Technical Reports on Information and
Computer Science from Kochi
Vol. 2 (2010), No. 4

# ソフトウェアの実行時間を考慮したハードウェア/ソフトウェア分割手法

Hardware/Software Partitioning Methodology with Consideration of Software Execution Time

松永 惇弥　　村岡 道明
Junya Matsunaga　Michiaki Muraoka
高知大学大学院　情報科学分野　村岡研究室

## 1. まえがき

近年、LSI の大規模化が進み、それに伴う設計期間の長期化が問題となっている。また、システムレベルにおいては、ハードウェア/ソフトウェアの最適な分割が大きな課題である。しかし、従来では、ハードウェア/ソフトウェアの分割を行う際にソフトウェアの実行時間を考慮するのが困難であった。そこで、本研究では、ソフトウェアの実行時間を考慮したソフトウェアの並列化およびハードウェア/ソフトウェア分割を容易に試行する手法を提案し、その手法を暗号化アルゴリズム AES に適用した。

## 2. ソフトウェアの実行時間を考慮したハードウェア/ソフトウェア分割手法

本手法は、次の 6 ステップより構成される。

(1) 時間精度付きモデルの作成：C 言語で記述されたアルゴリズムを対象として、時間精度付きモデル(時間精度付きモデル:C コードにターゲットプロセッサの実行サイクル数を付与したもの)を作成する。
(2) 時間精度付きモデルのシミュレーション：(1) で作成した時間精度付きモデルのシミュレーションを行い、プロファイリングに必要なデータを算出する。
(3) プロファイリング：(1)で作成した時間精度付きモデルと(2)で算出したデータを用いて、そのターゲットプロセッサにおけるアルゴリズムの内部動作やテーブルの動作回数、アクセス頻度、サイクル数などを求める。
(4) 並列化向きビヘイビア分割：(3)の性能結果から、マルチコアを前提とした並列動作によりソフトウェア上で高速化を図る。
(5) ハードウェア/ソフトウェア分割：(4)で得たモデルを SER(Specification, Exploration & Refinement Environment)上で様々なアーキテクチャを作成する。
(6) 性能評価：ハードウェア/ソフトウェア分割およびソフトウェアの並列化を行ったモデルを、(1) 時間精度付きモデルの作成、(2) 時間精度付きモデルのシミュレーションを行い、プロファイリングから分割後のモデルの性能を評価する。

以上により、評価した結果が性能上問題なければ終了となる。しかし、並列化向きビヘイビア分割が適切でない場合には、(4)に戻る。次に、並列化向きビヘイビア分割の性能向上が難しい場合は(5)に戻りハードウェア化の検討を行う。

## 3. 提案手法の適用

今回提案する本手法を暗号化アルゴリズム AES に適用した。本手法を適用(2.(3)参照)するにあたり、Visual Spec (vesion4.1.6)上で時間精度付きモデルの作成を行った。条件として、ターゲットプロセッサは TX49H2、コンパイラは MULTI for MIPS Ver 4.2.3 を使用した。クロック周波数は、25MHz とした。以上を用いてシミュレーションを行い算出したデータを基にプロファイリングした結果を表 1 に示す。表 1 より、AES 暗号のボトルネック部分は MixColumns 処理であり、全体の 60.15%を占めていることが分かった。次に、MixColumns の逐次的処理を並列処理にすることにより高速化を図った。表 1 から Columns0～3 は、ほぼ同じ実行サイクル数であり、並列処理を行っても問題ないため、Columns0～3 をビヘイビアに分割し、4 並列ソフトウェア化を行った。

表 1. AES のプロファイリング結果

| ファンクション名 | ライン数 | 呼び出し回数 | 処理サイクル数(1回) | 累積サイクル数(各ファンクションが占める割合) |
|---|---|---|---|---|
| KeyExpansion | 74 | 1 | 1301 | 1301(11.25%) |
| AddRoundKey | 95 | 11 | 56 | 616(5.33%) |
| SubBytes | 269 | 10 | 181 | 1810(15.65%) |
| ShiftRows | 109 | 10 | 66 | 660(5.17%) |
| MixColumns | 214 | 9 | 773 | 6955(6015%) |
| Columns0 | 174 | 9 | 159 | 1427 |
| Columns1 | 184 | 9 | 159 | 1431 |
| Columns2 | 194 | 9 | 158 | 1425 |
| Columns3 | 204 | 9 | 158 | 1421 |

## 4.ソフトウェア並列化の結果

3.から MixColumns から Columna0~3 をビヘイビアに分割し、4 並列ソフトウェア化したものを図 1 の左側に示す。また、元のソフトウェア記述とその性能を比較した結果を図 1 の右側に示す。

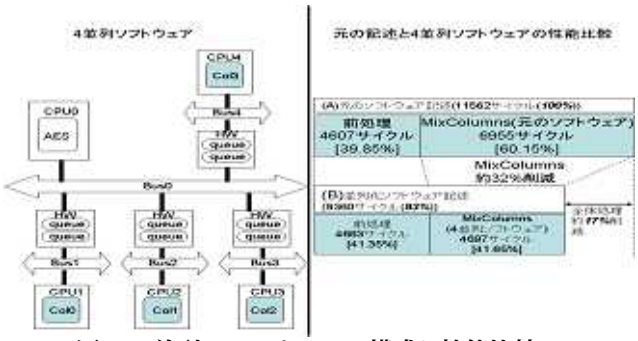


図 1. 4 並列ソフトウェアの構成と性能比較

図 1(左側)に示すように、各ビヘイビアを分割し、CPU0～CPU4 に割り当てることによりソフトウェア上で並列化することができた。また、図 1(右側)に示すように、ボトルネック部分である MixColumns の実行時間を 32%削減でき、また、全体の実行時間は、17%削減することができた。

## 5.まとめ

本論文では、ソフトウェアの実行時間を考慮したハードウェア/ソフトウェア分割手法を提案し、本手法を暗号化アルゴリズム AES に適用し、評価を行った。その結果、元のソフトウェアのボトルネック部分は、4 並列ソフトウェア化を行うことにより、実行時間を元のアルゴリズムよりも 32%を削減できた。また、全体の実行時間においては、元のソフトウェア記述に比べて 4 並列ソフトウェアでは 17%を削減できた。