RISC-Vが混載する新しいFPGA：SLMLET

天野英晴, 矢内洋祐, 奥原颯 （ 慶應義塾大学）
小島拓也 （東京大学） 飯田全広, 尼崎太樹, 久我守弘, 中里 優弥 （ 熊本大）
趙謙 （九州工業大）

開発動機：FPGAの闇落ち

- FPGAが登場したとき、僕らは自由なデバイスの登場に熱狂した
- 30年間で、規模、性能、IP取り込みと共にその自由度は増大
  →できないことはなくなった
- しかし、いつしか2大ベンダはCPUベンダに取り込まれ、、
  →やりたいことは何でもできる。ただし枠の中で引いた路線に乗っているならばな、、、
  - FPGAはアクセラレータなんだ、AI用途だGPU、TPUや他のDSAと戦うんだ、、
  - 高価なチップ、不要な装備、複雑で使いにくい開発環境、Discloseされた見えない部分、、、
- もっと自由なFPGAを作ろうぜ
  - 完全オープン、ポータブルな階層　SLM+RISC V+Hyperbus

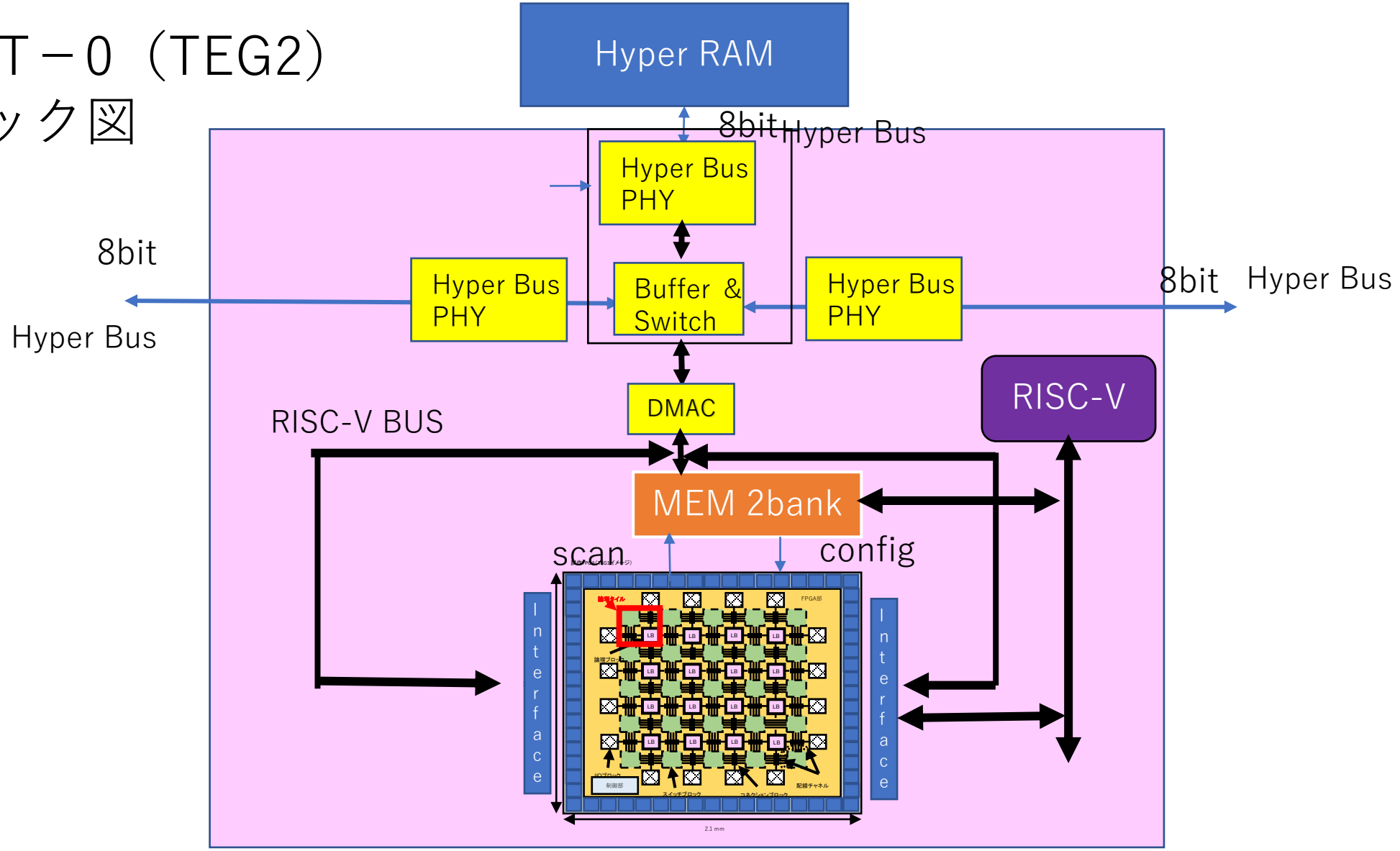
  →　SLMLET (最初はMCMは無理です。すいません。基板上でチップを並べます)

  このプロジェクトは、CREST Society5.0を支える革新的コンピューティング技術「MEC用マルチノード統合システムの開発」の一部です。

SLMLETの設計思想

- IoT用低電力・低コストFPGA
  - 個々のチップのロジック量は（コスト次第だが）小さいが
- HyperBusの接続によりスケールアップ
  - HyperRAMも接続できるので、DRAMよりも低電力でそれなりの容量は得られる。
- マルチチップ間のストリーム処理が簡単に実装可能
  - IoTデバイスの暗号化、匿名化、MQTTブローカのオフロード
- チップ内に複数のConfiguration Dataを圧縮形で持ち、高速に入れ替え可能
- 内部ロジックも含めマイグレーション可能
- RISC-Vとロジックの協調でそれなりの処理も実現可能
- 設計環境も含めすべてがオープンIP、オープンソフトウェア
  - HLS、マルチチップ設計はCyber Work Benchで行い、この部分は未定。
  - 全てが公開されているので、現在のFPGAの持つ「非公開部分」がない。

SLMLET－0（TEG2）のブロック図



eFPGA (embedded-FPGA)とは？

- IPとしてチップに組み込むFPGA
  - ASIC、SoCの一部として柔軟性の向上
    - 変更、バグ対応、I/Oへの適用、製品の長寿命化、全体としての電力削減
- 商用製品が普及、国プロも進行中
  - Flex Logix : ソシオネクストが5G基地局向きASICに搭載、エッジ推論向きコプロセッサ　InferX X1
  - Achronix Speedcore embedded Technology

  →　ハードIP、高性能、先端プロセス向け、高額
  - Preferred Network NEDOプロジェクト　FPGA　IPと可変精度演算コアの組み合わせによるAIチップ
  - ソフトコア
    - Quicklogic eFPGA IP, IP generator Australis
    - SPINNAKAR Systems, Synthesizable Programmable Core

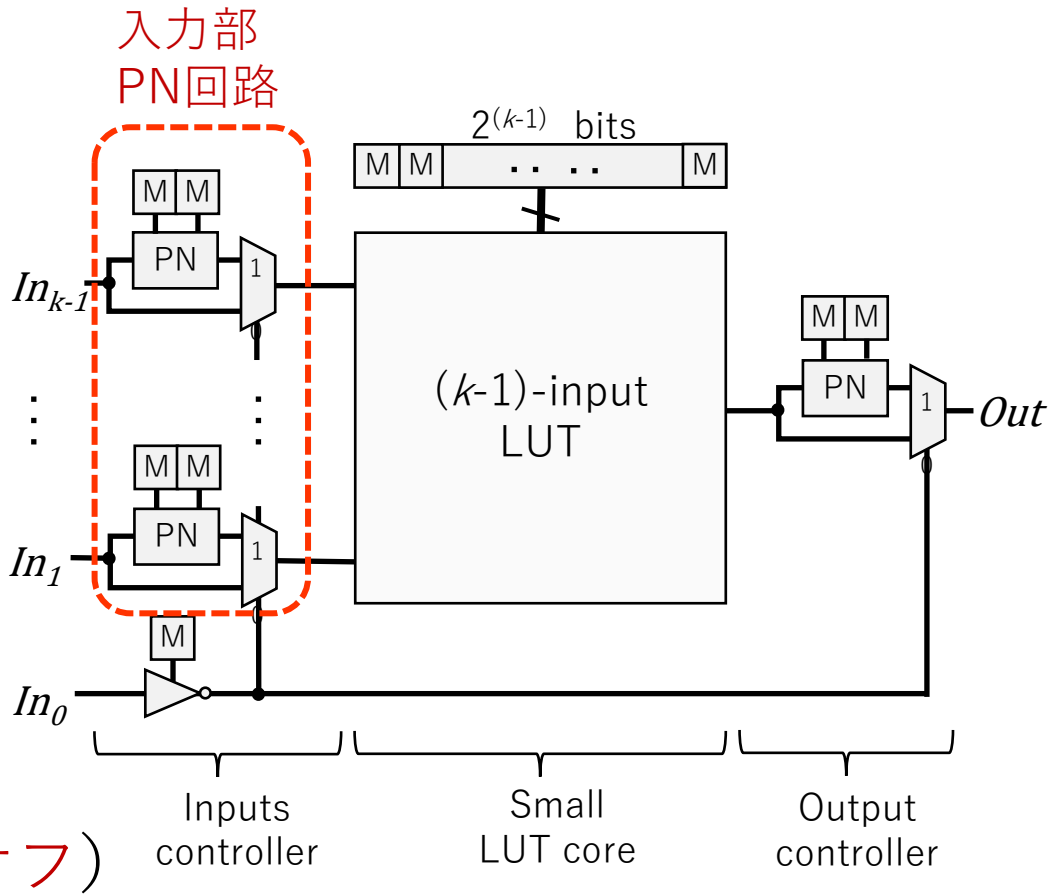
eFPGAは難しい

- FPGAの構成情報設定、スキャン、故障検出にはかなりのノウハウが必要
- FPGAのセルは配置、配線が通常のロジックとは異なる
  - レイアウト作業が大変
  →ハードIPは高額、プロセス依存になる
  →通常のHDLで書いたソフトIPでは役に立たない
- 設計用ソフトウェアをCレベル設計からConfiguraiton Data生成までをそろえるのは大変
- SLMLETでは
  - SLMは通常のFPGAに比べて面積効率が優れている。
  - SLMは長年の研究によりノウハウが蓄積されている。
  - CADもバックエンドは全てフリーソフトウェア、Dockerにより移植性が高い
  - HLSはNECのCyber Work Benchを利用
    - マルチチップの設計環境をXilinxのFPGA対象に構築中

SLM(Scalable Logic Module) とは？

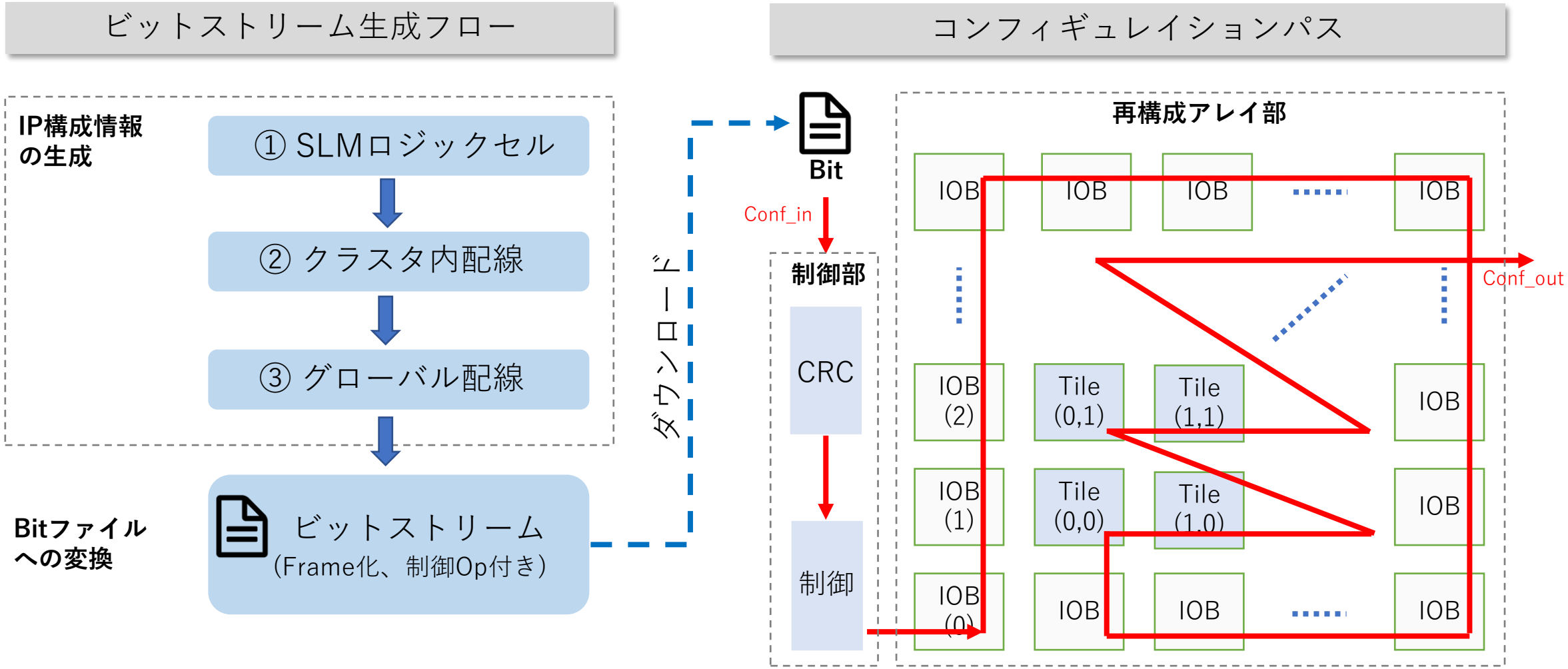
熊本大学で長年を研究開発をしている新しいFPGA（FPL2014）

- シャノン展開による部分関数分解時の性質を利用して構成メモリを抑える（NPN表現）
- k入力SLMの構成メモリのオーダーは$O(2^{k-1})$
- SLM向けCADツールを開発済
- オプションとして入力部のPN回路の有無は選択可能（実装可能論理と面積とのトレードオフ）



k入力SLMセル

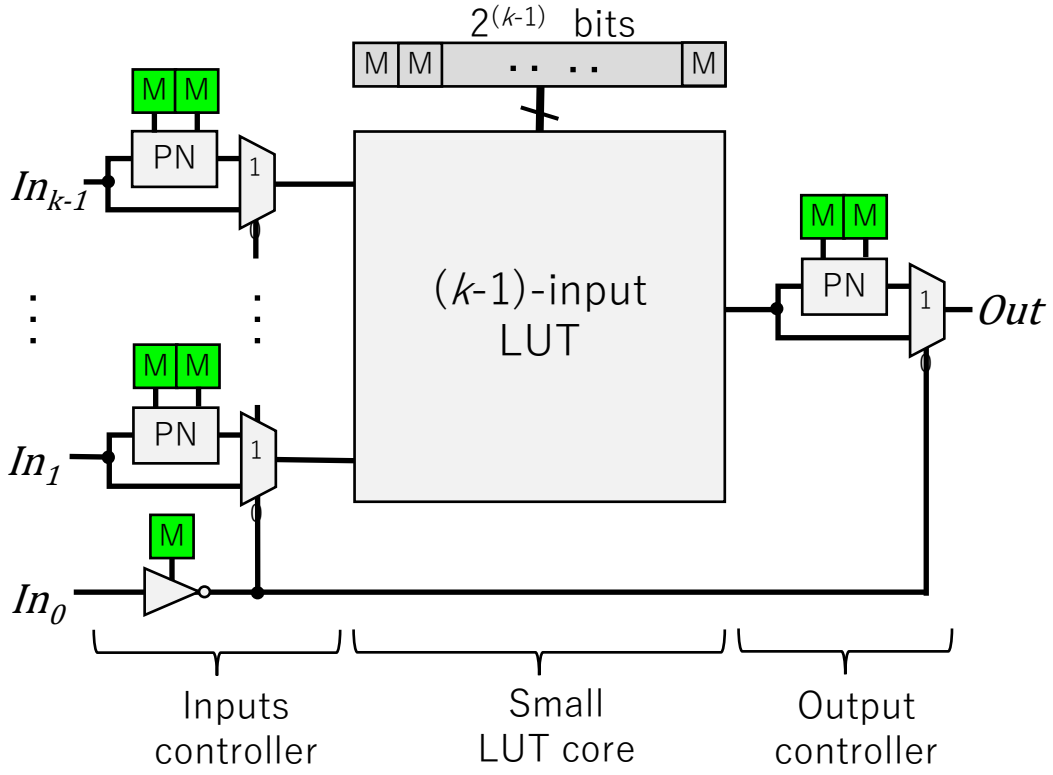
ビットストリームの生成とコンフィギュレイション



配線・I/O部のビットストリーム生成は従来と同様だが、
SLMセル構成情報の決定は新たな手法が必要

SLMセル構成情報の決定

SLMロジックセル



LUTコアの構成情報Mと入出力制御の構成情報Mを決める必要がある

SLM構成情報の生成方法

実装ロジックをシャノン展開後のサブLUTの真理値表によりMの値を決定

Mのパターン探索を行う

全Mのビットパターンを更新

SLMと実装ロジックが等価？

No

Yes

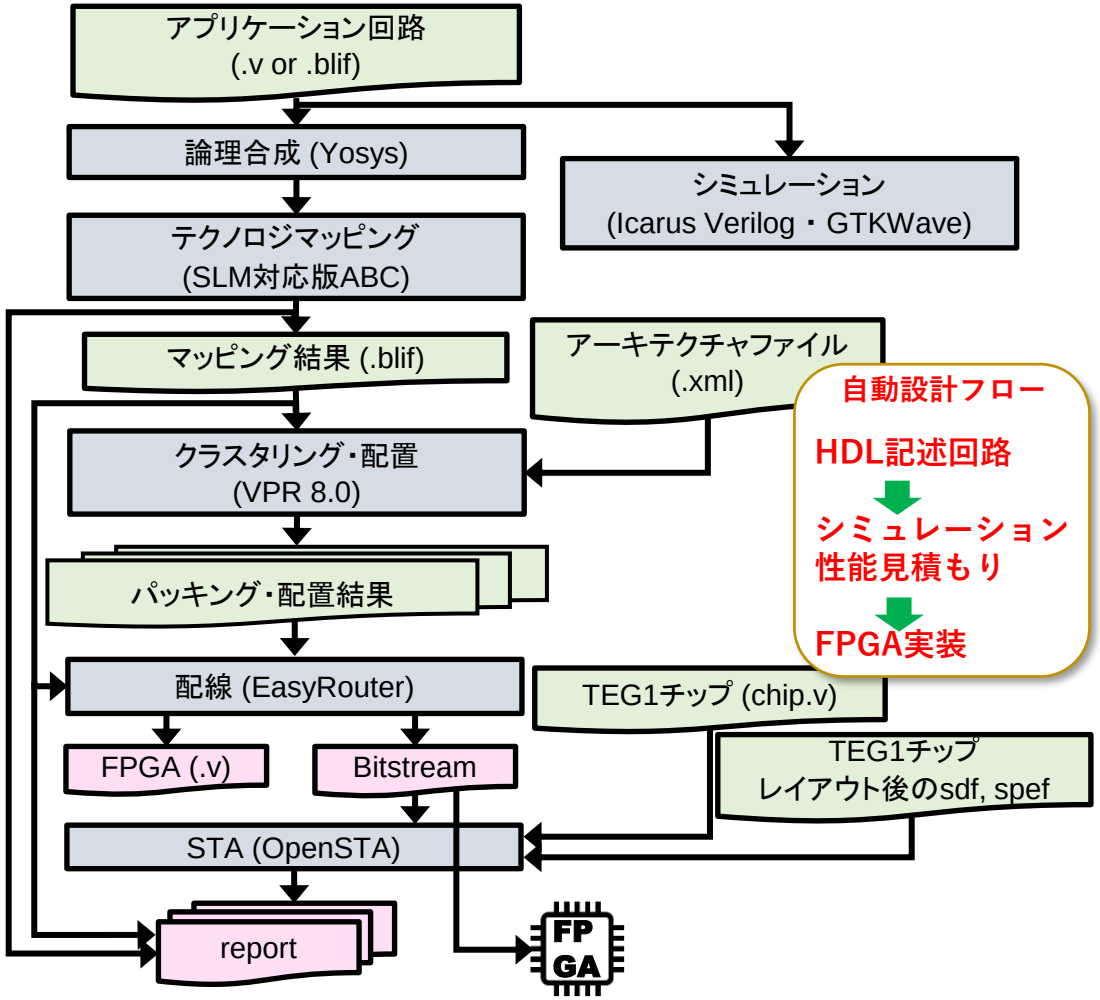
Mのパターンを決定

Mと Mをあわせて、実装ロジックと論理的に等価するSLMの構成情報が分かる

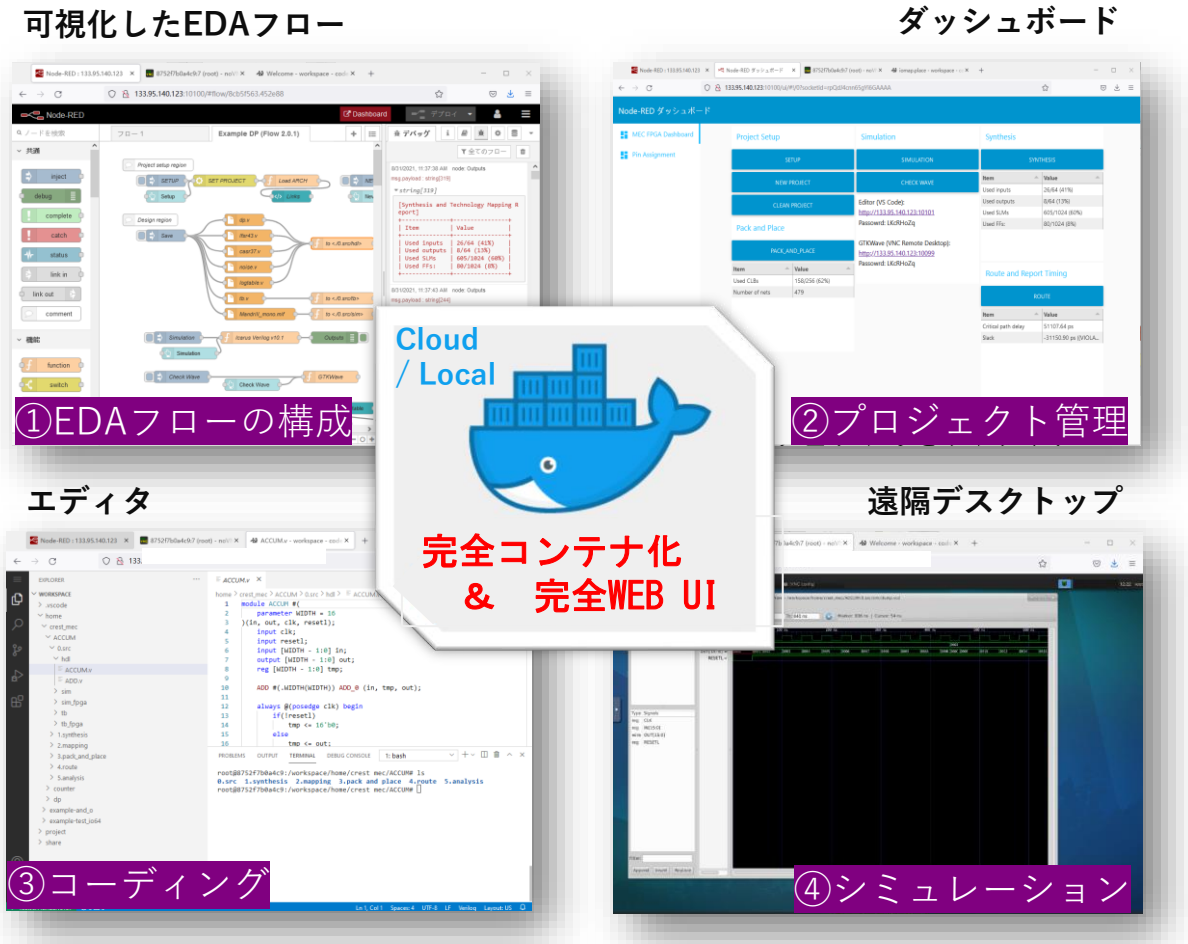
ブラウザベース設計環境の構築

～ EDAフローの確立と設計環境の整備 ～

EDAフロー



設計環境

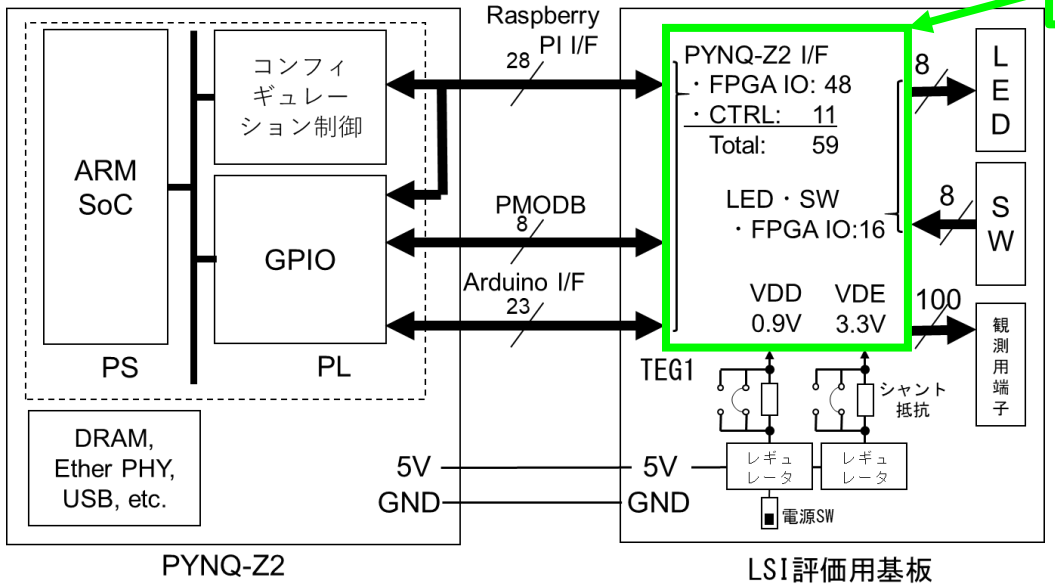


実アプリケーションを実装・評価する環境が整った

TEG1チップ評価（USJC 50nm利用）

LSI評価用基板の開発

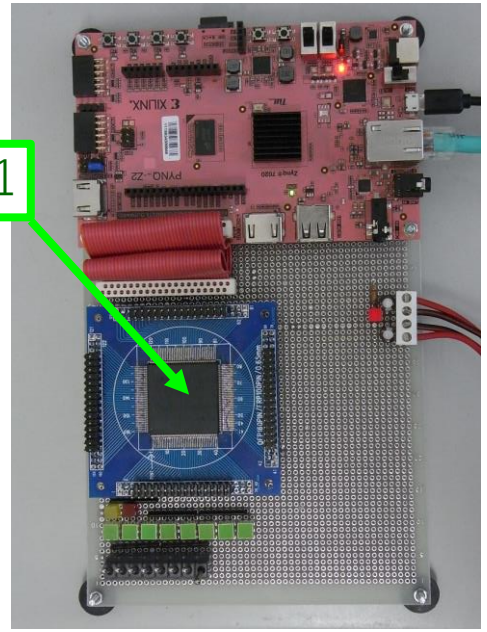
- 設計方針
  評価チップの制御信号およびFPGAの入出力を柔軟に制御できるようにする
  - ✓ PYNQ-Z2ボード(Xilinx Zynq XC7Z020)のドータボードとして設計
  - ✓ PYNQからPythonより制御
  - ✓ FPGA制御信号11本，入出力48本の計59本をPL部に接続



TEG1

TEG1評価

- 評価用基板のモックアップボードを作成
  評価回路：24ビットバイナリカウンタ
  動作確認：コンフィギュレーション後，
  問題なく動作していることを確認



評価用基板モックアップ

リソース消費：
FF：24/1024 (3%)
BLE：38/1024 (4%)

STA解析結果：
FF-to-FFクリティカルパス遅延
5.658ns (176.7 MHz)

動作周波数：(実測)
Max 約83 MHz (上記評価回路)

消費電流：(カウンタ動作時)
コア電源(0.9V)：90mA
I/O電源(3.3V)： 36mA
(電源に内蔵の電流計による)

LSI評価基板製造が間に合わなかったのでモックで評価

RISC-Vの役割

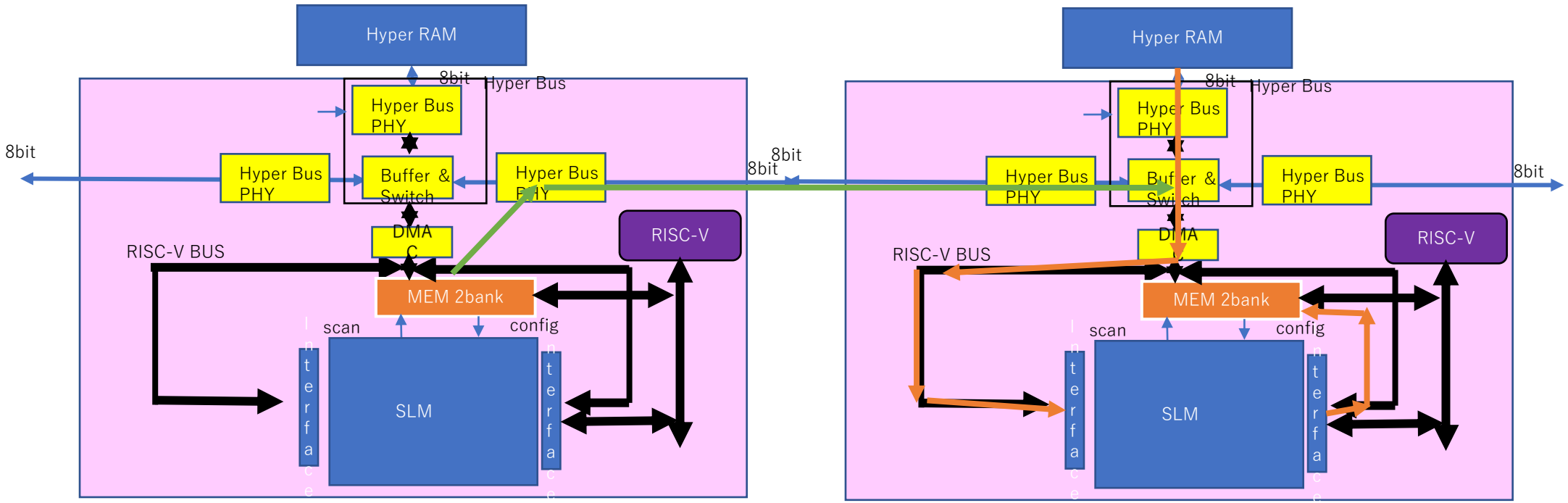
- チップのコントローラ
  - 基本機能はROMに置く予定→なるべく簡単にSLMをConfigurationして起動
    - Configuration Dataの取り込み
    - SLMへの設定
    - SLMの起動
    - HyperbusのDMA起動
    - ボディバイアス制御による電力削減
- RV32I＋RV32Mのみ
- TEG2ではキャッシュなしOSなし、Cでのプログラムが動作可能な最低レベルの構成
- マルチサイクルか簡単なパイプライン構成
- 128KBの2バンクのデータメモリ、16KBの命令メモリを利用
- SLMでRISC-Vの命令拡張を行うことは、とりあえず考えていない
  - ま、そのうち、、、

Hyperbusとは？

- 入出力各8ビット、1.8Vで両方向エッジ（最大200MHｚ）転送を行うI/Oバス
  - SPIの上位規格
  - 簡単な制御ビット
  - 低消費電力
  - 基本的にDMAでブロック転送をする
- Cypress社のHyperRAMが接続可能
  - ４Ｍ－16Mビット×8程度
  - 低電力（特にスリープ時）

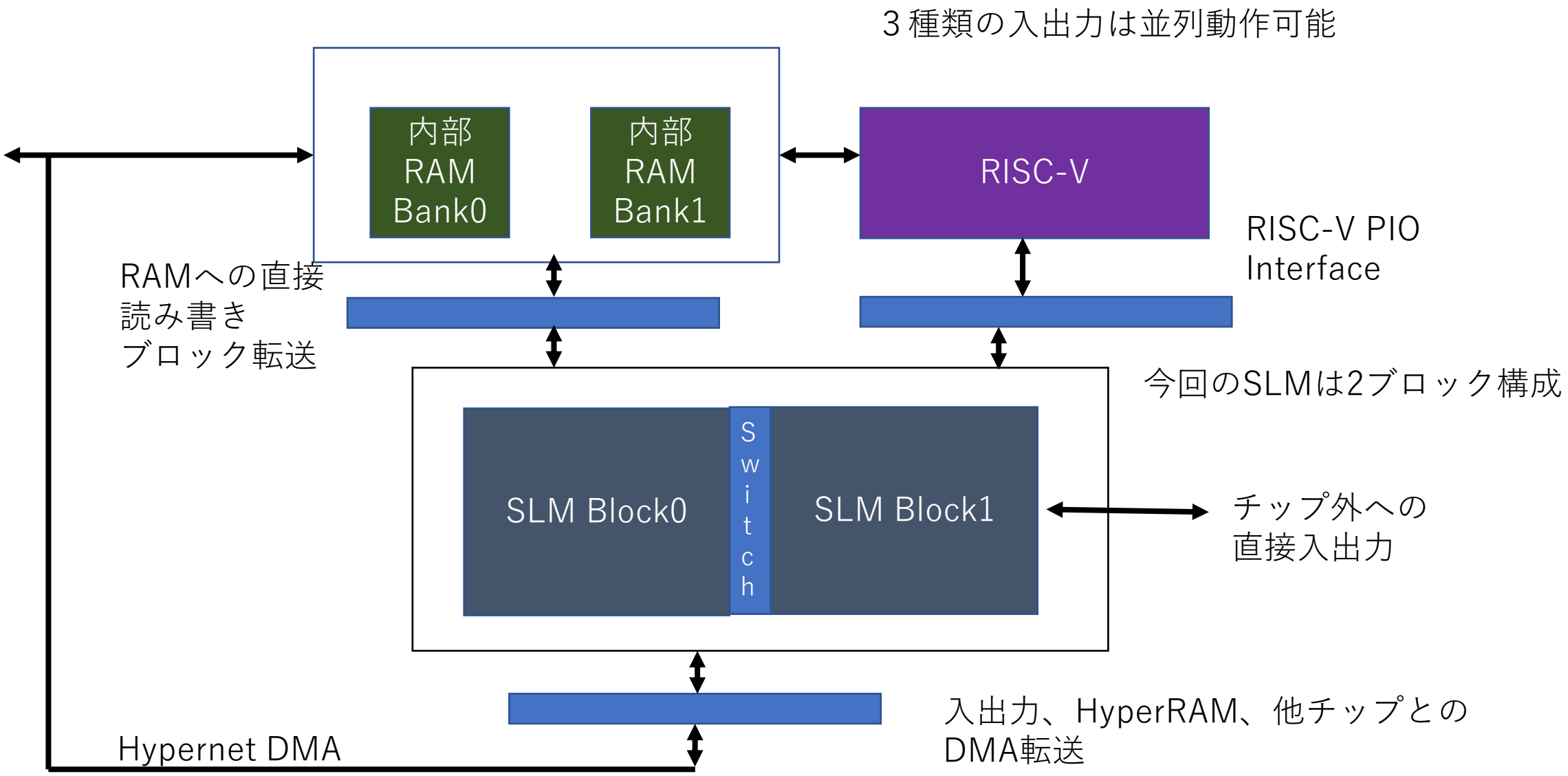
SLMLETのチップ間転送

- 自チップHyperRAM、他チップの内部メモリに転送可能
- 自チップのSLM、他チップのSLMに直接転送可能

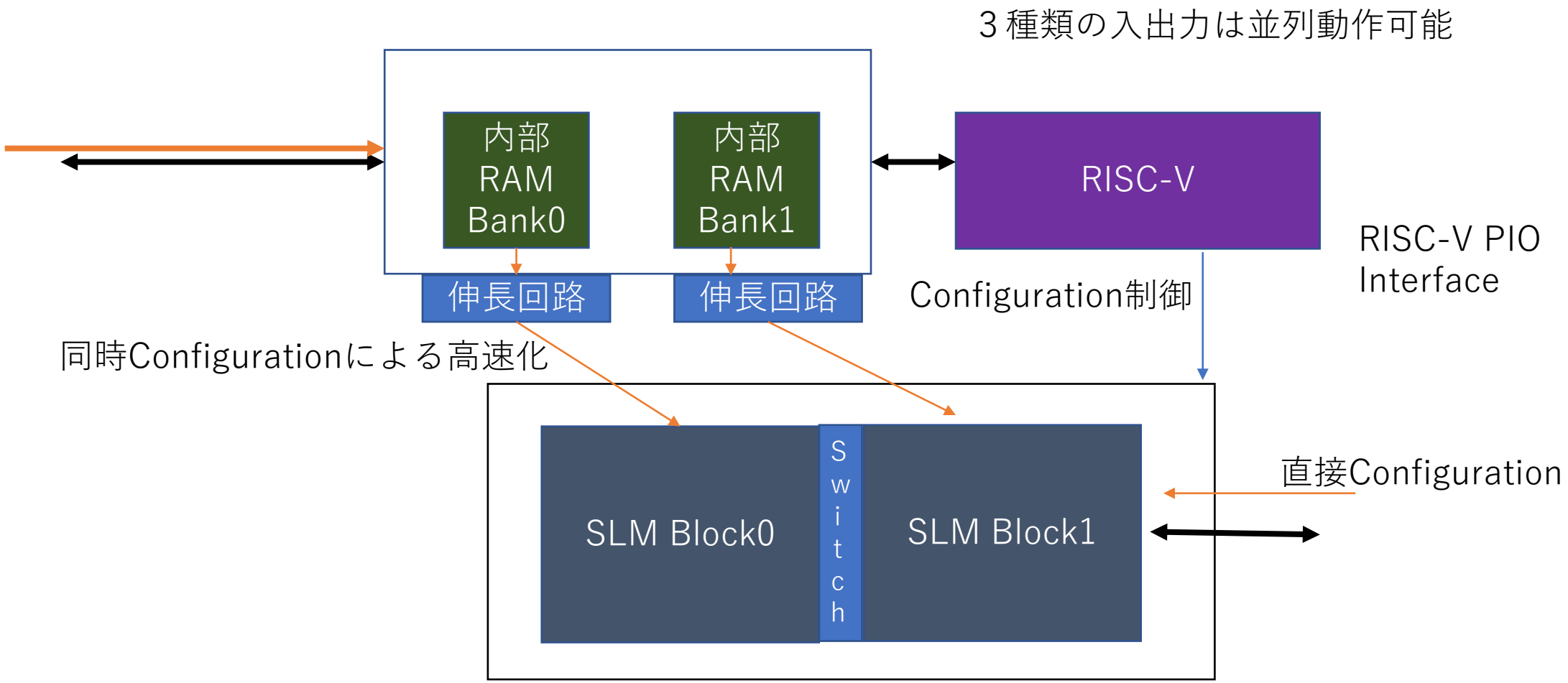


マルチチップのストリーム処理がスムーズにできる

SLMの入出力

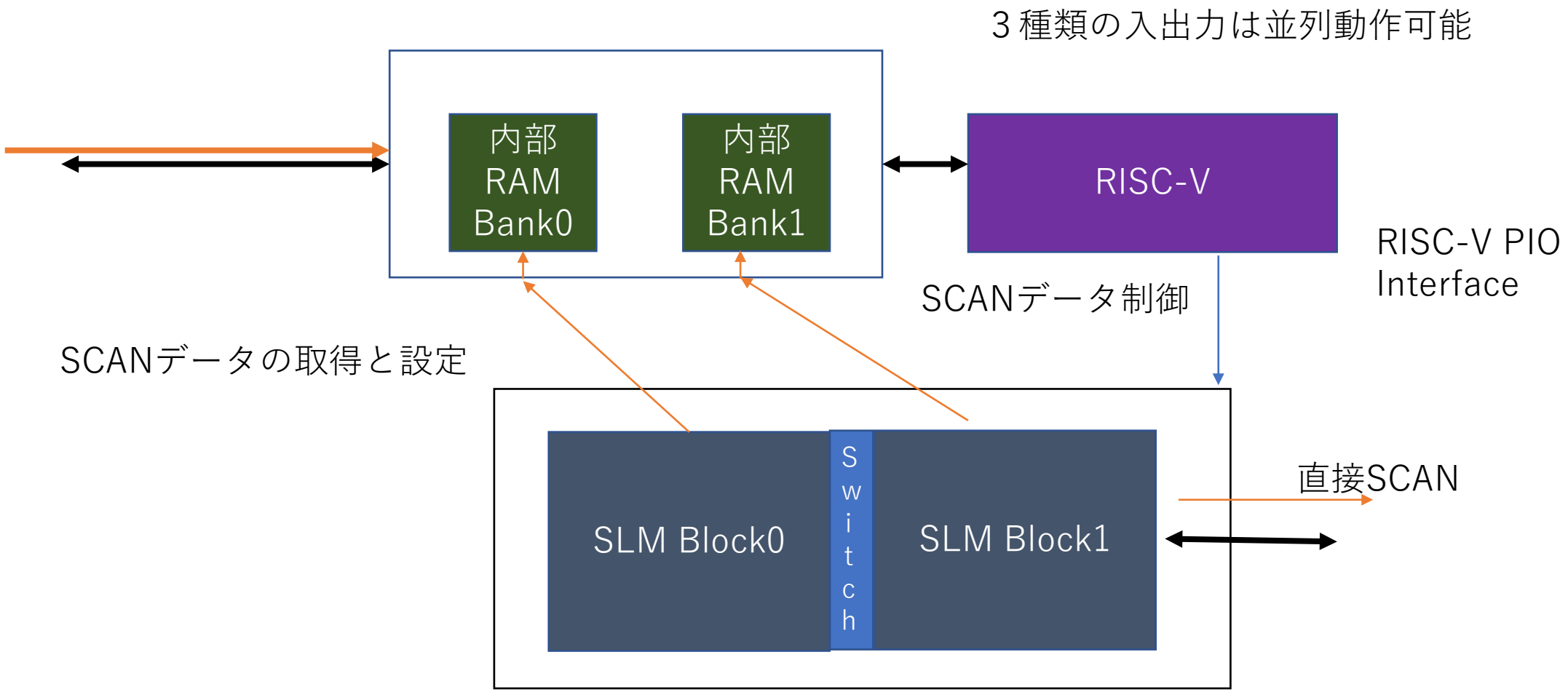

3種類の入出力は並列動作可能
内部
RAM
Bank0
内部
RAM
Bank1
RISC-V
RISC-V PIO
Interface
RAMへの直接
読み書き
ブロック転送
今回のSLMは2ブロック構成
SLM Block0
Switch
SLM Block1
チップ外への
直接入出力
入出力、HyperRAM、他チップとの
DMA転送
Hypernet DMA

SLMのConfiguration



チップ内に圧縮したConfiguration Dataを複数セット溜めておき入れ替えて並列Configurationをする。IoT用に面積有効活用。機能を切り替え利用可能
全てをプロセッサのメモリにマップしたXC6200よりも現実的？な方法

ハードウェアコンテキストマイグレーション



SCANデータを読み込み、書き込みを内部RAM経由で可能。Configuration Dataと共に
他チップに送って、ハードウェアコンテキストマイグレーションができる

おわりに

- USJC50nm　2.1mm角で設計したTEG1（SLM部のみ）は稼働
  - 50MHz程度で動作
  - Spartan3 XC3S50よりやや大きい程度
- 現在、SLMLET-0（TEG2）を4.2mm角で設計中
  - 来年3月テープアウト予定
  - SLM部はTEG1の2倍サイズ（2個のLogic Region）
  - RISC-VはRV32I、RV32Mで、キャッシュ、TLBなしのコントローラ
  - TEG3以降、キャッシュを付けて簡単なOS稼働を狙う
- SLMLETのIPはオープンソース
  - RISC-Vを使ったチップのIPとして使ってもらえることを期待