動的再構成可能プロセッサVulcanの評価

橋永, 寿彦
九州大学大学院システム情報科学府

Gauthier, Lovic
Fukuoka Industry, Science & Technology Foundation, FLEETS

神戸, 隆行
財団法人福岡県産業・科学技術振興財団福岡知的クラスター研究所

Ferreira, Victor Mauro Goulart
Fukuoka Industry, Science & Technology Foundation, FLEETS

他

出版情報：電子情報通信学会技術研究報告, VLD2005-98, pp.7-11, 2006-01. IEICE(RECONF)
バージョン：
権利関係：

動的再構成可能プロセッサ Vulcan の評価

橋永　寿彦[†]　LovicGAUTHIER[††]　神戸　隆行[††]

VictorMauro Goulart FERREIRA[††]　薄田竜太郎[††]　平木　哲夫[†]

山崎　陽介[†]　長野　孝昭[†]　村上　和彰[†††]

† 九州大学大学院システム情報科学府

〒 816–8580 春日市春日公園 6–1 九州大学大学院システム情報科学府研究院 E 棟 4F

†† 財団法人 福岡県産業・科学技術振興財団 福岡知的クラスター研究所

††† 九州大学大学院システム情報科学研究院

E-mail: †redefis@fleets.jp

あらまし　特定用途向けプロセッサとは，アプリケーションに特化した命令を実行することによって，汎用プロセッサに対して高性能を実現するものである．本稿では，特定用途向けプロセッサの実現方式として，データパスに動的再構成可能ハードウェアを用いたプロセッサ Vulcan を提案する．また，実際にアプリケーションを実装した結果を，汎用プロセッサで同様のアプリケーションを実行した結果と比較し，Vulcan の評価を行った．

キーワード　特定用途向けプロセッサ，動的再構成可能プロセッサ，AES

A Study of the Dynamically Reconfigurable Processor Vulcan

Toshihiko HASHINAGA[†], Lovic GAUTHIER[††], Takayuki KANDO[††],

Victor MAURO GOULART FERREIRA[††], Ryutaro SUSUKITA[††], Tetsuo HIRAKI[†], Yosuke

YAMAZAKI[†], Takaaki NAGANO[†], and Kazuaki MURAKAMI[†††]

† Department of Informatics, Kyushu University

†† Fukuoka Industry, Science & Technology Foundation, FLEETS

††† Division of Informatics, Kyushu University

E-mail: †redefis@fleets.jp

Abstract　Application specific extensions of a processor provide higher performance. In this paper, the authors propose “Vulcan” the Application specific processor with dynamically reconfigurable datapath, and demonstrate the efficiency of the proposed processor.

Key words　Application Specific Instruction Processor, Dynamic Reconfiguration Processor, AES

1. はじめに

近年，携帯電話やデジタルカメラ，PDA などの急速な普及に伴い，それらの製品に用いられるシステム LSI には高性能や柔軟性，短 Turn Around Time(TAT) が求められている．これらの要求に対して，従来は汎用プロセッサと専用の Application Specific Integrated Circuit(ASIC) を搭載したシステム LSI を用いることで高性能の要求を満たしてきた．しかし，この手法では ASIC のレジスタ転送レベル設計や ASIC を並列に動作させるための並列プログラミングに開発工数がかかり，短 TAT を実現することが困難になってきている．そこで，アプリケーションに特化した命令セットを備えた特定用途向けプロセッサをアプリケーション毎に生成することによって，高性能と設計期間の短縮を実現しようとする手法が研究されてきた．

特に最近では，汎用プロセッサに機能ユニットやコプロセッサを追加することによって特定用途向けプロセッサを実現するシステムが注目されている．これらは構成可能プロセッサと呼ばれ，汎用プロセッサに新規命令を追加するための機能ユニットやコプロセッサを新規に設計，あるいはデータベースから選択し，ハードウェアとして実現するという特徴がある．これらはプロセッサの高速化という面に関しては強力な手法であるが，一方で命令を追加するための機能ユニットをハードウェアとし

て実現するため，一度生成したプロセッサは単一のアプリケーションに対してのみ有効であるということや，頻繁な規格変更があった場合は，また一からハードウェアを作り直さなければならないという問題点がある．そこで筆者らは，特定用途向けプロセッサを再構成可能ハードウェアを用いて実現することで，複数のアプリケーションに対応可能かつ，チップ製造後にもハードウェアを変更可能なプロセッサが実現可能なのではないかと考え，その実現可能性について研究を行っている [1], [2], [3].

本稿では，2 章で再構成可能ハードウェアを用いた特定用途向けプロセッサの実現方式を述べ，3 章で，その一例であり，筆者らが提案する動的再構成可能プロセッサ Vulcan について紹介する．4 章では，Vulcan を用いた評価実験を説明，5 章で実験結果とその考察を述べ，最後に 6 章でまとめる．

2. 再構成可能ハードウェアを用いた特定用途向けプロセッサ

特定用途向けプロセッサの実現方式としては，コプロセッサ型とデータパス型が考えられる．コプロセッサ型とは，ベースとなるプロセッサのアクセラレータとして再構成可能ハードウェアを追加したもので，コプロセッサで実行する命令群を，ベースとなるプロセッサの拡張命令と見なすものである．データパス型は，ベースとなるプロセッサのデータパスに再構成可能ハードウェアを用いた機能ユニットを追加したもので，機能ユニットで実行する命令を，拡張命令と見なすものである．また，それぞれのアーキテクチャにおいて，再構成可能ハードウェアが再構成を行うタイミングが動的であるか，静的であるかによってもプロセッサの特徴が異なると考えられる．

既存の手法では，コプロセッサ型で静的な再構成を行うものは汎用的な CPU と FPGA を組み合わせたもの，動的な再構成を行うのは IPFlex 社の DAP/DNA [4] などが挙げられる．データパス型で静的な再構成を行うものとしては，ALTERA 社の NIOS II [5] などが挙げられる．しかし，データパスに動的再構成可能なハードウェアを用いて命令セットを拡張する方式はこれまであまり提案されていない．そこで，我々の研究グループでは，共同研究先が開発中のデバイス Vulcan をデータパスに動的再構成可能なハードウェアを用いたプロセッサの一例として用い，評価を行っている．

データパスに動的再構成可能なハードウェアを用いたプロセッサでは，拡張した命令セットを実行するためのデータパスを時分割で実現することができるので，静的に再構成を行う場合よりも，小面積で多くのアプリケーションに特化した命令を実現できるのではないかと考えられる．しかし，実行時に頻繁な再構成が必要となるので，性能的には再構成のオーバーヘッドを十分考慮する必要がある．

3. 動的再構成可能プロセッサ Vulcan

3.1 Vulcan の概要

Vulcan は動作時に再構成可能なデータパスを持ち，順序制御をプログラムで行う動的再構成可能なプロセッサである．

Vulcan は 1 ステップ毎にコンフィギュレーション・データの再設定を行いながらアプリケーションを実行する．ここでステップとは，コンフィギュレーション・データの設定と演算の実行を合わせた，1 命令の実行にかかる期間のことをいう．また，1 命令を実行するための再構成可能なデータパスの構成情報をコンフィギュレーション・データと呼ぶ．Vulcan が保持できる命令数は 128 個までである．また，Vulcan の演算実行にかかるクロック・サイクル数は，命令毎にユーザが設定することができる．以下，Vulcan のアーキテクチャと実行フローについて述べる．

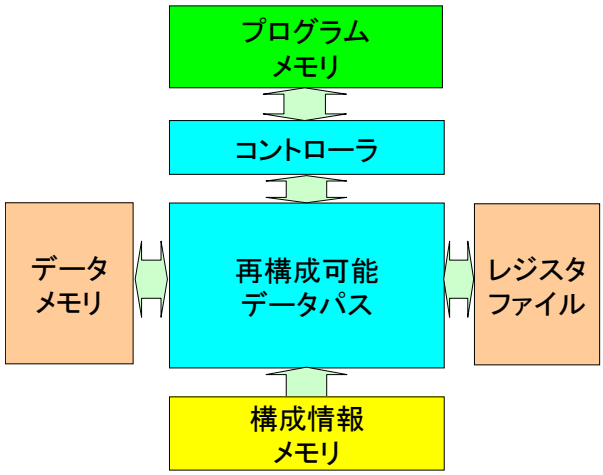

図 1　Vulcan のアーキテクチャ概観

3.2 Vulcan のアーキテクチャ

図 1 に Vulcan のアーキテクチャの概観を示す．Vulcan はコントローラ，メモリ，構成情報メモリ，レジスタファイル，再構成可能データパス (Reconfigurable Data Path: RDP) からなっている．各ユニットの機能は以下のとおりである．

- コントローラ:

コンフィギュレーション・データのロード，メモリ内のデータのロードなどシステム全体の実行管理を行う．内部に 22bits のインデックス・レジスタを 7 本備えており，インデックス・レジスタ内の値を用いてメモリにアクセスすることも可能である．また，インデックス・レジスタ内の値を格納するスタックと接続している．

- メモリ:

Vulcan の命令フォーマットを表 1 に示す．Vulcan の命令では，通常のオペレーションの代わりに各命令の構成情報を示す番号を指定することによって所望の命令を実行する．オペランドは，ソースとデスティネーションの 2 つのレジスタを指定することができる．また，Vulcan の命令では，分岐などの制御命令は演算命令と別に指示することができる．分岐を行う際は，分岐条件を別に指定する．ジャンプ命令などは，相対アドレスを示すこともできる．

- 構成情報メモリ:

コンフィギュレーション・データを格納するメモリ．命令のコンフィギュレーションを 128 個まで保持できる．

- レジスタファイル:

後述する RDP 部の各 PE での演算結果を保存し，異なる命令に引き渡すことができる．1 ワード 256bits 幅である．レジスタファイルは RDP 部の PE に接続されており，データが出力される PE の位置によって格納されるレジスタファイル中の bit 位置が決定される．

- 再構成可能データパス:

図 2 に RDP 部の構成を示す．RDP 部には Programmable Element(PE) が 16 行 8 列で配置されている．PE は 6 入力 2 出力であり，入力に対して任意の論理関数を割り当てることができる．各 PE 間は，垂直方向の Vertical Line(VL) と水平方向の Horizontal Line(HL) で接続されている．VL が 1 列の各 PE に接続しており、各 VL は HL で接続されている．VL から HL，あるいは HL から VL への信号の受け渡しはスイッチ部が制御している．VL は 64bits 幅で 8 本，HL は 64bits 幅で 7 本である．また，メモリから RDP 部へは一度に 64bits のデータをロードすることができる．

表 1　Vulcan の命令フォーマット

演算命令 (構成番号)	オペランド ×2	制御命令	分岐命令	メモリアドレス

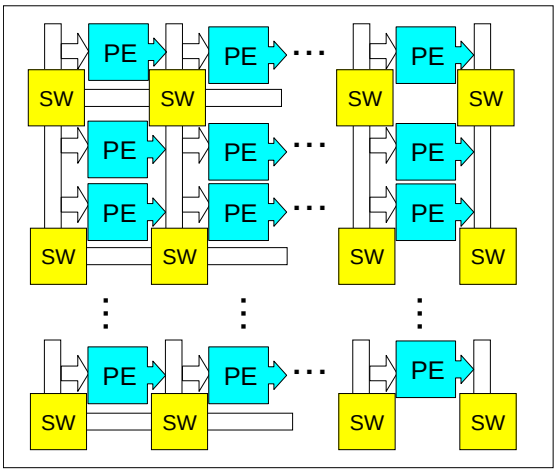

図 2　RDP 部概観

3.3　実行フロー

Vulcan では，まずコントローラがメモリからプログラムを読み出し，演算命令であれば，コントローラが命令中に指定された構成情報メモリのアドレスから適切なコンフィギュレーション・データを読み出し RDP 部にロードする．RDP 部は，コンフィギュレーションが確定すると入力データに対して処理を行う．各 PE で実行された演算結果は VL に出力されるとともにレジスタに書き込むことができ，中間結果として別の命令にデータを引き渡して使うことができる．レジスタを用いることで大きな処理を複数命令に分割して実行する．また，制御命令であれば，プログラムカウンタの値に対して命令中に指示された操作を行う．

表 2　シミュレータ設定

命令発行方式	アウト・オブ・オーダ
命令セット	PISA
分岐予測器	
Type	2 レベル (bimod, 2K エントリ)
BTB サイズ	512 エントリ，4 ウェイ
命令発行幅	4 命令/cc
命令デコード幅	4 命令/cc
IFQ サイズ	4 エントリ
RUU サイズ	16 エントリ
LSQ サイズ	8 エントリ
キャッシュ・メモリ (容量，レイテンシ)	
L1 データキャッシュ	32KB(4 ウェイ，128 エントリ)，1 cc
L1 命令キャッシュ	32KB(1 ウェイ，512 エントリ)，1 cc
L2 共有キャッシュ	64KB(4 ウェイ，1024 エントリ)，6 cc
メモリバンド幅	8 Bytes
メモリポート数	2
整数演算器 (ユニット数，実行，発行レイテンシ)	
ALU	4, 1 cc, 1 cc
Mult.	1, 3 cc, 1 cc
Div.	1, 20 cc, 19 cc

(cc: clock cycle (s))

4.　実　　験

4.1　実験の目的

特定用途向けプロセッサのねらいとしては，汎用プロセッサよりも特定のアプリケーションに対して高性能であるということと，専用回路よりも設計が容易であるという 2 点である．そのため，筆者らの提案する再構成可能ハードウェアを用いた特定用途向けプロセッサの評価を行うためには，まず，汎用プロセッサとの性能比較，ASIC，FPGA などハードウェアとのコスト，パフォーマンスによる比較が必要である．また，その後，他の実現方式による再構成可能ハードウェアを用いた特定用途向けプロセッサとの比較，評価も必要となると考える．本稿では，その中で汎用プロセッサとの性能比較のみを，実際にアプリケーションを実装した例を用いて行った．

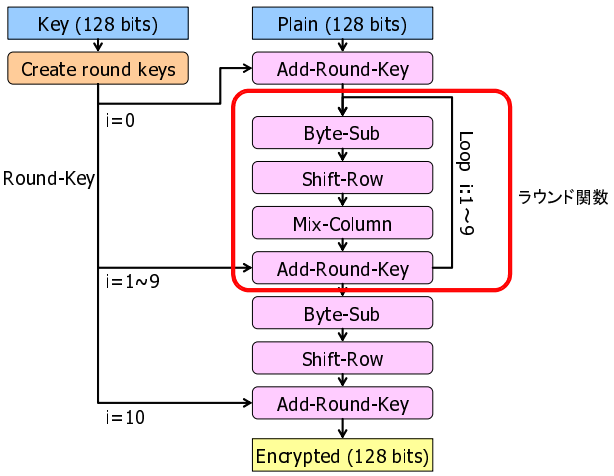

図 3　AES 暗号化の実行フロー

4.2　実 験 内 容

実験では，プロセッサ Vulcan と汎用プロセッサに同様のアプリケーションを実装し，性能を計測，結果について比較評価を行う．実装するアプリケーションには，AES(Advanced Encryption Standard) を用いる．

AES [6] は，NIST(National Institute of Standards and

Technology) により DES(Data Encryption Standard) の後継として採用された暗号化アルゴリズムである．鍵長，ブロック長ともに 128，192，256bits が利用可能で，暗号化と復号化に同じ鍵を用いる，共通鍵暗号方式である．今回の実験では，鍵長，ブロック長ともに 128bits とした．図 3 に AES の暗号化処理の流れを示す．AES 暗号化では，平文を 8bits 単位に区切り，4×4 行の配列に配置した入力に対して，ラウンドと呼ばれる処理単位を規定回数実行し，暗号文を得る．128bits の鍵長を扱う場合，ラウンド数は 10 回である．各ラウンドは，以下の 4 つの処理からなっている．

- Add-Round-Key

入力とラウンド鍵の排他的論理和

- Mix-Column, Shift-Row

ビットシフトなど，線形変換

- Byte-Sub

S-Box と呼ばれる非線形変換

なお，10 回目のラウンドでは，Mix-Column は行われない．また，ラウンド毎にビットシフトと S-Box 変換を用いてラウンド鍵が生成される．

Vulcan 上へのアプリケーション実装は，C 言語で記述されたソースコードから人手で命令を作成し，同時に Vulcan 用のプログラムを記述する．実行結果は，Vulcan の命令セットシミュレータによって計測可能である．比較対象としては，プロセッサシミュレータ Simplescalar [7] から，アウト・オブ・オーダ実行可能な sim-outorder を用いる．採用した命令セットは，32bits 版 MIPS 互換の PISA である．実験に用いたシミュレータの構成パラメータは，表 2 に示す．Simplescalar 上での AES 暗号化に用いたプログラムは，Vulcan への実装の際に参照したソースコードを gcc-2.6.3 により，最適化オプション O2 でコンパイルした．

5. 実験結果と考察

5.1 実 験 結 果

Vulcan 上での AES 暗号化では，1 ブロック (128bits) の平文の暗号化に 309 命令を要し，うち鍵のスケジューリングを除いた命令数は 79 命令であった．また，作成した命令の種類は 38 種類であった．表 3 に暗号化 1 ラウンドの処理を行う命令と使用した PE 数を示す．今回の実装では，1 ラウンドの処理を 8 命令で実行することができた．特に 128bits の平文とラウンド鍵の排他的論理和を行う Add-Round-Key では，全ての PE を使用して一命令で実行できている．S-BOX 処理では，4byte 分の処理を PE124 個用いて行っているので，128bits 分の処理を行うために 4 命令が必要であった．また，行方向のシフト処理である Shift-Row は 1 命令で実行できたが，列方向の Mix-Column では，2 命令を実行に要した．

表 4 に 1 ブロックの平文を AES 暗号化するのにかかった実行命令数と，実行クロック・サイクル数を示す．表 4 より，命令セットが PISA である MIPS プロセッサに対して，Vulcan は約 36 分の 1 の実行命令数であった．また，両者の動作周波数を同じと考えると，命令セットが PISA である MIPS プロセッサに対して，Vulcan は AES 暗号化を約 5.3 倍高速に実行可能であると言える．

表 3　Vulcan 上での AES 暗号化 1 ラウンドの実行命令と使用 PE 数

実行命令	使用 PE 数
Add-Round-Key	128
S-BOX1	124
S-BOX2	124
S-BOX3	124
S-BOX4	124
Shit-Row	64
Mix-Column1	104
Mix-column2	104

表 4　AES 暗号化の性能比較

プロセッサ	実行命令数	実行クロック・サイクル数
Vulcan	309	1497
MIPS(Simplescalar)	11258	7902

5.2 考　　察

図 4，図 5 に Simplescalar と Vulcan 上で AES 暗号化処理を行った際の実行命令の内訳をそれぞれ示す．Simplescalar 上での AES 暗号化では，ロード，ストア命令が約 40%と最も頻繁に実行されていることが分かる．これは，今回の実装で S-Box 変換と鍵生成処理の際のラウンド定数を表参照を行う用に実装したことと，C 言語で記述されたソースコードをコンパイルしたものであるので，平文を暗号化する際の中間結果もメモリに保存するようになっているからであると考えられる．Vulcan では，暗号化の中間結果は全て内部のレジスタに保存しており，S-Box 変換も RDP 部の LUT を用いて実行しているので，ロード，ストア命令は，16 命令のみであった．

Simplescalar 上での AES 暗号化処理の実行命令で次に多いのは加算命令である．これは，ループや行列のインデックス計算の際に用いられていると考えられる．Vulcan では，ラウンド関数などを全て展開しているので，ループでのインクリメントなどは行われない．

論理演算は，Add-Round-Key で用いられる排他的論理和が最も多かった．Vulcan では，Add-Round-Key での 128bits の排他的論理和を 1 命令で行っている．また，シフト演算に関しても，Simplescalar での実行では，byte 単位でデータを処理しているのに対して，Vulcan では，1 ラウンドの Shift-Row を 1 命令，Mix-Column を 2 命令で実行している．

また，Vulcan ではムーブ命令の実行回数が 40%以上を占めている．これは，表 1 に示すように，Vulcan ではオペランドを 2 つしか指定できないため，複数の入力が必要な場合，同一のワードに入力を揃えなければならないからである．

これらの要因から，今回の Vulcan での AES 暗号化は，Simplescalar と比べて大幅な実行命令数の削減が実現できたと考えられる．しかし，約 36 分の 1 の命令削減率に比べると，実行クロック・サイクル数の削減率は約 5 分の 1 と少ない．ここで，Vulcan での AES 暗号化にかかった実行クロック・サイク

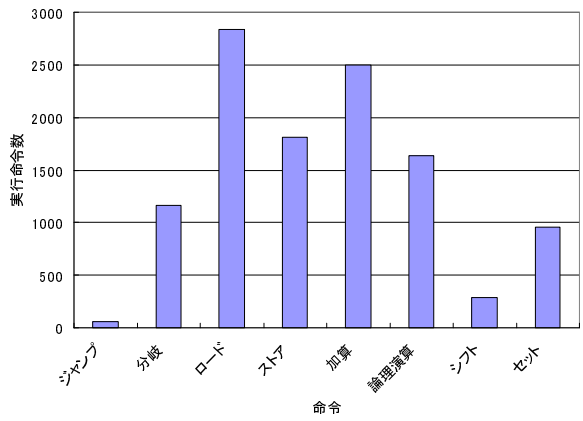

図 4　Simplescalar 上での AES 暗号化の実行命令数内訳

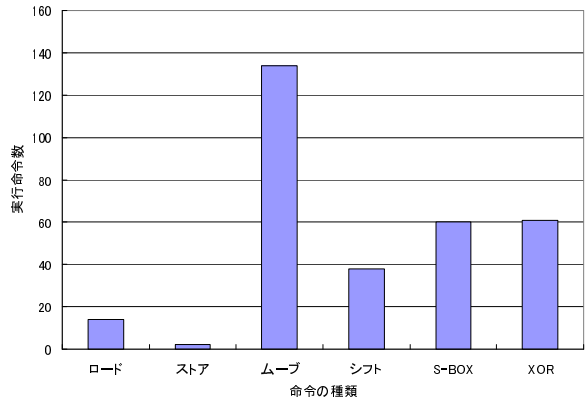

図 5　Vulcan 上での AES 暗号化の実行命令数内訳

ル数の内訳を図 6 に示す．

図 6 より，約 50%が命令フェッチにかかるクロック・サイクル数であることが分かる．今回の実装では，ラウンド関数を全て展開したため，1 ブロックの実行では命令の時間的局所性が無く，命令キャッシュがミスを起こしたためである．命令キャッシュのヒット率は 87.34%であった．命令キャッシュの容量が十分で，同様のプログラムを複数回実行すれば，命令フェッチにかかったクロック・サイクル数の全体に対する割合は削減されると考えられる．

また，Vulcan では，1 命令毎に再構成を行うため，命令の実行にかかるクロック・サイクル数だけ再構成にかかるクロック・サイクル数が消費されてしまう．これを解決する方法としては，RDP 部を分割して一方が実行中に他方が再構成を行うパーシャル・リコンフィギュレーション，あるいは，一命令で実行可能な演算の規模を多くすることで，再構成にかかるクロック・サイクル数が全体の実行クロック・サイクル数に占める割合を減少させるなどの手法を今後検討していく必要がある．

6. おわりに

本稿では，動的再構成可能ハードウェアを用いた特定用途向けプロセッサを提案し，その一実装例であるプロセッサ Vulcan を用いて，動的再構成可能ハードウェアを用いた評価実験につ

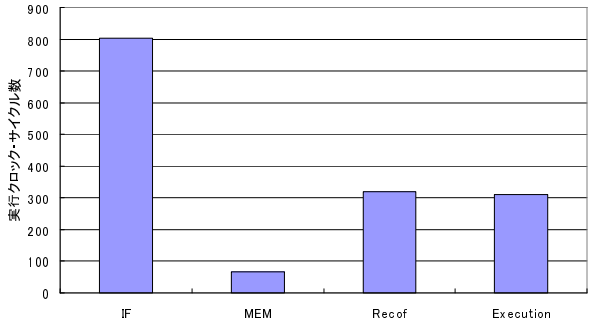

図 6　Vulcan 上での AES 暗号化の実行クロック・サイクル数内訳

いて述べた．評価実験においては，Vulcan 上での AES 暗号化は，4 ウェイのアウト・オブ・オーダ実行を行うプロセッサに対して高速に実現できる可能性があることを示した．しかし，再構成のオーバーヘッドなどの要因から，実行命令数の削減率に対して，十分実行クロック・サイクル数の削減率が得られないことが分かった．今後は，Vulcan において再構成のオーバーヘッドを削減する手法の検討を行うとともに，今回実装したアプリケーションと性質の異なる様々なアプリケーションについて同様の実験を行い，FPGA や他の再構成可能デバイスとの比較，評価も行う予定である．

謝辞　本研究の一部は，平成 14–18 年度文部科学省知的クラスター創成事業「次世代システム LSI アーキテクチャの開発」に拠る．

文　献

[1] 首藤真，松尾拓真，橋永寿彦，森江善之，Lovic Gauthier，村上和彰，“命令セット再定義可能プロセッサ「Redefis」～ユーザ機能の実装に適した SoC プラットフォームの提案～”，電子情報通信学会技術研究報告 CPSY2004-25 ～31，pp7-12，2004

[2] 橋永寿彦，首藤真，松尾拓真，森江善之，Lovic Gauthier，村上和彰，“Vulcan～Redefis の一実施例とそれへのユーザ機能実施例の紹介” 電子情報通信学会技術研究報告 CPSY2004-25～31，pp13-18，2004

[3] 松尾拓真，首藤真，橋永寿彦，森江善之，Lovic Gauthier，村上和彰，“Redefis 開発環境の概要～開発環境の概要とアプリケーション開発例～” 電子情報通信学会技術研究報告 CPSY2004-25～31，pp19-24，2004

[4] IPFLEX 社，
http://www.ipflex.com/

[5] ALTERA 社，
http://www.altera.com/

[6] NIST, FIPS PUB 197,
“Advanced Encryption Standard(AES),”
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, Nov. 2001.

[7] SimpleScalar LLC,
http://www.simplescalar.