社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＳＬＤＭ－113　（１）
2004／1／22

プロセッサにおける配線の再構成可能性の利用について

原田　恭典†　　木村　晋二††　　柳澤　政生†

† 早稲田大学理工学部電子・情報通信学科
〒 169-8555 新宿区大久保 3-4-1
†† 早稲田大学大学院情報生産システム研究科
〒 808-0135 北九州市ひびきの 2-7
E-mail: {harada, kimura, yanagi }@yanagi.comm.waseda.ac.jp

あらまし　本稿では，プロセッサにハードウェア的な再構成可能性を導入する手法を提案する．プロセッサの性能を左右する要素として，演算器の性能の他に，演算器やレジスタを接続する配線構造がある．これまでプロセッサの構成を変更する場合は，演算に着目されることが多かったが，応用によっては配線の影響が大きいものも多い．そこでここでは配線の再構成可能性に着目し，ビットレベルのデータ処理に必要な再構成可能な配線機構を導入する．これは，再構成可能なハードウェアの代表である FPGA などで用いられているスイッチングマトリクスに対応するものである．我々はこのハードウェアをバレルシフタとの類似性からバレルイクスチェンジャと呼ぶ．バレルイクスチェンジャは，n 入力 n 出力で，任意の入出力間の接続ができるものである．これを装備したプロセッサでは，ワードデータのビットの完全置換で，設定のための命令を考慮しても 10 倍以上の高速化が得られる．本稿では，設計による性能評価の他に DES などの応用に対してバレルイクスチェンジャの有効性を示す．
キーワード　応用指向プロセッサ，再構成可能接続，バレルイクスチェンジャ，ビット置換，配線装置

Reconfigurable Interconnection and Its Application to the Bit-Exchange Unit in a Processor

Yasunori HARADA†, Shinji KIMURA††, and Masao YANAGISAWA†

† Dept. of Electronics, Information and Communication Enginneering, Waseda University
3-4-1 Okubo, Shinjukuku, Tokyo 169-8555, Japan
†† Grad. School of IPS, Waseda University
2-7 Hibikino, Kitakyushu, 808-0135 Japan
E-mail: {harada, kimura, yanagi }@yanagi.comm.waseda.ac.jp

Abstract　This paper proposes a reconfigurable interconnect unit for a general (and/or embedded) processor. The performance of a processor depends not only on the operational units but also on the interconnection between registers and operational units. When configuring a procesor architecture, we usually focus on the application specific operational units, but there are not a few applications on which the effect of the interconnection is larger than that of the operational units. So we focus on the reconfigurability in the interconnect architecture and we introduce a reconfigurable interconnect unit for the bit-level data processing. The unit corresponds to a switch-matrix in FPGA and is called as a barrel-exchanger because of the similarity to a barrel-shifter. An n-bit barrel-exchanger has n inputs and n outputs, and any connection between inputs and outputs can be obtained. A processor with a barrel-exchanger gains more than 10 times speed-up for the bit substitution and for DES encription. We also show the area estimation of 8, 16, 32 and 64 bit barrel exchangers.
Key words　Application Specific Processor, Reconfigurable interconnect, Barrel exchanger, Bit substitution, Wriring unit

1. はじめに

近年のプロセッサの進歩により，これまでハードウェアの専用回路が必要と思われていたものがプロセッサで処理可能となりつつある．集積回路技術の発展により利用できる利用可能なハードウェアの規模が拡大したため，専用回路でしか使用できなかった演算器をプロセッサに内蔵できるようになったことが大きな要因である．乗算回路やバレルシフタ，マルチメディア演算回路の内蔵があげられる．

プロセッサの性能を左右する要素として，演算器の性能のみでなく配線構造も重要な要素の一つである．特に応用指向の専用ハードウェアでは，レジスタを含むデータパスの構造による部分が大きい．この配線の自由度の考え方は，再構成可能なハードウェアとして広く用いられている FPGA でも採用されている．FPGA で任意のハードウェアが実現できるのは，各素子の論理が自由に設定できるだけでなくそれらの結線が自由にできることが大きい．

配線構造に関しては，データパスレベルからビットレベルまでのレベルが考えられる．ここでは，ビットレベルの配線構造の可変性に着目して処理の高速化を考える。シフタとして広くプロセッサに搭載されているバレルシフタは，配線のつなぎ換えを用いてビットデータの移動を自由化したハードウェアというとらえ方もできる．これにより 1 ビットずつのシフトの繰り返しで行う従来のハードウェアに比較して，数十倍の高速化を達成している．

バレルシフタよりも柔軟な配線構造をプロセッサに導入することで，種々の応用において高速化が期待できる．例えば映像向けのプロセッサにおいては，映像処理そのものもビット演算を含んでいるが，それだけではなく，映像データを安全に配信するためのスクランブル処理，暗号化処理などもビット処理を多く含む．とくに世界中で広く用いられている共通鍵ブロック暗号 DES[1] やその後継の Rijndael [3], [4] などは Feistel 構造 [5] を含むが，その主たる処理はワード内でのビットの入れ替えである．

そこで本稿では，このようなビット処理を高速に実行するためにビットの完全入れ替えを行うことができるハードウェアについて述べる．これは n 入力 n 出力で，これらの間の接続を任意に指定できるようなモジュールであり，FPGA の論理ブロックを接続するスイッチマトリクスに対応する．すなわち，出力毎にどの入力と接続するかを制御できる．これによりシフト動作，ビットの入れ替えおよびコピーが自由に行える．文献 [2] や最新のプロセッサなどでは，バイト単位の入れ替えを行うようなハードウェアが導入されていることが多いが，それをビット単位で行えるようにしたものは少ない．この点に関し，提案するハードウェアの優位性があると考えられる．

以下，プロセッサ向けに新たに導入した再構成可能な配線機構の構造と，それを制御するための命令セットについて述べ，この機能モジュールの面積評価や性能評価を示す．ワード単位のビットの完全置換を考えると，出力毎の接続を決定するための命令列を考慮しても 10 倍以上の高速化が達成される．このような性能向上は，DES などの実応用でも確認できた．また，この再構成可能な配線機構を実際に設計し，論理合成を行って面積と速度の評価を行った．32 ビットの場合でも 100 MHz で動作することと，扱うビット数が 2 倍になった時に，面積が 3 倍になるという結果を得た．

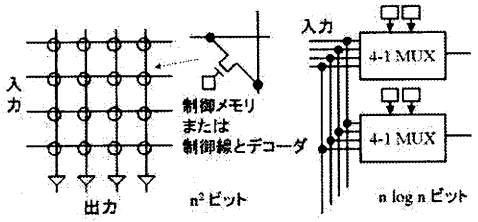

図 1　スイッチングアレイ実現方法

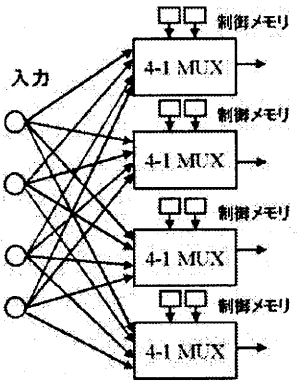

図 2　バレルイクスチェンジャ

2. アーキテクチャモデル

FPGA のクロスポイントスイッチやスイッチマトリックスに見られるスイッチングアレイをプロセッサに導入することにより，ビット毎のシフトおよび置換処理の高速化を実現できると考えられる．

2.1 バレルイクスチェンジャ

スイッチングアレイ実現方法として，図 1 に示した従来の FPGA に見られる SRAM 方式と出力の選択をマルチプレクサで行うセレクタ方式が考えられる．

シフト機能を有する既存のハードウェアとして，バレルシフタが広く用いられている．バレルシフタは入出力線の接続をマルチプレクサにより変更することで，配線の再構成を実現していると考えることが出来る．そこでセレクタ方式での実現を考えた場合，バレルシフタを拡張することにより実現できると考えられる．図 2 に示すようなバレルシフタの拡張回路であるバレルイクスチェンジャの実現を目指している．これによりビットのシフトのみではなく，任意のビットの置換やコピーが可能となる．

2.2 既存ハードウェアとバレルイクスチェンジャ

既存のハードウェアであるバレルシフタの各マルチプレクサ

は，共通のメモリによって制御される．任意のビット数のシフトが可能であり，n 入力 n 出力の場合マルチプレクサの制御メモリは log(n)+1 ビット必要となる．制御メモリが共通であるため，1 命令で設定を行うことが出来る．

バレルイクスチャンジャもバレルシフタ同様，マルチプレクサにより配線の再構成を実現している．各マルチプレクサは個々のメモリによって制御を行う．バレルイクスチェンジャにより，出力ビットごとにどの入力か選択可能となる．n 入力 n 出力の場合各マルチプレクサの制御メモリは log(n) ビット必要となり，制御には合計 n log(n) ビット必要となる．制御メモリの設定には複数命令を要する．

2.3 命令セットについて

プロセッサにおける入力データの全置換処理の実行方法を述べる．プロセッサとしてデータ幅 32 ビット，1 命令を 1 クロックで実行できるようなパイプラインプロセッサを想定する．命令長 32 ビットの内容は，命令コードが 7 ビット，データ領域を 25 ビットと定義する．

3. バレルイクスチェンジャとビットの置換

3.1 従来の 32 ビットデータ置換処理

バレルイクスチェンジャを持たないプロセッサにおける，32 ビットの置換処理を述べる．この場合置換処理を行うには，ビット毎に以下の 3 手順を行う必要がある．

（1） マスク処理を行い該当ビットのデータを抽出

（2） 該当データのシフト

（3） シフト後のデータを出力データと OR 演算

32 ビットデータの置換処理を行う場合，合計で 32 × 3 の 96 命令要する．

3.2 ビット置換アルゴリズム

3.2.1 32 ビットバレルイクスチェンジャ

32 ビット入力 32 ビット出力のバレルイクスチェンジャによる置換処理を考える．各マルチプレクサの制御メモリは，5 ビット (log32) のデータで設定することが可能である．1 命令につき 25 ビットのデータ領域を利用できるので，1 命令につき制御メモリ 5 個分のデータを設定することが可能となる．32 出力分の制御メモリの設定には，7 命令必要となる．置換処理全体としては，イクスチェンジャによるデータの入れ替え命令を加えた 8 命令で処理を行うことが出来る．

3.2.2 16 ビットバレルイクスチェンジャ

16 ビット入力 16 ビット出力のバレルイクスチェンジャによる 16 ビットデータの置換処理を考える．各マルチプレクサの制御メモリは 4 ビット (log16) のデータで設定できるため，1 命令で 6 個分の制御メモリを設定可能である．計 16 個のマルチプレクサの設定には，3 命令必要となる．置換処理全体は 4 命令で処理することが出来る．

次に 16 ビットバレルイクスチェンジャによる，32 ビットデータ処理を考える．データの置換処理は図 3 のアルゴリズムに従って行われる．処理には合計で 27 命令を要する．各入力ビット幅のバレルイクスチェンジャの置換にかかる命令数を，表 1 に示す．

入力: 置換対象のデータ
出力: 置換処理後のデータ

Step 1. シフトにより上位データ抽出 (1 命令)
Step 2. 下位データの抽出 (1 命令)
Step 3. 上位から下位にへ移動するデータをバレルイクスチェンジャによってまとめる (4 命令)
Step 4. Step 3. と同様に下位から上位へ移動するデータをまとめる (4 命令)
Step 5. 上位のデータと上位に移動するデータのマージ (3 命令)
Step 6. Step 5. と同様に下位のデータと下位に移動するデータのマージ (4 命令)
Step 7. マージ後の上位データの入れ替え (4 命令)
Step 8. マージ後の下位データの入れ替え (4 命令)
Step 9. 上位データのシフト及び下位データとのマージ (2 命令)

図 3 16 ビットバレル X c h によるデータ置換アルゴリズム

表 1 ビット置換に必要な命令数

データ長	32bitCPU のみ	CPU & 32bitXch	CPU & 16bitXch	CPU & 8bitXch
32bit	96 命令	8 命令	27 命令	46 命令
16bit	48 命令	4 命令	4 命令	19 命令
8bit	24 命令	3 命令	2 命令	2 命令

4. 命令セット

4.1 バレルイクスチェンジャ実現のための命令拡張

バレルイクスチェンジャ実現に際し，従来の命令にバレルイクスチェンジャ専用命令を追加する必要がある．専用命令として，以下の命令が必要となる．

（1） 初期化および制御メモリへのセット開始命令

（2） 制御メモリへの順次書き込み命令

（3） イクスチェンジャによってビット置換を行う命令

初期化および制御メモリへのセット開始命令とは，制御メモリへのセットの最初の命令を指す．2 の書き込み命令とは命令コードのみ異なる．制御メモリへの順次書き込み命令は，各出力の制御メモリへ順番に書き込む．イクスチェンジャによってビット置換を行う命令によって，レジスタからデータを読み出した後，イクスチェンジャを経由させてレジスタへ書き込みが行われる．

4.2 バレルイクスチェンジャ専用命令

バレルイクスチェンジャの設計にあたり追加する専用命令として，START 命令，DATA 命令，D-END 命令，Xch 命令，Change 命令の 5 つを定義した．各命令は前節 4.1 で述べた 3 種類に分類することが出来る．START 命令は 1 の初期化および制御メモリへのセット開始命令，DATA 命令および D-END 命令は 2 の制御メモリへの順次書き込み命令，Xch 命令および Change 命令は 3 のイクスチェンジャによってビット置換を行う命令に分類することが出来る．各命令ともに 1 クロックで処理を完了することが出来る．各命令の構成を図 4 に示す．

START 命令は内部カウンタ値の初期化およびメモリへの最初の書き込みを行う．命令の構造は命令コードと制御メモリ 5 個分の 25 ビットを持つ，あとの DATA 命令とは命令コードの

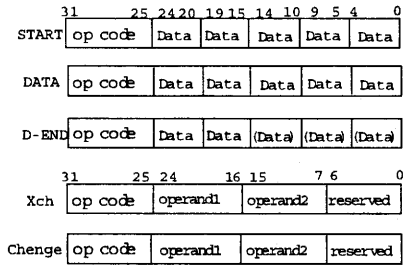

図 4 命令構成

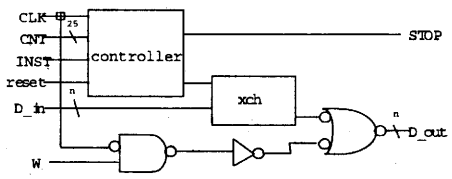

図 5 バレルイクスチェンジャ

表 2 各バレルイクスチェンジャの面積比較

面積 [mm^2]	8bitXch	16bitXch	32bitXch	64bitXch
controller	0.012582	0.003856	0.082764	0.188723
xch	0.007720	0.037600	0.138592	0.541824
total	0.0212670	0.074382	0.224870	0.737452

入力: 64 ビット長の平分, 64 ビット長秘密鍵
出力: 64 ビット長の暗号文

Step 1. 初期設定
Step 2. ビットスライス
Step 3. 平文に対する初期置換 IP
Step 4. 鍵に対する初期置換 PC1
Step 5. 鍵に対する巡回シフト
Step 6. 内部鍵 K の生成処理 PC2
Step 7. f 関数の生成処理
Step 8. 次段データの設定
Step 9. Step 5. から Step 8. の処理を 16 回繰り返す
Step 10. 暗号文に対する逆置換 IP^{-1}
Step 11. 出力処理

図 6 DES 暗号化アルゴリズム

み異なる.

DATA 命令は内部カウンタ値に従い, 5 個の対応するメモリに並列書き込みを行う. 命令構造は DATA 命令と比較して, 命令コード部のみ異なる.

D-END 命令は DATA 命令とほぼ同じ処理を行う. しかし D-END 命令が入ってきた場合, メモリへの書き込みはこの命令で完了したと見なされる. 32bitXch で 16 ビットデータを扱う場合など, 3 命令目にこの命令を入れることにより残りの 16 ビット分の出力指定をせずに済む.

Xch 命令はデータを該当するレジスタから取り出し, バレルイクスチェンジャにかける. 得られたデータをまた指定されたデータへ格納する. データ構造としては, 命令コードと格納先レジスタ, 読み出し先のレジスタの情報をもつ. 格納先レジスタおよび読み出し先のレジスタは各々operand1, operand2 で指定される.

最後の Change 命令は先に実行された Xch 命令と同様の置換を異なる入力に対して行う場合に用いる. この命令により, メモリへの書き込み命令のオーバーヘッドを削減することが可能となる.

5. ハードウェア実装

5.1 構成モジュール

図 5 の様な構成をもつバレルイクスチェンジャの設計を行った. 実装バレルイクスチェンジャである BXCH は, 内部モジュール

・controller
・xch

によって構成される. x c h 部は図 2 に示したようにマルチプレクサが入力ビット数分並列に配置されており, 各マルチプレクサには全ての入力信号が接続されている. 制御信号は, controller 部によって設定された制御メモリから与えられる.

controller 部はメモリ機能と状態遷移による制御部をまとめたものである. 内部状態は入力命令によって変化し, 入力命令および内部カウンタ値に従って対応するメモリへ書き込みを行う. メモリには一時メモリと制御メモリが存在し, 内部の書き込み信号がアサートされた場合のみ一時メモリの内容を制御メモリに書き込を行う.

5.2 論理合成結果

入力が 64 ビットから 8 ビットまでの各々の場合のバレルイクスチェンジャを VHDL により記述し, 面積見積もりを行った. 見積もりに際して, 論理合成ツール Synopsys Design Compiler, VDEC Rohm 0.35 μ m プロセス向けライブラリを用いた. 各々の面積を表 2 に示す. 動作周波数は 100MHz で合成した。controlller, Xch は各部分の面積を, BXCH は全体の面積を表す. 表からわかるように, ビット数が 2 倍になると面積は約 3 倍になっている.

6. バレルイクスチェンジャを用いた DES 暗号化処理

6.1 DES 暗号化処理

バレルイクスチェンジャの評価対象として, DES 暗号化アルゴリズムを用いた. DES 暗号化処理は、図 6 に示す手順によって行われる. f 関数生成処理には拡大置換 E、内部鍵 K との XOR, 8 分割, S ボックスによる置換, 置換 P の各処理が含まれる.

DES 暗号化アルゴリズムを MIPS RS2000 アセンブリ命令に則したプログラミングを行い, 処理にかかる命令数を計測したところ表 3 の様になった. Step 5. から Step 8. の実行命令数は 16 回分をまとめたものである. バレルイクスチェンジャを用いることで改善できると考えられる処理部も併せて表 3 に

表 3 DES 暗号化処理に必要な命令数

	全命令	改善可能部
初期設定	146	—
ビットスライス	1604	1604
初期置換 IP	1607	1607
初期置換 PC1	1408	1408
巡回シフト	24172	—
生成処理 PC2	18512	18512
f 関数部	71876	63860
次段データの設定	14464	14464
逆置換 IP^{-1}	1606	1606
出力処理	1411	—
合計	136806	101457
割合	—	74.16 %

記載した．f 関数における S ボックスはデータ内の置換処理よりはメモリからのデータ読み出し処理の色合いが強く，バレルイクスチェンジャによる置換効果が薄い．S ボックスに対して，バレルイクスチェンジャの有効な利用法を検討中である．またプロセッサにおいてはバレルシフタが一般的に広く用いられており，バレルシフタとにた構造をもつバレルイクスチェンジャを用いた場合も巡回シフトに関して同程度の性能を発揮できると考える．そこで，S ボックスおよび巡回シフトを改善可能部から除外した．改善可能箇所が命令全体の約 74 %にのぼることから，バレルイクスチェンジャが有効であると考えられる．

6.2 実行命令数

6.2.1 従来の DES 暗号化処理

バレルイクスチェンジャを用いずに DES 暗号化処理を行った場合について考える．バレルイクスチェンジャとの比較のため，改善可能部のみ考慮する．初期置換 IP を行った場合，64 × 3 = 192 命令必要となる．初期置換 PC1 は 56 × 3 = 162 命令かかる．生成処理 PC2 には 48 × 3 = 144 命令かかる．拡大置換 E は 48 × 3 = 144 命令かかる．置換 P は 32 × 3 = 96 命令必要である．逆置換 IP^{-1} は 64 × 3 = 192 命令必要となる．

6.2.2 32 ビット入力バレルイクスチェンジャ

32 ビットバレルイクスチェンジャを用いた DES 暗号化処理を示す．初期置換 IP および逆置換 IP^{-1} は入力がともに 64 ビットであるため，32 ビットづつ 3．3 の手順に従い分けて処理する必要がある．上位と下位 32 ビットの抽出に 2 命令必要となる．上位ビット内の入れ替えに 8 命令必要となる．下位ビット内入れ替えは，Change 命令を用いることにより 1 命令で済む．上位の移動とマージに 3 命令，下位の移動とマージに 4 命令かかる．マージ後の上位ビットの入れ替えに 8 命令，下位ビットの入れ替えには 1 命令かかる．最後の上位データのシフトとマージには 2 命令必要なる．初期置換 IP および逆置換 IP^{-1} はともに 27 命令で処理される．

他の処理も同様に，初期置換 PC1 は 27 命令，生成処理 PC2 は 20 命令，置換 P は 8 命令，拡大置換 E は 14 命令必要となる．

6.3 提案手法による処理の高速化

各入力ビット数のバレルイクスチェンジャを用いる場合と用いない場合の各処理に必要な命令数を，表 4 に示す．バレルイクスチェンジャを用いることにより，命令数を約 1/10 程度まで削減できると考えられる．表 3 における改善有効部の命令数も同様に 1/11.24 になるとすれば，全体の命令数は 46755 命令まで削減可能である．DES アルゴリズム全体では，命令数を約 1/3 にまで削減することができる．tripleDES を用いた場合でも，命令数を約 1/3 に削減できると考えられる．

表 4 実行命令数の比較

	32bitXch	16bitXch	8bitXch	Xch なし
初期置換 IP	27	78	480	196
初期置換 PC1	27	78	280	162
生成処理 PC2	20 × 16	44 × 16	204 × 16	144 × 16
拡大置換 E	14 × 16	44 × 16	204 × 16	144 × 16
置換 P	8 × 16	27 × 16	46 × 16	96 × 16
逆置換 IP^{-1}	27	78	480	196
合計	752	2174	8504	6698
比率	11.24	32.45	126.5	100

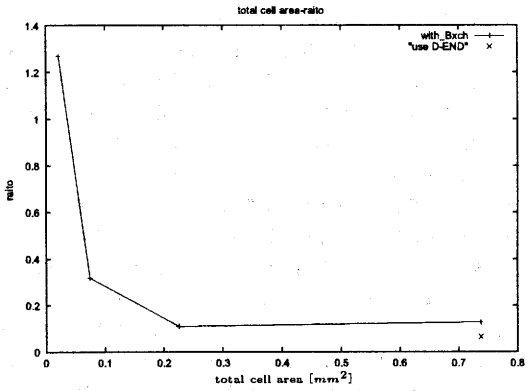

図 7 面積／実行命令の削減比

7. 性能評価

表 2 に示した各バレルイクスチェンジャの面積比較および表 4 に示した実行命令数の比較をもとに，“面積／実行命令の削減比”のグラフを求めた (図 7)．面積と実行速度がトレードオフの関係にあることが分かる．8 ビット入力のバレルイクスチェンジャを除いては，バレルイクスチェンジャ導入により約 1/3〜1/10 程度まで命令数の削減が実現された．8 ビット入力バレルイクスチェンジャに関しては，分割の細分化に伴うマスク抽出およびマージ処理の増加によりバレルイクスチェンジャ導入による削減効果が打ち消されている．

また図 7 の 64 ビット入力バレルイクスチェンジャの導入結果において，トレードオフの関係から外れている．原因として，DES 暗号化処理における平均置換ビット長は 52 ビットであるため，冗長の制御ビット書き込みが要求されたことがあげられる．1 回の置換処理において，平均して無駄に 3 命令発行されていることになる．ここで先に提案した D-END 命令を用いる

ことで無駄な書き込み命令を削除することが可能となり，必要命令数を 445 命令まで命令数を削減できる．バレルイクスチェンジャ導入前の約 6.6 %に相当する．

バレルイクスチェジャを演算器として使用する場合，面積を考慮すると置換を行うデータの平均ビット数以下を扱う最も大きいもしくはもう 1 サイズ小さいバレルイクスチェンジャを用いることが最適であると考えられる．2 種の内どちらを選択するかは，ターゲットの制約によって決定される．

以上より，バレルイクスチェンジャ導入の効果を示した．

8. むすび

配線自由度の実現に用いられるスイッチングアレイのプロセッサへの導入手法を提案した．自由なビットデータ移動の実現により，シフトおよびビットレベルの置換に関して 10 倍程度の演算の高速化が可能となる．今後の課題としては，任意の制御ビットへの自由な書き込み，バイト単位でのデータ処理の実現などが考えられる．またターゲットアーキテクチャとして想定する映像プロセッサへの演算器としての導入を考えた場合の，マルチスレッド対応への拡張を含めた検討が考えられる．

謝辞 日ごろから御討論いただく早稲田大学理工学部柳澤研究室の皆様に感謝します．論理合成ツール，ライブラリの使用について VDEC および（株）ロームに感謝します．また、映像向けプロセッサについて御討論いただくシャープ (株) 今井繁規様，李剛烈様に感謝します．本研究は一部文部科学省知的クラスタプロジェクト，日本科学技術振興会科学研究費，シャープ，NEC の研究補助金による。

文　献

[1] National Institute of Standards and Technology, "Data Encryption Standard(DES), "FIPS PUB 46-2, Dec. 1993.

[2] 木村 和也，大橋 岳洋，高木 直史，高木 一義，"専用回路を用いたマイクロプロセッサにおけるパーミュテーションの高速化，" 情報処理学会研究報告，SLDM112-26，2003.

[3] National Institute of Standards and Technology, Announcing the Advanced Encryption Standard (AES)," FIPS PUB 197,Nov. 2001.

[4] J. Daemen and V. Rijmen, AES Proposal: Rijndael," AES Al- gorithm Submission, http://csrc.nist.gov/encryption/aes/Rijndael.pdf, Sep. 1999.

[5] J. A. ブーフマン，暗号理論入門，2001