Feb. 29th 2024

組込みマルチコア研究発表会

組込みソフトウェア向け
消費電力見積もりSHIMスキーマの提案

コウ　エツ
HOU YUE
埼玉大学大学院　理工学研究科
数理電子情報専攻　情報工学プログラム
安積研究室　博士前期課程

Saitama University

Outline

■研究背景

■前提知識

■解決したい課題

■アプローチ

■実験結果

■まとめ

組込みシステム

■ 組込みシステムの大規模化・複雑化が進行

- 高性能・低消費電力のマルチ/メニーコアプロセッサの導入が望まれている

■ 消費電力の見積もりの必要性

- 組込みソフトウェアでは、通常、最大電力が与えられる
  - 消費電力がこの制限内に収まるように設計する必要がある
- 消費電力の見積もりを行うことで、電力を多く消費する処理を判定
  - 消費電力を削減する時に、再設計する処理の判断する

ロードマップ

■ 研究の進捗

SHIM（Software-Hardware Interface for Multi-Many-Core）

■ SHIMの特徴

- ボードやチップの特徴をパラメータ化して、XML形式で記述される
- 基本的なプロセッサ情報など，ソフトウェア開発において重要となることが記述されている
  - レイテンシ，FIFO レジスタ通信のような通信、メモリアクセスの情報や，クロック，命令セット，キャッシュサイズ

SHIMにある電力に関する内容

■ PowerConfiguration

- SHIM 2.0では、開発者とツールが最適な作業を行うために必要な適切な情報を提供するために、パワーモデリングのサポートが導入されている

	説明
PowerConfiguration	電力に関連する設定や構成を表す
PowerConsumptionSet	動作点の電力消費量と、電力消費が定義されているコンポーネントを定義する
PowerConsumption	電力消費値を定義する
PowerConsumerRef	親のPowerConsumptionSetによって定義されるコンポーネントの電力消費を参照するために使用される

解決したい課題

- モデルベース開発で、組込みソフトウェアの消費電力を見積もる研究がない
- SHIM2.0では、定式化する機能がない
- 「組込みソフトウェア向け消費電力見積もりSHIMスキーマの提案」というテーマを提案する

アプローチ

■ 問題を解決するための主なステップは以下の通り：

1. 実機を使用した消費電力の測定
   - 見積もりの基準となるデータを取得する

2. 干渉要素と規則性の調査
   - 定式化のために、両者の関連性を理解する

3. 消費電力を表すスキーマの作成
   - スキーマを作成することで他のハードウェアにも適用する

4. スキーマを使用した各計算処理に対する消費電力の見積もり
   - 実態にあった消費電力の見積もりを行う

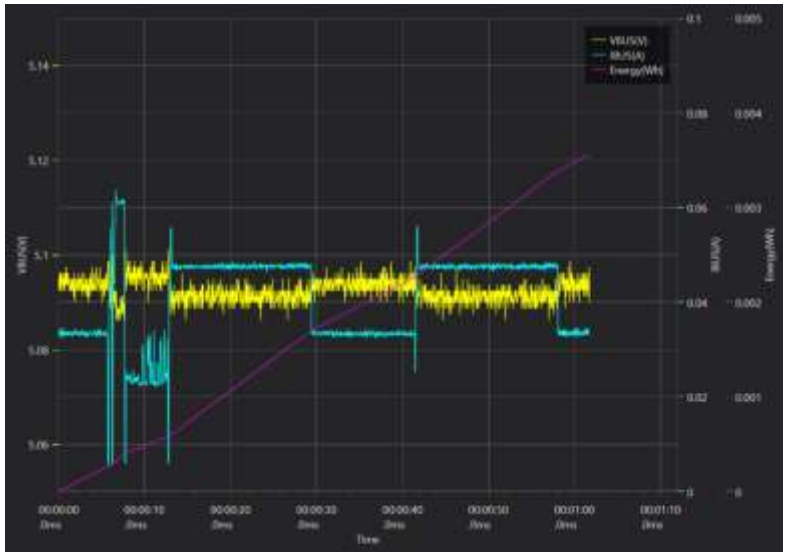
5. ソフトウェアの消費電力の見積もり
   - ハードウェア要件に違反していないかを確認する

実験結果

■ 実験１

- 実験環境
  - ターゲットデバイス：SONY Spresense (ARM Cortex M4F)

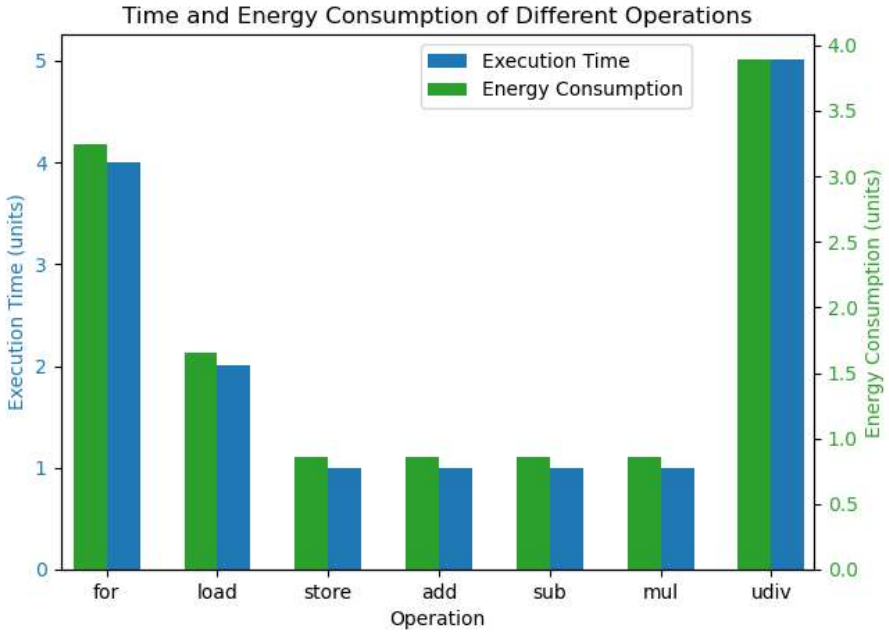
  - 測定デバイス：AVHzY CT-3 USB テスター



実験結果

■ 実験

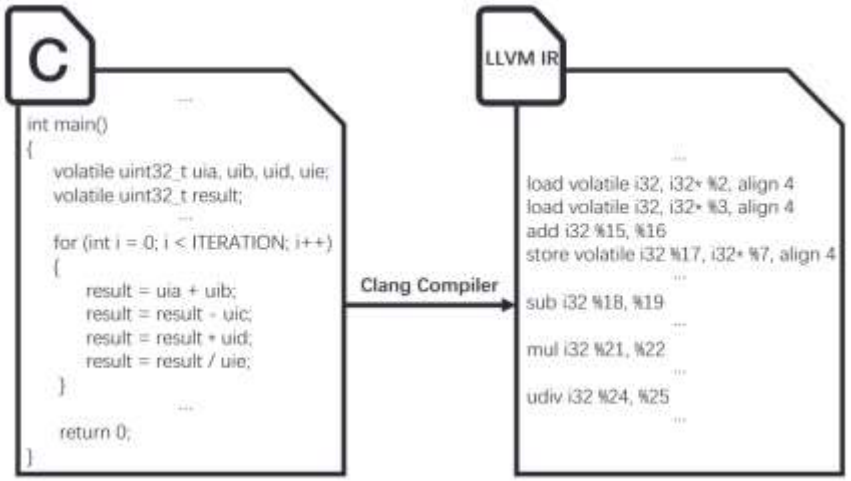
- 基礎評価
  - ターゲットデバイスの基本命令の消費電力量と実行時間を測定する
  - シングルコア環境における実際の実行時間と消費電力を取得するためのテストスクリプトを作成する
  - 計測された単一命令データに基づき、LLVM-IR命令レベルでの予測を行い、計測値と比較することで予測精度分析を行う



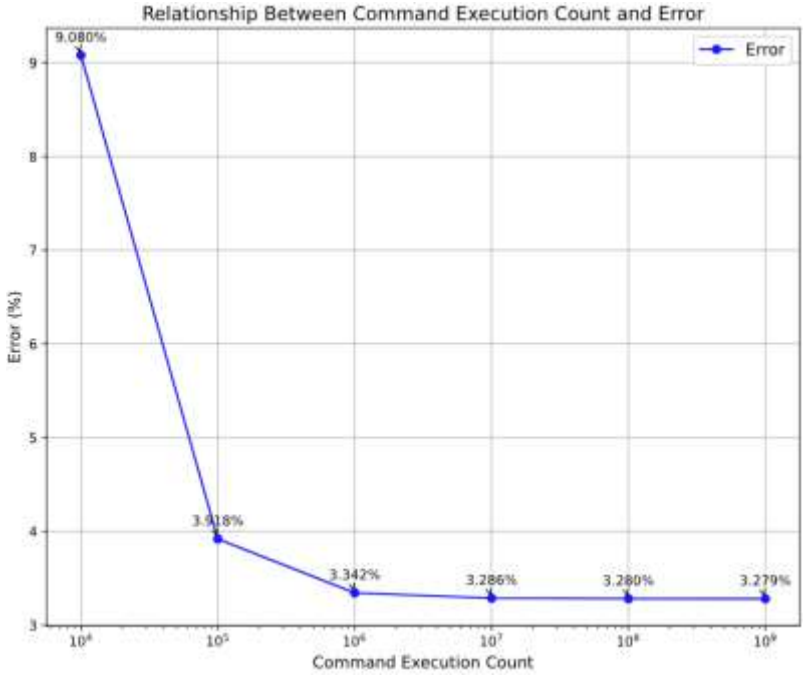
実験結果

■ 実験

- 基本評価
  - テストスクリプト
    - 四則演算



  - 消費電力の予測
    - for文の部分に注目する



実験結果

■ 実験

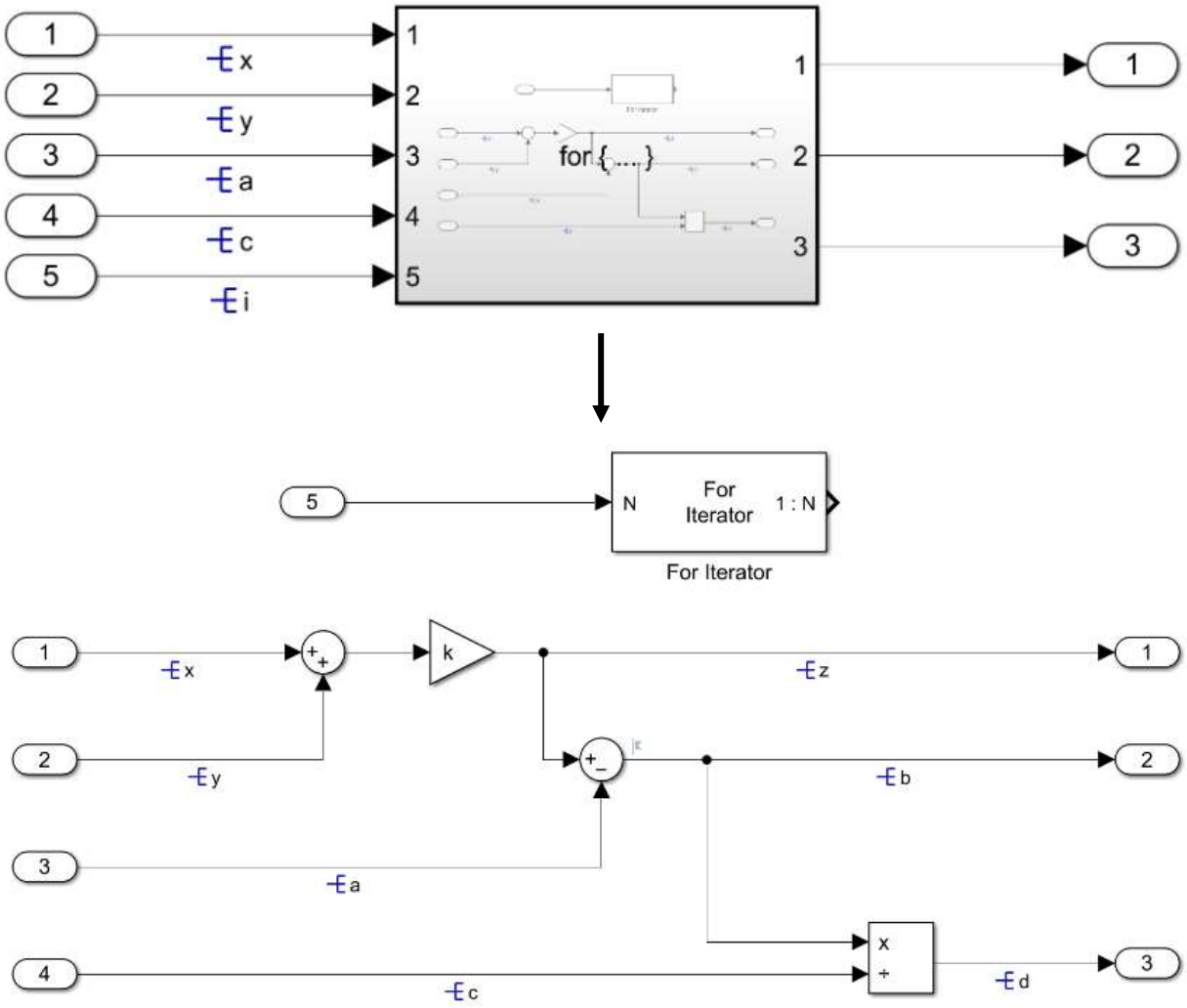
- 入力データの構造（3層構造）
  - CommonInstructionSet
    - 記述スキーマ全体のルート要素として機能し、1つまたは複数の命令のエネルギー消費情報を含む
  - Instruction
    - name属性を通して特定の命令名を識別し、各命令のエネルギー消費情報を表現する
  - PowerConsumption
    - 最も具体的な情報レイヤで、各命令に関連するエネルギー消費のコストを直接記述する。
    - Impactは、マルチコアシステムで行われる将来のエネルギー予測のための展望として考えられている
  - 例：

```
<CommonInstructionSet>
    <Instruction name="load">
        <PowerConsumption>
            <Cost>1.656</Cost>
            <Impact>0.000</Impact>
        </PowerConsumption>
    </Instruction>
<CommonInstructionSet>
```

実験結果

■ 実験

- モデルを用いる評価



実験結果

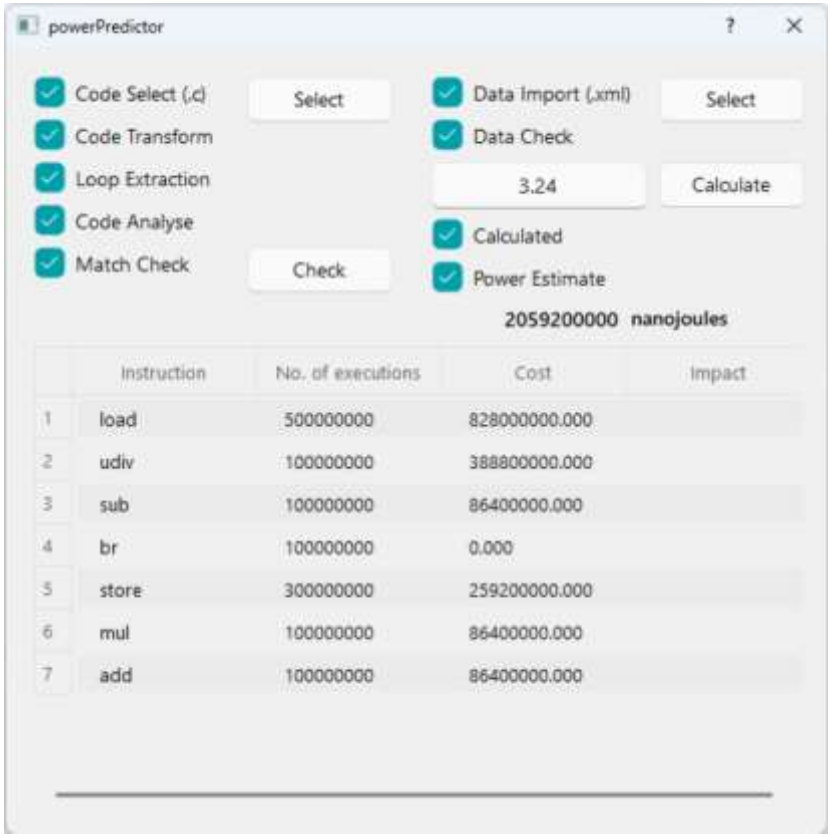
■ 実験

- モデルを用いる評価
  - Embedded Coderで生成されたコード
  - 関数呼出しでユーザーのコードで実行する
  - LLVM-IR命令に転換する
  - 予測ツールで予測を行う
  - 結果エラー：４％

```
for (sl_iter = 1; sl_iter <= tmp; sl_iter++) {
  /* Gain: '<S1>/Gain' incorporates:
   *  Inport: '<Root>/Inport'
   *  Inport: '<Root>/Input'
   *  Sum: '<S1>/Sum'
   */
  z = (x + y) * k;

  /* Sum: '<S1>/Sum1' incorporates:
   *  Inport: '<Root>/Inport1'
   */
  b = z - a;

  /* Product: '<S1>/Divide' incorporates:
   *  Inport: '<Root>/Inport2'
   */
  d = b / c;
}
```



まとめ

■研究背景

- 組込みシステムは通常最大電力が与えられるため、消費電力の見積もりが必要

■提案手法

- モデルベース開発でLLVM-IR命令レベルの電力消費量を導入して、組込ソフトウェアの消費電力を予測する

■結論

- LLVM-IR命令レベルの消費電力見積もり手法は利用可能

■将来的な問題

- マルチコアへの対応
- 外部デバイスの電力消費量の導入手法

■本研究の方針

- 組込みソフトウェアに向ける静的な消費電力見積もりに適用するスキーマと見積もり手法を提案する

補足資料

■ 他の実験

- 実験環境
  - ターゲットデバイス：ESP32 (Xtensa LX6)

- 一部のデータ

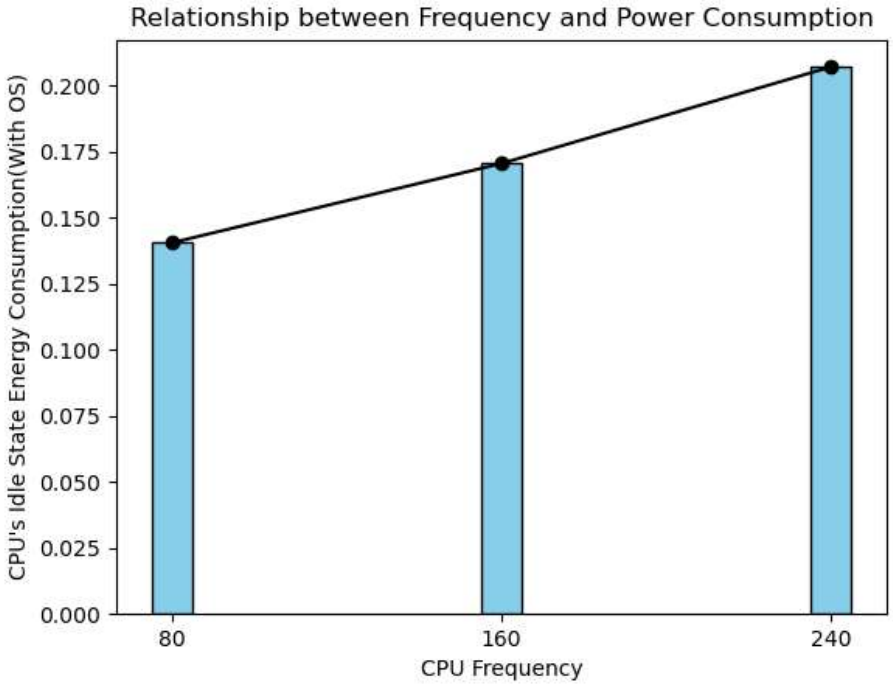
add	利用中のコア数	各コアの実行回数	総実行回数	コストパワー(J)
80	1	200000000	200000000	8.928
160	1	200000000	200000000	6.084
240	1	200000000	200000000	5.4
80	2	100000000	200000000	5.4
160	2	100000000	200000000	3.924
240	2	100000000	200000000	3.816

補足資料

■ 他の実験

- コストエネルギーの分割
  - ソフトウェア消費量（アプリケーション）
  - 基礎消費量（OS、コンポーネントの静的消費）

- 基礎消費電力とCPU周波数の関係（２コアがIdle状態）



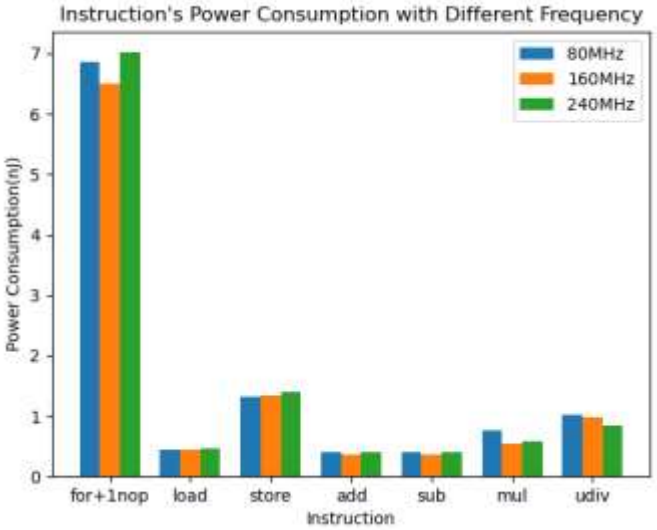
補足資料

■ 他の実験

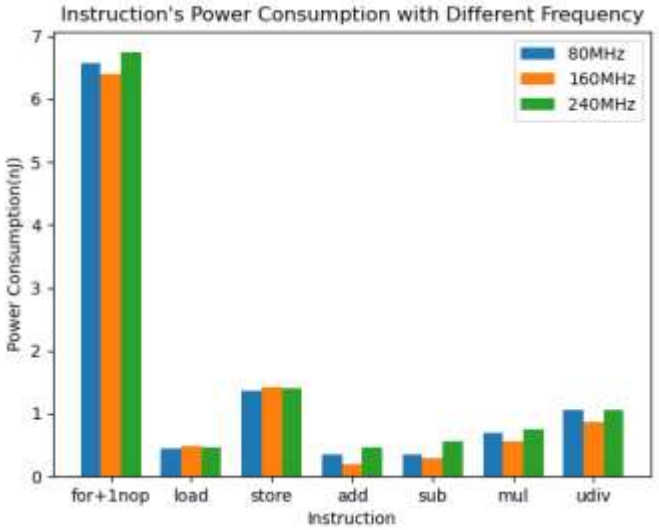
- 分割した後のデータ

add	利用中のコア数	各コアの実行回数	総実行回数	基礎消費量	ソフトウェア消費量
80	1	200000000	200000000	7.037	1.890
160	1	200000000	200000000	4.262	1.822
240	1	200000000	200000000	3.448	1.951
80	2	100000000	200000000	3.561	1.839
160	2	100000000	200000000	2.131	1.792
240	2	100000000	200000000	1.907	1.909

- 各処理のコスト



シングルコア



デュアルコア