C-034

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

梅原 直人† 古川 達久† 的場 督永† 山崎 勝弘† 小柳 滋†

立命館大学大学院 理工学研究科†

1．はじめに

本研究では、今後のシステム LSI 設計において、欠かせない技術であるハードウェア/ソフトウェア・コデザインをテーマとし、その中でもハードウェアとソフトウェアの最適な切り分けを見出す分割手法に焦点を当てる。そこで本稿では、ソフト・マクロ CPU である MicroBlaze を用いて、ハードウェアとソフトウェアの分割を考慮してシステムを設計し、FPGA に実装する。その実験過程と結果から対象アプリケーションの特性と、要求に合った最適な切り分けとの関連性を見出し、今後の有用な分割手法の導出を目指す。今回の対象アプリケーションは AES 暗号、JPEG エンコーダである。それらのシステム設計とその評価結果について述べる。

2．ソフト・マクロ CPU による検証システム

ソフト・マクロ CPU とは FPGA ベンダが提供する RTL の HDL 記述として用意されたプロセッサコア IP である。他のロジックと CPU も含めて論理合成をかけ FPGA 上に組み込むことができるため、自由度の高い FPGA の利用が可能となる。本研究では Xilinx 社のソフト・マクロ CPU MicroBlaze を用いる。以下にその特徴を示す。

- 100 Dhrystone MIPS
- ハーバード・アーキテクチャ
- 汎用レジスタ：32bit×32 個
- 命令長：32bit
- 2 アドレッシングモード
- 回路規模：900LUT
- 3 オペラント命令
- 3 段パイプライン
- アドレス空間：4GB
- ビッグエンディアン

このソフト・マクロ CPU を用いて検証システムを構築した。MicroBlaze と自作モジュールは Xilinx 社の提供するツール、EDK6.3i と FoundationISE6.3i を使用して設計し、FPGA ボードに実装した。これらのツールを使用して回路のゲート数や仕様メモリ量を測った。ハード/ソフト・コデザインにおけるソフトウェア部分やシステムの制御部を MicroBlaze に担当させることにより、短期間でのシステム設計や実機での検証を簡単にすることが可能となる。

3．AES 暗号処理システムの設計と検証

3.1 AES Rijndael アルゴリズム

AES は米国商務省国立標準技術局(NIST)によって選定作業が行われた米国政府の次世代標準暗号化方式である。

Rijndael アルゴリズムは入力データを 4 つの変換方式を用いて、一定回数のデータ変換を経ることで暗号化を行う。鍵長は 128,192,256 ビットの 3 種類であり、暗号化に使う鍵も拡張を行うことで暗号の強度を上げている。鍵長によってデータのブロックサイズや暗号化のループ回数が決定される。

図 1 に示すように、データ変換には AddRoundKey 変換、SubBytes 変換、ShiftRows 変換、MixColumns 変換の 4 種類がある。さらに鍵拡張部（KeyExpansion）を加えることで大きく 5 つの機能ブロックに分けられる。

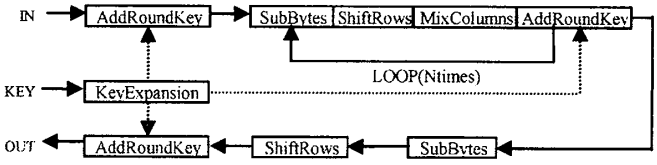

図 1：AES 暗号化フロー

KeyExpansion 変換は暗号化に使用する鍵を拡張する処理である。KeyExpansion 変換はシフト、テーブル参照、排他的論理和といった複雑な処理を経る。しかし、この処理はデータの大きさに関わらず一定なので、データが巨大になれば処理全体に占める割合も小さくなる。

暗号化のための変換部分については、まず AddRoundKey 変換では入力データと鍵データの排他的論理和をとる。SubBytes 変換は本来、数学的で複雑な処理であるが、入力に対して出力が一意に決まるので、テーブル参照で実現している。ShiftRows 変換は入力データを 2 次元配列と見て、各行でシフトを行う。ただしシフト量は行によって異なる。最後の MixColumns 変換は、数学的なガロアフィールド理論に基づいた行列演算をしなければならない。ガロアフィールド理論に基づく演算は非常に難解であるが、行列による積和演算を行うので加算と乗算についてのみ考慮すればよい。加算は単純な排他的論理和で実現できるが、乗算は入力値によって演算方法の場合分けが必要であり、AES 暗号処理システムでは最も難解かつ複雑な部分となっている。

3.2 AES 暗号処理評価システム

設計した AES 暗号処理システムを MEMEC 社の FPGA ボード DS-KIT-MB-S2E6LC-EURO に実装した。図 2 に示すように、MicroBlaze プロセッサには高速の LMB（Local Memory Bus）を介した BlockRAM を接続し、これに命令とデータを格納した。ハードウェア処理に当たる部分や評価用のクロックカウンタなどは、それぞれのモジュールを Verilog-HDL で設計し、GPIO(General Purpose Input/Output)を通じで OPB(On-Chip Peripheral Bus)に接続しており、プログラマブル I/O 転送でデータ転送を行う。

Design and Verification of AES Cryptosystem and JPEG Encoder in terms of Hardware/Software Best Partitioning, Naoto Umehara, Tatsuhisa Furukawa, Tokuei Matoba, Katsuhiro Yamazaki and Shigeru Oyanagi † Graduate school of Science and Engineering, Ritsumeikan University

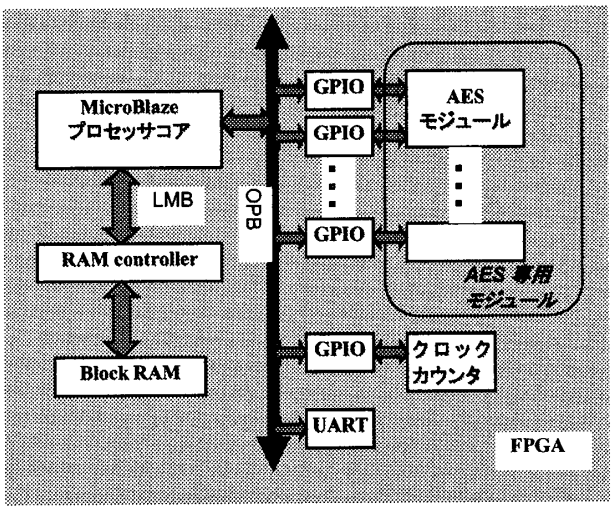

図 2:AES 暗号処理評価システムの構成

3.3 ハード/ソフト分割方法

ハードウェアとソフトウェアの分割探索のために、まずソフトウェアでシステムを実現して、そのモジュールごとの CPU 負荷を計測した。結果は図 3 の通りである。各処理ブロックで必要なクロックサイクル数を計測した結果、SubBytes 変換と MixColumns 変換が特に CPU の負荷が高いことが分かった。これは SubBytes 変換はテーブル参照、MixColumns 変換は条件判定付の演算などを含むことが原因と考えられる。

図 3:AES 暗号処理システムの各モジュールの CPU 負荷

表 1:AES 暗号システムの分割パターン

	ハードウェア処理部	ソフトウェア処理部
S		SubBytes, MixColumns, ShiftRows, AddRoundKey, KeyExpansion, 制御
A	MixColumns	SubBytes, KeyExpansion, ShiftRows, AddRoundKey, 制御
B	SubBytes	MixColumns, AddRoundKey, ShiftRows, KeyExpansion, 制御
C	SubBytes, MixColumns	ShiftRows, AddRoundKey, KeyExpansion, 制御
D	SubBytes, MixColumns, ShiftRows	AddRoundKey, KeyExpansion, 制御
E	SubBytes, MixColumns, AddRoundKey	ShiftRows, KeyExpansion, 制御
F	SubBytes, MixColumns, ShiftRows, AddRoundKey	KeyExpansion, 制御
G	SubBytes, MixColumns, ShiftRows, AddRoundKey, KeyExpansion	制御

この結果に基づいて、AES 暗号処理システムのハードウェアとソフトウェアの分割パターンについて検討した(表 1)。表 1 では上にある分類、つまり"S"に近いほどソフトウェアの占める割合が大きくなり、下に行けばハードウェアの割合が大きくなる。

3.4 実験と考察

表 1 の 8 通りの分割パターンを図 2 の評価システム上に実現し、128 ビットのデータを暗号化してクロックサイクル数、使用メモリ量、回路規模を計測した。各分割パターンの実験において、ハードウェア処理部が AES 暗号専用モジュールに搭載されている。実験結果を図 4 に示す。なお、クロックサイクル数には通信時間も含まれるが、1 回の通信に数クロック程度しか掛からないので、それほど影響はないとみなしている。

	S	A	B	C	D	E	F	G
クロックサイクル数	38075	26959	25859	14743	11313	13979	10549	6650
使用メモリ(Byte)	4387	3700	4024	3572	3496	3812	3736	3108
回路規模（gate）	0	324	4800	5124	5124	5316	5316	10932

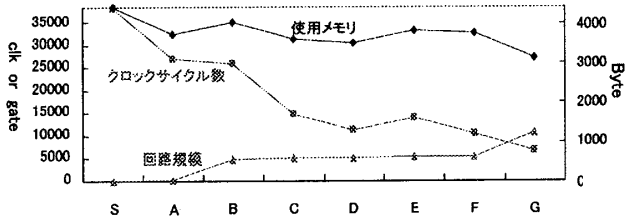

図 4:AES 暗号システムの実験結果

クロックサイクル数は自作のクロックカウンタを MicroBlaze システムに接続して計測した。なお、回路規模は AES 暗号化システムのみの結果であり、MicroBlaze やペリフェラルの I/O などは含まれていない。

S→A や S→B に注目すると、SubBytes 変換と MixColumns 変換をハードウェア化したときのクロックサイクル数の削減が非常に大きいことがわかる。また、メモリ使用量の削減に注目すると MixColumns 変換が多く、次に SubBytes 変換が続く。これは MixColumns 変換が複雑な計算を行うためと、SubBytes 変換がテーブルを用いているのが要因と考えられる。

4. JPEG エンコーダの設計と検証

4.1 JPEG エンコード フロー

JPEG は静止画像の圧縮方式である。JPEG エンコードのアルゴリズムは図 5 に示すように、色空間変換（YUV 変換）・離散コサイン変換（DCT）・量子化・ハフマン符号化の順に 4 つの変換を経てエンコードを行う

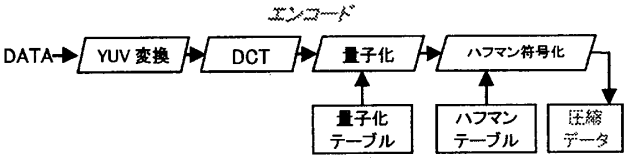

図 5:JPEG エンコード フロー

4.2 JPEG エンコーダ評価システム

設計した JPEG エンコーダを FPGA に実装して、評価するためのシステム構成を図 6 に示す。MicroBlaze が実行す

る命令は BlockRAM に、対象データは画像であるために少なくとも数百 KByte から MByte の容量が必要となると思われるので、外部の SDRAM に格納した。また、JPEG 専用ハードウェアと OPB との間にバッファ用の RAM を用意し、対象データを SDRAM とバッファの間で DMA 転送をさせる。そして、JPEG 専用ハードウェアはバッファ RAM とデータを送受信しながら処理を行う。

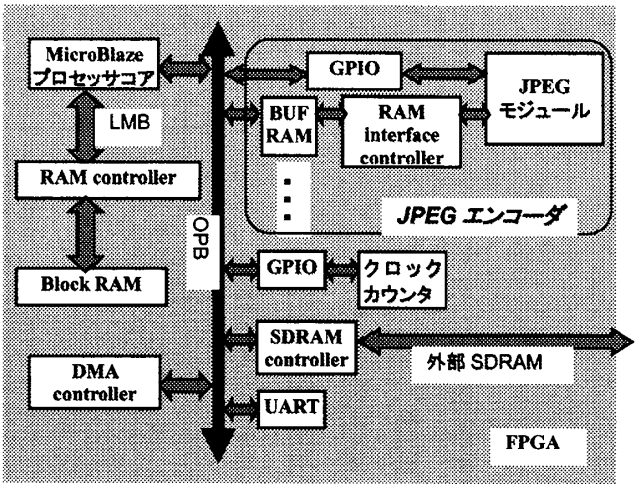

図 6:JPEG エンコーダ評価システムの構成

4.3 ハード/ソフト分割方法

AES 暗号処理システムと同様に、こちらも CPU 負荷を計測して分割パターンを選出する。この JPEG エンコーダのソフトウェアプログラムはハードウェアとの比較のために、浮動小数点数(および演算)は用いていない。JPEG エンコーダの CPU 負荷を図 7 に示す。

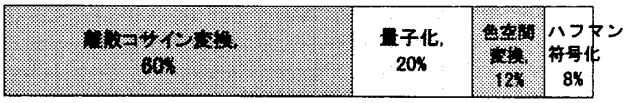

図 7:JPEG エンコーダの各モジュールの CPU 負荷

表 2:JPEG エンコーダの分割パターン

	ハードウェア処理部	ソフトウェア処理部
S		色空間変換, DCT, 量子化, ハフマン符号化, 制御
A	量子化	色空間変換, DCT, ハフマン符号化, 制御
B	色空間変換	DCT, 量子化, ハフマン符号化, 制御
C	DCT	色空間変換, 量子化, ハフマン符号化, 制御
D	DCT, 量子化	色空間変換, ハフマン符号化, 制御
E	色空間変換, DCT	量子化, ハフマン符号化, 制御
F	色空間変換, DCT, 量子化	ハフマン符号化, 制御
G	DCT, 量子化, ハフマン符号化	色空間変換, 制御
H	色空間変換, DCT, 量子化, ハフマン符号化	制御

図 7 より離散コサイン変換が全体の CPU 負荷の 6 割を占めていることが分かった。離散コサイン変換は積和演算を中心として演算を行っており、このことが原因と思われる。この結果を基に選出したハード/ソフト分割パターンを表 2 に示す。

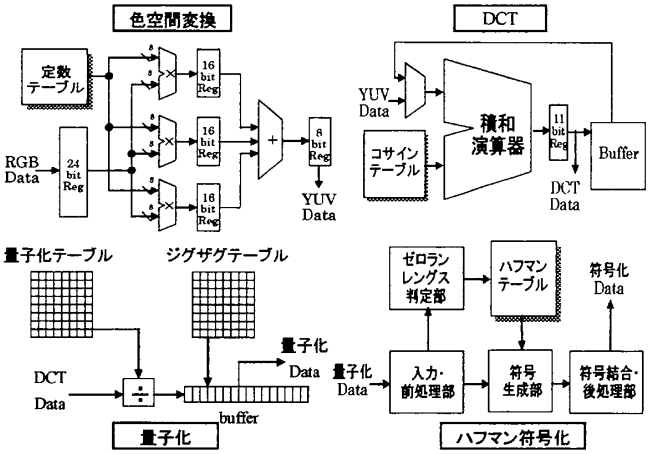

図 8:JPEG エンコーダの各モジュール構成

図 8 に JPEG エンコーダのアーキテクチャの概略として、各モジュールの構成を挙げる。まず色空間変換モジュールは RGB 値に対して 3×3 の定数との積和演算を行うものである。離散コサイン変換(DCT)モジュールは 8×8 の定数(コサインテーブルの出力値)との積和演算を持つ 1 次元 DCT を、2 度行う構成となっている。計算としても回路としても最も重い部分である。つぎに、量子化モジュールは量子化テーブルの定数との除算を行う。ただし、出力はジグザグテーブルという左上から順序をジグザグに割り振ってある出力規則に従ってハフマン符号化モジュールに出力しなければならない。最後にハフマン符号化モジュールは、ハードウェアにしづらいアルゴリズムであり内部が非常に複雑になっている。入力を前処理してゼロランレングス部に渡し、ハフマンテーブルと前処理の結果から符号を生成して、生成された符号を結合して出力する。乗算などの遅い演算は特にないが、状態遷移が複雑になっている。

各モジュールのハードウェア化の予測を行うと、色空間変換モジュールと DCT モジュールは乗算を行っているので、ハードウェア化による速度向上の効果も大きいと見られる。ただし特に DCT において、その分回路規模が大きくなる恐れもある。また量子化は、演算自体は除算のみとシンプルだが除算はソフトウェアの演算でも最も負荷の大きいものの一つなので、ハードウェア化の効果もそれなりに期待できる。ハフマン符号化は制御が複雑すぎる上に、演算自体も簡単なものが多いのでハードウェアにする意味が少ないと見られる。

4.4 実験と考察

表 2 の 9 通りの分割パターンを図 6 の評価システム上に実現し、16×16 画素の RGB 画像(各色は 8 ビット表現)をエンコードしてクロックサイクル数、使用メモリ量、回路規模を計測した。各分割パターンの実験において、ハードウェア処理部が JPEG 暗号専用モジュールに搭載されている。実験結果を図 9 に示す。

	S	A	B	C	D	E	F	G	H
クロックサイクル数	781522	611439	697408	321998	160787	227262	66057	98889	4125
使用メモリ (Byte	8372	6988	6772	6508	5332	4820	3644	3019	2244
回路規模 (gate)	0	4403	4986	35120	39523	40106	44509	49857	54843

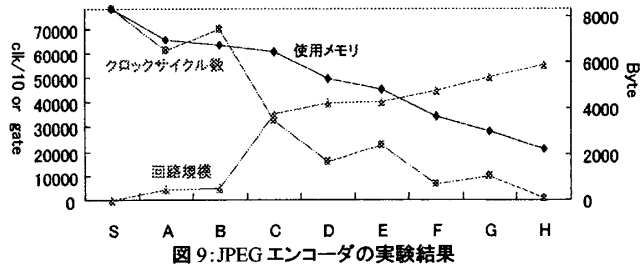

図 9：JPEG エンコーダの実験結果

まず目に付くのは、離散コサイン変換のみをハードウェア化した C の切り分け結果である。S→C の変化に注目すると、ハードウェア規模は大きく増大しているが、クロックサイクル数短縮の効果も大きく出ている。また、演算に用いる領域とコサインテーブルがメモリ上に必要なくなったため、使用メモリ量も削減できている。

次に量子化のみをハードウェア化した S→A と、色空間変換のみをハードウェア化した S→B の変化に注目すると、回路規模は同程度であまりゲートを消費せずに、クロックサイクル数を削減できている。特に量子化のみをハードウェア化した時のクロックサイクル数の短縮量は、先述した離散コサイン変換による短縮量の約 3 分の 1 もある。

一方、色空間変換のみをハードウェア化したときのクロックサイクル数は、あまり削減できていない。よって、離散コサイン変換と量子化をハードウェア化した D が最適であると言える。

5. ハード/ソフト分割パターンの評価式による検証

システム LSI 設計の観点から、得られた実測値から早期に要求に合った最適な切り分けパターンを適切に見つけなければならない。その設計初期の最適な分割へのフィードバックを得るために、実験した分割パターンの優劣を判断する必要がある。そこで、実測値がパターン内でどの程度の優先度を持つかを数値として表す評価式を考案して評価を行った。

$$Priority_{pattern} = \sum_{\substack{Item=gate,\\ memory,\\ clock\cdots}} \left(Weight * \frac{Value_{Worst} - Value}{Value_{Worst} - Value_{Best}} \right)_{Item}$$

Priority ： **あるパターンの優先順位度**
Item ： **項目、本研究ではクロックサイクル数(clock)、回路規模(gate)、使用メモリ量(memory)**
Weight ： **項目(Item)ごとの重み、ただし全項目の Weight の総和は常に 1**
Value ： **実測値、添え字の"Worst"は該当項目するパターンの最大値、"Best"は該当するパターン項目の最小値、添え字のないものは該当するパターンの実測値**

この式は全パターンにおいて、項目毎に最悪の値（最大値）と最良の値（最小値）を抽出して、最大値を 1、最小値を 0 とした。その上で評価値を導出したいパターンの実測値が最大から最小の間で、どのあたりに位置するかを割り出して重み付けをする。その後、対象としている項目の優先順位を、評価式を用いて計算し、任意のパターンでの評価対象項目の総和を取って、パターン全体の優先順位を決定する。

例えば、JPEG エンコーダの切り分け結果に対して、回路規模の制限 43000 Gate 以下、重み付けを「クロックサイクル数：回路規模：使用メモリ=0.8：0.1：0.1」という制約を与えた場合の結果を図 10 に示す。

	S	A	B	C	D	E	F	G	H
Priority	0.1	0.289	0.203	0.539	0.716	0.655	0.832	0.799	0.9

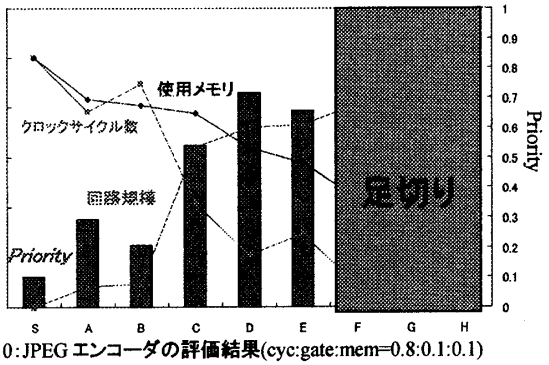

図 10：JPEG エンコーダの評価結果(cyc:gate:mem=0.8:0.1:0.1)

図 10 は速度重視で評価を行っている。図 9 の回路規模の実測値より足切り条件の 43000 Gate を超えている F、G、H の 3 項目は採用枠から除外され、残りの S～A の中で最も評価の高い D がこの場合には最適と考えられる。

このように評価をすることによって、機能ブロックごとの特性や重要度、効率性が判別しやすくなる。この結果は研究の発展に役立つと思われる。

6. おわりに

本研究では、ハード/ソフト協調設計におけるハードウェアとソフトウェアの最適な分割を検証するために、AES 暗号化システムと JPEG エンコーダを設計した。また、Xilinx 社の提供するソフト・マクロ CPU である MicroBlaze を用いて、FPGA ボード上に両システムを実装して、実行速度、使用メモリ量、回路規模の観点から評価した。さらに、制約条件を満たす最適な分割パターンを見出すための評価式を提案した。

今後の課題として、今回の実験結果と評価結果を利用し、命令毎の動的実行数を観測して分析を行い、より精密で設計初期での予測を行うことがあげられる。さらに、MPEG などの異なったアプリケーションに対して本手法を適応し、その有効性を検証したい。

参考文献

[1] 下村高範,安部公輝:暗号アルゴリズム Rijndael のハードウェア実装と評価,情報処理学会研究報告.SLDM, Vol.2003, No.7, pp13-17, 2003.01.
[2] 小野定康,鈴木順司:わかりやすい JPEG/MPEG2 の技術,オーム社,2001.
[3] 貴家仁志:よくわかるディジタル画像処理,CQ 出版,1996.