	数	備考
CPU	1	RISC-V RV32Iに、4SIMD 8bit浮動小数点ベクトル機構、ハードウェアによるマルチタスク制御機構、多重分岐命令（特許取得済、Prolog,Lisp,JavaVMの実行を加速）と、それらを操作する命令を付加
RAM	1	プログラム・コードRAM 64KBytes。reset後、bootloaderにより、SIO経由でプログラム・コードを配置可能。CPUコアによるコードRAMの内容変更はできない
	1	データRAM 64KBytes
SIO(UART)	3	SIO0(UART0)は、bootloadingとデバッグ・コンソールに使用される。UART1,2はユーザが自由に使用できる
GPIO	38	18本はPullUP指定可能。20本はPullDown指定可能。2本は、SIO1と兼用。2本は、SIO2と兼用。18本は外部イベント源として使用可能。（※外部イベントとは、一般CPUにおける割込のようなもの。外部イベントが発生すると、イベントについて待機しているスレッドが起床する）
AWG	1	16bit出力,1ch。出力ピンは、PWMと兼用。PWMと排他的に使用
PWM	8	出力ピンは、AWGと兼用。AWGと排他的に使用
Ether I/F	1	NICはオンチップであり。PHY I/Fを持つ（PHYは外付け）
ROS2通信機能	1	"ROS2rapper"という名称の、新開発のハードウェアによるROS2通信機能。CPUから初期化することで、自動的に通信を行う
外部SRAM I/F	1	32bit,入力/出力独立。ただし、AX1001では外部ピンの制約により使用できない



(株)アックス(1992年創業)の基本ソフトの採用実績



国スパコン富岳 OS研究

富岳にはMcKernelが採用されている



自動運転Autware
(ROS使用,OSS)

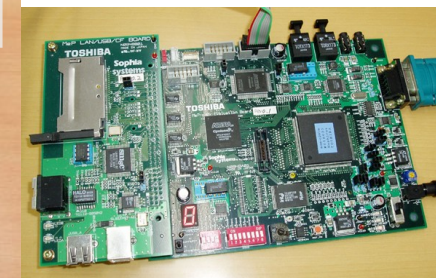
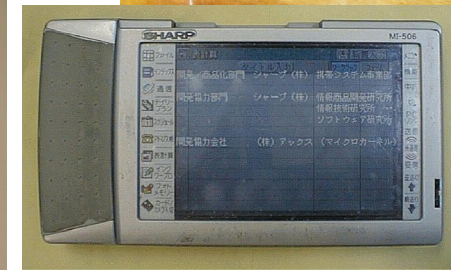


ロボット(産総研)

小さなもの/IT家電用OS/Linux

資本金+資本準備金=5億3千万円
Nextyエレからも資本が入っている
村井純先生(インターネット殿堂入り)も株主だ

・日本の独自CPUの基本ソフトウェアをサポート
東芝MeP, セイコーエプソンC33,C17、ルネサス(旧日立製作所)SH-Mobile,SH2A,
ルネサス(旧NEC)V850, 富士通 FR/V, シャープLH795xx



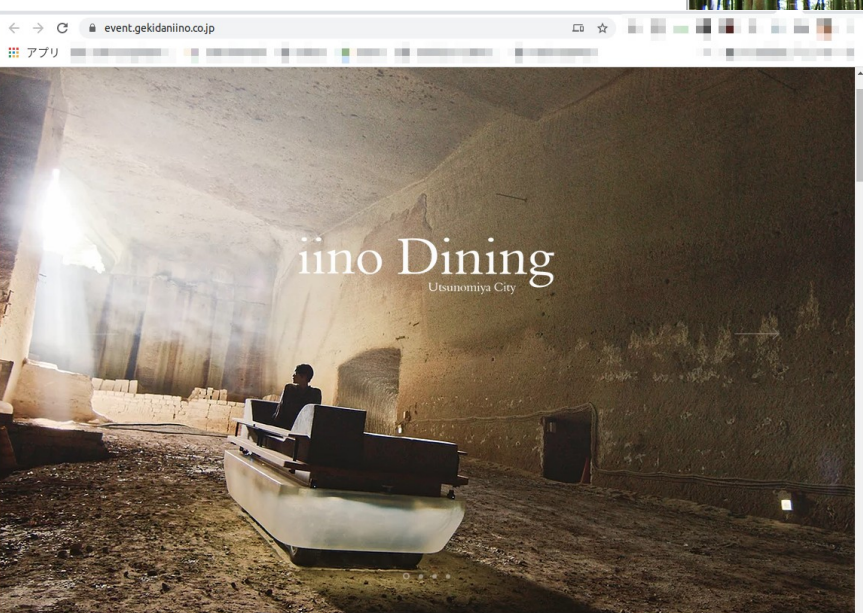
Autoware(ROS使用)応用 iinoプロジェクト

• iino(ゲキダン イイノ)

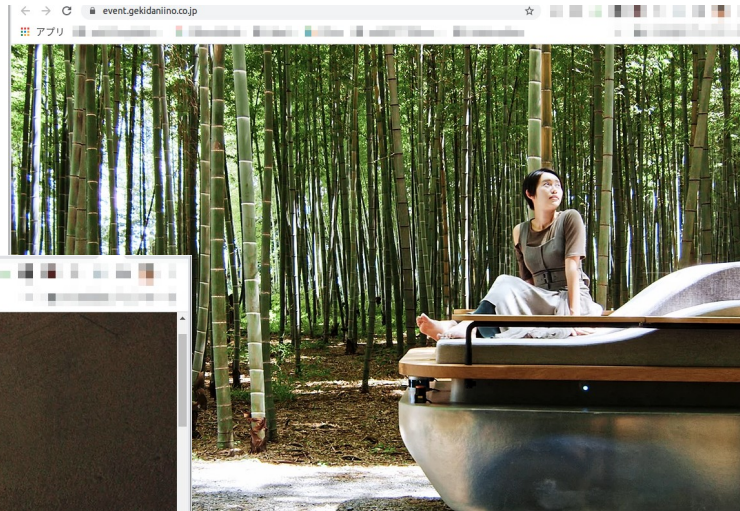
- 関西電力の新規事業プロジェクト
- 時速 5km/h でゆっくり走行

<https://gekidaniino.co.jp/>

<https://iinomob.jp/>



宇都宮 大谷資料館(採石場跡地)



宇都宮 竹林若山農場
月岡 不心



2023年11月 御堂筋で公道走行

<https://gekidaniino.co.jp/wp/wp-content/uploads/2023/11/%E3%83%88%E3%83%83%E3%83%97%E3%83%9F%E3%83%8B.png>より引用

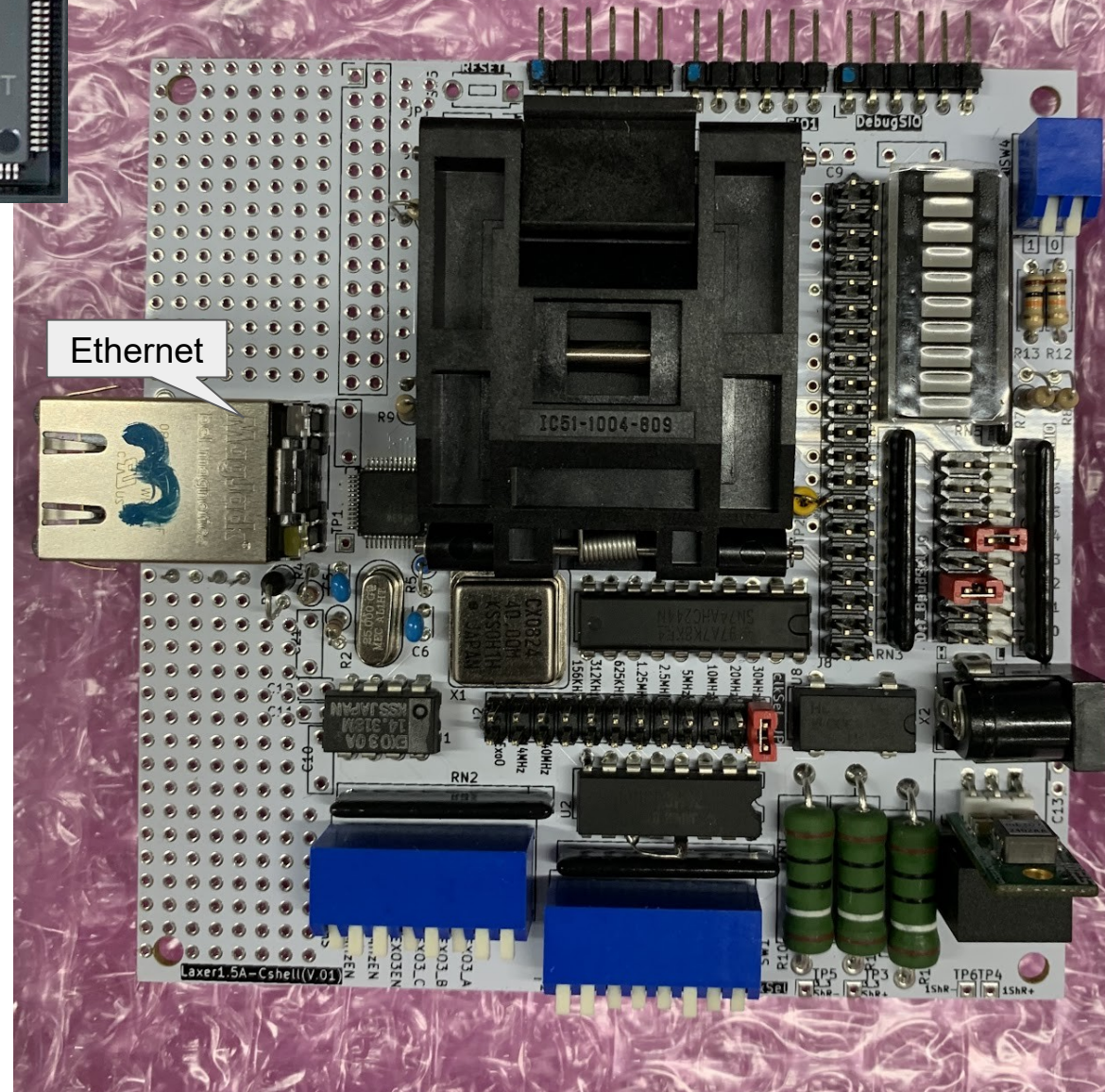
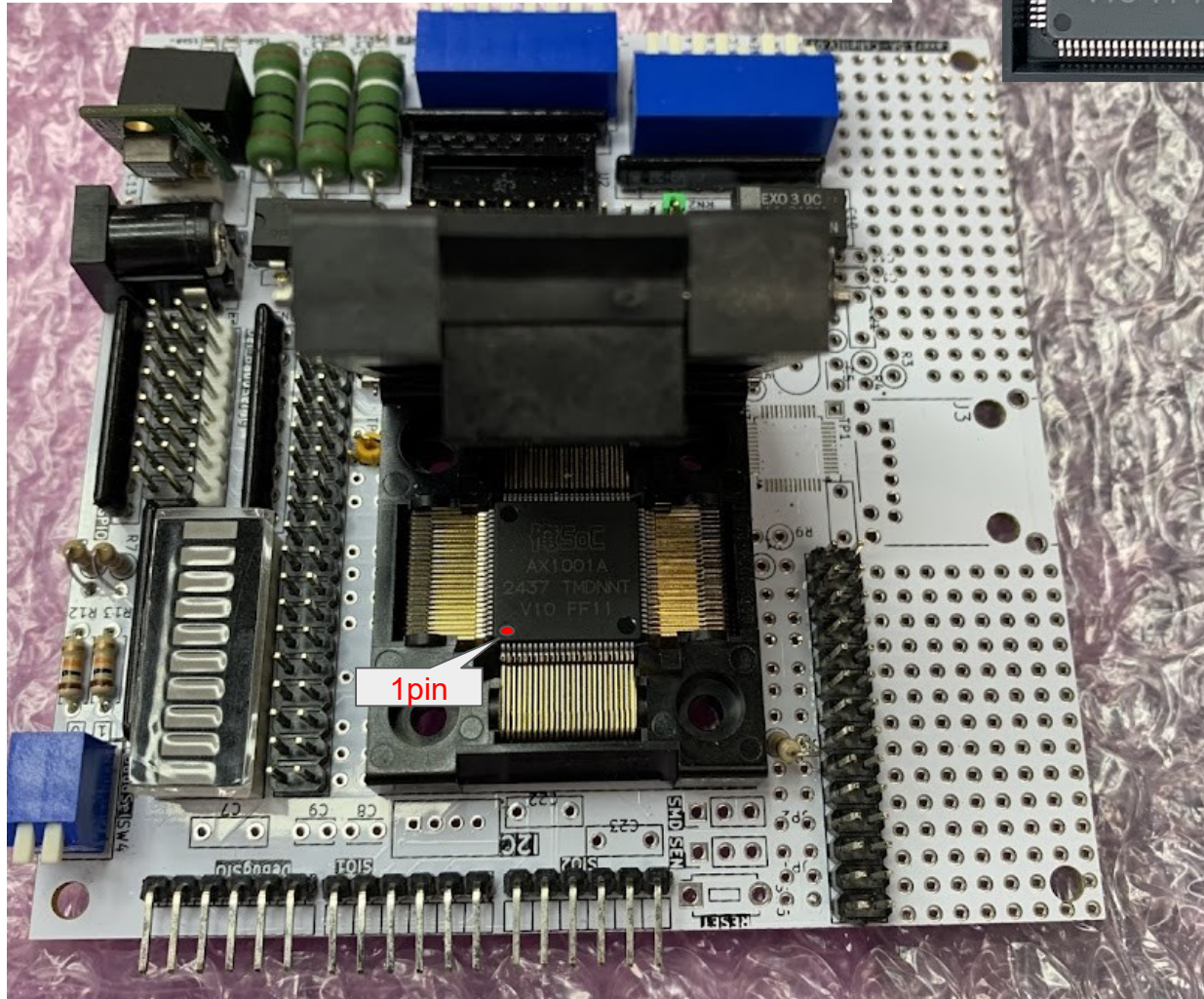


- 街なかを移動
- チョイ乗り

AXEの”俺SoC” AX1001

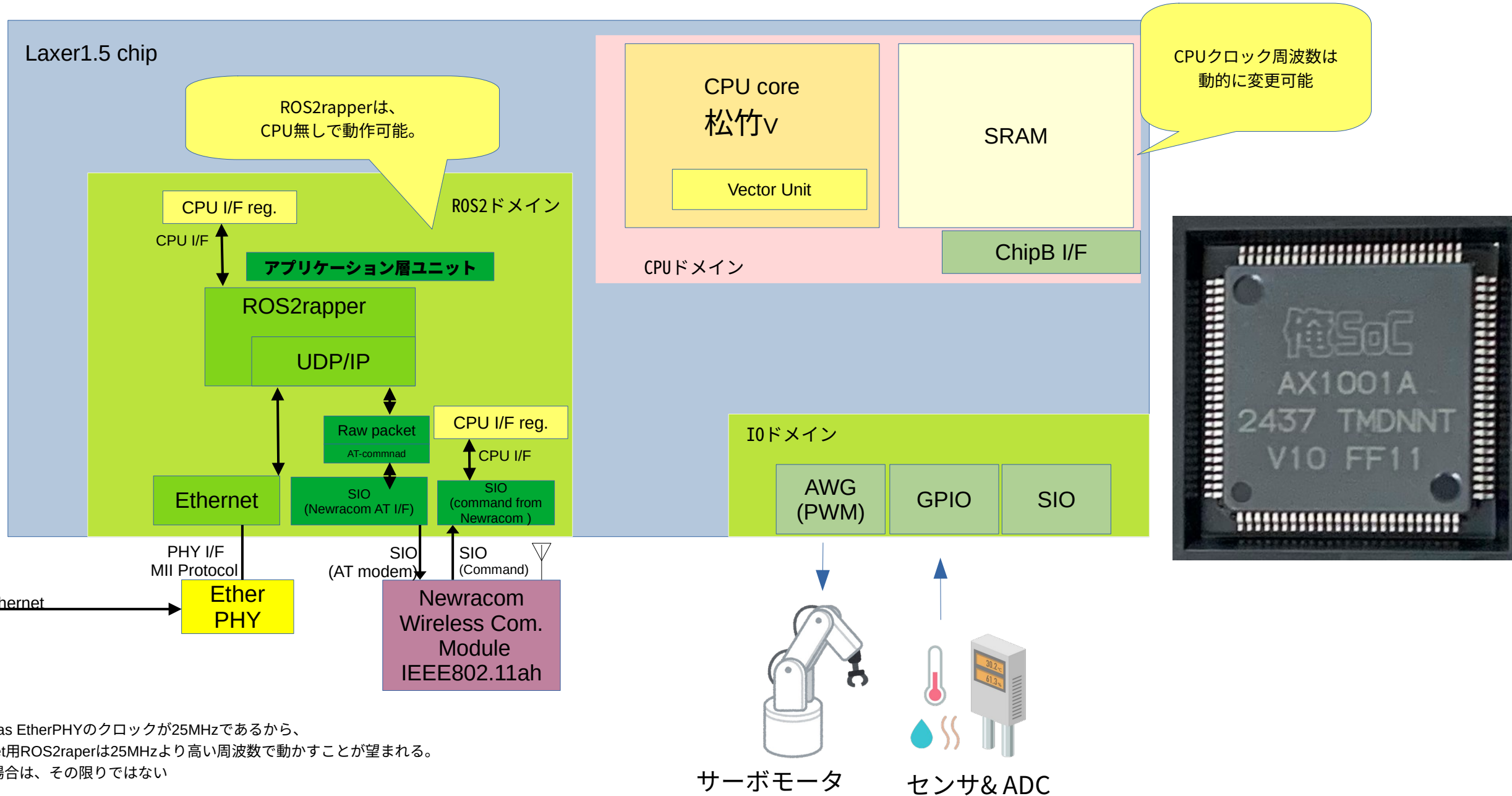
LEDチカチカ https://drive.google.com/file/d/10z89qI9ZQaNe8txQFIMjL5HRLol7bTh4/view?usp=drive_link
ROS2Subscribe https://drive.google.com/file/d/1M2Jt1u0B1dGReELrE8qoQbPvnz0AZI3c/view?usp=drive_link
ROS2Publish https://drive.google.com/file/d/1CCb1PVTXruBFXhebgIjcXFeVvfWzMloC/view?usp=drive_link

テスト基板
Ethernetで、ハードウェアROS2が動作



LEDチカチカ https://drive.google.com/file/d/10z89qI9ZQaNe8txQFIMjL5HRLol7bTh4/view?usp=drive_link
ROS2Subscribe https://drive.google.com/file/d/1M2Jt1u0B1dGReELrE8qoQbPvnz0AZI3c/view?usp=drive_link
ROS2Publish https://drive.google.com/file/d/1CCb1PVTXruBFXhebgIjcXFeVvfWzMloC/view?usp=drive_link

Laxer1.5 ChipA(AX1001)ブロック図



“俺SoC”, とことんOS無し

- ・ 省電力、省メモリ、堅牢かつ高速
- ・ ロボットの部品モジュールがローコストで簡単に作

俺SoC



オレ達のCPU「松竹V(しょうちくぶい)」

- ・ 機械学習AI 加速用 ベクトル計算ユニット(8bit float, 4SIMD, パイプライン)
- ・ 論理推論加速 機構をRISC-Vコアに追加
 - ・ 特許取得済み(第7506718号, 2024年6月18日 (東京エレクトロンと共同特許))
- ・ GnuPrologのコンパイルド・バイナリを加速
- ・ ハードウェア・マルチスレッド機構
 - ・ OSソフトウェア一切なしで、スレッド切り替え
 - ・ ハードウェア・セマフォで排他/同期。LR/SCもある
 - ・ 外部ピンからの入力で、スレッド起床(ハードウェアのみで)
 - ・ 割り込みなし(割り込み相当の処理は、専用スレッドで)

エッジデバイスでも
大脳的処理を!

OSプログラム・コード
OSワーキング・エリア
不要!

R0S2通信ハードウェア”R0S2rapper”を搭載

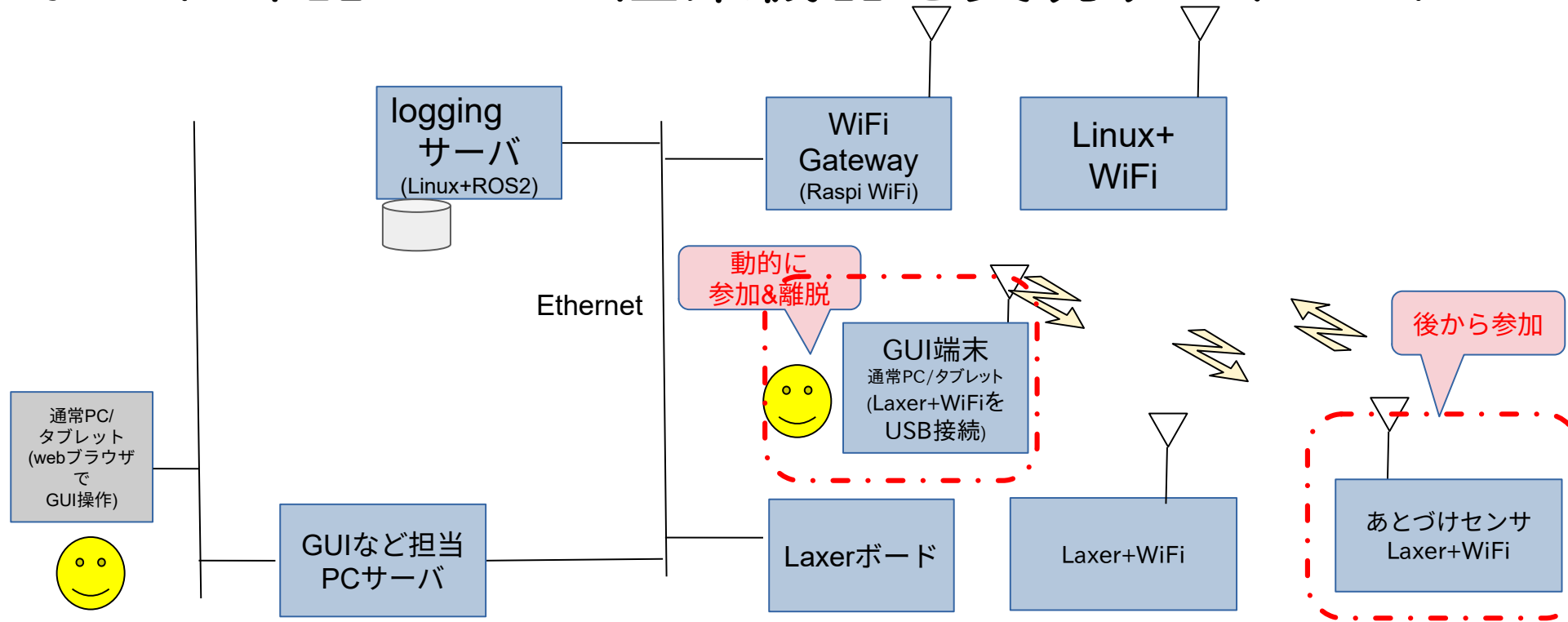
- ・ CPUの助けなしにR0S2通信

OS 不要のROS!

外部I/O: Ethernet I/F, 任意波形生成器(AWG(PWMにもなる)), GPIO



LaxerAX1001は、ROS2通信で、 ロボット部品/エッジ産業機器を実現する、マイコン・チップだ

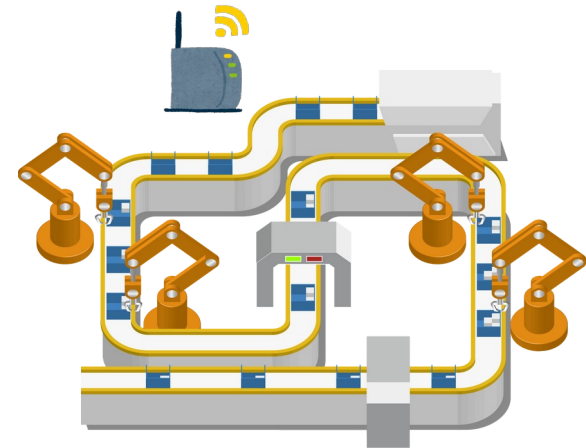


全ノード、ROS2で通信し、動的に参加者(センサなど)を発見できる



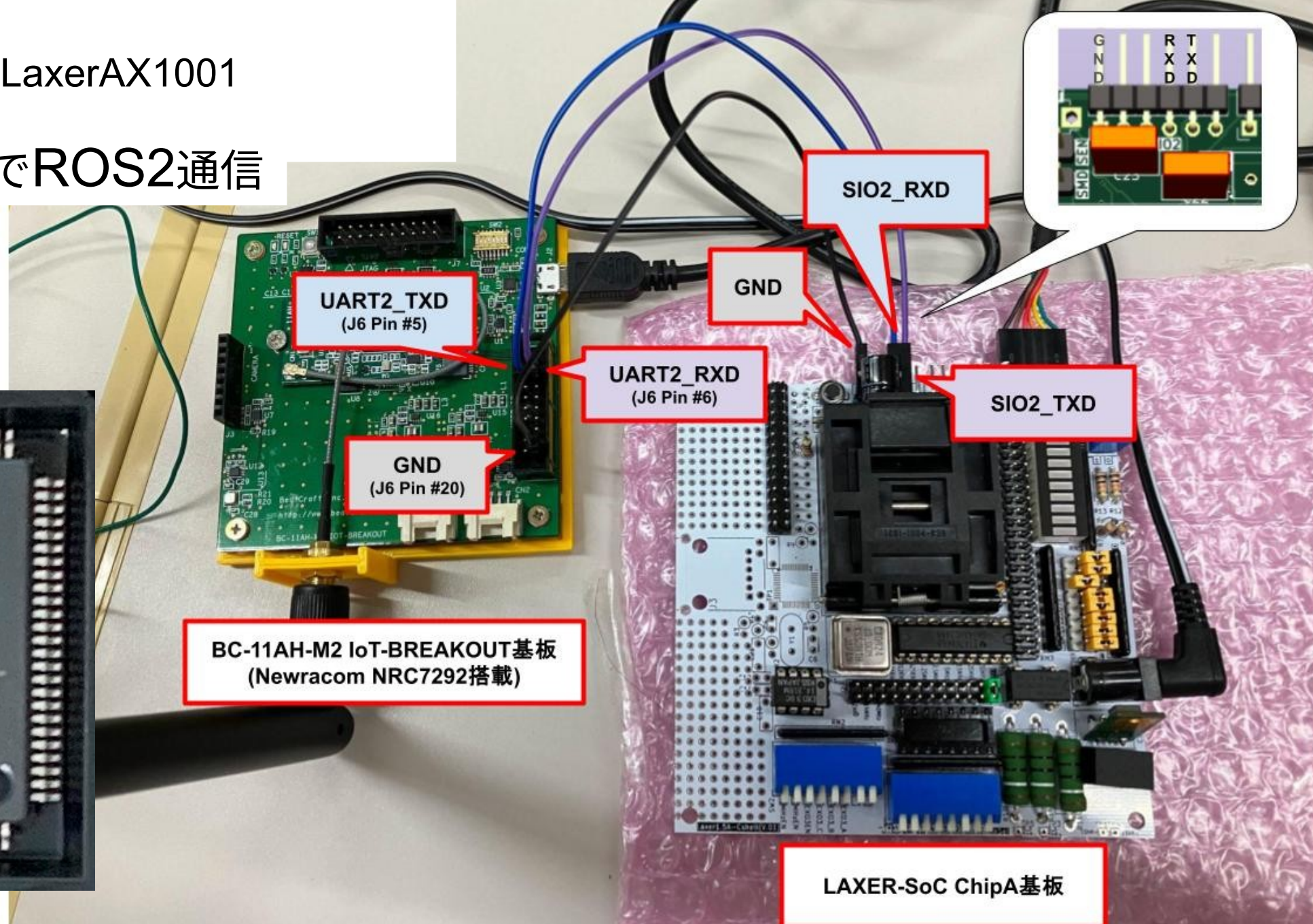
小さなデータであれば、帯域も問題無し
工場、製造業(FA)も、OK

無線センサが、ROS対応だといいね!



Newracom(802.11ah)+LaxerAX1001

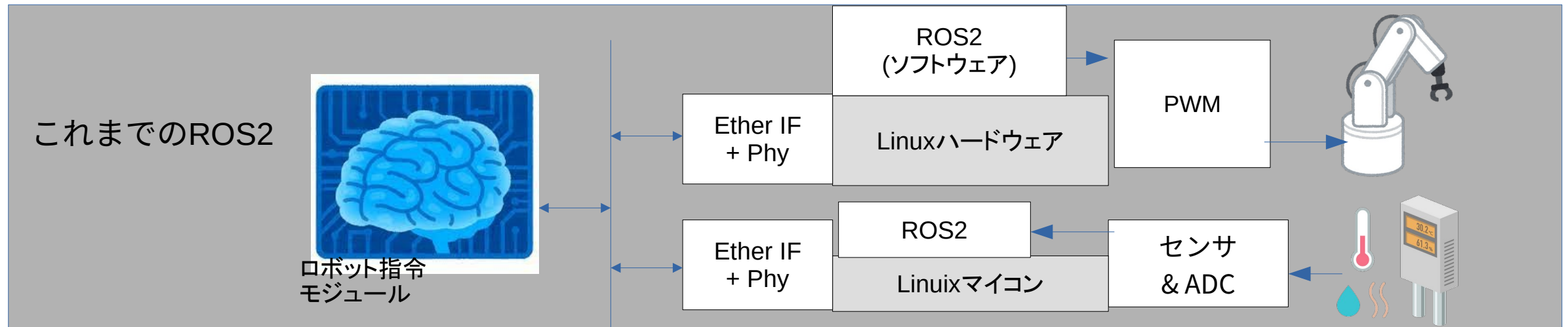
- IEEE802.11ahでROS2通信



ROS2プロトコル

- ROS2は、ロボット業界のデファクトスタンダードになろうとしている
 - ロボットの部品モジュールはROS2プロトコルで結びつける
 - ロボットの部品モジュールの流通性を高める
 - 自動運転でも使用されている
- ROS2は、これまで Linux、一部のRTOSで動作するソフトウェアだった
 - × Linuxは、MMUのついた高級なCPUで動作。メモリが多く必要。高電力消費。

AXEでは、ROS2プロトコルを完全ハードウェア化した”ROS2rapper”を開発

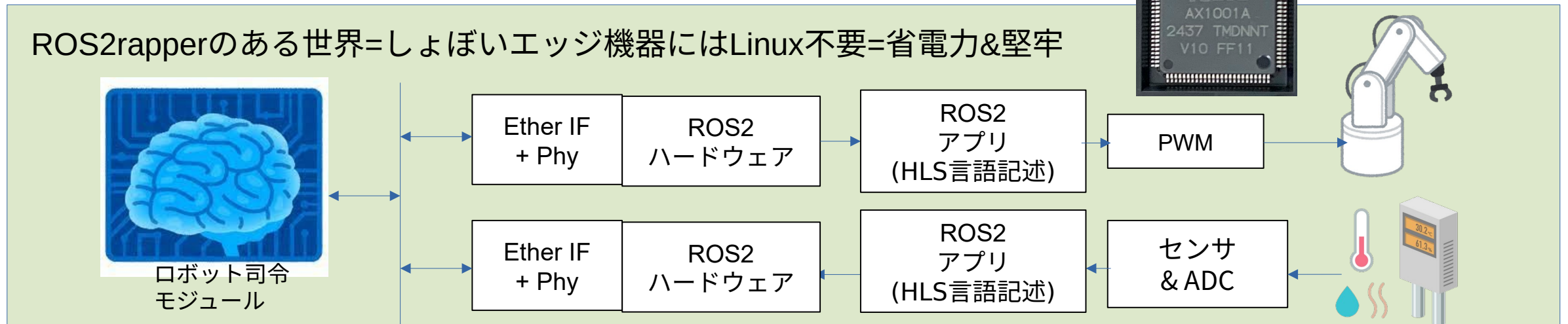


ROS2プロトコルを完全ハードウェア化した…

AXE製、“ROS2rapper”

- CPU無しで、ロボットの部品モジュールができる
 - センサとROS2プロトコルHWだけで、センサ・モジュール
 - PWMとROS2プロトコルHWだけで、アクチュエータ・モジュール
 - アプリケーションはC言語で書いておけば、すぐハードウェア論理に合成
- ロボット部品が、ゴミのようなLSIでできる ← CPU不要
- 高速、堅牢、低消費電力

“ROS2rapper” IPは、
オープンソース・ハードウェアで配布中。
<https://github.com/AXE-jp/ros2rapper>



オレ達のCPU「松竹V(しょうちくぶい)」の排他制御

“俺SoC”, とことんOS無し

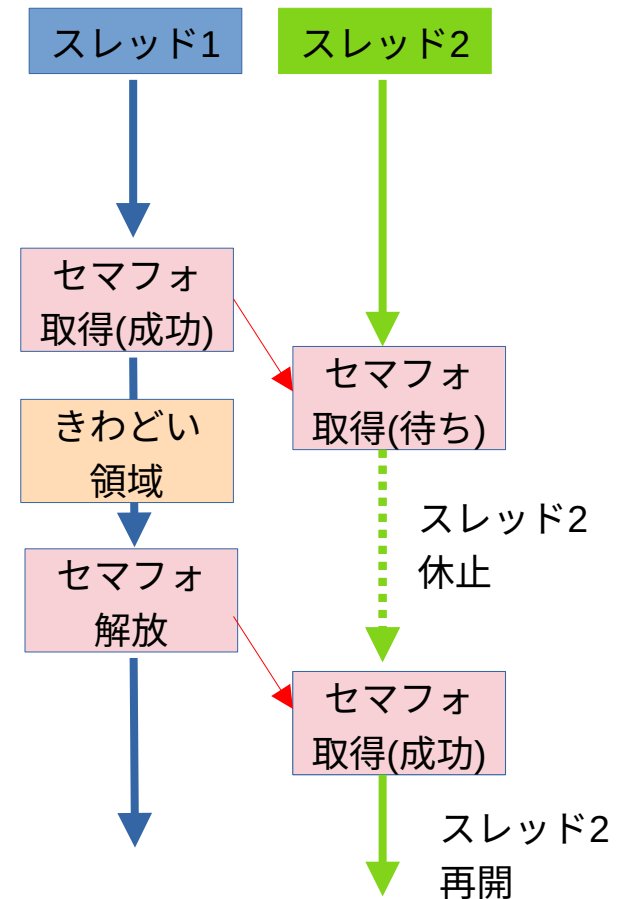
- ・ 省電力、省メモリ、堅牢かつ高速
- ・ ロボットの部品モジュールがローコストで簡単に作れる

マルチタスクだが、
OSプログラム・コード
OSワーキング・エリア
不要!



ハードウェア・マルチスレッド機構

- ・ OSソフトウェア一切なしで、スレッド切り替え
- ・ ハードウェア・セマフォで排他/同期(重要)
- ・ LR/SC(RISC-Vの排他制御プリミティブ)もある
- ・ 外部ピンからの入力で、スレッド起床(ハードウェアのみで)
 - ・ 割り込みなし(割り込み相当の処理は、専用スレッドで)
- ・ ラウンドロビン・スケジューリング

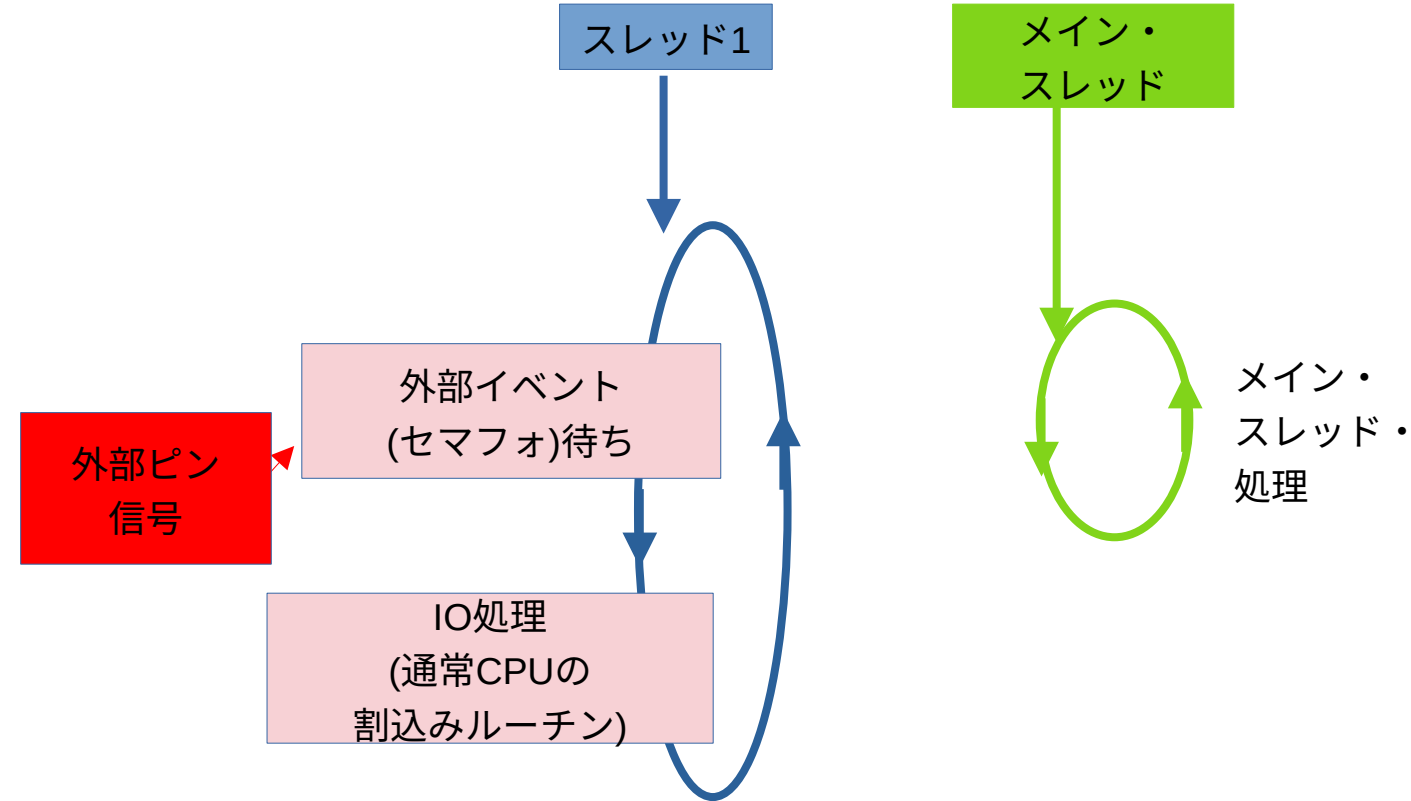


オレ達のCPU「松竹V(しょうちくぶい)」の外部イベント (割り込み相当)

ハードウェア・マルチスレッド機構

- OSソフトウェア一切なしで、スレッド切り替え
- 外部ピンからの入力で、スレッド起床(ハードウェアのみで)
 - 割り込みなし
- 割り込み相当の処理は、専用スレッドで。
- 割り込み起動より、高速
 - レジスタ退避などしないから

マルチタスクだが、
OSプログラム・コード
OSワーキング・エリア
不要!



スレッド操作、セマフォ操作命令

6.5.7. カスタム命令

ニーモニック	命令フォーマット	opcode	funct3	funct7	imm
task_start	R-type	01010 (custom-1)	000	0000000	-
task_terminate	R-type	01010 (custom-1)	000	0000001	-
task_getid	R-type	01010 (custom-1)	000	0000010	-
task_read_xreg	R-type	01010 (custom-1)	000	0000011	-
task_write_xreg	R-type	01010 (custom-1)	000	0000100	-
task_sleep	R-type	01010 (custom-1)	000	0000101	-
task_yield	R-type	01010 (custom-1)	000	0000110	-
sem_init	R-type	01010 (custom-1)	000	0000111	-
sem_signal	R-type	01010 (custom-1)	000	0001000	-
sem_wait	R-type	01010 (custom-1)	000	0001001	-
task_setts	R-type	01010 (custom-1)	000	0001010	-

オレ達のCPU「松竹V(しょうちくぶい)」

機械学習AI 加速用 ベクトル計算ユニット

エッジデバイスでも
AI処理を!

- 8bit float, 4SIMD, ベクトル・パイプライン
- 動的クロック切り替え機構で500KHz~320MHzで、動作
 - ※最高 480MHz (動的クロック切り替え非対応)
- Vector ExtensionのImplementation-defined Constant Parameters(実装固有パラメータ)
ELEN:32 , VLEN: 1024



- ベクトルレジスタ
1024ビットのベクトルレジスタ×32個
ベクトルレジスタはすべてのタスクで共有(実体は1つ)
- 機械学習AIの推論に最適な8bit浮動小数点演算をベクトル処理

- エッジ機器内での、AI処理



- データ通信量の削減
- 分散処理による、全体の負荷の分散

32ビットのスカラ浮動小数点数レジスタ
(Fレジスタ): 32個も備える

8bit浮動小数点 ベクトル演算命令

- AX1001の8bit浮動小数点数のフォーマット
 - 仮数部:3bit, 符号:1bit 指数部:4bit, 指数バイアス=7

6.7.8. サポートするCSR

Vector Extensionで規定される、以下のCSRをサポートしている。

CSRアドレス	アクセス	名前	概要
0xC20	R	vl	Vector length
0xC21	R	vtype	Vector data type register
0xC22	R	vlenb	VLEN/8 (vector register length in bytes)

6.7.6. サポートする命令

- V拡張
 - vsetvli
 - vsetivli
 - vsetvl
 - vle.v
 - vse.v
 - vlse.v
 - vsse.v
 - vlm.v
 - vsm.v
 - vadd.{vv, vx, vi}
 - vsub.{vv, vx}
 - vand.{vv, vx, vi}
 - vor.{vv, vx, vi}
 - vxor.{vv, vx, vi}
 - vsll.{vv, vx, vi}
 - vsrl.{vv, vx, vi}
 - vsra.{vv, vx, vi}
 - vfadd.{vv, vf}
 - vfsb.{vv, vf}
 - vfmul.{vv, vf}
- F拡張
 - flw

オレ達のCPU「松竹V(しょうちくぶい)」

- 省電力、省メモリ、堅牢かつ高速
- ロボットの部品モジュールがローコストで簡単に作れる

論理推論 AI加速 機構をRISC-Vコアに追加

- 特許取得済み
- 第7506718号, 2024年6月18日 (東京エレクトロンと共同特許)
- GnuPrologのコンパイルド・バイナリを加速
- 詳細は、付録参照のこと



エッジデバイスでも
大脳的処理を!

その他の カスタム命令

6.4.2. カスタム命令

ニーモニック	命令フォーマット	opcode	funct3	funct7	imm
mov_reg_regin_direct	I-type	00010 (custom-0)	000	-	Don't care
mov_regindirect_reg	I-type	00010 (custom-0)	001	-	Don't care
branch_reg_in_direct	I-type	00010 (custom-0)	010	-	Don't care
jump_reg_indirect	R-type	00010 (custom-0)	011	0000000	Don't care

6.4.2.3. branch_reg_indirect

プログラムカウンタ相対で分岐を行う。プログラムカウンタに加算する値をソースオペランドをレジスタ間接で指定する。後続命令のアドレス（**pc+4**）をレジスタrdに格納する。

本命令は、次の操作を行う。

```
x[rd] <- pc + 4
pc <- pc + x[x[rs1]]
```

6.4.2.4. jump_reg_indirect

分岐を行う。分岐先のベースアドレスをレジスタ間接で指定する。後続命令のアドレス（**pc+4**）をレジスタrdに格納する。

本命令は、次の操作を行う。

```
rd <- pc + 4
pc <- x[x[rs1]] + x[rs2]
```

Laxer AX1001 消費電力概要

	テストプログラム名	CPUクロック周波数・消費電力(mW)					
		320MHz			1.79MHz		
-	-	実測(1.0v)mA	実測(3.3v)mA	1.0V+3.3v計 (mW)	実測(1.0v)mA	実測(3.3v)mA	1.0V+3.3v計 (mW)
	メモリテスト	177.0	5.0	193.5	10.0	0.0	10.0
	有意義なことはしていない	113.0	11.0	149.3	30	10	63.0

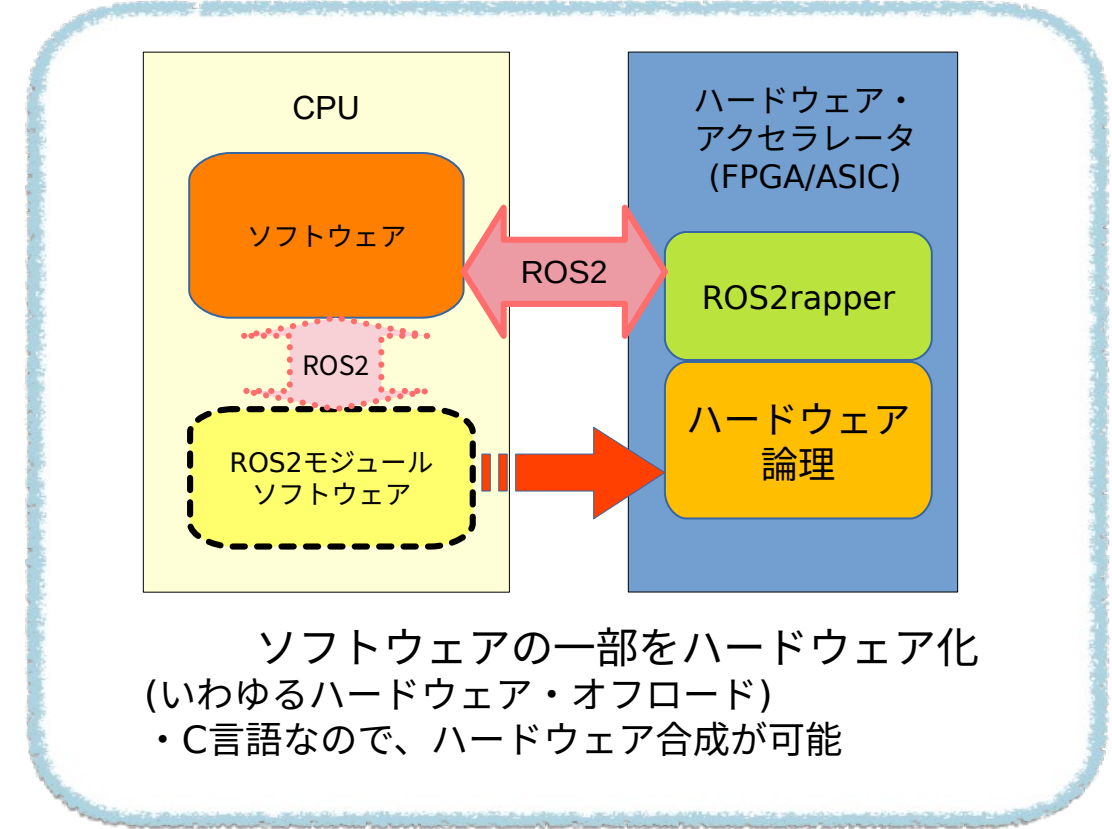
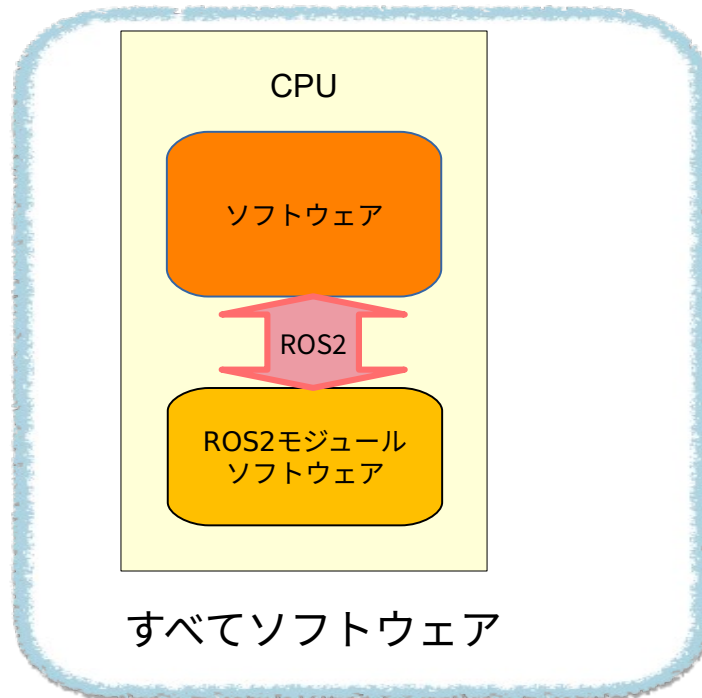
参考	※実際には、DC-DCコンバータなどで損失が出るので、仮の理想値	電池容量 (mW換算)	320MHz(時間)	1.79MHz(時間)
	単1形アルカリ電池容量 約10,000 mAh/1本, 3本4.5v使用	135,000	697.67	13,500.00
	単3形アルカリ電池 容量約1,000~2,900mAh/1本, 3本4.5v使用	39,150	202.33	3,915.00
	コイン電池CR2450 (3V) 620mAh, 2個(6V)使用 (※最大電流30mA/1個)	7,440	(38.45 ※電流が足りない)	744.00

高位合成で
時代が変わった

高位合成によりHW⇔SWの行き来が自在

そこで、ROS2便利

- HLS(C言語風高位記述言語)で書くと、どんな形でも、どこでも実行できる
- ROS2インターフェースで柔軟



ROSインターフェース=インターフェース部分の書き換え不要

・デバイス・ドライバなどの開発が不要

ROS2プロトコルを完全ハードウェア化

AXEが、ROS2プロトコルを完全ハードウェア化”ROS2rapper”

オープンソース・ハードウェアで配布中.

<https://github.com/AXE-jp/ros2rapper>

- ソフトウェア技術者がハードウェア論理を書き、LSI化。現実には!
- OSSとして、配布を準備中
 - LGPLとAXEプロプライエタリ、ダブル・ライセンスを検討中
- CPU無しで、ロボットの部品モジュールができる
 - センサとROS2プロトコルHWだけで、センサ・モジュール
 - PWMとROS2プロトコルHWだけで、アクチュエータ・モジュール
- アプリケーションはC言語で書いておけば、
すぐハードウェア論理に合成

- ロボット部品が、ゴミのようなLSIでできる ← CPU不要
- CPU脳の敗北
- ハードウェアなので、堅牢かつ高速。そしてコンパクト

※ROS2ハードウェアには、コンフィギュレーション用のPROMがあることが望ましい

Arty A7-35ボード(xc7a35ti-csg324-1Lチップ)において

FPGA使用資源

- LUT: 33666

- FF: 13087

最大周波数: 121.01MHz

送信パケット生成 所要時間:

- IPデータグラム生成複数サイクル版:

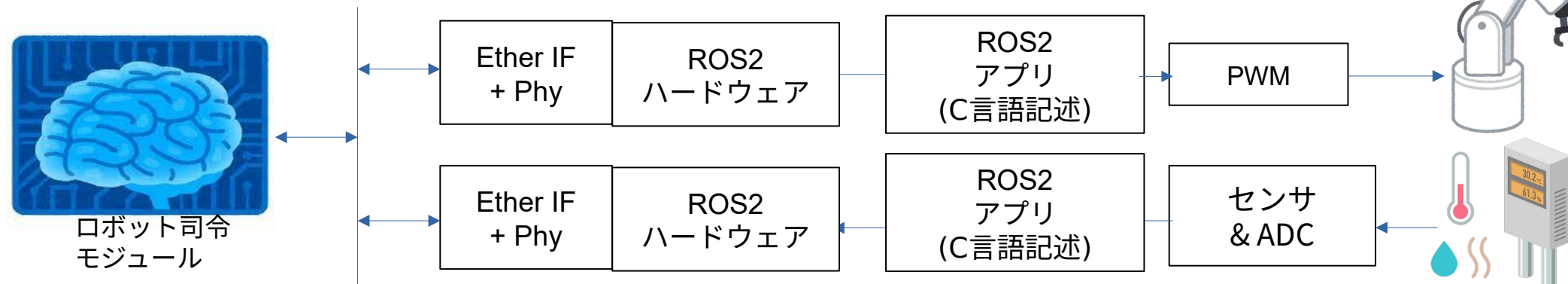
- 14サイクル=140nsec@100MHz

- IPデータグラム生成1サイクル版):

- 8サイクル=160nsec@50MHz

受信 処理時間:

- 46サイクル=460nsec@100MHz

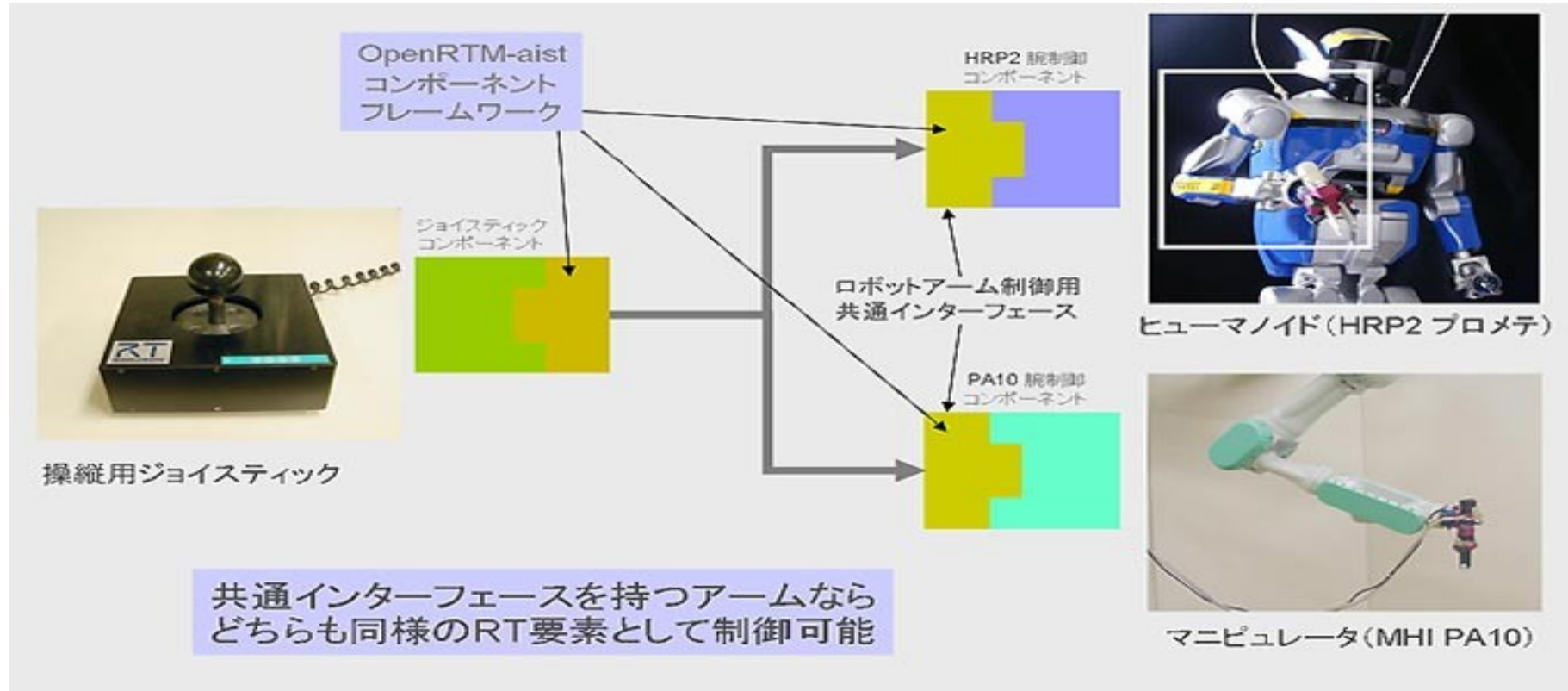


ロボット用ミドルウェア(ROS, RTミドルウェア など)

ROS ロボット・ミドルウェア = ソフトウェア・バス

ロボットのモジュールの流通性が高まる

日本のロボット業界は、ROSをデファクト・スタンダードにしようと活動している

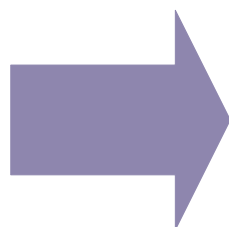


CPU脳を
たたき直す

もう、CPUは(簡単には)速くならないよ

- ついに、微細化 限界
- 微細化 による
 - 高集積
 - 高クロック周波数
 - コア数 増加

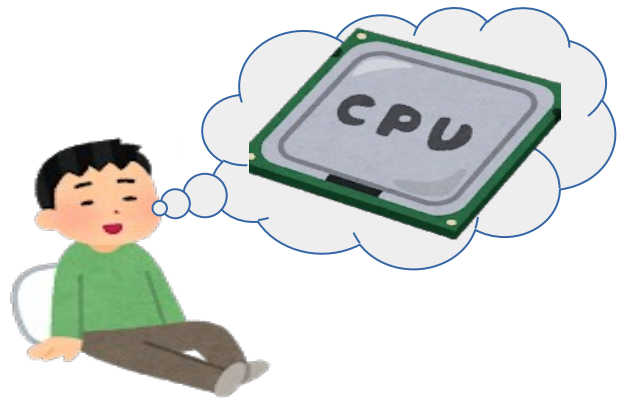
は終了



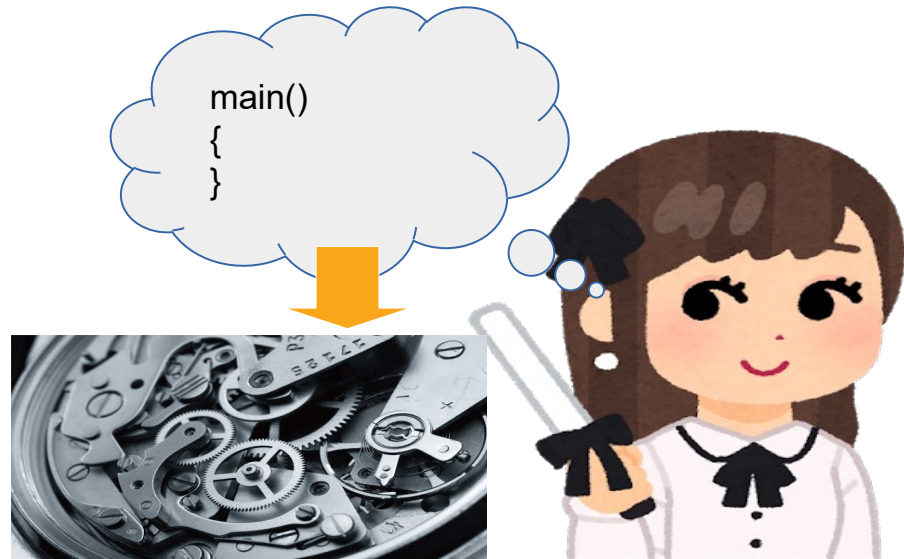
- 専用 ロジック回路による
 - 高速化
 - 省 消費電力
- を行うしか

自由ASIC時代

- 「CPU脳」な人類を、パラダイムシフト
- CPU抜き アーキテクチャを、すすめる
- CPU抜き システムのアーキテクチャ決定をサポートすべし



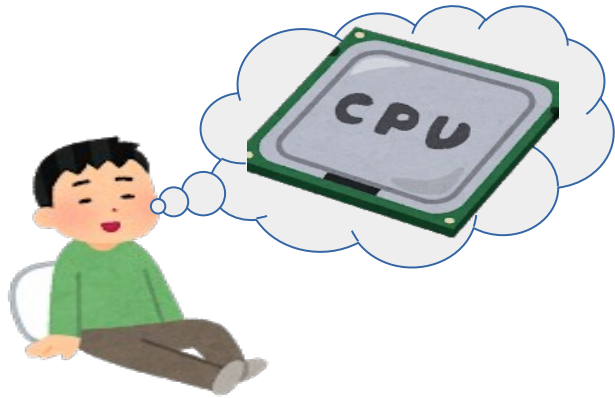
旧人類 CPU脳



ハードウェア
(状態数が たいへんに大きい精密機械)

脱CPUアーキテクチャを推進する

- 新しい細粒度 高並列アーキテクチャの時代
 - 1port RAMをやめさせる、D-FFを使わせる
- 同時並列にバラバラにデータ・アクセスできる
- 教育&コンサルテーションを提供します!



CPU脳プログラマ



ツールと教育

ソフトコアCPU
改造 承ります

RISC-Vのトラストゾーン

TEE実装 Key stone が、どう動くか調査

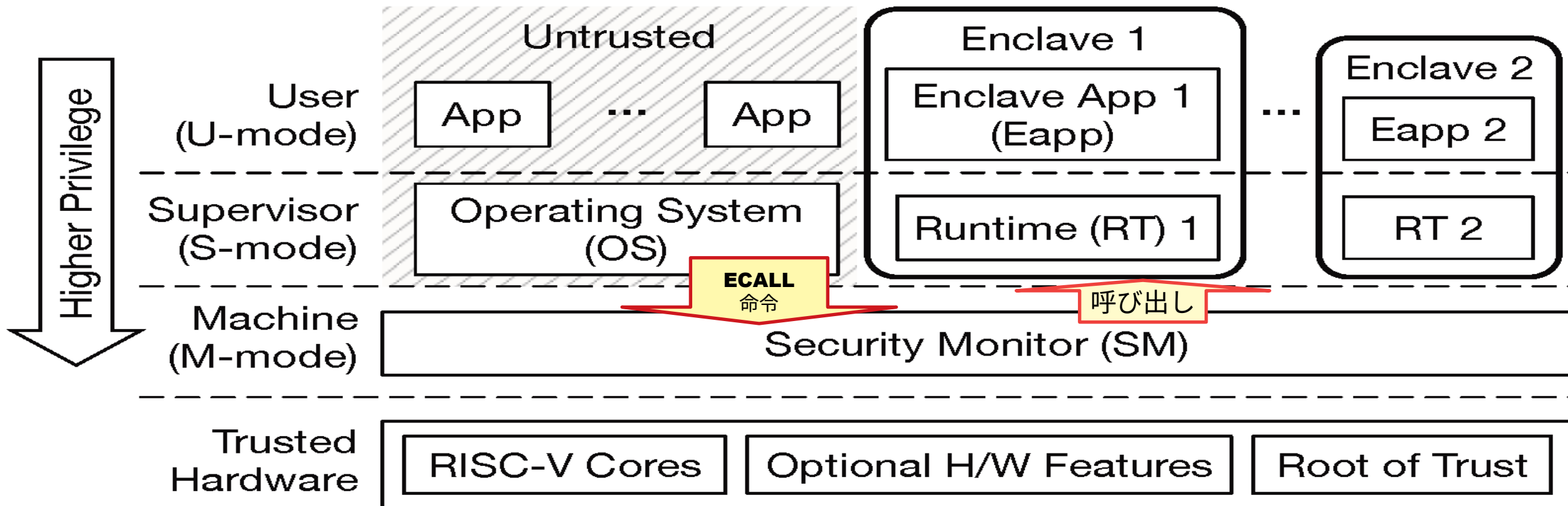
<http://docs.keystone-enclave.org/>

オープンソース・プロジェクト

ARMでいう Trust Zone と同等のものを、RISC-Vで実現

- OSSなので、ソースを読んで調査

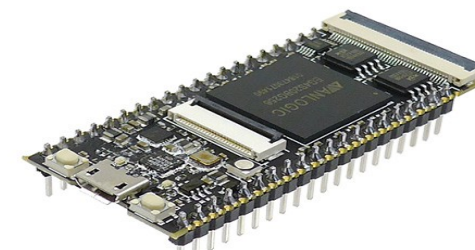
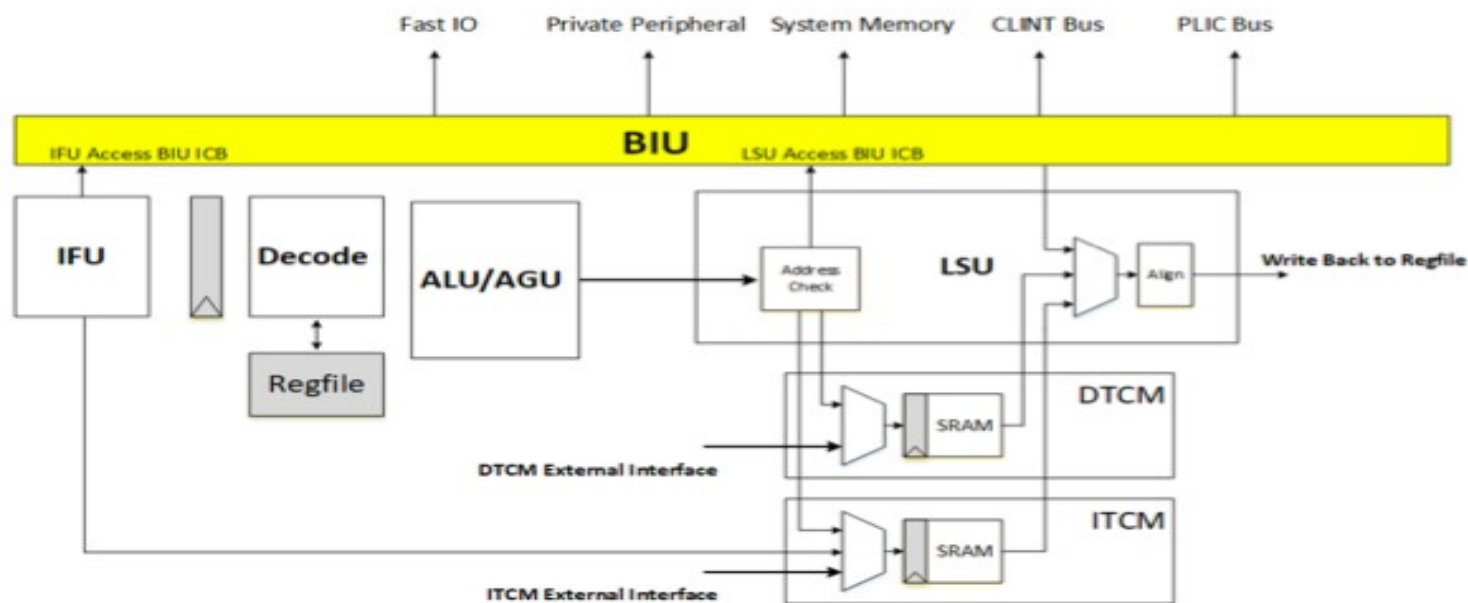
- RISC-V(Key stoneが使用の実装) には、3つのレベルがある
- U-mode (User) / S-mode (Supervisor; OS) / M-mode (Trusted)
- M-modeのみ物理空間で、プロテクトできる(TrustWorld)
- U-mode や S-mode は仮想空間(通常のOSが使用)
- M-mode に入るには、ECALL 命令を使用



Humming bird E203 core改造

- e203には、ビット操作 拡張命令 “B”が入っていない
- オレオレ 命令 を追加

population, parity, clz(count leading zero), ctz(count trailing zero),
Float add, bit reverse, half word exchange, quarter word exchange



E203はTang Primer FPGAボードで動作する
RISC-Vソフトコア

図は下記より引用:

https://content.riscv.org/wp-content/uploads/2018/07/Shanghai-1110_HummingBirdE200

• ITCM and DTCM is integrated inside Core

SPARCもオープンソースなソフトコアあり

- Open Sparc

<https://www.oracle.com/servers/technologies/opensparc-overview.html>

- LEONシリーズ

- 欧州宇宙機関(ESA)が積極開発

- Open Sparc の継続

<https://en.wikipedia.org/wiki/LEON>

- Len3, 3FT,4,5

- LEON3FT : Fault-tolerant processor

<https://www.gaisler.com/index.php/products/ipcores>

<https://www.gaisler.com/index.php/products/processors/leon3>

- Leon3 はGPL

- SPARC v8 が FPGAでも動作

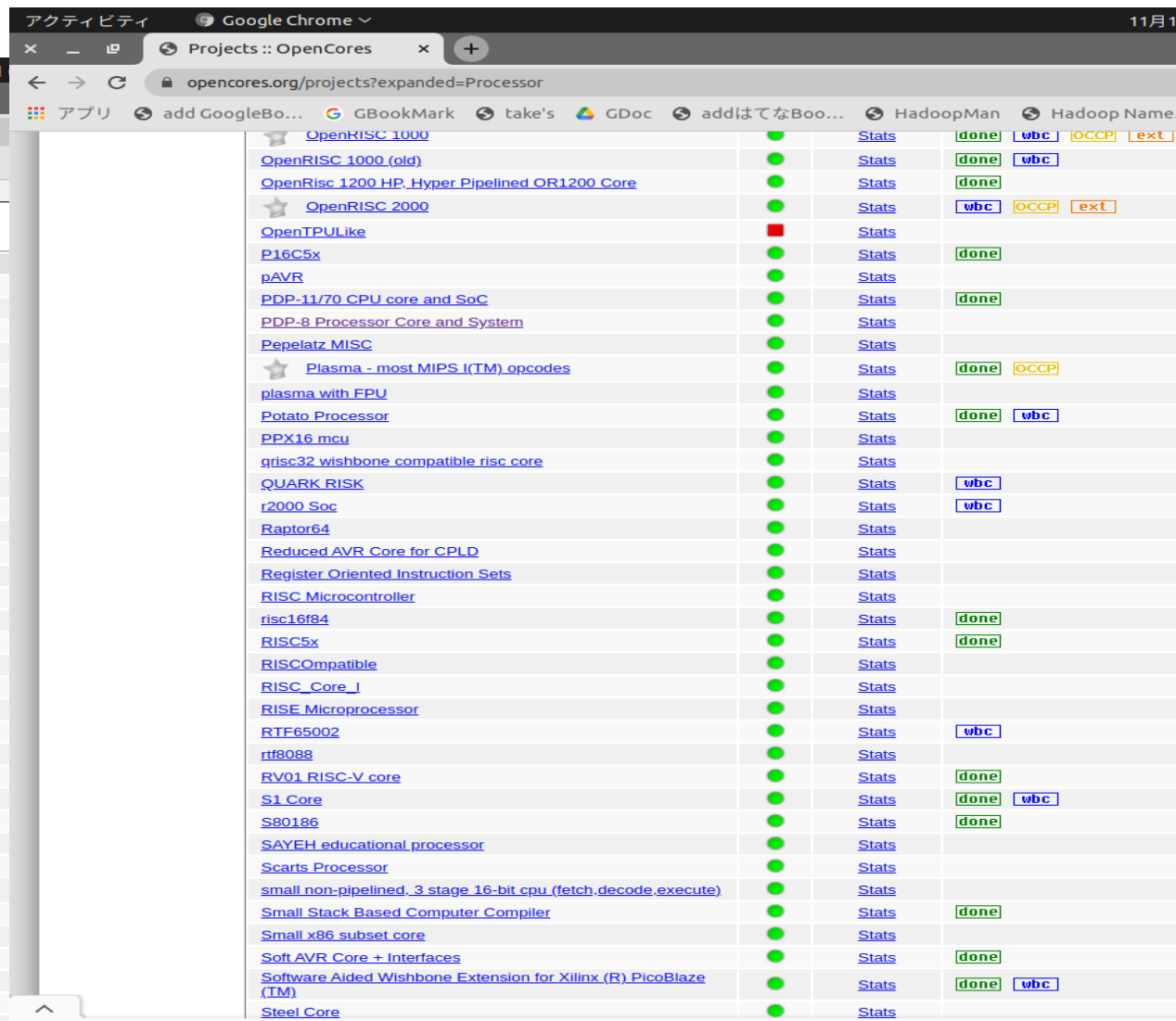
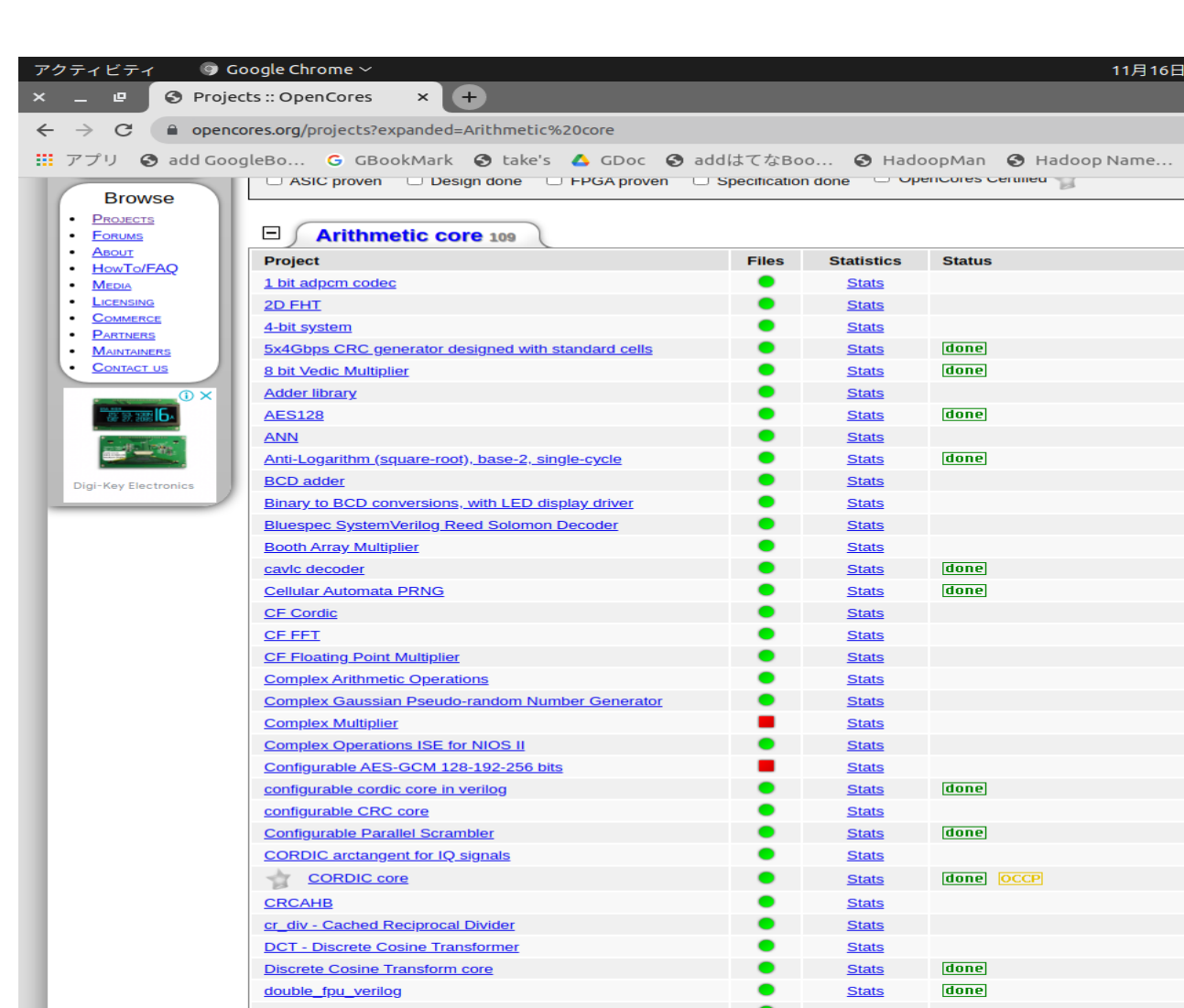
オープンソースなソフトコア

- Opencores

<https://opencores.org/>

オープンソース・プロジェクトのコア

有名コアのRTL記述 多数アリ ※ライセンスに注意



以上

AIもやっています 「ごまめ[®]」
健康のためのAI

お問い合わせ

<https://www.axe.bz/>

mail to: eigyo@axe.bz