RISC-V CPUチップ評価の実践と再現性検証

― FPGA・Google Open MPW、TinyTapeoutによるJASA独自RISC-Vオープンプラットフォーム開発とGithub公開

2025年12月4日

JASA 技術本部ハードウェア委員会 RISC-V WGメンバ

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

1-1. 発表者紹介

小檜山智久

全体概要

JASA RISC-V WG主査 ((株)日立産機システム)

黒川能毅

JASAチップ1の内容

WGメンバ　個人開発者

小林正隆

デモンストレーション

WGメンバ ((株)日立産業制御ソリューションズ)

河崎俊平

JASAチップ2のGithub公開

WGメンバ (SHコンサルティング)

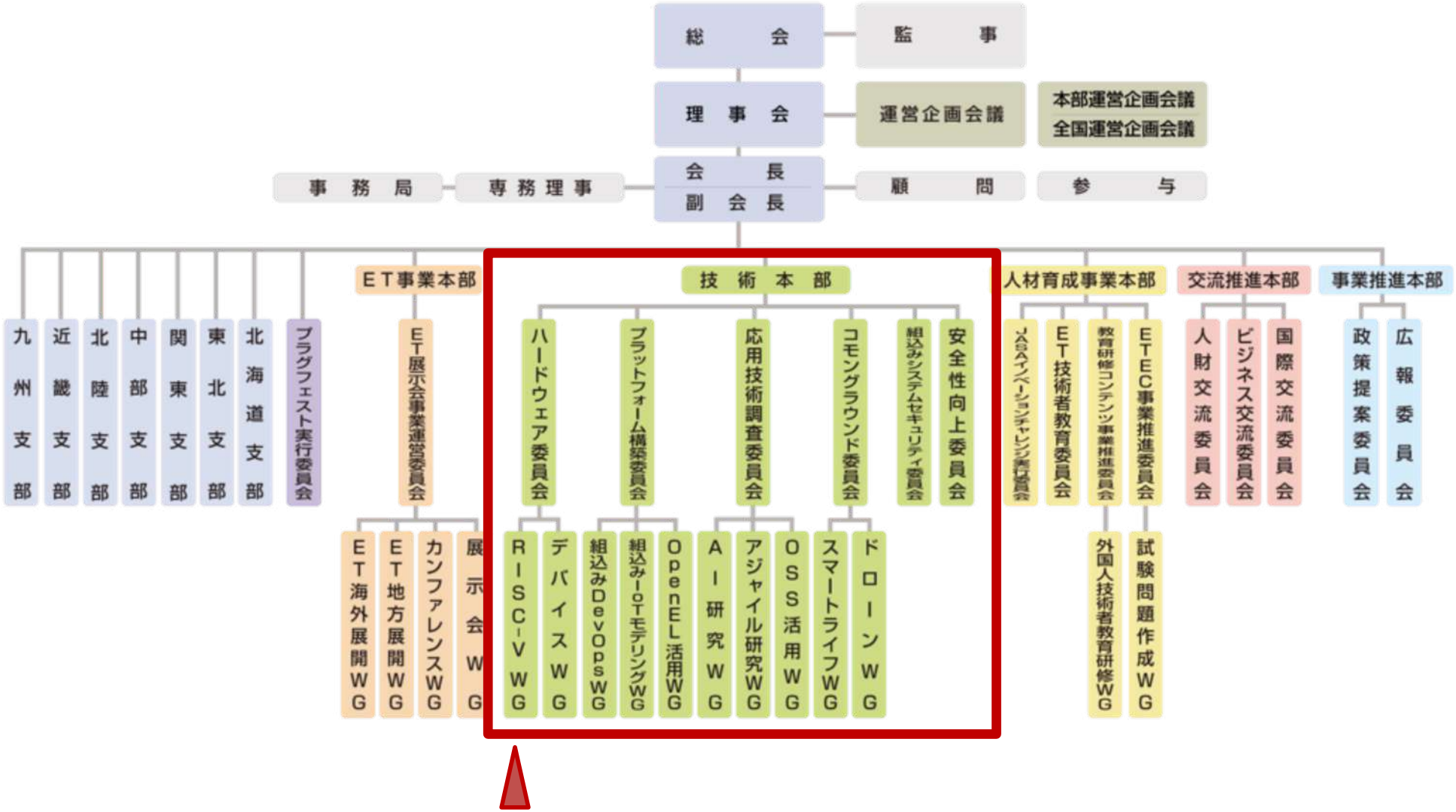
1-2. JASAについて

About JASA

名称	一般社団法人 組込みシステム技術協会 (Japan Embedded Systems Technology Association　略称「JASA」)
会　長	竹内 嘉一
事務局	本部：東京都 入船 1-5-11 弘報ビ 支部：北 関東、中部、北 九州
目　的	組込 ソフトウ システ に の健全な発 の向上 的とする
	組込みシステム る品質、生産性 ュリティ等に関する技 開発 及び標準化の （２）組込みシステム技術に関する人材育成、地域振興及び国際交流の推進 （３）組込みシステムに係る技術・環境・経営及び貿易・投資に関する調査研究並びに情報の提供 （４）組込みシステム技術などに関する内外関係機関との情報交流及び連携の推進 （５）組込みシステム応用技術の普及啓発 （６）本会の会員に対する福利厚生に関する事業の推進 （７）その他本会の目的を達成するために必要な事業
会員数	正会員・支部会員：145社、賛助会員：28社 、 学術会員：3団体、個人会員：9名（2023年4月1日現在）

1-3. RISC-V WGの位置づけ ①

Positioning of RISC-V WG



一般社団法人 組込みシステム技術協会
Japan Embedded Systems Technology Association

1-4. JASA RISC-V WGの活動について

About the activities of JASA RISC-V WG

《WGの活動方針》

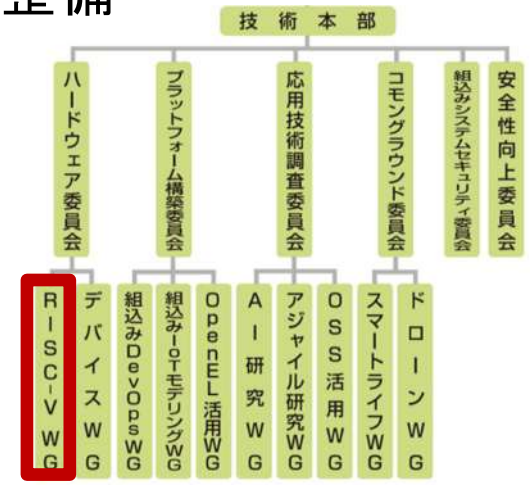
・オープンな仕様で会員が自由に活用できるRISC-Vプラットフォーム
　を会員の協力で整備し、組込み分野でのRISC-V普及に努める
・関連団体とのコラボによりプラットフォームの応用範囲を広げる

《活動内容の項目》

- ◆ 月例WGの開催
- ◆ RISC-V著名人を講師にお迎えし、Webセミナーを開催
- ◆ 組込みに使えるRISC-Vプラットフォームの整備
- ◆ RISC-V関連団体との協創

《WGメンバ》

・委員：15社, 3校、26名



一般社団法人 組込みシステム技術協会
Japan Embedded Systems Technology Association

1-4. JASA RISC-V WGの活動について

About the activities of JASA RISC-V WG

《WGの活動方針》

・オープンな仕様で会員が自由に活用できるRISC-Vプラットフォーム
　を会員の協力で整備し、組込み分野でのRISC-V普及に努める

・関連団体とのコラボによりプラットフォームの応用範囲を広げる

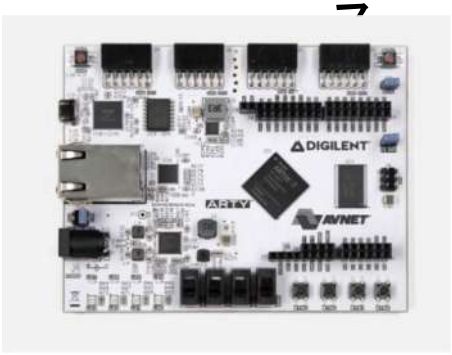
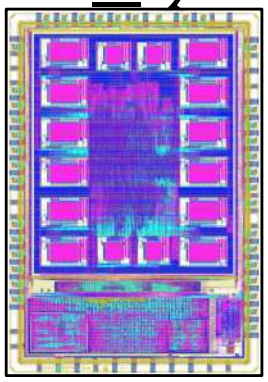
《活動内容の項目》

- ◆ 月例WGの開催
- ◆ RISC-V著名人を講師にお迎えし、Webセミナーを開催
- ◆ 組込みに使えるRISC-Vプラットフォームの整備
- ◆ RISC-V関連団体との協創

1-5. 開発ロードマップと実績

Development schedule for last 6 years

《過去6年間の活動》

2020-22年度	2023年度～
・32ビット版(Arduino)/64ビット版 (LINUX)RISC-VコアFPGA実装 ・産学連携 ・成果のWeb公開	・JASA版RISC-V SoC開発 **- ターゲット:セキュアなIoTエンジン** **- オープンシリコン開発環境の活用** ・ボード作成, RTOS実装 ・IoTクラウド接続
FPGAベース	カスタムLSI(SoC)ベース
 市販FPGA評価ボード Arty A7 35T:32/Arduino 100T:64/LINUX	 eFabless chipignite (…で計画)

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

1-6. RISC-Vプラットフォーム開発の狙い

Aim of developing the RISC-V platform

《初めて取り組む人の課題》

・オープンなのでどこかに情報は公開されている

・それぞれの場所に、それぞれの言語で点在する情報を丹念に集め、

　断片をつなぎ合わせ、試行錯誤しながら理解を深めて作り上げる

　には時間と執念とある程度の知識が必要

　⇒ 初心者にはつらい！

《WGが提供する価値》

◆ WGメンバで実際に検証して実績がある手順を確立

◆ 日本語/英語のバイリンガルで参照できるWebページの公開

◆ 開発環境の準備から完成まで一気通貫で説明

◆ 困ったらWGのオブザーバになってわからないことを聞ける

◆ 開発環境等の更新にできる限り対応

1-7. 開発ロードマップ

Development roadmap

《長期計画》

JASAチップ１
RoT外付
アナログ外付
無線通信外付

デュアルRISC-V
IoT管理チップ

2023
eFabless 130nm

JASAチップ2
RoT内蔵
アナログ内蔵
無線通信外付

デュアルRISC-V
IoT管理チップ
RoT含

2026
AiSoI
フラッシュプロセス

JASAチップ3
RoT内蔵
アナログ内蔵
無線通信内蔵

IoT管理チップ
RoT含
無線通信含

2029
AiSoI
フラッシュプロセス

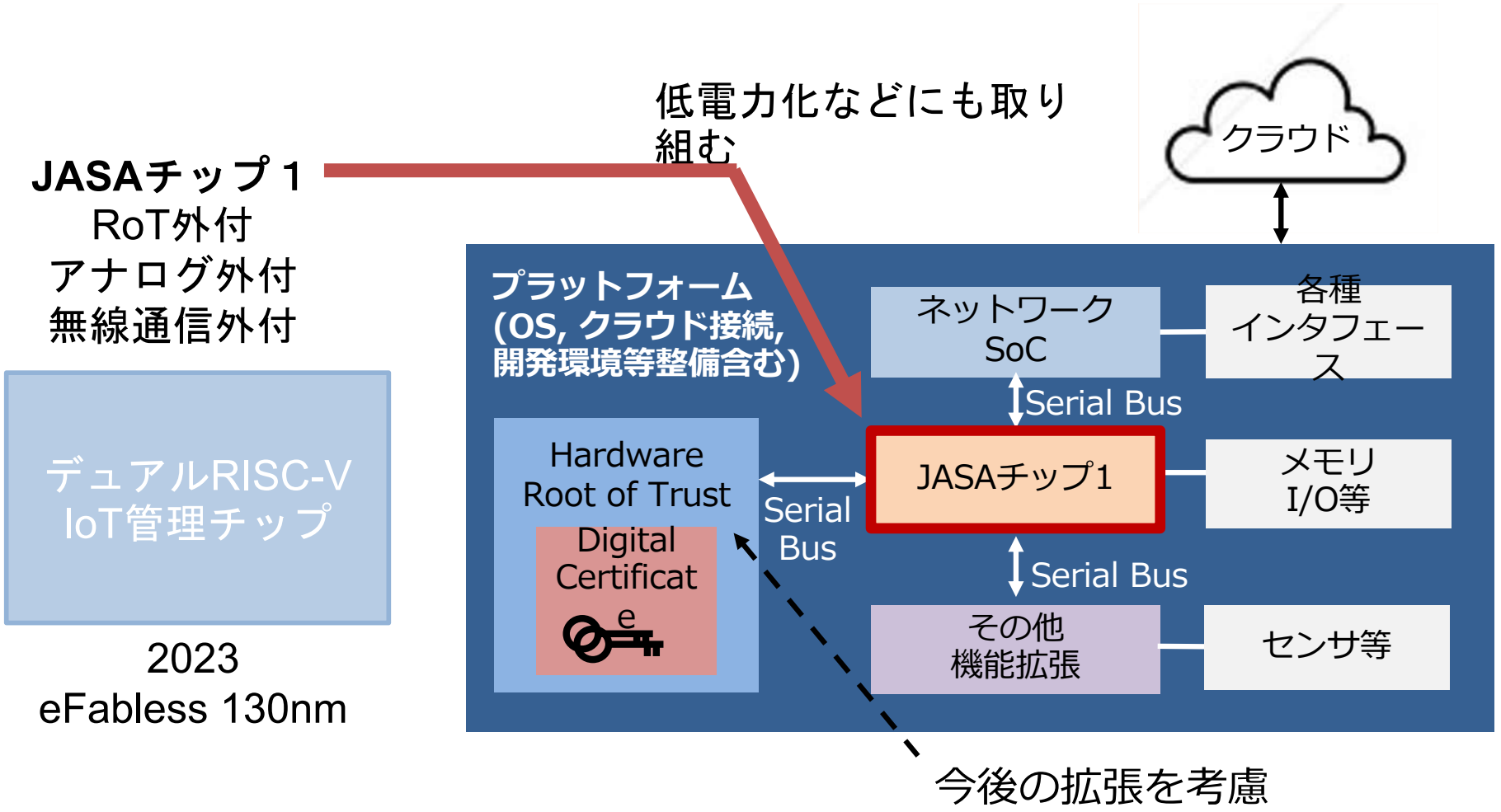
RoT: Root of Trust

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

1-8. 開発のスコープ

Development Scope

《IoTシステム化までがターゲット》



RoT: Root of Trust

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

1-9. 開発ロードマップ

Development roadmap

《長期計画》

JASAチップ1
RoT外付
アナログ外付
無線通信外付

デュアルRISC-V
IoT管理チップ

2023
eFabless 130nm

JASAチップ2
RoT内蔵
アナログ内蔵
無線通信外付

デュアルRISC-V
IoT管理チップ
RoT含

2026
AiSoI
フラッシュプロセス

JASAチップ3
RoT内蔵
アナログ内蔵
無線通信内蔵

IoT管理チップ
RoT含
無線通信含

2029
AiSoI
フラッシュプロセス

コレ

・eFabless活動停止によりTinyTapeout用に軽量プロセッサを開発し、新JASAチップ1とする

・従来検討していたJASAチップ1(Marmot)に今後RoTを実装してJASAチップ2とする

RoT: Root of Trust

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

２．
Tiny Tapeoutを活用した独自RISC-V CPUチップの試作

2-1.ベースとなるCPU：教科書通り

Base CPU: By the book

RISC-V CPU 個人開発の経緯

・2020年春コロナ禍初GW、1週間の休みでも外出できない

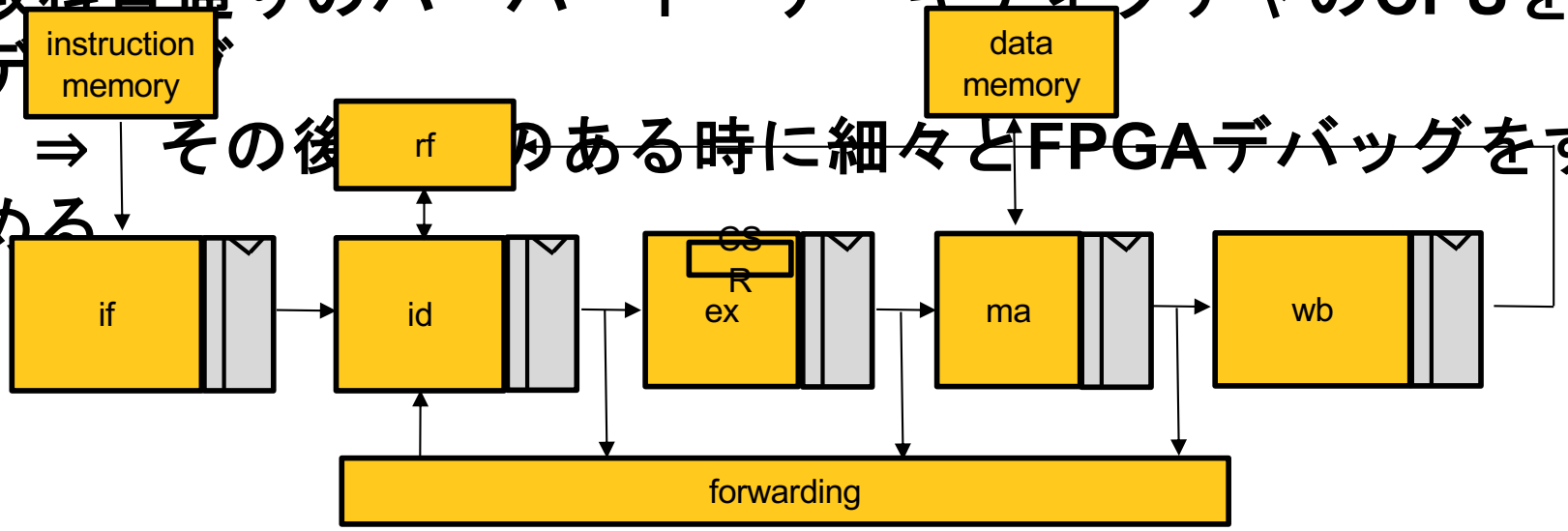
・時間つぶしのためCPUでも作ってみようか？

　　単純な5段パイプラインRISC-Vを Verilogで書いてみよう・・・

　　⇒　沼の始まり

・教科書通りのハーバード・アーキテクチャのCPUをコーディング

　　⇒　その後、余裕のある時に細々とFPGAデバッグをすすめる



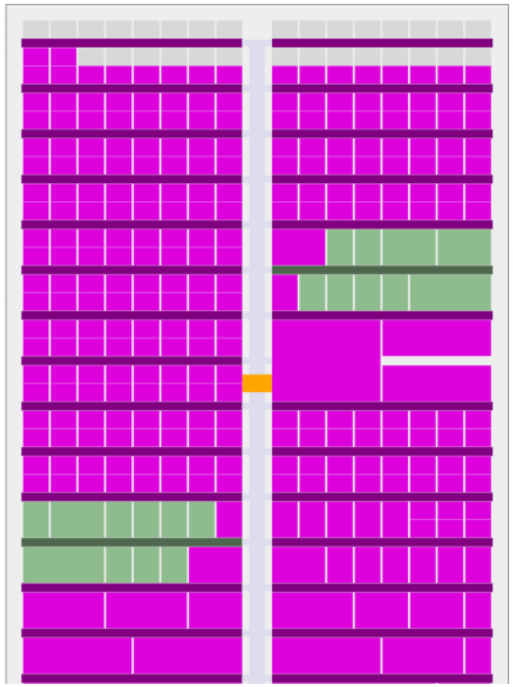
2-2. Tiny Tapeoutについて

About Tiny Tapeout

Tiny Tapeout：　回路デザインを実際のチップ上に製造することを

簡単かつ安価に行うことができる教育プロジェクト

チップ上のデザイン可能領域を167 x 108 μmの微小なタイル領域
に分割してタイル毎に契約することで、多
チップに
相乗りさせ、またDAツールとしてオープ
ェイン(Open Road)を利用することで、安
フローを
実現。



Fabとしては　Skywater社 sky130など
オープンソースとなっているPDKを使用

2-3. Tiny Tapeout試行

Tiny Tapeout Trial

2025年春、JASA RISC-V WGにて：
　前述のTiny Tapeoutのサービスを知る
　果たしてTiny Tapeoutで自作CPUは作れるのか？

手元の5stage RISC-VでTiny Tapeoutを試行
面積削減のため命令,データメモリを64ワードまで減らして試す
結果：最大のタイルサイズ8x2に対して4.6倍のサイズとなり
　　遠く及ばず　（1タイルサイズ = 167 x 108 µm)
原因:　①メモリ5184ビット全てがFlipFlop(FF)となる
　　②また約9000超ビットがパイプラインFFなど
で存在

⇒ 少なくとも今の論理で入りきるものは作れず、
非パイプライン型の小さなCPUを設計することを決心

8x12サイズに無理やり入れると右のようなレイアウトになる　⇒

2-4. 省回路のCPU：State Machine方式

Small Area CPU : State Machine Type

以下の方針でCPUを組みなおす

・ステートマシンで実行ステージを決定しそのステージを処理

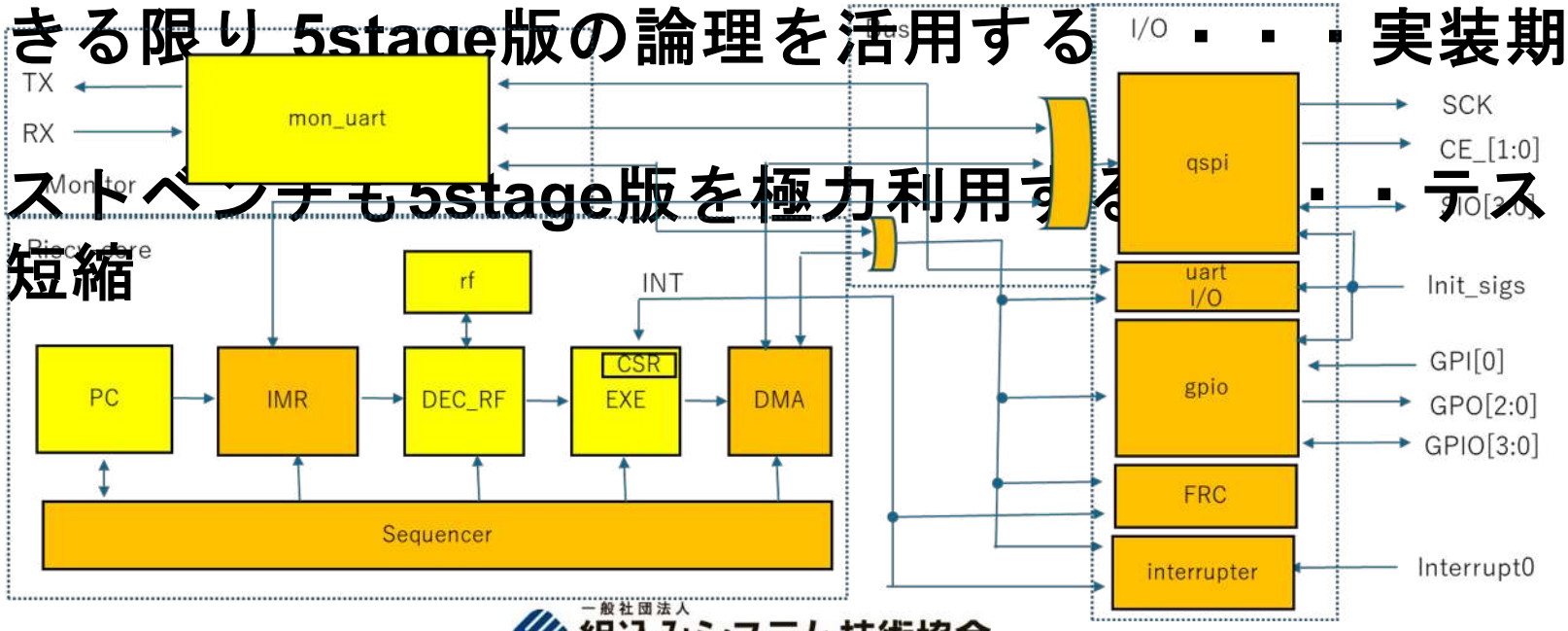
⇒　ステージ間のFFを削除

・命令,データメモリを外部のQSPIメモリに代用させ、メモリはRFのみ

⇒　メモリFFを大幅削減

・できる限り 5stage版の論理を活用する・・・実装期間の短縮

・テストベンチも5stage版を極力利用する・・・テスト期間の短縮



一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

2-5. 論理セル数比較

Comparison of Number of Logic Cells

設計と実装：個人開発のため、2025/7月の土日祭日と早朝を使い

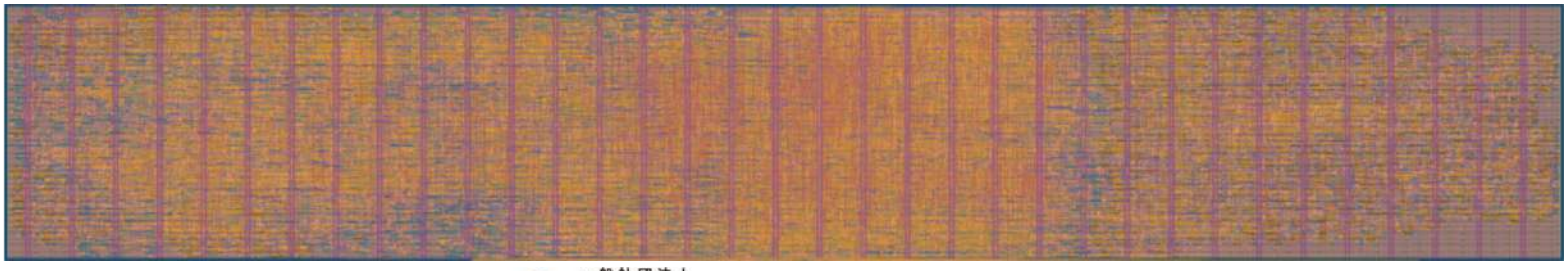
約2週間でざっくりFPGAテストまで進める

⇒ Tiny Tapeoutのフローに乗せてみる

8x2のサイズ花1に成功！5Stage版の1/6の面積サイズ

表1 論理セル数比較

#	セル種別	5stage	State machine	倍率
1	メモリ部分FF	5184	1088	1 / 4.76
2	それ以外のFF	9334	1616	1 / 5.78
3	ロジックゲート	63541	9543	1 / 6.65



2-6. 開発CPUスペック表

Specification Table of Developed CPU

項目	内容
命令セット	RV32I （RISC-V命令セットの最小構成）
プロセッサ構成	State Machineタイプのシングルプロセッサ
動作モード	Mモードのみ
搭載メモリ	なし　外付けメモリを使用
サイズ	1336 μm x 216 μm
ゲート規模	約40 KGates（2nand換算）
目標動作周波数	25～50MHz（ただしI/Oの影響で低下する可能性あり）
信号ピン	8 input 8 output 8 bi-direction
外付けメモリ	QSPI PSRAM/Flash ROM
タイマー	64bitフリーラン
I/O	4bit GPIO, SPI, UART, 割込み1ピン
モニター機能	UARTにハード組み込み

2-7. モニター機能

Monitoring Function of CPU

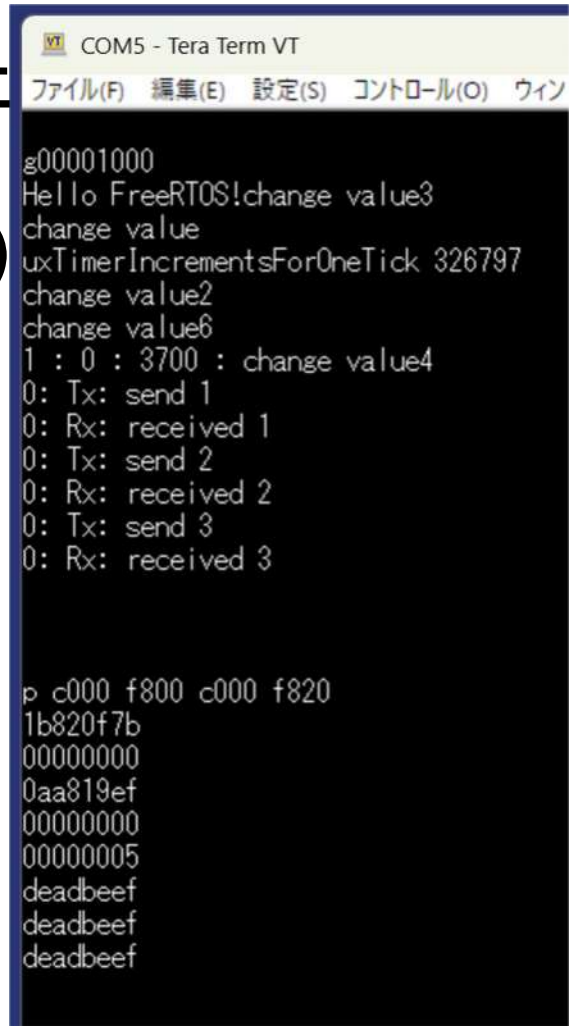
初代のCPU時代に開発

・フルVerilog簡易モニター機能をUARTに

・実装機能 (最低限デバッグに必要な機能)

- メモリのリード・ライト
- CSR/RF/IOレジスタのリードライト
- CPU実行、停止
- 現在アドレスの表示
- ブレイクアドレスの設定・解除
- CPU実行時はCPUのUARTとして機能

・簡易機能であるが、FPGA使用の
実機テストで威力を発揮



2-8. Tiny Tapeout向け検証

CPU Testing for Tiny Tapeout

以下の検証環境を整備

・FPGA上の開発環境一式を5 stage版流用

・自作のRISC-V RV32Iのアセンブラ

・risc-v toolchain　Cプログラムでベアメタル動作する環境を構築

・I/Oやタイマー、割り込みについてはCでテスト作成

⇒　FPGA向けの動作を楽しむ個人開発環境としてはOK

これだけでは修正不可能なASICの検証としては不足

しかし、プロ同様の多量の検証ベンチセットの準備は到底不可能

⇒　目標をFPGA上でOSSのFree RTOSのデモを動かす事に設定

理由：動作要素として、ecallやタイマー割込みなどが入っており、

2-9. Free RTOSとの格闘

Struggle of Free RTOS on Developed CPU

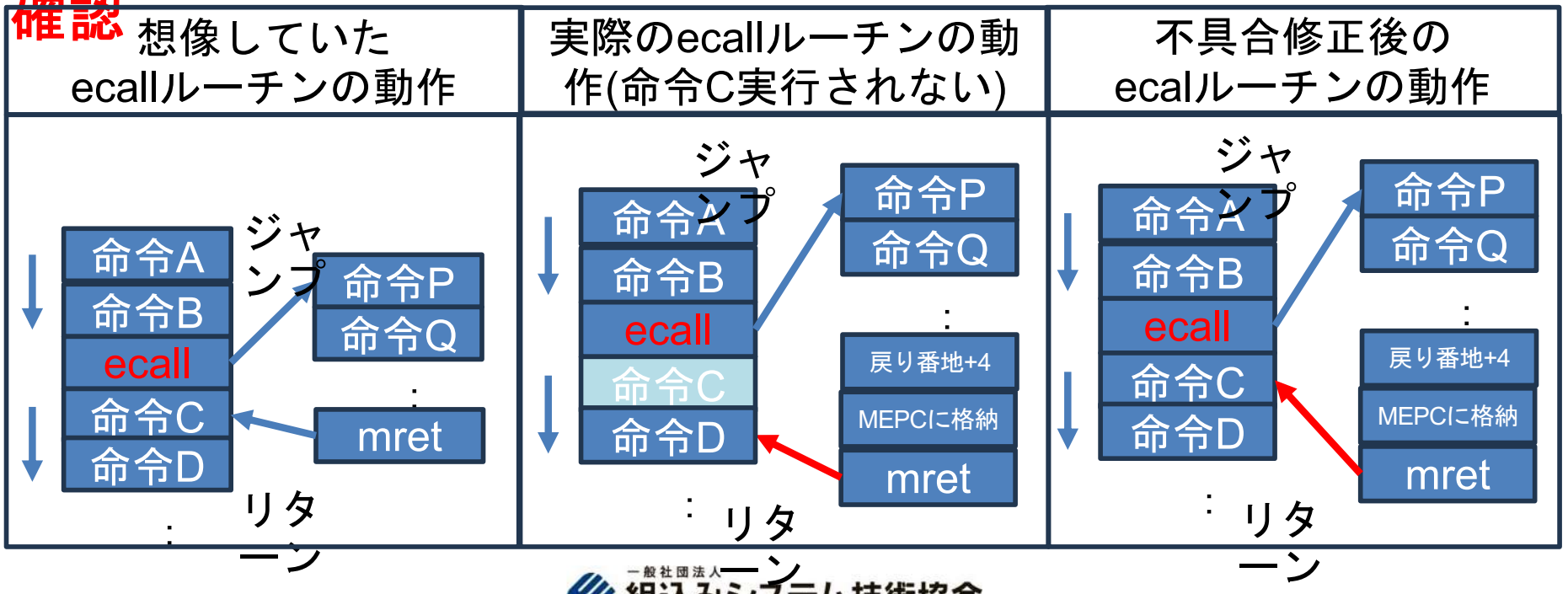
発見した不具合：条件により数秒～数時間で停止

原因：ecallの仕様の理解の間違い(詳細は下図)

仕様をきちっと理解していれば避けられたのですが、

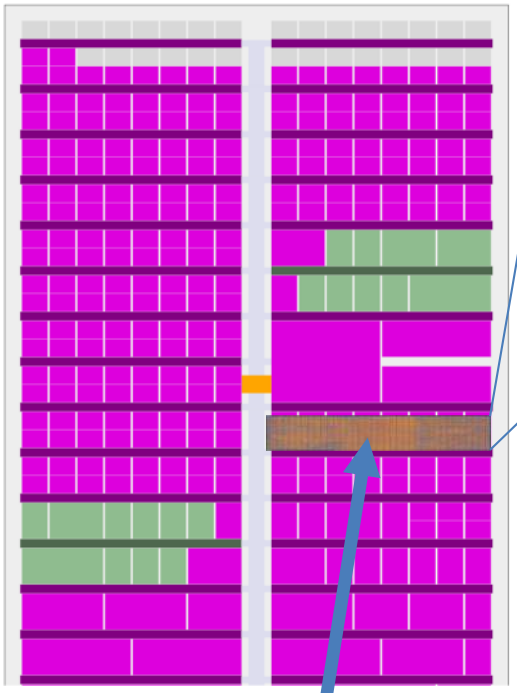
ゆるい個人開発では抜けが出てしまったのでした・・・

⇒　この不具合修正により**Free RTOSデモの1日連続動作を確認**

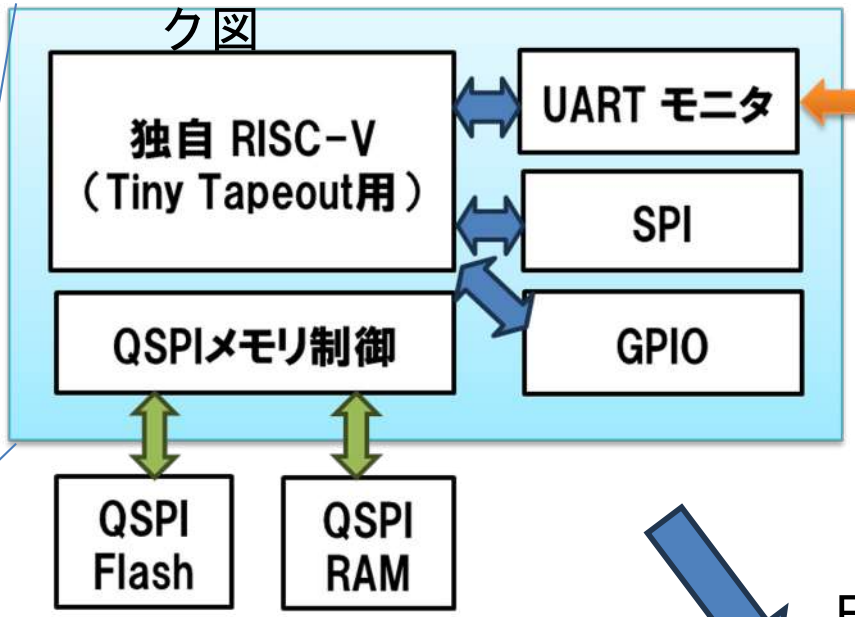


2-10. JASAチップ1の全体構成

Tiny Tapeoutチップイメージ



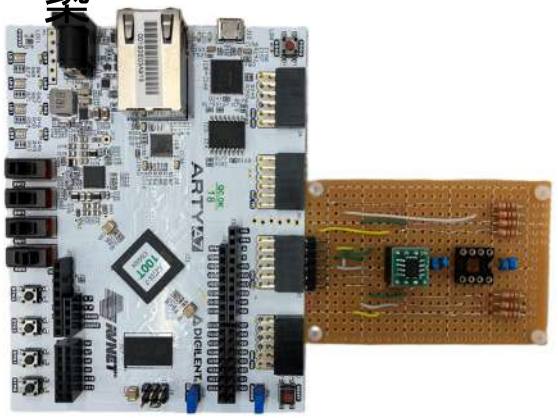
JASAチップ1の構成ブロック図



PCから制御

- プログラムロード
- プログラム実行
- プログラム停止
- I/Oリード・ライト
- メモリリード・ライト
- レジスタダンプ

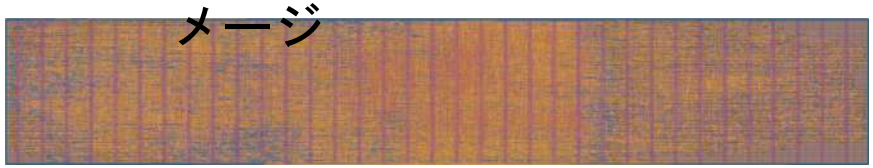
FPGAの検証環境構築



Open Roadによる合成・レイアウト

同一Verilogから生成

JASAチップ1のレイアウトイメージ



2-11. Tiny Tapeout向けCPU開発まとめ

Conclusion of Small Area CPU Development

結論

・Tiny Tapeout使用可能なRISC-V(RV32I)互換CPUを個人で開発

・State Machineタイプを選択し、約1/6に論理サイズを縮小

・他開発の部品とテストプログラムを再利用し開発工数を削減

・Free RTOSをポーティングし論理テストの網羅性を少しでもカバー

今後の展開

・2026/3のTiny Tapeoutを目指して実機検証を継続

・SDカードサポート（超低速でもストレージをサポートできないか？）

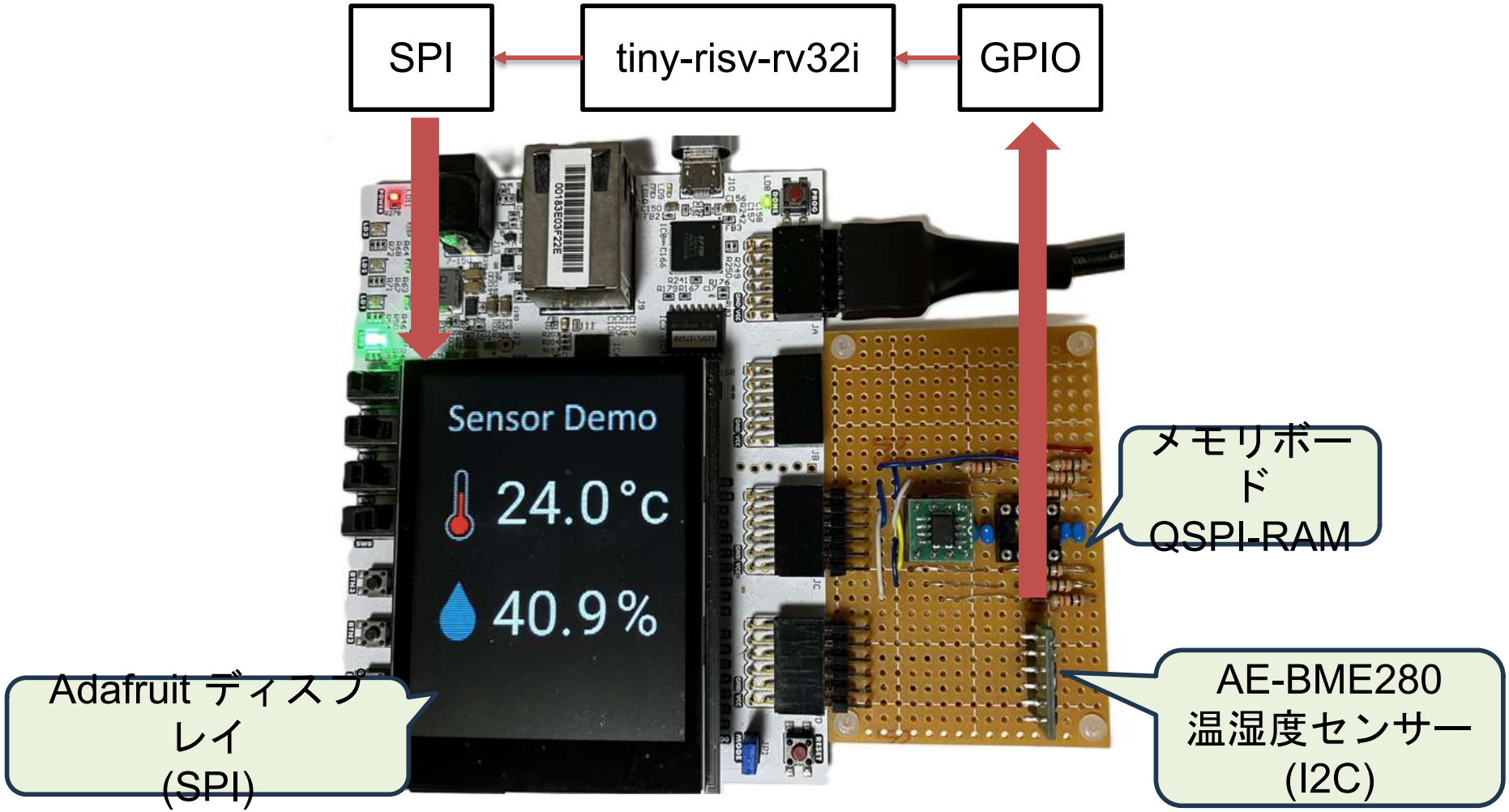
・そのほか未対応のCSRのサポート

3.
デモンストレーション

3. デモ概要

Summary of Demonstration

JASAチップ1の tiny-riscv-rv32i で動作するデモを作成しました。



4.
Marmotのソース公開と今後の展開に向けた課題

・eFabless活動停止によりTinyTapeout用に軽量プロセッサを開発し、新JASAチップ1とする
・従来検討していたJASAチップ1(Marmot)に今後RoTを実装してJASAチップ2とする

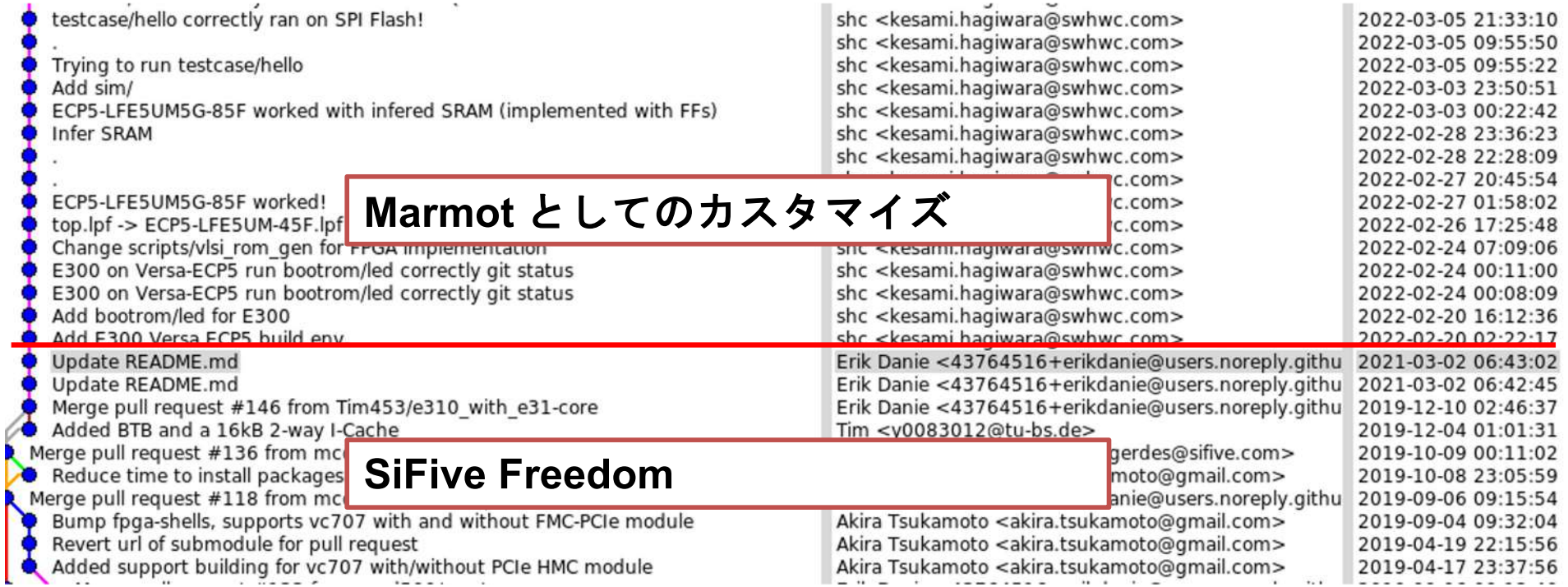
4-1. Marmotのソース公開

Marmot Source Release

以下のサイトで公開

https://github.com/OpenProcessors/marmot_asic

SiFive FreedomをベースにCaravel 対応



testcase/hello correctly ran on SPI Flash!	shc <kesami.hagiwara@swhwc.com>	2022-03-05 21:33:10
.	shc <kesami.hagiwara@swhwc.com>	2022-03-05 09:55:50
Trying to run testcase/hello	shc <kesami.hagiwara@swhwc.com>	2022-03-05 09:55:22
Add sim/	shc <kesami.hagiwara@swhwc.com>	2022-03-03 23:50:51
ECP5-LFE5UM5G-85F worked with infered SRAM (implemented with FFs)	shc <kesami.hagiwara@swhwc.com>	2022-03-03 00:22:42
Infer SRAM	shc <kesami.hagiwara@swhwc.com>	2022-02-28 23:36:23
.	shc <kesami.hagiwara@swhwc.com>	2022-02-28 22:28:09
.		2022-02-27 20:45:54
ECP5-LFE5UM5G-85F worked!		2022-02-27 01:58:02
top.lpf -> ECP5-LFE5UM-45F.lpf		2022-02-26 17:25:48
Change scripts/vlsi_rom_gen for FPGA implementation		2022-02-24 07:09:06
E300 on Versa-ECP5 run bootrom/led correctly git status	shc <kesami.hagiwara@swhwc.com>	2022-02-24 00:11:00
E300 on Versa-ECP5 run bootrom/led correctly git status	shc <kesami.hagiwara@swhwc.com>	2022-02-24 00:08:09
Add bootrom/led for E300	shc <kesami.hagiwara@swhwc.com>	2022-02-20 16:12:36
Add E300 Versa ECP5 build env		2022-02-20 02:22:17
Update README.md	Erik Danie <43764516+erikdanie@users.noreply.githu	2021-03-02 06:43:02
Update README.md	Erik Danie <43764516+erikdanie@users.noreply.githu	2021-03-02 06:42:45
Merge pull request #146 from Tim453/e310_with_e31-core	Erik Danie <43764516+erikdanie@users.noreply.githu	2019-12-10 02:46:37
Added BTB and a 16kB 2-way I-Cache	Tim <y0083012@tu-bs.de>	2019-12-04 01:01:31
Merge pull request #136 from mc	gerdes@sifive.com>	2019-10-09 00:11:02
Reduce time to install packages	moto@gmail.com>	2019-10-08 23:05:59
Merge pull request #118 from mc	anie@users.noreply.githu	2019-09-06 09:15:54
Bump fpga-shells, supports vc707 with and without FMC-PCIe module	Akira Tsukamoto <akira.tsukamoto@gmail.com>	2019-09-04 09:32:04
Revert url of submodule for pull request	Akira Tsukamoto <akira.tsukamoto@gmail.com>	2019-04-19 22:15:56
Added support building for vc707 with/without PCIe HMC module	Akira Tsukamoto <akira.tsukamoto@gmail.com>	2019-04-17 23:37:56

4-1. Marmotの課題

Issues of Marmot

Marmot には Freedom と同じ課題が存在

1. submodule のURLが古い
2. 古いビルド環境が必要

他にも課題があるかもしれない

4-3. 課題① submoduleのURLが古い

Issue of Marmot #1: Expired URL of Submodule

課題

- submodule のURLが古い

path	現状	修正
roms/vgabios	git://git.qemu-project.org/vgabios.git/	https://github.com/qemu/vgabios.git
roms/seabios	git://git.qemu-project.org/seabios.git/	https://github.com/qemu/seabios.git
roms/SLOF	git://git.qemu-project.org/SLOF.git	https://github.com/qemu/SLOF.git
roms/ipxe	git://git.qemu-project.org/ipxe.git	https://github.com/qemu/ipxe.git
roms/openbios	git://git.qemu-project.org/openbios.git	https://github.com/qemu/openbios.git
roms/openhackware	git://git.qemu-project.org/openhackware.git	https://github.com/qemu/openhackware.git
roms/qemu-palcode	git://github.com/rth7680/qemu-palcode.git	https://github.com/qemu/qemu-palcode.git
roms/sgabios	git://git.qemu-project.org/sgabios.git	https://github.com/qemu/sgabios.git
pixman	git://anongit.freedesktop.org/pixman	https://gitlab.freedesktop.org/pixman/pixman.git
dtc	git://git.qemu-project.org/dtc.git	https://github.com/qemu/dtc.git
roms/u-boot	git://git.qemu-project.org/u-boot.git	https://github.com/qemu/u-boot.git

対策案

- submodule を取得するためのスクリプトファイルを準備

4-4. 課題② 古いビルド環境が必要

Issue of Marmot #2:Keeping Old Development Environment

課題

- 古いビルド環境が必要（Ubuntu 18.04）

必要なツール

名称	version	説明
sbt	1.2.1	Chisel のビルドに必要 バージョンの特定：scala 2.12.4に対応したバージョンを検索
Java (openjdk)	11	バージョンの特定：以下のサイトに記載あり https://docs.scala-lang.org/overviews/jdk-compatibility/overview.html Ubuntu 18.04 では、openjdk-11をapt-get で取得可能
verilator	3.922	Verilogのシミュレーション用 バージョンの特定：README.md に記載あり
RISC-V toolchain	10.2.0	バージョンの特定：SiFive社のツールチェーンの最新版 https://github.com/sifive/freedom-tools/releases

対策案

- ビルド環境を構築するためのスクリプトを準備

4-5. 活動成果の公開

Publication of Development Results

◆ 活動成果へのアクセス方法

・RISC-V WGのホームページ最下部のリンクをクリック

RISC-V WGページ

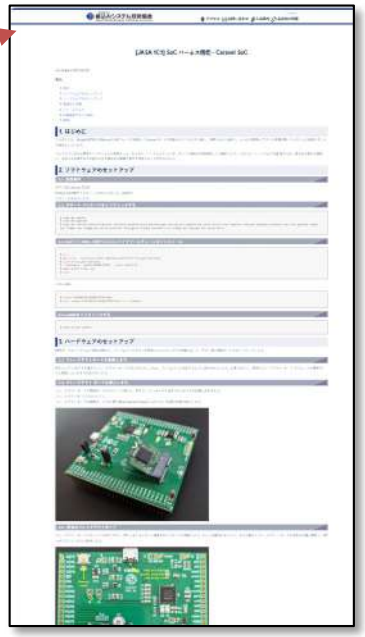
・日本語リンク

・英語リンク

・FPGAリンク

・SoCリンク

・詳細項目リンク



スマホをお持ちの方はこちら ⇒

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

RISC-V CPUチップ評価の実践と再現性検証

2025/12/4 発行

発行者　一般社団法人 組込みシステム技術協会
東京都 中央区 入船 1-5-11 弘報ビル5階
TEL: 03(6372)0211 FAX: 03(6372)0212
URL: https://www.jasa.or.jp/

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association