

# 高位合成処理システムCCAPを用いた AES暗号処理の高速化

神原 弘之<sup>†</sup> 梅原 直人<sup>††</sup> 中谷 嵩之<sup>††</sup> 石浦 菜岐佐<sup>‡</sup> 富山 宏之<sup>‡‡</sup>  
<sup>†</sup> 京都高度技術研究所 <sup>††</sup> 立命館大学 <sup>‡</sup> 関西学院大学 <sup>‡‡</sup> 名古屋大学

我々は C Compatible Architecture Prototyper(CCAP) 高位合成処理系を開発している。CCAP は、ANSI C で記述された組込みソフトウェアについて、プロセッサによる解釈実行より高速であり、かつ回路規模がコンパクトな専用ハードウェアを、もとのプログラムを変更することなく生成することを目指している。CCAP では、生成された専用ハードウェアとプロセッサの主記憶へのアクセスを制御するCCAP 専用の調停回路を用意している。これにより、生成された専用ハードウェアは、ハードウェア設計者の助力がなくともプロセッサと一体化されてもとのソフトウェアと同じように実行結果を得ることができる。本稿では、この CCAP を用いたシステム設計手法を AES 暗号処理の高速化に適用した結果について報告する。

## Speed Improvement of AES Encryption using hardware accelerators synthesized by C Compatible Architecture Prototyper(CCAP)

Hiroyuki KANBARA<sup>†</sup> Naoto UMEHARA<sup>††</sup> Takayuki NAKATANI<sup>††</sup>  
Nagisa ISHIURA<sup>‡</sup> Hiroyuki TOMIYAMA<sup>‡‡</sup>  
<sup>†</sup> ASTEM RI <sup>††</sup> Ritsumeikan University  
<sup>‡</sup> Kwansai Gakuin University <sup>‡‡</sup> Nagoya University

The authors are developping a high-level synthesizer called C Compatible Architecture Prototyper(CCAP). CCAP compiles ANSI C program which is a part of embedded software and generates an application specific hardware accelerator in HDL. Synthesized accelerator executes faster than a cpu and accesses to main memory like a cpu. CCAP offers an arbiter circuit which makes it possible for the synthesized accelerator and a cpu to access main memory in parallel. In this paper we report the speed improvement of AES Encryption using CCAP

### 1 はじめに

携帯電話に代表される今日の組込みシステムは、音楽再生や動画の撮影といった多大な計算処理を行うデジタル家電に進化してきた。組込みシステムのさらなる将来像としては、高度な信号処理能力と通信機能を備えたユビキタスセンサーが大量にネットワークに接続され、相互に通信を行うことが想定されている。

このようなデジタル家電やユビキタスセンサーを安価にかつ大量に供給するには、マイク、カメラといったセンサーからの情報を信号処理し、その結果をネットワークに送信する組込みシステムをシステム LSI で実現することが必須となっている。商用交流電源ではなくもっぱら電池で動作す

るデジタル家電やユビキタスセンサーでは、供給できる電力が限られるため、組込みプロセッサの動作クロック周波数は、汎用 PC の十分の一程度に制限される。このようなデジタル家電やユビキタスセンサーで汎用 PC と同等の信号処理もしくは通信能力を持たせるには、ソフトウェアでは性能のボトルネックになるような機能について、特定の演算に特化した専用ハードウェアで実現することが行われている。例えば、音声や画像について Gauss や Sobel といったフィルタで信号処理を行ったり、演算結果を送信するにあたり AES や DES アルゴリズムによる暗号化を行う場合、ソフトウェアではなく専用ハードウェアで処理することで一桁以上の高速化がはかれる。このような専

用ハードウェアの設計と、専用ハードウェアを組み込みソフトウェアから制御するドライバプログラムの作成、そしてドライバプログラムと専用ハードウェアを組合わせた際の動作の確認には多大な労力が必要とされており、専用ハードウェアを組み込みソフトウェアから自動生成する技術の確立が望まれている。

我々は C Compatible Architecture Prototyper(CCAP) 高位合成処理系を開発している<sup>1)</sup><sup>2)</sup><sup>3)</sup>。CCAP は、ANSI C で記述された組み込みソフトウェアについて、プロセッサによる解釈実行より高速であり、かつ回路規模がコンパクトな専用ハードウェアを、もとのプログラムを変更することなく生成することを目指している。CCAP では、生成された専用ハードウェアとプロセッサの主記憶へのアクセスを制御する CCAP 専用の調停回路を用意している。これによりハードウェア設計者の助力がなくても、生成した専用ハードウェアと組み込みプロセッサを一体化して、もとのソフトウェアと同じ実行結果をより高速に得ることが可能になることを目指している。

本稿では、今後の通信において用いられる秘密鍵方式の AES 暗号化について、CCAP を用いてソフトウェアの一部を専用ハードウェアとして実現し、高速化をはかった実験結果について報告を行う。ANSI C で記述された AES 暗号化プログラムについて、プログラム中の主だった処理の手順を関数単位で切出し、各々を専用ハードウェア化を行った場合の速度の向上を評価した。組み込みプロセッサの 60% 弱の回路規模の専用ハードウェアを追加することで、5.0 倍弱の処理速度の向上をはかることが確認された。

## 2 高位合成処理システム CCAP

高位合成処理システム CCAP を用いて、(組み込みプロセッサで解釈実行される)「組み込み用ソフトウェア」から、性能のボトルネックとなる機能を抽出し、「専用ハードウェア」として生成するまでの手順を Fig. 1 に示す。

組み込みシステムの設計者は、デジタル家電やユビキタスセンサーの組み込みシステムに搭載する、信号処理や通信などの機能を一般的な組み込み用のプログラミング言語である ANSI C で実装し(「ソフトウェア(信号処理、通信)」)、汎用 PC 等を用いて動作確認を行う。この「ソフトウェア(信号処理、通信)」で記述された機能の一部について専用ハードウェアで実行した場合の性能を『組み込み用

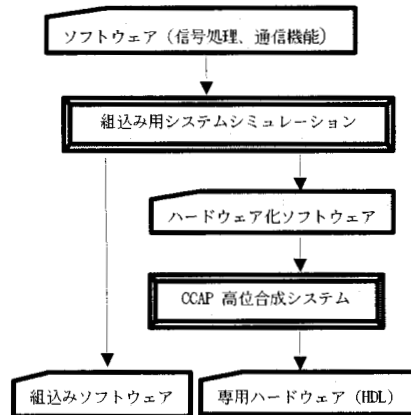


Fig. 1 CCAP 高位合成システムを用いたシステム設計手法

システムシミュレータ』を用いて評価する。「専用ハードウェア」化することにより高速化が可能な信号処理や通信機能は、関数といった単位で「ハードウェア化用」のソフトウェアとして切り出す。我々はこのようなソフトウェアを「ハードウェア関数」と呼んでいる。「ハードウェア関数」については、記述を変更することなくそのまま『高位合成システム CCAP』に入力し、逐次的なソフトウェア処理の記述から高速な専用ハードウェアの回路記述(「専用ハードウェア (HDL)」)を生成する。システム設計者は、「ソフトウェア(信号処理、通信)」から「ハードウェア関数」を除いた「組み込み用ソフトウェア」を組み込みプロセッサ用にコンパイルし直す。

Fig. 2 に示すように、高位合成で生成された「専用ハードウェア」と組み込みプロセッサは FPGA を用いて、プロトタイプハードウェアとして実現することを、我々は想定している。FPGA を用いたプロトタイプで、コンパイルし直した「組み込み用ソフトウェア」を動作させることで、

- もとの「ソフトウェア(信号処理、通信)」を実行した際と同じ機能を実現していること
- システム LSI 化に十分な性能が達成されていること

を確認する。このような手順でシステム設計者が「専用ハードウェア」と「組み込み用ソフトウェア」

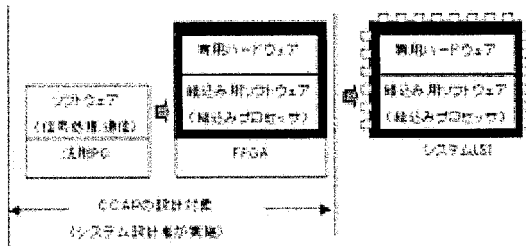


Fig. 2 CCAP 高位合成システムの設計対象

を同時に設計し、組込みシステムのプロトタイプまで完了することができるようになる。

このような設計手法を実現するため、高位合成システム CCAP は以下のような特徴を備えている。

- もとのプログラムを変更することなく専用ハードウェア化するため、組込みシステムで幅広く用いられているプログラミング言語である ANSI C のプログラム記述そのものから、ハードウェア (の HDL 記述) を生成する
- プロセッサで解釈実行されるソフトウェアと合成される専用ハードウェアが共有するデータに高速にアクセスできるよう、ポインタや外部変数といったデータ形式についても高位合成の対象とする

以降では、AES 暗号処理プログラムを CCAP 高位合成システムを用いて高速化を行った設計実験について報告する。

### 3 AES 暗号について

AES 暗号は、2001 年に NIST (National Institute of Standards and Technology) によって、次期米国標準の秘密鍵暗号として選定された秘密鍵によるブロック暗号の方式である。AES 暗号化の処理の手順を Fig. 3 に示す<sup>4) 5)</sup>。AES 暗号化では、128bit (16byte) の入力データ (IN) に対し、「SubBytes」「ShiftRows」「MixColumns」「AddRoundKey」の 4 種類の基本処理を繰り返し行うことで、暗号化された出力データ (OUT) が得られる。繰り返し (loop) の回数は秘密鍵 (KEY) の語長によって異なり、128bit 鍵では 9 回、192bit 鍵では 11 回、256bit 鍵では 13 回になる。

以下では、Fig. 3 中の各基本演算処理について説明する。「KeyExpansion」では、秘密鍵から「AddRoundKey」処理に入力する拡張鍵の生成を

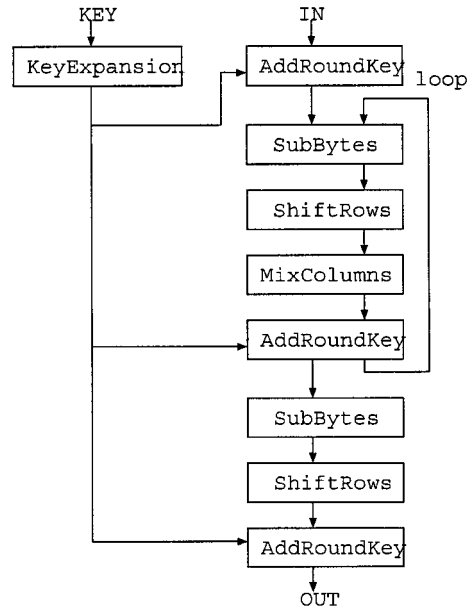


Fig. 3 AES 暗号化の処理の手順

行う。一つの入力ストリームの暗号化について、最初に「KeyExpansion」処理により拡張鍵を用意すれば良い。「SubBytes」処理では、データの各バイトに S-Box と呼ばれるテーブル参照に基づく変換が行われる。「ShiftRows」処理では、16byte の入力データを 4byte 単位で区切った際の 4 つの行の各行毎に、0 から 3byte 単位での巡回シフト演算を行う。「MixColumns」処理では、ガロアフィールド理論に基づいた行列演算を行う。「AddRoundKey」処理では、拡張鍵と入力データの排他的論理和 (XOR) 演算を行う。

### 4 専用ハードウェアを用いた AES 暗号処理の高速化

ソフトウェアによる処理の一部を専用ハードウェアで置き換えることによる、AES 暗号化の高速化については、

- AES 暗号化の ANSI C ソフトウェアを組込みプロセッサで解釈実行
- 手順中の「ShiftRows」演算を専用ハードウェアで実行し、残りの処理を組込みプロセッサが解釈実行
- 手順中の「SubBytes」演算を専用ハードウェア

アで実行し、残りの処理を組込みプロセッサが解釈実行

- 手順中の「MixColumn」演算を専用ハードウェアで実行し、残りの処理を組込みプロセッサが解釈実行
- 手順中の「ShiftRows」「SubBytes」「MixColumn」演算を専用ハードウェアで実行し、残りの処理を組込みプロセッサが解釈実行

の5種類の構成について、128bit長の共通鍵を用いた際の128bitの入力データの暗号化に必要なサイクル数とその際の回路規模を測定して評価した。

「KeyExpansion」は、一つのストリームについて処理の最初に実行すれば良いので、専用ハードウェア化による高速化の全体処理の効率改善への寄与は少ないと考え、ソフトウェア処理のまま残した<sup>6) 7)</sup>。

実行サイクル数の測定には、Mentor社のVerilogシミュレータ：ModelSim XEを用いて、「組込みプロセッサ」と「専用ハードウェア」の動作を模擬して行った。「組込みプロセッサ」には、Verilog HDLで記述したソフトコアのMIPS社のR3000互換プロセッサを用いた。評価に用いた「組込みプロセッサ」にはMMUとキャッシュメモリは含まれていない。AES暗号化ソフトウェアのR3000命令セット向けのコンパイルにはgcc(GNU C Compiler)を用いた。

回路規模の評価には、Xilinx社のISE Foundationを用いて、Vertexシリーズ向けの論理合成とマッピングを行った。専用ハードウェアと組込みプロセッサの構成をFig. 4に示す。回路規模の評価には、Fig. 4中の「主記憶」は含まれていない。また、「組込みプロセッサ」と「専用ハードウェア」の「主記憶」へのアクセスを制御する「調停回路」は高位合成の対象ではなく、人手で予め設計して用意した。今回用意した「調停回路」は、「組込みプロセッサ」に最大で3個の「専用ハードウェア」を追加接続し、「組込みプロセッサ」と「専用ハードウェア」が同時並行的に「主記憶」にアクセスすることができる仕様になっている。

Table 1に組込みプロセッサとハードウェア関数の回路規模と遅延時間を示す。最大遅延時間については、Verilog HDLで記述されたR3000互換ソフトコアの「組込みプロセッサ」が最も大きく、「調停回路」や「専用ハードウェア」を追加することによるシステム全体での動作クロック速度の低

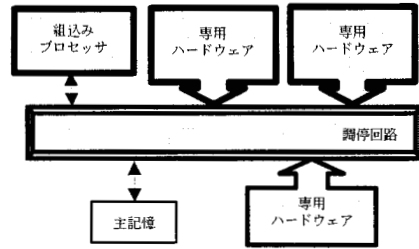


Fig. 4 高速化の評価を行ったハードウェアの構成

下は起こらないことが判明した。また「調停回路」とAES暗号化処理中の「ShiftRows」「SubBytes」「MixColumn」の3つの処理の「専用ハードウェア」の回路規模の総和は、「組込みプロセッサ」の60%弱であった。

Table 2に組込みプロセッサと専用ハードウェアの回路規模とAES暗号化処理に要したクロックサイクル数を示す。実行サイクル数の改善には、「SubBytes」処理の専用ハードウェア化が最も効果があった。これは、専用ハードウェアの内部に256byteの定数テーブルを4つ搭載することで、S-Boxテーブルによる変換処理が並列化され、高速化されたと考えられる。ただし定数テーブルを4つ備えることによる専用ハードウェアの回路規模の増大も最も大きかった。

回路規模の増大と比較しての実行効率の改善については、「MixColumn」処理の専用ハードウェア化による効率の改善が大きいとみなされる。これは「MixColumn」処理で行われる「4byteのデータ4行からなる行列の縦横の並び替え」が、組込みプロセッサによるソフトウェア処理では逐次的に行われるが、「MixColumn」処理の「専用ハードウェア」内部ではこの配列の並び替えを配線のみで並列化できたことによるものと考えられる。

「ShiftRows」「SubBytes」「MixColumn」の3つの処理を専用ハードウェア化した場合は、もとのAES暗号化プログラムの「組込みプロセッサ」による解釈実行と比べて5.0倍弱の実行効率の改善が計れた。この場合の回路規模の増大は、もとの「組込みプロセッサ」の60%弱に収まっており、マルチプロセッサ化と比較してCCAP高位合成システムによるシステム設計の有利性が確認された。

Table 1 組込みプロセッサと専用ハードウェア化したハードウェア関数の回路規模と最大遅延時間

モジュール名	回路規模 (Slices)	最大遅延時間 (ns)
組込みプロセッサ	2825	12.4
調停回路	528	5.71
専用 HW(ShiftRows)	210	1.95
専用 HW(SubBytes)	470	1.97
専用 HW(MixColumn)	367	2.01

Table 2 実行サイクル数の速度向上と回路規模の増大

モジュール名	実行速度		回路規模	
	サイクル数	速度向上比	Slices	規模の比率
組込みプロセッサのみ	26794	1.00	2825	1.00
組込みプロセッサと調停回路	-	-	3353	1.18
プロセッサと調停回路と専用 HW(ShiftRows)	24964	1.07	3563	1.26
プロセッサと調停回路と専用 HW(SubBytes)	21964	1.21	3823	1.35
プロセッサと調停回路と専用 HW(MixColumn)	12047	2.22	3720	1.32
プロセッサと調停回路と専用 HW(ShiftRow)と専用 HW(SubBytes) と専用 HW(MixColumn)	5387	4.97	4400	1.58

## 5 まとめ

本報告では、今後のインターネット上での通信に幅広く用いられる秘密鍵方式の AES 暗号化について、ソフトウェアで実現された機能の一部を取出して専用ハードウェア化することによる実行効率の改善と、回路規模との増大との関係について評価を行った。CCAP 高位合成システムにより、組込みソフトウェア中の特定の処理を容易にハードウェア化し、実行の高速化を行うことが可能であることが確認された。今後は CCAP 高位合成システムが生成する回路の品質のさらなる向上を目指す予定である。

## 6 謝辞

本研究に際し、AES 暗号化プログラムを開示頂いた立命館大学工学部電子情報デザイン学科の山崎勝弘教授に感謝致します。また御討論頂いた関西学院大学工学部石浦研究室の諸氏と京都大学の杉原有理氏に感謝致します。

## 参考文献

1) 西口健一, 石浦菜岐佐, 西村啓成, 神原弘之, 富山宏之, 高務祐哲, 小谷学, ソフトウェア互換ハードウェアを合成する高位合成システム CCAP における変数と関数の扱い, 信学技報, VLD2005-12, 2005

2) 西村啓成, 石浦菜岐佐, 石守祥之, 神原弘之, 富山宏之, 高位合成システム CCAP におけるハードウェア関数からのソフトウェア関数の呼び出し, 情処関西支部大会, C-05, 2006

3) 石守祥之, 奥野裕, 石浦菜岐佐, 西村啓成, 神原弘之, 富山宏之, C プログラムと中間表現の実行比較に基づく高位合成システム CCAP のテスト, 情処関西支部大会, C-06, 2006

4) Daemen, J. and Rijmen, AES Proposal : Rijndael, <http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>

5) National Institute of Standards and Technology (NAIST), Advanced Encryption Standard (AES), FIPS Publication 197, 2001, <http://csrc.nist.gov/encryption/aes/index.html>

6) 梅原直人, 古川達久, 的場督永, 山崎勝弘, 小柳滋, ソフト・マクロ CPU を用いた回路設計とハードウェア/ソフトウェア最適分割法の検討, 情処関西支部大会, C-06, 2005

7) 梅原直人, 古川達久, 的場督永, 山崎勝弘, 小柳滋, ハード/ソフト最適分割を考慮した AES 暗号システムと JPEG エンコーダの設計と検

証, 第4回情報技術フォーラム (FIT 2005),  
C-034, 2005