FPGAベース並列マシンRASHでのDES暗号解析処理の改良

浅見廣愛[†] 飯田全広[††]
中島克人[†] 森伯郎[†]

我々はFPGAを主体とする可変構造型の並列計算機RASH（Reconfigurable Architecture based on Scalable Hardware）を試作し，DES（Data Encryption Standard）をはじめとする秘密鍵暗号の鍵探索処理がこのRASHによって高速に行えることを実証した．今回，RASHでのDESの鍵探索処理のさらなる高速化を目的として，FPGA上での回路について検討と改良を行った．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あたり最大4倍向上することができた．合計48個のFPGAを搭載する1ユニットのRASHシステムでは1.39G鍵/秒の性能となる．

Improvement of DES Key Search on FPGA-based Parallel Machine "RASH"

Hiroaki Asami,[†] Masahiro Iida,[††] Katsuto Nakajima[†] and Hakuro Mori[†]

"RASH" (Reconfigurable Architecture based on Scalable Hardware) is a reconfigurable parallel machine constructed with multiple FPGAs, which can perform exhaustive key search of DES (Data Encryption Standard) at high speed. In this paper, we show how we could obtain about four times speed-up. In order to improve the performance of key search of DES, we have enhanced the DES circuit on the FPGA. With own new DES circuit, one unit system of RASH with 48 FPGAs can execute key search at a rate of 1.39 G key/second.

1. はじめに

プログラマブル・ロジック・デバイス（PLD, Programmable Logic Device）は，早くからその有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り高速化と高集積化が著しい．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）を開発した[1),2)]．

RASHの一応用分野として暗号解析（暗号鍵探索）があげられる．FPGAに暗号回路を実装することにより，汎用のマイクロプロセッサよりも高速な処理が行える．また，FPGAは専用LSIに比べて処理性能は劣るが，解析する暗号を容易に変更できる．さらに，鍵探索処理は完全な並列性を持つため，FPGAの並列化により容易に高速化できる．このため，我々はDES（Data Encryption Standard）の鍵探索処理をRASHで実現し，性能評価を行った[3),4)]．

FPGAは書き換え可能であるため，一度実装した回路であっても，改良して再実装することにより性能を向上させることが可能である．今回，DESの鍵探索処理のさらなる高速化を目的としてFPGAでの回路構成の改良を行い，大幅に性能を向上することができたので報告する．

2. RASHのアーキテクチャ構成

2.1 EXEボード

RASHの基本構成要素は，CompactPCI（Peripheral Component Interconnect）基板上に1チップ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

† 三菱電機株式会社
Mitsubishi Electric Corp.
†† 三菱電機エンジニアリング株式会社
Mitsubishi Electric Engineering Co., Ltd.

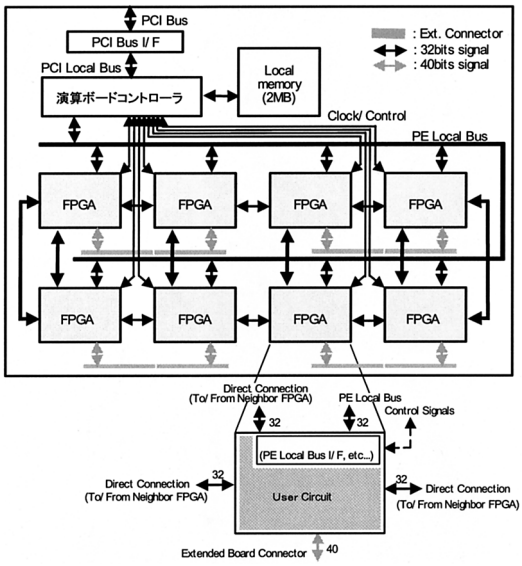

図 1 EXE ボードの構成
Fig. 1 Structure of EXE board.

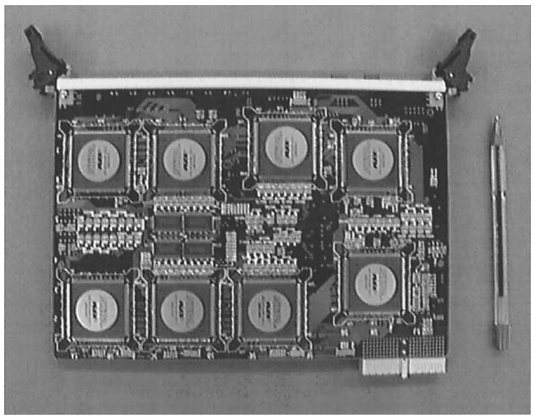

図 2 EXE ボードの外観（FPGA 搭載面）
Fig. 2 The appearance of EXE board (FPGA side).

（240 ピン QFP）を使用した（**図 1**，**図 2** 参照）．EXE ボードには PCI バスインタフェース回路と 2 MB の SRAM からなるローカルメモリが搭載されコントローラに接続されている．また，コントローラは各 FPGA とバス接続（32 bit）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保持することができ，1 つの FPGA あたり 190 ms 程度で再構成が可能である．

FPGA 間はこれとは別に 32 bit 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 チップ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

表 1 使用可能なクロック周波数
Table 1 Available clock frequency.

4.92	24.58	36.00	48.22
9.68	28.64	39.50	50.00
14.32	30.00	42.00	55.00
19.35	33.15	45.00	60.00

単位：MHz

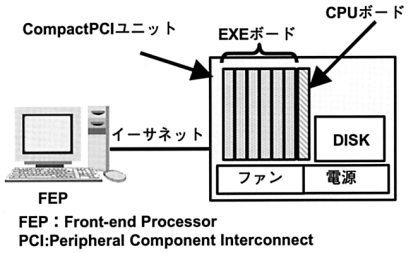

図 3 RASH のユニット構成
Fig. 3 Structure of RASH.

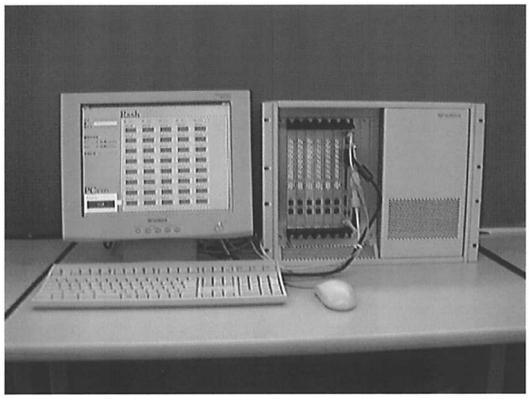

図 4 RASH ユニットの外観
Fig. 4 The appearance of RASH.

可能となる．後者のような用途を考慮し，各 FPGA には 1 種類のローカルクロックのほかにもう 1 種類のグローバ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 1** のように約 4.9 MHz から 60 MHz 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 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 3**，**図 4** 参照）．

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）としてのパソコン等が接続される．また，複数ユニット間もネットワーク接続される．これにより，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可能である．

2.3 拡張ボード

各FPGAからは直接40 bit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．たとえばメモリや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よい．このような実装形態をとることにより，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は56 bit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64 bitの平文ビット列 x を長さ56 bitの鍵ビット列 K で暗号化し，長さ64 bitの暗号文ビット列 y を出力する．

このアルゴリズムを次に示す（**図5**）．

(1) 与えられた平文を初期転置 IP（Initial Permutation）により変換し，64 bitのビット列 x_0 を得る．ここで，x_0 の上位32 bitを L_0，下位32 bit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．

$$L_i = R_{i-1} \quad (1)$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i) \quad (2)$$

（$\oplus$ は排他的論理和，$1 \leq i \leq 16$）

k_1〜k_{16} はそれぞれ長さが48 bit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32 bit，L_{16} を下位32 bit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32 bitのビット列 R_{i-1} と48 bitのビット列 k_i を入力とし，32 bit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（**図6**）．

(1) 32 bitの入力データ R_{i-1} を決められた拡大関数 E（Expansion function）により48 bitに拡大する．

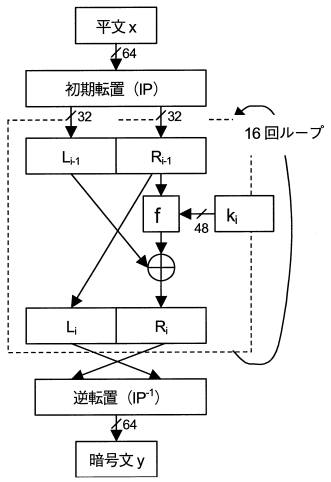

図5 DESの基本アルゴリズム
Fig. 5 The basic DES algorithm.

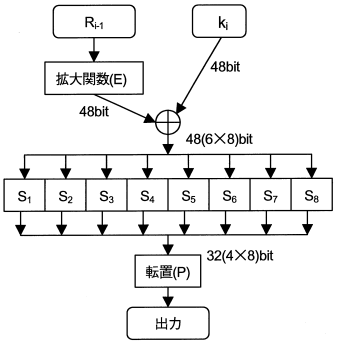

図6 f 関数
Fig. 6 The f function.

(2) $E(R_{i-1}) \oplus k_i$ を計算し，結果を6 bit単位の8個のビット列 B_1〜B_8 に分ける．

(3) 各ビット列 B_1〜B_8 を8個のS-Box S_1〜S_8 で処理し，4 bitのビット列 C_1〜C_8 を得る．S-Boxは6 bitの入力に対して決められた表をもとに4 bitを出力する関数である．

(4) C_1〜C_8 を1つの32 bit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．

3.2 鍵スケジュール

鍵スケジュールでは，56 bitの鍵 K から48 bitの副鍵を16個生成し（k_1〜k_{16}），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．なお，以下で $c_i d_i$ $(0 \leq i \leq 16)$ は56 bitのビット列であり，c_i はそのビット列の上位28 bitを表し，d_i はそのビット列の下位28 bitを表すものとする．

(1) 56 bitの鍵 K を置換し56 bitのビット列 $c_0 d_0$

を得る．$PC-1$ はあらかじめ定められた 56 bit から 56 bit へのビットの置換である．

$$c_0 d_0 = PC - 1(K) \quad (3)$$

(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（$1 \leq i \leq 16$）．ls_i は i の値によってあらかじめ定められた 1 bit もしくは 2 bit の左への巡回シフトである．

$$c_i = ls_i(c_{i-1}) \quad (4)$$

$$d_i = ls_i(d_{i-1}) \quad (5)$$

(3) 56 bit のビット列 $c_i d_i$ を置換し 48 bit の副鍵 k_i を生成する．$PC-2$ はあらかじめ定められた 56 bit から 48 bit へのビットの選択的置換である．

$$k_i = PC - 2(c_i d_i) \quad (6)$$

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 7** 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フェース回路からなる．DES コアは 56 bit 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 5 の破線部分，以降 f 関数 1 段回路と称する），56 bit の鍵から 48 bit 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．

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，すなわち初期鍵と探索数が与えられる．また，鍵探索を行う一組の指定平文と指定暗号文も与えられる．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．生成された暗号文は指定した暗号文と一致判定を行い，一致する場合は正しい鍵が発見されたと判断する．鍵が発見されず鍵生成回路が探索範囲の鍵生成を終えた場合は，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．このときの各回路の LE（Logic Element）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 2** である．**表 3** にこのときの暗号化性能を示す．表 2 から f 関数 1 段回路 1 個の LE 使用率が 20%程度，それ以外のバスインタフェース回路＋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．

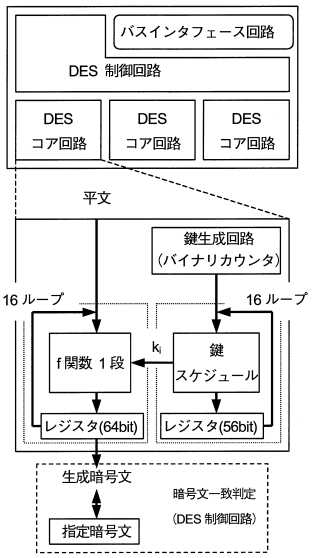

図 7 従来の実装方式
Fig. 7 Previous DES implementation.

表 2 従来実装での LE の使用
Table 2 LEs utilization of previous DES implementation.

回路	使用 LE 数	LE 使用率
バスインタフェース回路	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 3 従来性能
Table 3 Performance of previous DES implementation.

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	39.5 MHz
鍵探索性能	7.41 M 鍵/秒/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 25 MHz で動作させたという報告が Hamer らによりなされている[5]．彼らは主に，S-Box と鍵生成回路を改良することにより回路規模を縮小し，FPGA 上に 1 組の f 関

数 16 段パイプラインを構成した．また，我々の従来の DES 回路は 39.5 MHz で動作している．このため，我々は 16 段パイプラインよりも少ない段数でパイプラインを構成しデバイスに余裕を持たせることにより動作クロックを向上できるのではないかと考えた．そのため，彼らの改良のアイデアを採り入れ，さらに f 関数 N (< 16) 段パイプラインの構成を検討し，性能評価を行うことにした．従来実装と同様に Verilog-HDL で回路を記述し MAX+plus II で合成を行った．

今回の改良の主なものを次にあげる．これらについて以下で説明する．なお，1 と 3 は Hamer らの行った改良点である．

(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) **パイプラインの最適化**
f 関数 N (< 16) 段パイプラインを検討し，デバイスに適した段数構成にする．

(3) **鍵生成回路の単純化**
鍵生成回路を単純化する．

(4) **副鍵の供給の最適化**
副鍵の供給方法をパイプラインの段数に合わせて最適化する．

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，f 関数 1 段回路の使用 LE 数を縮小できる．

S-Box は 3 章でも述べたように 8 個の 6 入力 4 出力の Look-Up Table（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）と見なせる．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 8** 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．この 6-LUT では，6 入力のうちの 4 入力を 1 段目の各 LUT に入力する．また，2 段目と 3 段目の LUT で 4 入力 1 出力のセレクタを構成し，6 入力のうちの残りの 2 入力をセレクト信号として，1 段目の出力をセレクタで選択する．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，3 段目の LUT を使ってこの演算も行う．

これにより，1 つの S-Box を 224 LE（$= 7 \times 32$）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いかない．

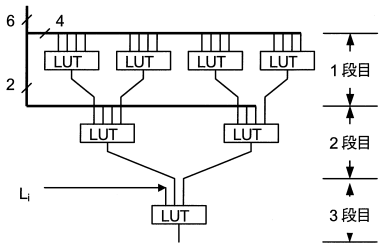

図 8 S-Box の最適化
Fig. 8 Optimization of S-Box.

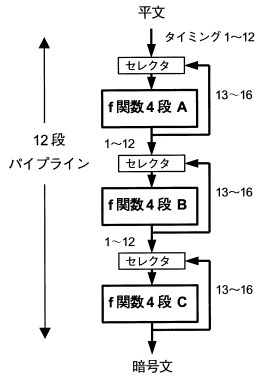

図 9 12 段パイプラインの構成
Fig. 9 12 stages pipeline.

5.2 パイプラインの最適化

f 関数 N (< 16) 段パイプラインを構成することを考える．$N = 2^h$（h：自然数）ならば N 段パイプラインでの処理を $16/N$ 回繰り返すことにより DES の処理が行えるので構成は容易である．$N \neq 2^h$ の場合は**図 9** のようにセレクタと n ($= 2^k$, $N = \alpha n$) 段パイプラインを組み合わせることにより構成できる．

図 9 には一例として 12 段パイプラインを示した．12 段パイプラインは 4 段パイプラインを 3 個直列に接続することにより構成できる．この場合，処理は 16 クロックで 1 サイクルであり，タイミング 1〜12 で平文と鍵が入力され，次のサイクルのタイミング 1〜12 で暗号文が出力される．また，タイミング 13〜16 で各 4 段パイプラインでの出力をセレクタを使い入力に戻す．これにより，タイミング 1〜4 で入力されたデータは 4 段パイプライン C での処理を 2 回，5〜8 で入力されたデータは 4 段パイプライン B を 2 回，9〜12 で入力されたデータは 4 段パイプライン A を 2 回繰り返す．これにより，各データに対して f 関数 16 回の処理が行われる．

FLEX10K100 では LE 数の制約から構成できる 4

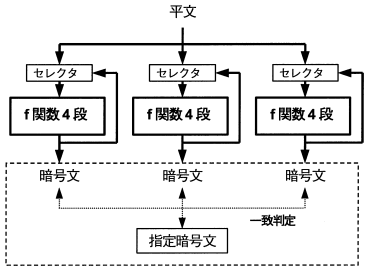

図 10 4 段パイプライン 3 並列の構成
Fig. 10 Structure of 3 parallel-4 stages pipeline.

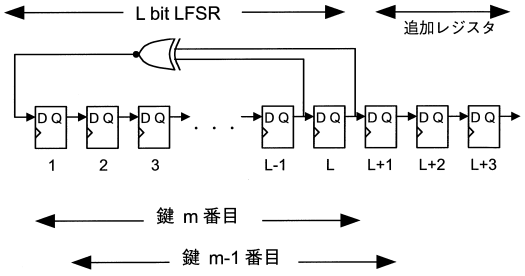

図 11 LFSR
Fig. 11 LFSR.

段パイプラインは 3 個程度である．これを**図 10** のように単純に並列に構成した場合に比べて 12 段パイプラインを構成したほうが，鍵の探索処理を行う場合の鍵の供給や暗号文の一致判定を行う回路が 1 つにできるため，回路の規模を縮小できる．また，以下の 5.4 節に示す副鍵の供給の最適化を適用する場合，4 段パイプラインでは各段に 4 入力 1 出力のセレクタが必要となり 3 並列のため処理は複雑になる．これに対し，12 段パイプラインでは 2 入力 1 出力のセレクタで単純に処理を行うことができるため回路規模を縮小できる．

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 32 bit 程度を固定して各 FPGA に供給する．したがって，上位 32 bit 程度はレジスタで保持すればよいが，下位 24 bit 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．パイプラインを構成した場合，つねに段数分の鍵を保持する必要があるため，さら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 11** のような構成であり，M 系列の乱数発生器として使用できる．L bit の LFSR は最大 $2^L - 1$ の周期を持ち，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 11 において LFSR の左端から L bit 目までを m 番目の鍵として使用した場合，1 つ前の $m-1$ 番目の鍵は 2 bit 目から $(L+1)$ bit 目までとなる．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，パイプラインで処理する場合のレジスタ数を大幅に抑制できる．

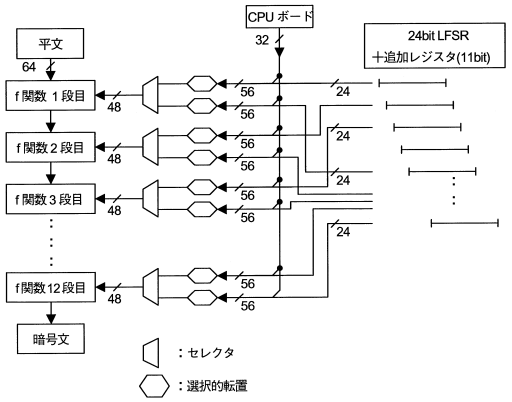

図 12 副鍵の供給
Fig. 12 Provision of sub-key.

5.4 副鍵の供給の最適化

従来の実装では，鍵スケジュールでの副鍵 k_i の生成では 3.2 節に示したビットシフトと選択的置換による演算を行っていた．この k_i を得るための演算は，ハードウェア的には鍵 K のビットを入れ替えるだけの処理である．そのため，f 関数の 16 段パイプラインを構成した場合，各段の f 関数に鍵 K からのビットの選択だけですべての副鍵を供給することにより，LE の消費を抑制できる．さらに，5.3 節の LFSR を用いた鍵生成回路を組み合わせることにより副鍵の供給を単純に行うことができる．

なお，12 段パイプライン等の場合，副鍵の供給方法が多少複雑になる．これは，5.2 節で示したようにタイミング 13〜16 で各 4 段パイプラインでの出力が入力に戻されるため，ある f 関数において入力に戻される前のデータと戻された後のデータでは供給する副鍵が異なるからである．このため，**図 12** のように LFSR に保持された鍵から作られる副鍵を各 4 段パイプラインの出力を入力に戻す前と後で切り替えて各段

の f 関数に供給する．セレクタには LE を使用するが，全体としては従来の実装方法に比べ少ない LE で鍵スケジュールが行えるようになる．

6. 回路構成と性能評価

6.1 回路の構成

5.1 節の S-Box の最適化により f 関数 1 段回路の回路規模を従来に比べて 450 LE 程度削減できた．また，5.2〜5.4 節の改良手法を組み合わせることにより，f 関数 1 段回路の回路規模を 100 LE 程度，DES 制御回路の回路規模を 500 LE 程度削減できた．その結果，f 関数 1 段回路の回路規模は 350 LE 程度に，DES 制御回路の回路規模は 800 LE 程度になった．これをもとにして，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），8 段，12 段の構成を作成し比較した．**表 4** に各構成での性能を，**表 5** に LE の使用数と使用率を示す．また，参考のため 1 チップに収まりきらなかった 16 段パイプラインの MAX+plus II での評価結果も合わせて表 5 に示す．

6.2 性能評価

上述の FPGA での単体性能と，汎用マイクロプロセッサ[6),7)] や DES 暗号専用 LSI[8)] 上で DES 暗号化を行った場合との性能比較を**表 6** に示す．今回の改良により，FPGA の単体性能で専用 LSI の 2 分の 1 程度まで性能を向上させることができた．

表 4 各構成での性能
Table 4 Performance of improved DES implementation.

パイプ段数	コア回路数	動作周波数	鍵探索性能（FPGA あたり）
1 段	5	42 MHz	13.1 M 鍵/秒
8 段	1	48.22 MHz	24.1 M 鍵/秒
12 段	1	39.5 MHz	29.6 M 鍵/秒

表 5 各構成での LE の使用
Table 5 LEs utilization of improved DES implementation.

パイプ段数	コア回路数	LE 使用数	LE 使用率
1 段	5	3946	79%
8 段	1	3943	78%
12 段	1	4506	90%
16 段	1	5447	109%

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 内の回路の N 段パイプライン化による性能評価を RASH 実機上で行った．これにより，FPGA 1 個あたり 29.6 M 鍵/秒，EXE ボード 6 枚構成の RASH 基本ユニットで 1.39 G 鍵/秒の性能が得られ，RASH での従来性能の 4 倍の性能を得ることが確認できた．

また，本稿では $N \neq 2^h$ の場合のパイプライン構成を示した．たとえば，12 段パイプラインを構成した場合，5.4 節に示す副鍵の供給手法と組み合わせることにより，4 段パイプラインを 3 個並列に構成する場合よりも回路規模をおさえることができる．このようなパイプライン構成は，回路規模と動作クロックのトレードオフをとりつつ適切な段数を選択できるため，機能拡張や FPGA の容量変更に対する柔軟性も高くメリットは大きい．

なお，Hamer らは 16 段パイプラインを 1 チップに収めることができたが，これは Altera 社の FPGA 専用の HDL である AHDL で記述したためであろう．我々の Verilog-HDL による記述では 12 段パイプラインで彼らの 16 段パイプラインの LE 数を超えた．1 つには 4 段パイプラインごとに挿入したセレクタ等に LE を消費したためであろうが，ファンアウト調整等においても論理合成の差が大きく出たのではないかと推察される．ちなみに，Hamer らの回路の動作周波数は我々のそれに比べ 63%程度と低い．彼らの論理合成結果と詳細な比較をしない限り断定できないが，クリティカルパスの短縮と使用 LE 数の縮小とが FPGA においてもトレードオフの関係になった可能性がある．

今後，さらに暗号解析性能の向上を図るとともに，他の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．

表 6 DES 暗号の性能比較
Table 6 Comparision of DES Chip.

対象	動作周波数	ゲート規模	性能（1 チップあたり）
FPGA（RASH：12 段，1 回路）	39.5 MHz	100 K	29.6 M 鍵/秒
Intel Pentium[6)]	300 MHz	—	0.83 M 鍵/秒
DEC α チップ[7)]	300 MHz	—	2.14 M 鍵/秒
FPGA（TM-2a：16 段，1 回路）[5)]	25 MHz	100 K	25.0 M 鍵/秒
DES 暗号 LSI（16 段，2 回路）[8)]	33 MHz	150 K	66.0 M 鍵/秒
FPGA（RASH 従来回路：1 段，3 回路）	39.5 MHz	100 K	7.41 M 鍵/秒

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 回情報処理学会全国大会論文集，1H-08 (1999).
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 回情報処理学会全国大会論文集，1H-09 (1999).
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 回情報処理学会全国大会論文集，5N-08 (1999).
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 回情報処理学会全国大会論文集，5N-09 (1999).
5) Hamer, I. and Chow, P.: DES Cracking on the Transmogrifier 2a, *CHES '99 (CHES: Workshop on Cryptographic Hardware and Embedded Systems)*, pp.13–24 (1999).
6) Schneier, B. and Whiting, D.: Fast Soft Encryption: Designing Encryption Algorithms for Optimal Software Speed on the Intel Pentium Processor, *Proc. 4th International Workshop FSE97*, Lecture Notes In Computer Science, Vol.1267, pp.242–259, Springer-Verlag (1997).
7) Biham, E.: 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).

(平成 12 年 1 月 30 日受付)
(平成 12 年 6 月 2 日採録)

浅見 廣愛（正会員）

1970 年生．1994 年早稲田大学理工学部物理学科卒業，1997 年同大学院修士課程修了．同年三菱電機（株）入社．現在，同社情報技術総合研究所にて，FPGA を用いた並列処理装置の研究開発に従事．

飯田 全広（正会員）

1964 年生．1988 年東京電機大学電子工学科卒業．同年三菱電機エンジニアリング（株）入社．オフィスサーバ，DB エンジン等の開発に従事．1995 年同社を休職し，九州工業大学大学院情報工学研究科に入学．1997 年博士前期課程修了．現在同社に復職．

中島 克人（正会員）

1953 年生．1977 年京都大学工学部電気第二工学科卒業，1979 年同大学院修士課程修了．同年三菱電機（株）に入社し，汎用・専用計算機の開発に従事．1982 年より第五世代コンピュータ・プロジェクトに参画し，推論マシンのアーキテクチャ/言語処理系等の研究開発に従事．1993 年よりリアルワールド・コンピューティング（RWC）プロジェクトに参画．現在，同社情報技術総合研究所において，並列・分散処理向けアーキテクチャおよびミドルウェアの研究開発等に従事．最適設計・スケジューリング技術・可変構造型計算機等にも興味を持つ．工学博士．電子情報通信学会，IEEE Computer Society 各会員．

森 伯郎

1942 年生．1969 年京都工芸繊維大学工学部電気工学科卒業．1971 年京都大学大学院修士課程修了．同年三菱電機（株）に入社．以来，通信制御装置，大型汎用計算機，汎用計算機用 VLSI，ディジタルニューロプロセッサーの研究開発に従事．現在，同社鎌倉製作所において，セキュリティシステム・システムエンジニア．電子情報通信学会会員．