コンピュータセキュリティ　14−15
（2001. 7. 25）

RijndaelのLSI実装における低消費電力化手法の提案

池 兼次郎† 櫻井 幸一‡ 安浦 寛人¶

† 九州松下電器株式会社
‡¶ 九州大学 システムLSI研究センター

† 〒814-0001 福岡市早良区百道浜2-4-16
‡ 〒812-8581 福岡県福岡市東区箱崎 6-10-1
¶ 〒816-8580 福岡県春日市春日公園6-1

E-mail: ike@msdo.kme.mei.co.jp,sakurai@csce.kyushu-u.ac.jp,yasuura@c.csce.kyushu-u.ac.jp

あらまし

RijndaelはDESの後継暗号方式として昨年選定された128bitブロック暗号である。RijndaelはSPN構造を有しており、特にハードウェア実装を行う上でS-boxは多くのゲート数を必要とし、そのため電力消費も大きくなることが予測される。本稿ではRijndaelのS-boxについて、消費電力を削減するためパストランジスタ論理を用いて実装し、CMOS論理で実装した場合との比較を行った。

キーワード　Rijndael,S-box,消費電力,パストランジスタ

A Proposal of Low Power LSI Implementation for Rijndael

Kenjiro IKE† Kouichi SAKURAI‡ and Hiroto YASUURA¶

† Kyushu Matsushita Electric Co.,Ltd.
‡ ¶System LSI Research Center, Kyushu University

† 4-16,2-Chome Momochihama Sawara-Ku Fukuoka 814-0001 JAPAN
‡ 6-10-1 Hakozaki, Fukuoka 812-8581, JAPAN
¶ 6-1 Kasuga-koen, Kasuga, Fukuoka 816-8580 JAPAN

E-mail: ike@msdo.kme.mei.co.jp,sakurai@csce.kyushu-u.ac.jp,yasuura@c.csce.kyushu-u.ac.jp

Abstract

Rijndael is 128bit symmetric block cipher and it was selected as a new standard block cipher. Rijndael has SPN structure. In hardware implementation, S-box requires a number of gates and much power consumption. In this paper, we propose a design of S-box for Rijndael using pass transister logic for reduction of the number of transister and power consumption.

key words　Rijndael,power consumption,pass transister

1 はじめに

米国National Institute of Standard and Technology(NIST)は，従来のブロック暗号方式のデファクト・スタンダードとして使用されてきたData Encryption Standard(DES)[1]の後継となる新しい暗号方式(Advanced Encryption Standard(AES))として，Rijndael[2]を2000年10月に選定した。Rijndaelのハードウェア実装に関しては，AES選定過程を含め，FPGAによる実装方法[3, 4, 5, 6, 7, 8, 13, 14]とASICによる実装方法[9, 10, 12, 15, 11, 16]について，様々な提案がなされてきた。しかし，面積，スループット等に関する評価は行われてきたものの，電力消費に関する評価はほとんど行われていなかった。今後Rijndaelは携帯機器，ICカード等での実装が考えられ，その場合は低消費電力化を考慮に入れる必要がある。

本稿では，Rijndaelにおける消費電力削減を図るための一つの手段として，暗号化処理の中でゲート数が多く，かつ消費電力が大きいと思われるS-boxをパストランジスタ論理で実現する回路の設計について述べる。S-boxをパストランジスタ論理で実装することにより，CMOS論理で実装した場合と比較してトランジスタ数並びに消費電力の削減が期待できる。

2 Rijndaelアルゴリズム

2.1 アルゴリズム概要

Rijndaelはブロック長，鍵長それぞれ128bit, 192bit, 256bit長を取ることができるが，現在NISTによりアナウンスされているAESに関するFederal Information Processing Standard(FIPS)ドラフト[17]では，ブロック長128bit，鍵長128bit,192bit,256bit長を取ることが記述されている。

Rijndaelは Substitution Permutation Network (SPN)型のブロック暗号方式に分類され，暗号化処理の場合，平文の各ブロックに対して換字(Substitution)，転置(Permutation)などから構成されるラウンド関数を繰り返すことにより暗号文を得る。繰り返しの回数(ラウンド数)は，鍵長により10回,12回,14回と変化する。

Rijndaelで繰り返し実行されるラウンド関数は以下の関数で構成される。

- SubBytes — 1byte毎に換字表(S-box)を用いて換字変換を行う。
- ShiftRows — 1つのブロックを1byte単位の4行4列の行列と見なしたとき，各行に対して固定オフセットの巡回シフトを実行する。
- MixColumns — 上記行列の各列を3次の多項式と見なし，多項式について決まった既約多項式を法とする乗算を行う。
- AddRoundKey — ブロックに対して鍵スケジュール部で生成されるラウンド鍵とのbitwise EXORを取る。

ただし，暗号化処理の最初にはまずAddRoundKeyが実行される。また，最終ラウンドでは上記ラウンド関数のうちMixColumnsは実行されない。

鍵スケジュール部は128bitのラウンド鍵を生成するため，32bit単位で処理が行われる。基本的に32bitのbitwise EXOR演算により生成されるが，128bit中のある32bitに対しては固定オフセット巡回シフト，S-box演算，ラウンド毎に異なる定数とのEXOR演算の順で処理が施される。

2.2 256bit鍵長でのハードウェア実装

Rijndaelの暗号化処理において電力消費が大きい処理を調査するため，256bit 鍵長暗号化処理を行う回路を0.35μmスタンダードセル方式で実装した。

回路設計にあたってデータランダム部はラウンド関数を1ラウンドのみ実装し，実装ラウンド関数をラウンド数回繰り返し実行するIterative Loopingアーキテクチャを採用した。鍵スケジュール部はラウンドの進行と平行してオンザフライで生成する。回路は以下の3つのモジュールで構成される。

- データパス部 — 1ラウンド分のラウンド関数，セレクタ，128bitブロックの実行途中データを保存するレジスタからなる。
- 鍵スケジュール部 — 256bit入力鍵を用いてラウンド鍵をオンザフライで生成する。
- コントロール部 — データパス部，鍵スケジュール部を制御する。

また，上記モジュールを構成する主な関数は以下のように実装した。

- SubBytes — S-boxは拡大体$GF(2^8)$上の乗法逆元を取った後，$GF(2)$上のaffine変換を適用することにより作成されている。しかし，ここではこれらの構造は考慮せず，[17]で記述されているS-boxの表を実現する8入力8出力の関数として実装している。

- MixColumns — 演算をすべてEXORに展開している。

実装にあたっては，まずVerilog-HDLでRTL記述し，論理シミュレータにて動作確認を行った後，論理合成ツールDesignCompilerにてゲートレベルネットリストを生成した。ただし，論理合成は配線容量は無視した条件で行っている。実装結果を表1に示す。

モジュール	ゲート数(2入力NAND換算)
データパス部	11,846
鍵スケジュール部	8,308
コントロール部	111
総計	20,265

表 1: 256bit鍵長暗号化処理回路ゲート数

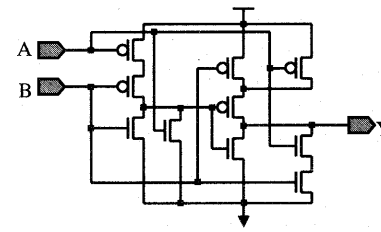
実装回路全体のうち，S-boxはデータパス部，鍵スケジュール部をあわせて13,260ゲート要しており，回路全体の65%近くを占めている。Rijndaelの暗号化処理を行うにあたって，データパス部，鍵スケジュール部を10〜14回のラウンド回数実行することを考慮に入れると，Rijndael全体の電力消費にS-boxの電力消費が占める割合が大きく，S-boxの低消費電力化を図ることがRijndael全体の低消費電力化に効果的であると考えられる。そこで，以下ではS-boxの低消費電力化を図るための手法として，S-boxをパストランジスタ論理で実装する手法について述べる。

3 パストランジスタ論理設計手法

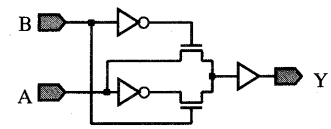
3.1 パストランジスタ論理

現在LSI設計における論理回路実装手法の主流はCMOS論理である。CMOS論理は一般的にpMOSトランジスタからなるプルアップネットワークとnMOSからなるプルダウンネットワークから構成されている。CMOS論理で2入力EXORを実装した例を図1(a)に示す。あらゆる入力信号に対してpMOSネットワークかnMOSネットワークのいずれか一方のみが導通することにより出力の値が決定される。

これに対し，パストランジスタ論理は，MOSトランジスタのソースに信号を入力して，それをゲートに入力した制御信号によって出力するか否かを決定するタイプのMOSトランジスタ回路形式である。パストランジスタ論理で2入力EXORを実装した例を図1(b)に示す。パストランジスタ論理はMOSトランジスタを用いた回路形式であるので，従来のCMOSプロセスでも実装可能である。



(a)CMOS論理による2入力EXORの例



(b)パストランジスタ論理(SPL)による2入力EXORの例

図 1: 2入力EXOR(Y=A ⊕ B)実装例

パストランジスタ論理を用いた論理回路の特徴として以下のことが上げられる[18, 19]。

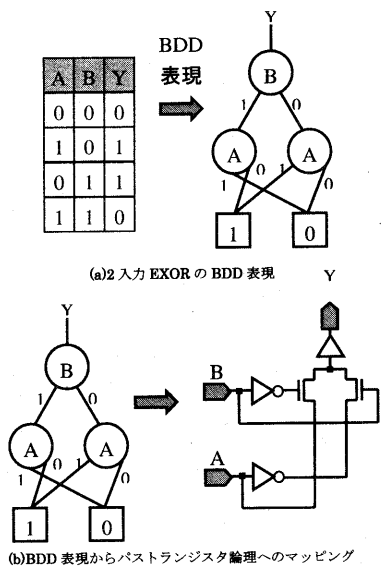
1. CMOS論理と比較して，回路実装に必要とするトランジスタ数が少なくなり，さらにトランジスタが配線の役割をも果たすので，配線数が少なくなる。

2. トランジスタ数が少なくなるため，実装面積が小さくてすみ，配線長が短くなる。

3.2 S-box設計手法

パストランジスタ論理には様々な方式が提案されているが、その中から、低消費電力特性に着目したSPL(Single-rail Pass-transistor Logic)回路方式[20]の使用を検討する。SPLはnMOSトランジスタ2個からなるセレクタを1単位として，複数個のセレクタのネットワークで論理を構成し，出力段にCMOSインバータまたはバッファを配置している構成になっている。図1(b)に示したパストランジスタ論理による2入力EXOR回路はSPL方式による回路である。

SPLを含めパストランジスタ論理を用いた論理回路設計手法として，BDD(Binary Decision Diagram)を用いた手法が提案されている[21]。BDDは2分決定木と呼ばれ，論理関数を表現するための有効な手法である[22]。2入力EXORのBDD表現の例を図2(a)に示す。1番上の節点(根)より始め，丸い節点に記された変数の値が0であれば右の枝を，1であれば左の枝をたどるという操作を繰り返す。最終的にたどり着く四角の接点に記された論理値が，変数の値に対する関数の値になる。論理関数を表現するBDDを得るためのツールとしてはBEM-II[23]やCU Decision Diagram Package (CUDD)[24]などが知られている。



(b)BDD表現からパストランジスタ論理へのマッピング

図 2: パストランジスタ論理設計

冗長な節点と等価な節点がすべて削除されたBDD(既約なBDD)は，変数順序を固定すれば論理関数の標準形になる。変数順序は任意の順序を取ることができるが，同一論理関数であっても変数順序によって節点数は大きく異なる。節点数が最小となる変数順序を求めることは計算困難な問題であることが知られている。

ある論理関数を実現するパストランジスタを得るためには，論理関数を表現する節点数最小のBDD表現を作成し，BDD表現の各節点をnMOSパストランジスタ2個からなるセレクタにマッピングする(図2(b)参照)。

Rijndaelで用いられるS-boxは8入力8出力であるため，現実的な時間で，すべての変数順序(8!=40,320)に対するBDD表現を求めて，その中から節点数最小のBDD表現を得ることができ，S-boxを実現するより実装に適したパストランジスタ論理回路を求めることができる。

筆者らはDESのS-boxに関して上記手法を適用してパストランジスタ論理回路を設計を行ったところ，トランジスタ数を約45%，消費電力を約半分に削減することができた[25]。

4 おわりに

本稿では，Rijdaelの低消費電力化を図るため，Rijndaelの関数において電力消費が大きいと思われるS-boxをパストランジスタ論理で設計するための手法について述べた。現在，パストランジスタ論理で実装したS-boxの消費電力評価を行っており，CMOS論理で実装した場合との比較を行う。また，Rijndaelの論理回路レベル，アーキテクチャレベルでの低消費電力化のための手法についても検討を行う。

謝辞

本研究の一部は東京大学大規模集積システム設計教育研究センター(VDEC)の協力で行われた。

参考文献

[1] National Bureau of Standards, FIPS PUB 46, *Data Encryption Standard*, U.S. Department of Commerce, 1977.

[2] J.Daemen and V.Rijimen, "AES Proposal: Rijndael," http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf, 1998.

[3] V.Fisher, "Realization of the Round 2 AES Candidates using Altera FPGA," The Third AES Candidate Conference (AES3), http://csrc.nist.gov/encryption/aes/round2/conf3/papers/24-vfischer.pdf, 2000.

[4] P.Mroczkowski, "Implementation of the block cipher Rijndael using Altera FPGA," Pub-

lic Comments on AES Candidate Algorithms – Round 2, http://csrc.nist.gov/encryption /aes/round2/comments/20000510-pmroczkowski.pdf, 2000.

[5] A.Dandalis, V.Prasanna, and J.D.P.Rolim, "A Comparative Study of Performance of AES Final Candidates Using FPGAs," The Third AES Candidate Conference(AES3), http://csrc.nist.gov/encryption/aes/round2 /conf3/papers/23-adandalis.pdf, 2000.

[6] N. Weaver and J. Wawrzynek, "A Comparison of the AES Candidates Amenability to FPGA Implementation," The Third AES Candidate Conference (AES3), http://csrc.nist.gov/ encryption/aes/round2/conf3/papers/13-nweaver.pdf, 2000.

[7] A.J.Elbirt, W.Yip, B.Chetwynd, and C.Paar, "An FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists," The Third AES Candidate Conference (AES3), http://csrc.nist.gov/encryption/aes/round2 /conf3/papers/08-aelbirt.pdf, 2000.

[8] K.Gaj, and P.Chodowiec, "Hardware performance of the AES finalists – survey and analysis of results," http://ece.gmu.edu /crypto/AES_survey.pdf, 2000.

[9] B.Weeks, M.Been, T.Rozylowicz, and C.Ficke, "Hardware Performance Simulation of Round 2 Advanced Encryption Standard Algorithm," http://csrc.nist.gov/encryption/aes/round2 /NSA-AESfinalreport.pdf, 2000.

[10] T.Ichikawa, T.Kasuya, and M.Matsui, "Hardware Evaluation of the AES Finalists," The Third AES Candidate Conference (AES3), http://csrc.nist.gov/encryption/aes/round2 /conf3/papers/15-tichikawa.pdf, 2000.

[11] 市川, 時田, 松井, "128ビットブロック暗号のハードウェア実装について(III)," 暗号と情報セキュリティシンポジウム2001(SCIS2001)予稿集, pp.669–674,2001.

[12] H.Kuo and I.Verbauwhede, "Architectural Optimization for a 1.82Gbits/sec VLSI Implementation of the AES Rijndael Algorithm," Proc. of Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), pp.53–67, 2001.

[13] M.McLoone and J.V.McCanny, "High performance Single-Chip FPGA Rijndael Algorithm Implementations," Proc. of Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), pp.68–80, 2001.

[14] V.Fischer and M.Drutarovský, "Two Methods of Rijndael Implementation in Reconfigurable Hardware," Proc. of Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), pp.81–96, 2001.

[15] A.Rudra, P.K.Dubey, C.S.Jutla, V.Kumar, J.R.Rao, and P.Rohatgi, "Efficient Rijndael Encryption Implementation with Composite Field Arithmetic," Proc. of Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), pp.175–188, 2001.

[16] 佐藤, 森岡, 宗藤, "Rijndaelの小型ハードウェア実装," マルチメディア.分散.協調とモバイルシンポジウム2001予稿集,pp.663–668,2001.

[17] National Institute of Standard and Technology, "Draft FIPS for the AES," http://csrc.nist.gov/publications/drafts/dfips-AES.pdf, 2001.

[18] 黒田, 桜井, "低電力, 高速, 小チップ面積と三拍子そろったポストCMOS論理, 普及始まる," 低電力LSIの技術白書—1ミリワットへの挑戦, 日経BP社, 1994.

[19] 瀧, "パストランジスタ論理に関する動向調査報告", DAシンポジウム'97論文集, pp.147–154, 1997.

[20] 瀧, 李, "パストランジスタ論理に基づく低消費電力回路方式と設計事例", 信学論, Vol.J80-A,No.5, pp.1–12, 1997.

[21] 小西, 岸本, 李, 瀧, "パストランジスタ論理SPL用論理設計CAD", DAシンポジウム'96論文集, pp.1–6, 1996.

[22] 石浦, "BDDとは," 情報処理, Vo. 34, No.5, pp.585–592, 1993.

[23] Shin'ichi Minato, "BEM-II: An arithmetic Boolean expression manipulator using BDDs", IEICE Transactions of Fundamentals, Vol.E76-A, No.10, pp.1721-1729, 1993.

[24] http://vlsi.colorado.edu/ fabio/ .

[25] 池, 櫻井, 安浦, "パストランジスタを用いたDES S-boxの実装評価," 2001年電子情報通信学会総合大会論文集 基礎・境界, p.213, 2001.