システムＬＳＩ設計技術　94－16
（2000．1．12）

# FPGAベース並列マシンRASHでのDES暗号回路の改良

浅見 廣愛[i], 飯田 全広[ii], 中島 克人[i], 森 伯郎[iii]
i)三菱電機(株)情報技術総合研究所,
ii)三菱電機エンジニアリング(株),iii) 三菱電機(株)鎌倉製作所
〒247-8501 神奈川県鎌倉市大船５丁目１番１号
tel. 0467-41-2526 , e-mail : hiroai@isl.melco.co.jp

あらまし

我々はFPGAを主体とする可変構造型計算機として，FPGAベース並列マシンRASH(Reconfigurable Architecture based on Scalable Hardware)を試作し，DES(Data Encryption Standard)を始めとする秘密鍵暗号の鍵探索処理が高速に行えることを実証した．

今回，RASHでのDESの鍵探索処理の更なる高速化を目的としてFPGA上での回路について検討，改良を行った．DESのアルゴリズムをFPGAに適した回路にすることにより，鍵探索処理性能をFPGA当り4倍まで向上させることができた．1ユニットのRASHシステムで1.39G鍵／秒の性能となる．

キーワード　並列処理, FPGA, 暗号, DES

# Improvement of DES circuit on FPGA-based Parallel Machine "RASH"

Hiroai Asami, Masahiro Iida, Katsuto Nakajima, Hakuro Mori
Mitsubishi Electric Corp. Information Technology R&D Center
Mitsubishi Electric Engineering Co. LTD. Mitsubishi Electric Corp. Kamakura Works
address: 5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan
tel.0467-41-2526 , e-mail : hiroai@isl.melco.co.jp

Abstract

"RASH" is a reconfigurable machine constructed with multiple FPGAs. We showed that exhaustive key search of DES can be performed very fast on RASH.

This time, we improved DES circuit on FPGA. We show that new DES circuit is about four times faster than previous one, and one unit system of RASH can execute key search at a rate of 1.39G key/second.

key words　parallel processing, FPGA, encryption, DES

## 1 はじめに

プログラマブル・ロジック・デバイス(PLD)は早くからその可用性について注目を浴びていた．特に近年，大規模回路用のPLDとして用いられるFPGA(Field Programmable Gate Array)は，最新デバイステクノロジの適用による高速化および大集積化の進展が著しい．

そこで，我々はFPGAを主体とした可変構造型計算機として，FPGAベース並列マシンRASH(Reconfigurable Architecture based on Scalable Hardware)を開発し[1] [2]，DES(Data Encryption Standard)を始めとする秘密鍵暗号の鍵探索処理が高速に行えることを実証した[3] [4]．

今回，RASHでのDESの鍵探索処理の高速化を目的としてFPGAでの回路構成について改良を行った．これにより，大幅な性能の向上が見られたので報告する．

## 2 RASHの構成

以下では，RASHのアーキテクチャの概要について説明する．

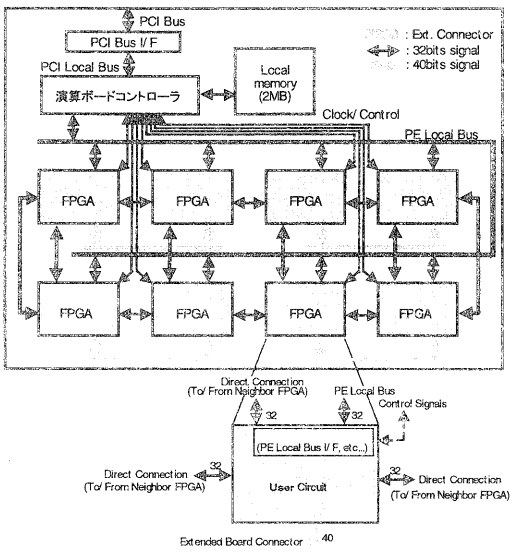


図2.1 EXEボードの構成

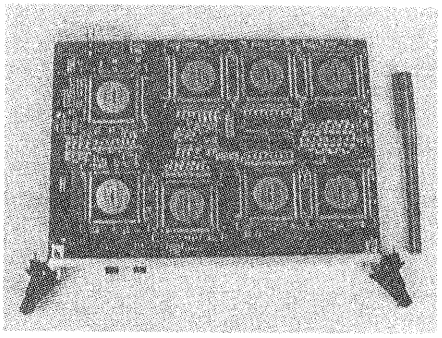

図2.2 EXEボードの外観（FPGA搭載面）

### 2.1 EXEボード

RASHの基本構成要素は，Compact-PCI(Peripheral Component Interconnect)基板上に1石10万ゲート規模相当のSRAMタイプのFPGAを8個搭載した演算ボード（EXE(EXEcution)ボード）である．FPGAにはALTERA社のFLEX10K100A-1(240ピンQFP)を使用した(図2.1,2.2参照)．EXEボードにはPCIバスインタフェース回路と2MBのSRAMからなるローカルメモリが搭載され，バス接続(32bit)された各FPGAとコントローラを介して，これらが接続されている．FPGAの回路情報はローカルメモリを経由してロードされる．ローカルメモリ上に複数種類の回路情報を常駐させることができ，1つのFPGA当り190ms程度で再構成が可能である．

FPGA間はこれとは別に32bitの信号線でメッシュ接続されている．これにより，実現したい機能を2石以上を使って搭載するような場合や，機能ブロック間の処理データをパイプライン的に流すような構成も可能となる．後者のような用途を考慮し，各FPGAには1種類のグローバルクロックの他にもう1種類のローカルクロックが供給される．グローバルクロックおよびローカルクロックは表2.1のように約4.9MHzから60MHzの16種類から選択できるようになっている．

### 2.2 ユニット構成

RASHでは，1つのCompactPCIユニットからなる構成を基本構成（1ユニット）としている．基本構成では，CompactPCIバス上で最大6枚のEXE

表2.1 使用可能なクロック

| | |
|---|---|
| 4.92 | 36.00 |
| 9.68 | 39.95 |
| 14.32 | 42.00 |
| 19.35 | 45.00 |
| 24.58 | 48.22 |
| 28.64 | 50.00 |
| 30.00 | 55.00 |
| 33.15 | 60.00 |

単位：MHz

ボードとそれらを制御するための1枚の汎用プロセッサボード(CPU ボード)が接続されている．また，基本構成には CPU ボード経由で接続される磁気ディスクやネットワークインタフェースも含まれている（図2.3）．

ネットワークはイーサネットとし，これを介して FEP(Front-End Processor) としてのパソコンなどが接続される．また，複数ユニット間もネットワーク接続される．ユニット間の通信量がそれ程多くなくできる場合は，多数のユニットを接続してより大きなシステムを構成することができる．

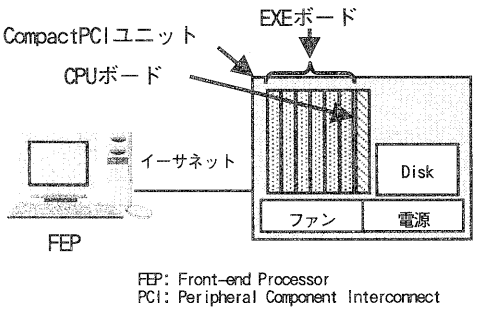


図2.3 RASH のユニット構成

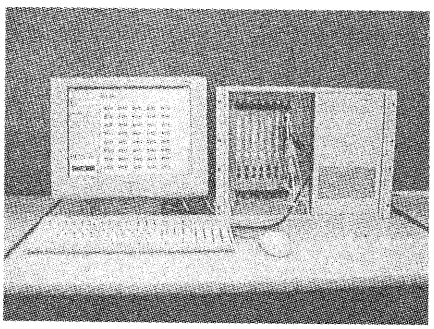

図2.4 RASH ユニットの外観

### 2.3 拡張ボード

各FPGAからは直接40bitずつの信号線が拡張ボードコネクタに接続されている．FPGA での実現が容量および速度の面で非効率な場合や，PCI バス経由では入出力のスループットが不足する場合には，拡張ボードをドータボードとして搭載させる．例えばメモリや I/O デバイスコントローラ等をドータボード上に実現すれば良い．このような実装形態を取ることにより，EXE ボード上でのアーキテクチャ上の制約の最小化と用途別の性能最大化の両立を図れる．

## 3 DES 暗号のアルゴリズム

DES(Data Encryption Standard)は56bit鍵の秘密鍵暗号であり，アメリカを中心として広く使われている．以下では，DES のアルゴリズムについて説明する．

### 3.1 暗号化のアルゴリズム

DES は，長さ 64bit の平文ビット列 x を長さ56bit の鍵ビット列 K で暗号化し，長さ 64bit の暗号文ビット列を出力する．

このアルゴリズムを次に示す（図3.1）．

1. 与えられた平文を初期転置 IP（initial permutation）により変換し，64bit のビット列 $x_0$ を得る．ここで，$x_0$ の上位 32bit を $L_0$，下位 32bit を $R_0$ とする．
2. 次の演算を1段として，これを 16 回繰り返し，$L_i$ と $R_i$ を計算する．
   $L_i = R_{i-1}$
   $R_i = L_{i-1} \oplus f(R_{i-1},K_i)$
   （⊕ は排他的論理和，$1 \leqq i \leqq 16$）
   $K_1 \sim K_{16}$ はそれぞれ長さが48bitの副鍵で，鍵スケジュールにより鍵 K から導き出される．鍵スケジュールと関数 f については後述する．
3. $R_{16}$ を上位 32bit，$L_{16}$ を下位 32bit とするビット列に逆転置 $IP^{-1}$ を行い，暗号文 y を得る．

関数 f は32bitのビット列 A と 48bitのビット列 J を入力とし，32bit のビット列を出力する．この関数の演算について次に示す．

1. 32bitの入力データ A を決められた拡大関数 E（expansion function）により 48bit に拡大する．

2. E(A) ⊕J を計算し，結果を 6bit 単位の 8 個ビット列 $B_1$～$B_8$ に分ける．
3. 各ビット列を 8 個の S-Box $S_1$～$S_8$ で処理し，4bit のビット列 $C_1$～$C_8$ を得る．S-Box は 6bit の入力に対して決められた表をもとに 4bit を出力する関数である．
4. $C_1$～$C_8$ を 1 つの 32bit のビット列として決められた転置 P により並び替える．これにより，得られたビット列が関数 f の出力になる．

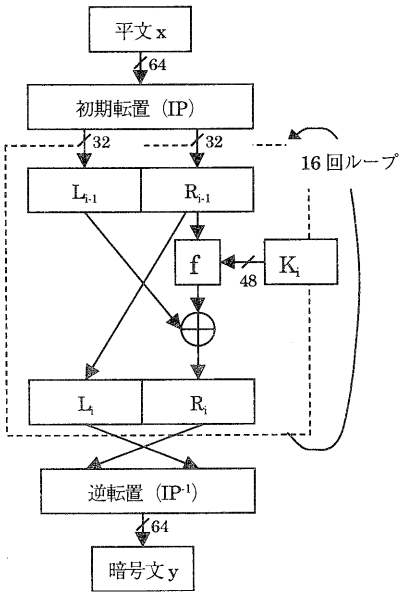


図 3.1 DES の基本アルゴリズム

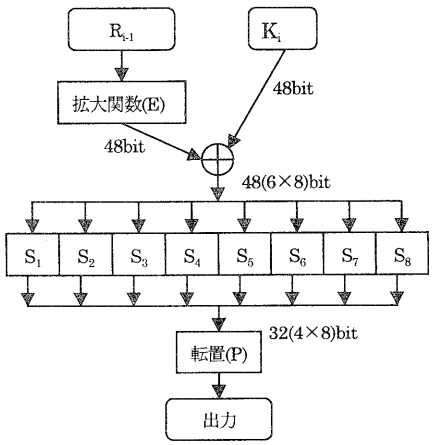


図 3.2 f 関数

### 3.2 鍵スケジュール

鍵スケジュールでは，56bit の鍵 K から 16 個の副鍵 48bit を生成し，上述の暗号化のアルゴリズムでの各段に供給する．DES の鍵スケジュールアルゴリズムについて以下で示す．

1. 56bit の鍵 K を置換 PC-1 で置換する．このビット列の上位 28bit を $c_0$，下位 28bit を $d_0$ とする．
2. 次の演算を 1 段として，これを 16 回繰り返し，$c_i$ と $d_i$ を計算する（$1 \leqq i \leqq 16$）．
   $c_i = ls_i(c_{i-1})$
   $d_i = ls_i(d_{i-1})$
   $ls_i$ は i の値による 1bit もしくは 2bit の左への巡回シフトである．
3. 次のようにして各 $c_id_i$ を置換 PC-2 で置換し 48bit の副鍵 $k_i$ を生成する．
   $k_i = PC\text{-}2(c_id_i)$

## 4 従来の DES 暗号の実装

RASH における FPGA(FLEX10K100A-1)への DES 暗号回路の従来の実装 [3] について図 4.1 に示す．

従来の回路は，3 個の並列動作可能な DES コア，DES コアの制御回路およびバスインターフェース回路からなる．DES コアは 56bit の候補鍵を生成する鍵生成回路（バイナリカウンタ）と，1 個の f 関数を含む回路（図 3.1 の破線部分，以降 f 関数 1 段回路と称する），56bit の鍵から 48bit の副鍵を作成する鍵スケジュール回路で構成されている．DES コアでは f 関数 1 段回路と，鍵スケジュール回路で演算を 16 回繰り返すことにより 1 回の暗号結果を得るようになっている．

RASH での実行時には，CPU ボードから，各 FPGA 上の上述の回路に鍵の探索範囲，即ち初期鍵と探索数が与えられる．鍵生成回路は，初期鍵を順次カウントアップして鍵を作成し，DES コアで暗号化を行う．鍵生成回路が探索範囲の鍵生成を終えると，CPU ボードから新たな探索範囲が与えられ正しい鍵が発見されるまで処理が継続される．従来の実装方式では，各 DES コアで独立して鍵探索を行

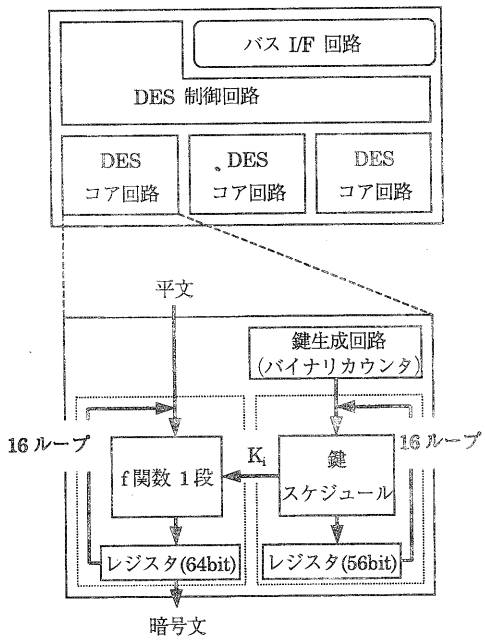


図 4.1 従来の実装方式

えるようにするために, DES コアそれぞれに鍵生成用回路を設けている.

上述の回路を Verilog-HDL で記述し MAX+PLUS II で合成を行い, RASH 上で実際の性能を測定した. この時の各回路の LE の使用数と使用率を MAX+PLUS II 上のフロアプランから見積もったのが表 4.1 である. 表 4.2 にはこの時の暗号化性能も示す. 表 4.1 から f 関数 1 段回路 1 個の LE 使用率が 20%程度, それ以外のバス I/F 回路+DES 制御回路の LE 使用率が 30%程度である. そのため, 1 個の FPGA に搭載する DES コア回路は 3 個となった.

表 4.1 従来実装での LE の使用

| 回路 | 使用 LE 数 | LE 使用率 |
|---|---|---|
| バス I/F 回路 | 89 | 1.8% |
| DES 制御回路 | 1357 | 27.2% |
| f 関数 1 段回路 3 個 | 2814 | 56.4% |
| (f 関数 1 段回路 1 個) | (938) | (18.8%) |
| 全回路 | 4227 | 84% |

表 4.2 従来性能

| 項目 | 内容 |
|---|---|
| DES コア回路 | f 関数 1 段の 16 ループ |
| f 関数の個数 | 3 個／FPGA |
| 動作周波数 | 39.5MHz |
| 鍵探索性能 | 474Mbps／FPGA |
| LE 使用率 | 84% |

## 5 回路構成の改良

DES 暗号の FLEX10K100 デバイスへの実装に関しては, AHDL の記述により f 関数 16 段のパイプラインを構成し (LE 使用率 86%), 周波数 25MHz で動作させたという報告が Hamer らよりなされている [5]. 今回の改良は主に彼らのアイデアに従っている. ただし, Hamer らは FPGA 上に 1 組の f 関数 16 段パイプラインを構成したが, 我々は更に, f 関数 N(<16)段パイプラインを構成しこれを並列化することを検討した.

今回の改良は主に次の 3 点である. これらについて以下で説明する.

1. S-Box の最適化
   S-Box の構成をデバイス (FLEX10K100) に適したものにする.
2. パイプライン化
   パイプラインにより処理と制御を単純化する.
3. 鍵生成回路の単純化
   鍵生成回路を単純化する.

### 5.1 S-Box の最適化

S-Box は f 関数 1 段回路の回路構成のほとんどを占めている. このため, S-Box をデバイス (FLEX10K100) に適した構成にして, 使用 LE 数を縮小し, f 関数 1 段回路の回路規模を縮小できる.

S-Box は 3 章でも述べたように 8 個の 6 入力 4 出力の Look-UpTable (LUT) で構成されている. これは 32 個の 6 入力 1 出力の LUT (6-LUT) とみなせる. これに対し, FLEX10K100 では, 1 つの LE に 4 入力 1 出力の LUT (4-LUT) が 1 つある.

このため, 図 5.1 のように 7 個の 4-LUT を使い, 6-LUT を構成するように, Verilog-HDL の記述を明示的に変更する. ここでは, 最初に 4bit 信号に対して 4 個の 4-LUT を使い, 次に各 LUT での出力を

残りの2bitと3個のLUTを使ってセレクタを構成する．また，3章にあるようにS-Boxでの出力は$L_i$と排他的論理和をとる．このため，最後のLUTを使ってこの演算も行う．

これにより，1つのS-Boxを224LE（=7×32）で構成する．

ちなみに，HamerらはAHDL記述により最終段のLUTに代えて，ANDカスケードチェインを使いセレクタを構成しているが，Verilog-HDL記述ではこの指定は必ずしもうまく行かない．

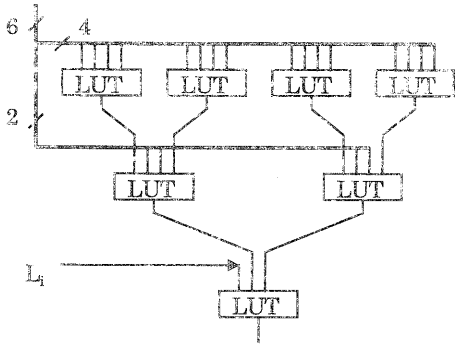


図5.1 S-Boxの最適化

### 5.2 パイプライン化

従来実装では，鍵スケジュールでの演算は16回のループの各ループでのビットシフトと選択的置換による演算である．この演算は，ハードウェア的には，各ループで鍵Kに対して1回選択的転置を行えば副鍵$K_i$を生成することができる．そのため，f関数の16段パイプラインを構成した場合，f関数の各段に図5.2のようにして副鍵を供給できる．

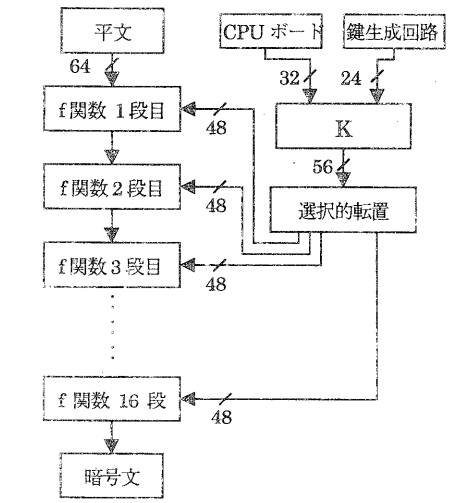


図5.2 副鍵の供給

### 5.3 鍵生成回路の単純化

RASHではCPUボードからのソフトウェア的な制御により，鍵Kの上位32bit程度を固定して各FPGAに供給する．従って，上位32bit程度はレジスタで保持すれば良いが，下位24bit程度で候補鍵を順次生成しなくてはならない．従来実装ではこれをバイナリカウンタで行っていた．図5.2のようなパイプラインを構成した場合，常に段数分の鍵を保持する必要があるため，更に多くのレジスタが必要になる．

このレジスタを減らすため，鍵生成にバイナリカウンタではなく，LFSR（Linear-Feedback Shift Register，線形フィードバックシフトレジスタ）を使用する．LFSRは図5.3のようなM系列の乱数発生器である．L bitのLFSRは最大$2^L$-1の周期をもち，単純な機能追加により$2^L$の周期にすることができる．したがって，探索範囲の大きさと同じ周期のLFSRをバイナリカウンタの代わりに使うことで，指定された探索範囲のすべての鍵を生成することができる．また，図5.3のようにLFSRで発生した鍵をm番目とすると，1つ前のm-1番目の鍵はm番目の鍵を1つシフトしたもの，m-2番目の鍵は2つシフトしたものとなる．したがって，N-1 bitのシフトレジスタを連結することで，現在LFSRで生成した鍵からN-1個前までの鍵，すなわちN段パイプラインに必要なすべての鍵を保持することが可能である．これにより，5.2節にあるようにパイ

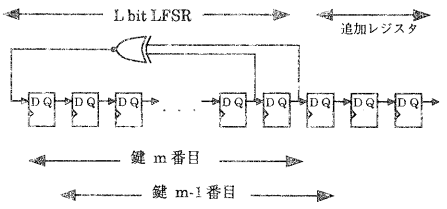


図5.3 LFSR

プラインで処理する場合のレジスタ数の増大を抑制できる.

## 6 回路構成と性能評価

### 6.1 回路の構成

前章に示した改良結果, f関数 1 段回路の回路規模を 350LE 程度に, DES 制御回路の回路規模を 800LE 程度にすることができた. この結果をもとにして, DES 暗号回路のパイプライン化を行い, RASH 上での性能評価を行った. パイプラインは f 関数 1 段 (従来方式) 4 段, 8 段の構成を作成した. 表 6.1 に各構成での性能を, 表 6.2 に LE の使用数と使用率を示す.

表 6.1 各構成での性能

| パイプ段数 | コア回路数 | 動作周波数 | 鍵探索性能/FPGA |
|---|---|---|---|
| 1 段 | 5 | 42MHz | 840Mbps |
| 4 段 | 3 | 39.5MHz | 1.85Gbps |
| 8 段 | 1 | 48.22MHz | 1.50Gbps |

表 6.2 各構成での LE の使用

| パイプ段数 | コア回路数 | LE 使用数 | LE 使用率 |
|---|---|---|---|
| 1 段 | 5 | 3946 | 79% |
| 4 段 | 3 | 4500 | 90% |
| 8 段 | 1 | 3943 | 78% |

### 6.2 性能評価

上述の FPGA での単体性能と DES 暗号専用 LSI や汎用マイクロプロセッサ上で DES 暗号化を行った場合との性能比較を表 6.2 に示す.

今回の改良により, FPGA の単体性能で専用 LSI [8] の 2 分の 1 程度まで性能を向上することができた.

表 6.2 DES 暗号の性能比較

| 対象 | 性能 |
|---|---|
| FPGA (RASH:4 段 4loop,3 回路,39.5MHz) | 1.85Gbps |
| Intel Pentium(300MHz)[6] | 53Mbps |
| DEC $\alpha$ チップ(300MHz)[7] | 137Mbps |
| FPGA (TM-2a:16 段 1 回路,25MHz)[5] | 1.6Gbps |
| DES 暗号 LSI(16 段 2 回路,33MHz)[8] | 4.2Gbps |

## 7 考察とまとめ

本稿では FPGA ベース並列マシン RASH での DES 暗号回路の改良について述べ, FPGA 内の回路並列化と回路のパイプライン化の組合せによる性能評価を RASH 実機上で行った. これにより, RASH での従来性能の 4 倍の性能を得ることが確認できた. 即ち, FPGA 1 個当り 1.85Gbps, EXE ボード 6 枚構成の RASH 基本ユニットで 1.39G 鍵/秒の性能が得られている.

Hamer らは, DES 16 段パイプラインを 1 個の FLEX10K100 デバイスに構成し, 25MHz で動作させ 1.6Gbps の性能を得ている [5]. これに対して, 4 段, 8 段パイプライン構成の回路を FPGA 内に複数搭載して, 動作クロックを向上させてることにより同等かそれ以上の鍵探索性能が得られることがわかった. パイプライン段数をおさえた多数の回路を FPGA に収納する方が FPGA の容量変更に対する柔軟性も高くメリットは大きい.

今後, 更なる DES の暗号解析性能の向上や, 暗号以外の他応用への適用を通じて FPGA の特性を活かした RASH の有用性評価を行う予定である.

### 参考文献


[1] 中島 克人, 森 伯郎, 佐藤 裕幸, 高橋 勝己, 浅見 廣愛, 水上 雄介, 飯田 全広, 新留 勝広, "ＦＰＧＡベース並列マシンＲＡＳＨの概要", 第 58 回情処全国大会, 1H-08, 1999-3.

[2] 浅見 廣愛, 佐藤 裕幸, 飯田 全広, 森 伯郎, 中島 克人, "ＦＰＧＡベース並列マシンＲＡＳＨのシステム機能と構成", 第 58 回情処全国大会, 1H-09, 1999-3.

[3] 飯田 全広, 水上 雄介, 高橋 勝己, 浅見 廣愛, 佐藤 裕幸, "ＦＰＧＡによる並列暗号解析装置の構成(1)－ＤＥＳ暗号等の鍵探索－", 第 58 回情処全国大会, 5N-08, 1999-3.

[4] 高橋 勝己, 飯田 全広, 水上 雄介, 中島 克人, 宮田 裕行, "ＦＰＧＡによる並列暗号解析装置の構成(2)－ＡＳＩＣとの比較－", 第 58 回情処全国大会, 5N-09, 1999-3.

[5] Ivan Hamer, Paul Chow, "DES Cracking on the Transmogrifier 2a", CHES'99(CHES: Workshop on Cryptographic Hardware and Embedded Systems), 1999.

[6] Bruce Schneier, Doug Whiting, "Fast Soft Encryption: Designing Encryption

Algorithms for Optimal Software Speed on the Intel Pentium Processor", Proceedings of 4th International Workshop FSE97, Lecture Notes In Computer Science 1267, Springer Verlag pp.242 –pp.259, 1997.

[7] Eli Biham, "Fast Software Encryption", 4th International Workshop, FSE'97 Proceedings, 1997.

[8] 高橋勝己, 飯田全広, 水上雄介, 山崎弘巳, 宮田裕行, 中島克人, 松本勉, "タイムメモリトレードオフ解読法に基づく暗号強度評価装置の実現性について", 情報処理学会論文誌, Vol.40, No.8, pp.3318-3328, 1999-8.