6N-8　FPGAを用いたAES暗号（Rijndael）のハードウェア化

清家　秀律　　黒川　恭一

防衛大学校情報工学科

1　はじめに

データ暗号化標準（DES）の安全性低下に伴い、アメリカ国立標準技術研究所（NIST）は、次世代暗号標準（AES）の標準化を行い、ベルギーのJoan Daemen氏とVincent Rijmen氏により開発されたRijndaelがAESに決定した。

そこで、本研究では、Rijndaelのblock sizeとkey sizeが可変であるという特徴を活用するために、Reconfigurable Hardwareである FPGAを用いて、Rijndaelのハードウェア化について考察をする。

2　Rijndaelの概要

Rijndaelは、入力されるblock sizeとkey sizeが4×4(128bit)、4×6(192bit)、4×8(256bit)のbyte(8bit)単位の行列として表現される。ラウンド関数と呼ばれる繰り返し処理もbyte単位で行われる。このラウンド関数内部では、AddRoundKey, ByteSub, ShiftRow, MixColumnの4つの変換があり、これらを組み合わせて暗号化及び復号を行う。図1及び図2に暗号化及び復号のフローを示す。

2.1　ラウンド数

ラウンド数(r)は、入力されたblock sizeとkey sizeにより決定される。表1にその回数を示す。

2.2　ByteSub(S-box)変換

ByteSub変換は、最初にGF(2^8)のinverseによる乗算をした後、byte単位で、

$b(x)=(x^7+x^6+x^2+x)+a(x)(x^7+x^6+x^5+x^4+1) \mod x^8+1$

の式によるaffine変換を行う。

2.3　ShiftRow変換

shift offsetは、入力されるblock sizeにより決定され、shiftは、その行列の行単位に行われる。ただし、1行目のshift offsetは0、shiftは循環shift leftである。暗号化及び復号におけるshift offsetを表2に示す。

2.4　MixColumn変換

変換内のColumnの値はGF(2^8)における多項式とみなし、固定した多項式 c(x)(1byte)と入力された多項式a(x)(1byte)との積をx^4+1でmodをとった乗算である。この式は、matrixの乗算として書くことができ、暗号化時に利用する固定した多項式

$c(x)='03'x^3+'01'x^2+'01'x+'02'$ は、

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

となり、複号時に使用する固定した多項式

$c(x)='0B'x^3+'0D'x^2+'09'x+'0E'$ は、

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

となる。

2.5　RoundKeyAddition変換

RoundKeyAddition変換は、RoundKeyとDataBlockとの単純なEXORである。そして、RoundKeyの長さは、Block長に等しい。また、逆変換は、それ自身がinverseである。

3　Reconfigurable Hardware(FPGA)

今回 Hardware化で使用したものは、Xilinx社のFPGA(Field Programable Gate Array) XCV1000である。これは、CLB Slice12288、block RAM4096×32の回路規模を持ち、外部ROMを接続して回路をDownloadしておけば、SelectMAPにより、FPGA内部の回路をその時の状態に応じて書き換える。

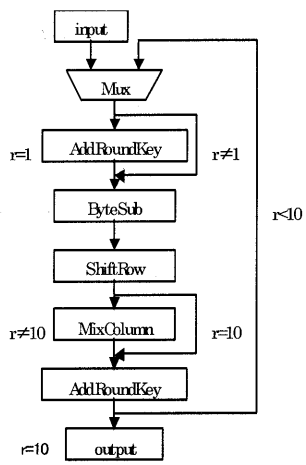

図1　暗号化フロー

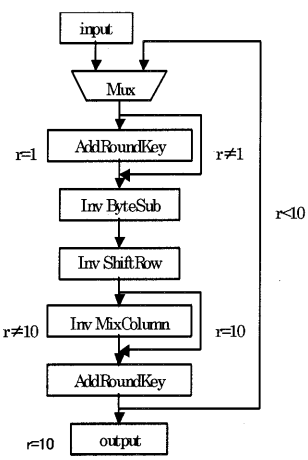

図2　復号フロー

表1　round数

Key size / Block size	128	192	256
128	10	12	14
192	12	12	14
256	14	14	14

表2　暗号化／復号のshift offset

Block size / 行	128	192	256
1行目	0/0	0/0	0/0
2行目	1/3	1/5	1/7
3行目	2/2	2/4	3/5
4行目	3/1	3/3	4/4

4　現段階での構成

本研究では，Rijndael のアルゴリズム及び Hardware における動作を確認するため、以下に示す構成での試作を行った。

(1)　Block size　：128bit 限定

(2)　Key size　：　128bit 限定

(3)　Round 数　：　10round 限定

Block size 及び Key size によって決定

(4)　Key scheduling 部なし

外部において拡大鍵を生成して、内部レジスタに格納して実行

(5)　ByteSub 変換

FPGA の特徴の 1 つである BlockRAM を利用して、入力された 8bit data に対応する値を初期値として RAM 内部に入れた。この際、入力された data を RAM の address の指定に使用し、その内部の data が変換後の値となる。

(6)　MixColumn 変換

ByteSub 変換と同様の方法を用いて BlockRAM を構成した。また、内部における多項式の乗算は GF(2^8)における計算であるため、まず最初、入力された多項式の係数である 2 進数を指数値に変換する。次に、指数部分の加算を行い、出た値に対応する多項式の係数の 2 進数に変換する。最後に、同じ指数部分同士の EXOR をとる。

(7)　RoundKeyAddition 変換

Byte 単位で data block と Roundkey の EXOR をとる。

(8)　コントロール部

図 1、図 2 で示した通り、暗号化及び復号は、変換自体は逆変換になるが、ブロック図全体の流れは同じであるため、コントロール部は、暗号化、復号とも同様の物を使用した。コントロール部のブロック図を図 3 に示す。

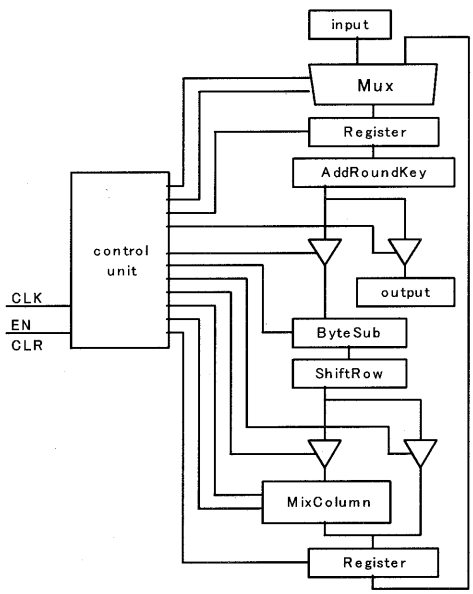

図 3　コントロール部のブロック図

5　Implement の結果

今回、暗号化及び復号を別々のチップに Implement をした。その結果について、表 3 に示す。また、チップ内部の配置配線を図 4，図 5 にそれぞれ示す。

表 3　暗号化、復号における Implement 結果

	暗号化	復号
CLB Slice	1196	1388
Block RAM	32	32
クロック（Mhz）	18.223	17.686

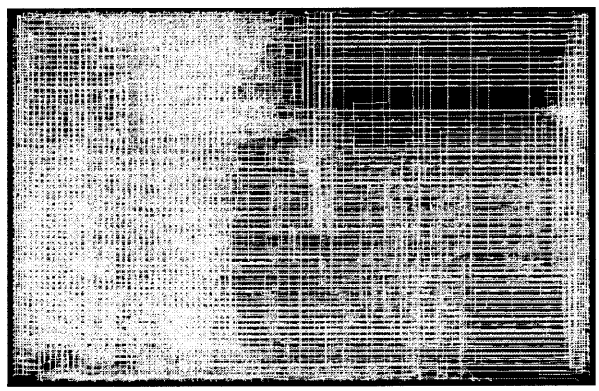

図 4　暗号化回路のフロアプラン

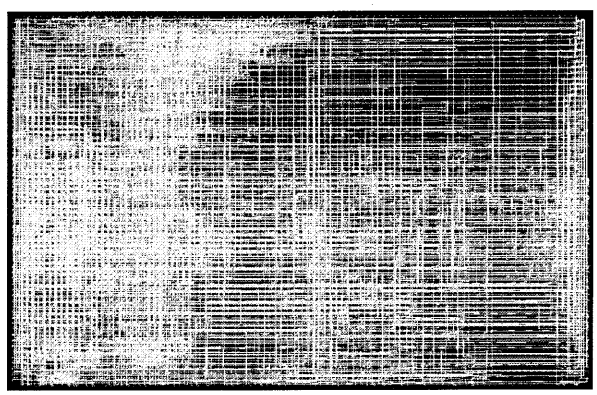

図 5　復号回路のフロアプラン

9　今後の課題

今回は，Rijndael のアルゴリズム及び動作を確認するための試作であった。今後は，Block size 及び Key size の 192bit、256bit への対応、暗号化、復号回路の同一チップへの配置、そして、Key Scheduling 部を内蔵していく。

参考文献

[1]Joan Daemen,Vincent Rijmen：
"AES Proposal：Rijndael",(1999.9.3)
http://www.esat.kuleuven.ac.be/~rijmen/rijndael/