社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

動的再構成可能プロセッサを用いた 組み込み向け複数暗号処理エンジンの実装

長谷川揚平† 阿部　昌平† 松谷　宏紀† 安生健一朗†† 粟島　亨†††
天野　英晴†

† 慶應義塾大学大学院理工学研究科　〒 223–8522 神奈川県横浜市港北区日吉 3-14-1
†† NEC エレクトロニクス　〒 211–8668 神奈川県川崎市中原区下沼部 1753
††† NEC システムデバイス研究所　〒 211–8668 神奈川県川崎市中原区下沼部 1753
E-mail: †drp@am.ics.keio.ac.jp, ††k.anjo@necel.com, †††awash@cp.jp.nec.com

あらまし　近年, 発達のめざましい動的再構成可能なデバイスは, 動的に回路構成を変更することが可能で, 開発期間の短縮およびハードウェアコストの削減を実現することができる. しかしながら, より大きく, より多くのアプリケーションを実現する場合には, チップ外部から回路構成情報を動的に設定することで, チップ内部に保持可能なハードウェアサイズを越える回路を実現する仮想ハードウェアのメカニズムが有効であると考えられる. 本稿では, NEC 社の動的再構成可能プロセッサ Dynamically Reconfigurable Processor (DRP) を用いて, 仮想ハードウェア機構により複数の暗号処理を動的にスイッチすることが可能な複数暗号処理エンジンを実装し, その性能評価を行った.
キーワード　動的再構成可能プロセッサ, DRP, 仮想ハードウェア, マルチアプリケーション, 暗号処理

Switchable Multi-Cryptographic Engines for Embedded Network Security on a Dynamically Reconfigurable Processor

Yohei HASEGAWA†, Shohei ABE†, Hiroki MATSUTANI†, Kenichiro ANJO††, Toru AWASHIMA†††, and Hideharu AMANO†

† Department of Information and Computer Science, Keio University
3-14-1, Hiyoshi, Kohokuku, Yokohama, 223–8522, Japan
†† NEC Electronics　1753 Shimonumabe, Nakahara, Kawasaki 211-8668, Japan
††† NEC System Devices Research Labs.　1753 Shimonumabe, Nakahara, Kawasaki 211-8668, Japan
E-mail: †drp@am.ics.keio.ac.jp, ††k.anjo@necel.com, †††awash@cp.jp.nec.com

Abstract　Recent dynamically reconfigurable processors, which have flexibility in changing their hardware structure and capability of parallel processing, are expected to reduce the hardware cost and turn around time. However, in order to implement larger and more applications than the capability of on-chip configuration size, the Virtual Hardware mechanism, which swap in/out the configuration data from/to external memory is required. In this paper, we implemented the multi-cryptographic engine which can switch multiple cryptographies dynamically by the virtual hardware on the NEC's Dynamically Reconfigurable Processor (DRP) and evaluated its performance.
Key words　Dynamically Reconfigurable Processor, DRP, Virtual Hardware, Multi-Applications, Cryptography

1. はじめに

近年, 我々の社会に出回っている家電製品には, 動画像処理や音声処理など多くの機能が登載されている. 最近は IPv6 の普及に伴ないインターネットに接続可能な製品も数多く登場している. このような情報家電は, 高性能かつ多種多様な機能が求められるが, 中でもプライバシやセキュリティの面から, 暗号化や認証の機能が重要である. マルチメディア処理や通信の暗号化は高い演算性能が要求されるため, 近年の System-on-a-Chip (SoC) では, Application Specific Integrated Circuits (ASIC) によるハー

ドウェア化が一般的な手法である. しかし, システムの大規模化, 要求機能の多様化, 高まる仕様標準化の動きに伴ない, ハードウェア化による下流設計段階での開発コストの増大が深刻な問題となっている.

このような課題を解決するハードウェアとして, マルチメディア処理や通信制御系の分野で動的再構成可能プロセッサが注目されている [1] [2]. 2002 年ころより急速に開発が進み, 複数の企業が開発に参入しており, 実用化が現実のものとなってきた [3] [4] [5]. 動的再構成可能プロセッサは, コンテキストと呼ばれる回路構成情報をチップ内部に複数保持し, これを高速に切り替えることによって様々な機能を実現する. アレイ状に配置した粗粒度の演算器や分散メモリによる並列処理によって, 通信制御分野やメディア処理などのいわゆるストリーム処理において高い処理性能を示す. また, マルチコンテキスト方式の動的再構成機能によって, 対象となるアプリケーションを時分割多重実行することにより, スケーラビリティや柔軟性を優れた面積効率で実現する. こうして, 動的再構成可能プロセッサは, SoC における専用ハードウェアや Digital Signal Processor (DSP) の代替として利用されることが期待される.

しかしながら, 動的再構成可能プロセッサの内部の構成情報メモリに保持することのできる回路構成情報には限りがあり, システムが大規模化した場合にはすべてのアプリケーションを単一の動的再構成可能プロセッサでは実行できない. また, チップ内のすべての機能が同時に稼働しているという状況はあまりなく, 面積の面からもハードウェアを増やすことは望ましくない.

ここで, 必要に応じてチップ外部より回路構成情報を設定するで, 複数のアプリケーションを実行する仮想ハードウェアのメカニズムを導入する. この機構により, 使用していないアプリケーションはチップ外部に退避させたり, 巨大なアプリケーションを時分割することでハードウェアコストを削減する.

本稿では, NEC 社の動的再構成可能プロセッサである Dynamically Reconfigurable Processor (DRP) を用いた複数暗号処理エンジンを提案する. これは, 複数の暗号処理を動的に切り替えることが可能なものであり, 仮想ハードウェア機構によってハードウェア資源の不足する組み込み分野において. 多くの機能の実現を目指すものである.

本稿の構成は以下の通りである. 節 2. で, 動的再構成可能プロセッサの概要を述べ, NEC 社の DRP のアーキテクチャについて述べる. そして, 節 3. では, DRP と組み込み CPU との協調動作システムについて述べる. 節 4. では, 実装した複数暗号処理エンジンについて述べ, 節 5. でその評価結果を述べる. 最後に, 節 7. で結論と今後の課題を述べる.

2. 動的再構成可能プロセッサ

2.1 動的再構成可能プロセッサの特徴

動的再構成可能プロセッサは, SoC や FPGA の問題点を解決する新しいアプローチである. SoC において専用ハードウェアや DSP を動的再構成可能プロセッサに置き替えることによって, 開発コストや開発期間を短縮することが可能となる. また, ハードウェア構成を容易に変更することができ, 新しい技術を容易に追加することも可能である. また, 単一のチップを大量に生産すればよいことから, チップ自体のコスト削減も期待できる.

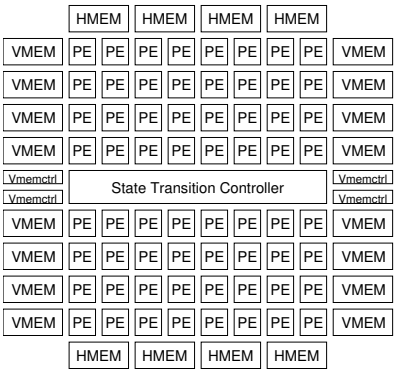

図 1　Tile の構造

動的再構成可能プロセッサは, 以下の 3 つの特徴をもつ.

- **粗粒度アーキテクチャ**　従来の FPGA が Look-Up-Table (LUT) による細粒度構成を採用していることに対し, 8bit から 32bit の粗粒度の演算器をアレイ状に配置したアーキテクチャを採用する. レジスタ, シフタ, マルチプレクサなどを組み合わせた構成をとり, 並列処理による高い処理性能を実現する.
- **ダイナミックリコンフィギュレーション**　チップ内部に複数のコンテキストと呼ばれる回路構成情報を用意しておき, これを次々に切り替えながら処理を進めていくマルチコンテキスト機能をもつ. 全体の処理を時分割多重化して実行することで高い面積効率を実現する. このようなマルチコンテキストデバイスは, 従来の FPGA などのプログラマブルデバイスの課題であったプログラマブル配線の増大を, 複数のコンテキスト間で配線資源を時分割多重化することによって解決する.
- **C 言語によるプログラミング**　動的再構成可能プロセッサは, 高水準言語による設計や, 高位機能合成技術との適合性が高い. これは, 高水準言語レベルで記述された演算要素を, 直接プロセッシングエレメントにマッピングすることが可能でありかつ, 時系列的な処理をマルチコンテキストの時系列処理にマッピングすることができるためである.

2.2 Dynamically Reconfigurable Processor (DRP)

DRP は, NEC エレクトロニクス社の動的再構成可能プロセッサのアーキテクチャである [3]. DRP の基本的なアーキテクチャを図 1 に示す. DRP は, Tile と呼ばれる基本ユニットをアレイ状に配置することで, 所望の規模の DRP コアを実現する. 各 Tile は, 演算器とレジスタファイルなどから構成される Processing Element (PE) を 8 × 8 の 2 次元アレイ状に並べ, その周辺に VMEM, HMEM と呼ばれる分散メモリを縦横に拡散配置し, さらに中央部に State Transition Controller (STC) を配置した構成となっている. Tile ごとに STC と PE のアレイが組み合わされているため, 各 Tile が独自のステートマシンの制御下で独立して動作することが可能である.

各 PE の構成を図 2 に示す. 各 PE は 8bit の ALU, シフトやデータ制御, 簡単な論理演算を行なう Data Management Unit (DMU), 8bit の Flip Flop, レジスタファイルから構成される. ま

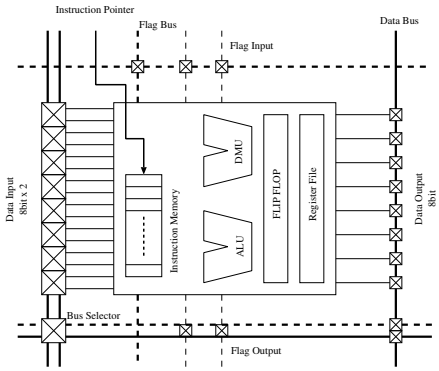

図 2　Processing Element (PE) の構造

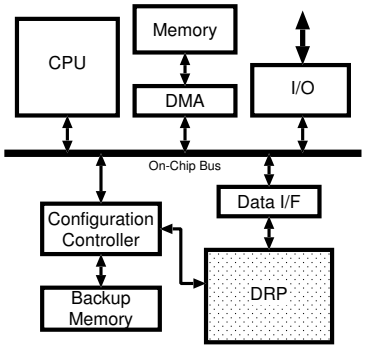

図 3　想定するシステムの構成

た, コンフィギュレーション時には命令データが命令メモリに書き込まれ, 実行中に STC が発行した命令ポインタを命令メモリが受け, 利用する命令データをロードすることでハードウェア構成を変更する.

DRP のプロトタイプチップである DRP-1 は Core の周辺部に 32bit の乗算器を 8 セット, メモリモジュール, PCI バスインタフェース, SDRAM/SRAM/CAM インタフェースを搭載したシステム LSI である. また, PCI バスインタフェース, SDRAM/SRAM/CAM インタフェースにより単独で PCI バスへの接続や外部メモリの制御が可能である.

DRP-1 では 4 × 2 個の Tile が配置されており, 全体では 512 個の PE をもつ. チップ中央には, DRP-1 全体の状態遷移を制御するためのシーケンサである Centeral State Transition Controller (CSTC) が配置されている. また, DRP-1 全体では VMEM を合計 80 セット, HMEM を合計 32 セットもつ.

DRP-1 は内部のメモリに最大 16 コンテキスト分の情報を蓄えることが可能で, クロックサイクル毎にコンテキスト切り替えが可能なマルチコンテキストデバイスである.

DRP の開発環境として, ソフトウェアライクな統合開発環境 Musketeer が提供されており, 短 TAT で効率的な開発を行うことができる [6]. DRP コンパイラは, C 言語をハードウェア記述用に拡張した Behavior Design Language (BDL) から, 動作合成, マッピング, 配置配線などの合成を経て DRP 上で実行可能なオブジェクトコードを生成する.

3. 組み込み CPU との協調動作

3.1 CPU/DRP 協調動作モデル

MPEG や JPEG をはじめとするマルチメディア処理などでは, 強力な演算能力を要求するものに加え, 性能要求の高くない処理や, システム管理やユーザインタフェース機能などの複雑な処理が混在している. 一般的な SoC では, 事前のプロファイリングにより負荷の高い処理を専用ハードウェアなどのコプロセッサで実行し, 高い演算能力は必要としないが, 複雑な処理を要するものは組み込み CPU で処理をする. このようにすることで, コスト, 性能, 消費電力の面で優れたシステムを実現する. しかし, 従来の専用ハードウェアと組み込み CPU を混載した SoC では, 用途別に様々なハードウェアが必要であったり, 次々と登場する新しい技術に対応するため, 開発コストが増大するという問題がある. そこで, このような SoC において, 専用ハードウェアにかわり動的再構成可能プロセッサを利用すれば, 組み込み CPU と協調動作させることにより, より柔軟でスケーラブルなシステムを構築することができる.

組み込み CPU と動的再構成プロセッサとの協調動作には, より密に結合して, プロセッサのデータパスの一部としてリコンフィギャラブルユニットを動作させ, 命令レベルでの協調動作を行う手法も提案されている [7]. この場合, リコンフィギャラブルユニットは, アプリケーションに最適化されたリコンフィギャラブル命令を実行するために利用され, CPU とのタスクの分散は命令単位に行われる. しかし, この方法は, CPU とリコンフィギャラブルユニット間の負荷分散が細粒度すぎるため, CPU のレジスタとリコンフィギャラブルユニット間で命令実行の度に大量のデータの授受が必要になり, 性能向上の妨げとなる. しかし, DRP がターゲットとする典型的なストリーム処理は, データを共有するタスクと呼ばれる処理単位から構成され, タスク間のストリームの授受が明確化されている. このため, 多くの場合, タスク単位の粗粒度でリコンフィギャラブルユニットに割り当て, 一定サイズのストリームをリコンフィギャラブルユニットの内部メモリに格納してまとまった処理を実行する方がデータ転送量を削減する点で有利である. そこで, ここでは, 演算性能を要求されるタスクは一定の大きさの Tile に, 演算性能を要求されないタスクは, 組み込み CPU で実行する手法を採用する.

3.2 想定するシステム

本稿で提案する複数暗号処理エンジンは, 複数の暗号処理を状況に応じて動的に切り替えることが可能で, 組み込み分野での IPsec や SSL/TLS などのセキュア通信に応用することを想定する. ここで, 各暗号処理はそれぞれがひとつのタスクとしてみなすことができる. そこで, ここでは, それぞれの暗号処理に相当するタスクを DRP で高速化し, それ以外の制御処理などを組み込み CPU で実行する.

本稿で想定する評価システムを図 3 に示す. これは組み込みプロセッサと DRP の基本構成ユニットである Tile を複数個結合したコプロセッサ型の協調動作システムである. 共有メモリと組み込み CPU, DRP 間のデータ通信は 64bit, 100MHz 程度のオンチップバスを通じて行われ, DMA 転送をサポートするものとする. また, DRP コアはストリーム処理を意識した独自の入

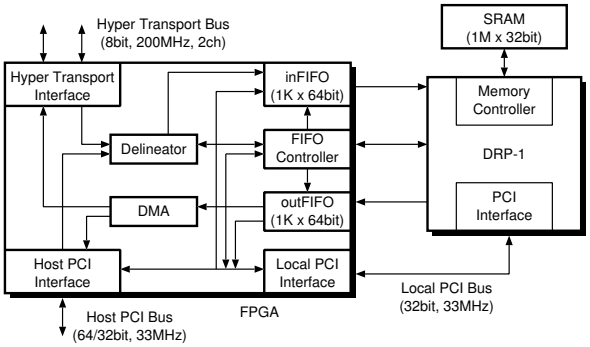

図 4　DRP 評価ボードの構成

出力インタフェースをもつものとする.

DRP のコンフィギュレーションは隣接する外部コントローラである Configuration Controller によって制御され, 各タスクのコンフィギュレーションデータは Configuration Controller を経由してロードされる. また, DRP コアの外部にはコンフィギュレーションデータ用のバックアップメモリ (キャッシュ) を保持する. これによって, 必要に応じて外部からコンフィギュレーションデータを設定し, DRP 上で仮想ハードウェアを実現する.

4. 暗号処理部の実装と評価

4.1 実 験 環 境

それぞれの暗号処理に対応するタスクの実装および検証には, DRP-1 を搭載した DRP 評価ボードを使用した. この評価ボードは, 図 4 に 示す通り, DRP-1 コア, DRP-1 の入出力を制御する FPGA, PCI インタフェース, 外部 SRAM などから構成される PCI ボードである. 動作検証には, ホスト PC から制御 API を用いて DRP-1 を制御し, ホスト PC と DRP-1 との協調動作による検証を行った.

DRP-1 と入出力を制御する FPGA は, ローカル PCI I/F を介して接続されている. ローカル PCI バスは, DRP-1 にコンフィギュレーションデータをロードする際に使用される. また, DRP-1 と FPGA は独自の入出力インタフェースをもち, データの入出力は FPGA の入出力 FIFO を介して行われる. また, この FPGA は, ホスト PCI I/F を介して, ホスト PC と接続される. DRP-1 のコンフィギュレーションの制御や, 入出力 FIFO へのアクセスはホスト PC から PCI バスを介して行なわれる.

なお, DRP 評価ボード上の制約から, DRP-1 のコンフィギュレーションは PCI バス経由で行なわれ, 部分再構成機能やコンテキストごとのコンフィギュレーションは利用することができない. このため, コンフィギュレーションの際には, DRP-1 を一度リセットしてから DRP-1 全体を書き替える必要がある.

4.2 DRP-1 上への実装

今回 DRP-1 上に実装した暗号処理は, 共通鍵暗号の DES, AES (Rijndael), CAST-128, CAST-256, RC6 と, 認証などで用いられる一方向ハッシュ関数の MD5 と SHA-1 である.

DRP-1 上へ実装するアプリケーションは, BDL を用いてアルゴリズムを記述し, 統合開発環境 Musketeer および DRP コンパイラを使用してコンパイルを行った. DRP コンパイラは, 論理的なコンテキスト (状態) のスケジューリングをコンパイラに委ねる自動スケジューリングモードと, クロックの境界を明示的に記述することでコンテキストのスケジューリングを行う手動スケジューリングモードをサポートする. 多くの暗号処理の場合, 1 クロックでどれだけの処理を実行できるかが重要であるため, 今回は手動スケジューリングモードでの実装を行い, 各コンテキスト内で実行する処理を明示することにした.

表 1　各暗号処理の DRP-1 上への実装結果

	Required Resources				Freq.	Throughput[Mbps]		
name	Ctxt	Max	All	Vmem	[MHz]	DRP-1	DSP	MIPS
DES	16	131	1378	10	29.7	111.8	84.2	40.3
CAST128	8	36	141	20	30.1	113.5	95.4	46.7
AES	6	129	343	34	56.9	364.0	16.9	46.4
CAST256	13	56	323	21	28.3	36.6	36.1	23.2
RC6	6	56	163	8	29.3	170.6	96.6	76.9
MD5	9	57	297	14	23.1	181.6	135.5	95.5
SHA-1	10	70	348	20	31.2	197.3	44.0	38.7

共通鍵暗号の中で, DES, AES は鍵スケジューラも DRP-1 でサポートするが, 他のアルゴリズムはソフトウェアで鍵スケジューリングを行う. また複数の鍵長を実行できるアルゴリズムは, 鍵長を指定できるように実装した. MD5 と SHA-1 は, 512bit ごとに実行する圧縮処理を行うもので, メッセージのパディングやメッセージ長の付加などの処理はソフトウェアで行う. また, 協調動作により, IPsec などで用いられる認証値生成処理である HMAC などの処理も実行可能である.

4.3 暗号処理部のコストと性能

実装した各暗号処理のコストおよび性能を表 1 に示す. この表では, DRP コンパイルフローにおける配置配線後の使用コンテキスト数 (Ctxt), 最大使用 PE 数 (Max), 総使用 PE 数 (All), 最大使用 VMEM 数 (Vmem) を示す. 最大使用 PE 数は, 使用コンテキストの中で最も多く使用した PE の数であり, 空間的に消費している PE の数を意味する. また, 総使用 PE 数は, 各コンテキストで消費した PE の数の総和であり, 時間的に消費している PE の数を意味する. なお, 表 1 の結果において, DES と AES は暗号処理モジュールに鍵スケジューラを組み込んだものである. この結果より, 各アプリケーションが, DRP-1 の 1 Tile (最大使用 PE 数 $\leqq$ 64) から 2 Tile 程度のリソースで実装可能であることがわかる. 最も PE を消費している部分は, 暗号化の処理のうちラウンド処理をおこなっている部分であり, そのコンテキストが最も遅延が大きいパスを含むコンテキストである.

また, 最も大きい遅延から動作周波数を算出しスループットを計測し, 各暗号処理の DRP-1 上でのスループットと DSP, MPU 上で実行した際のスループットを示す. このスループットは, 共通鍵暗号の場合には 10000 個のブロックを処理した際の最少クロック数から計測し, MD5, SHA-1 の一方向ハッシュ関数は, 512 bit の処理を 10000 回行い, その最小クロック数より算出した.

ここで, アクセラレータとしての比較対象とする DSP は, TEXAS INSTRUMENTS 社の TMS320C6713 を使用した. これは VLIW アーキテクチャを採用した浮動小数点 DSP で, 最大動作周波数 225MHz で動作する. 各々4KB の命令キャッシュとデータキャッシュ, 計 256KB の L2 キャッシュをもつ. C 言語で

記述したソースコードを, 専用に提供されている開発環境 Code Composer Studio C6713 Version2.20.05 により最適化オプションをつけてコンパイルした.

また, DRP-1 と組み合わせる対象と考える組み込みプロセッサには, NEC 社の MIPS 64 アーキテクチャのマイクロプロセッサである VR5500 を用いた. VR5500 は 10 段のパイプライン構造をもつ 2 way out-of-order SuperScalar 型のプロセッサで, 最大動作周波数は 400MHz である. また, 各々 32KB のデータキャッシュと命令キャッシュをもつ. 各暗号処理の評価プログラムは, C 言語で記述し T-Engine 開発環境において MIPS 用 gcc クロスコンパイラを用いてコンパイルした. コンパイル時の最適化オプションには -O3 を用いた.

評価結果より, DRP-1 上に実装した暗号処理は, すべて MIPS の 2 倍から 5 倍の性能を達成しており, アクセラレータとして利用可能な性能を実現していることがわかる. また, アクセラレータとして利用する場合, DSP よりも高速であり, 性能面で有利である. コストの点を考えると, 各暗号処理は DRP-1 の 1-2 Tile 程度のリソースで実現可能であり, 小規模のハードウェアでアクセラレーションが可能であることがわかる. また, 動作周波数は約 30MHz 程度と低いため, 消費電力の面でも有利である.

5. 仮想ハードウェアを用いた協調処理の評価

5.1 協調処理の評価

前節の評価により, DRP-1 の 2Tile 程度でアクセラレーション可能であることがわかった. そこで, 今回は, 前節で評価した NEC 社の VR5500 に, DRP-1 を 2Tile 接続した構成を想定する. CPU と DRP の Tile 間のデータ転送能力, 構成情報の転送方式は, 節 3. に示した数値を用いて評価した.

評価アプリケーションは, IPsec における認証値生成処理である HMAC-MD5-96 と HMAC-SHA-1-96 であり, これらのアプリケーションの内部で最も負荷の高い MD5 および SHA-1 を DRP-1 上で実行するものとする. また, 処理するデータは, 0 バイトから 8192 バイトまでの範囲で, 256 バイト刻みでメッセージサイズを決定し, 乱数によるランダムなメッセージを作成した.

事前のプロファイリングの結果より, コアとなるハッシュ関数の処理時間が全体の 90% 以上を占めることがわかっている. また, 処理するメッセージサイズが大きくなるに従って, ハッシュ関数の処理する割合いが増加する傾向にある.

HMAC-MD5-96 と HMAC-SHA-1-96 を MIPS 上のソフトウェアのみで実行した場合に対する, 協調動作による性能向上率を図 5 に示す. 横軸は処理するメッセージのサイズ [byte] である. 縦軸は協調動作時の処理時間を, ソフトウェアのみで実行した場合の実行時間で正規化した性能向上率である. ここで, DRP-1 上で実行するハッシュ関数の実行時間は, 前節で示した個々の実装結果を利用する. また, 協調動作させることにより, プロセッサと DRP-1 の間でデータ通信が発生するが, 今回は, MIPS, DRP-1 と共有メモリはオンチップバスを介して接続され, DMA 転送を行いデータを通信するものとしている. オンチップバスは 64bit のデータ幅で 100MHz で動作すると仮定し, 上記の処理データの通信に要する時間を見積もった. また, DRP-1 のコンフィギュレーションは既に完了しているものとして, 純粋な処理部分のみの評価を行うことにする.

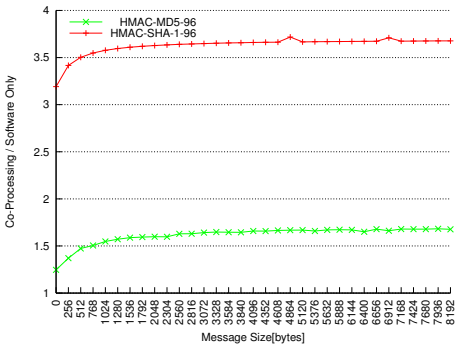

図 5　協調動作による性能向上率

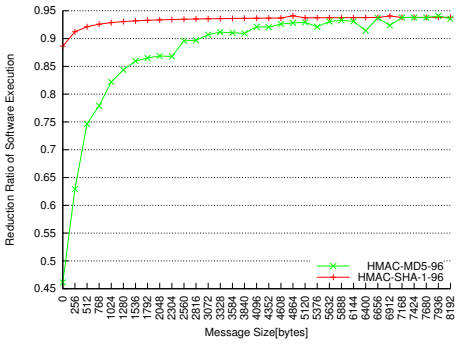

図 6　ソフトウェア処理部の減少率

図 5 から, MIPS と DRP-1 を協調動作させた場合, MIPS のみで実行させた場合に比べて, 性能を向上させることが可能であることがわかる. 特に, 処理するメッセージのサイズが大きくなるに従ってその効果は大きく, HMAC-MD5-96 では約 1.65 倍, HMAC-SHA-1-96 では約 3.6 倍の性能向上を達成した.

次に, MIPS 上のソフトウェアのみで実行した場合の実行時間に対する, 協調動作を行うことによるソフトウェア処理部の実行時間の減少率を図 6 に示す. 横軸は処理するメッセージのサイズを示し, 縦軸は協調処理を行うことによるソフトウェア処理部の実行時間の減少率を示す. 図 6 から, HMAC-MD5-96 では, メッセージサイズが小さい場合には, 50% ほどの削減率に留まっているが, 処理するメッセージサイズが大きくなるほど協調処理の効果は大きく, 3072 バイト以上の場合には 90% 以上のソフトウェアの処理時間を削減している. HMAC-SHA-1-96 は, HMAC-MD5-96 に比べ, SHA-1 に要する時間が大きな割合いを占めるため, 協調処理による影響が大きい. メッセージサイズが小さい場合でも, 90% 以上のソフトウェア処理を削減することが可能であることがわかる.

5.2 アプリケーションスイッチのオーバヘッド

前節の評価結果により, コンテキスト数の制限を考えると, 全ての暗号回路をオンチップに搭載することは現実的ではない. そこで, 各暗号回路のコンフィギュレーションデータを DRP-1 外部のメモリに蓄え, 必要に応じて DRP-1 にロードする仮想ハードウェア機構を用いる. このため, システム全体の評価には, 仮想ハードウェア機構においてコンフィギュレーションに要する

表 2　コンフィギュレーションのオーバヘッド

Application	Config.Size[bit]	Config.Time[μs]
DES (Enc + Key)	331104	51.7
CAST128 (Enc)	104928	16.4
AES (Enc + Key)	201184	31.4
CAST256 (Enc)	207744	32.5
RC6 (Enc)	90240	14.1
MD5	113760	17.8
SHA-1	134560	21.0

時間を含めた評価を行わなければならない. 特に複数の通信相手と IPsec 通信を行うような場合, 頻繁にコンフィギュレーションが発生するため, このようなコンフィギュレーションのオーバヘッドが問題となる.

今回実装した暗号処理のそれぞれのコンフィギュレーションデータのサイズを表 2 の Config.Size に示す. 大きいものでは 40KB 程度のデータを DRP-1 に転送する必要がある. DRP-1 は, 利用していないコンテキストに対して実行中にコンフィギュレーション情報を設定することが可能である. したがって, ある処理の実行中に次の実行に必要なコンフィギュレーションの設定をバックグラウンドで行うことができれば, 実行を妨げることなくアプリケーションの切り替えが可能である.

ここで, DRP-1 のコンフィギュレーションを 64bit, 100MHz のバンド幅で行うことができると仮定する. このときのコンフィギュレーションに要する時間の見積りを表 2 の Config.Time に示す. 各暗号処理に対して, 実行時間が Config.Time 以上であれば, コンフィギュレーション時間を隠蔽することが可能である. しかし, コンフィギュレーションの時間がアプリケーションの実行時間を上回る場合には, 処理をストールしなければならない. この際には, コンフィギュレーションキャッシュをチップ内に置いたり, コンフィギュレーションデータを圧縮することでコンフィギュレーションのバンド幅を向上させる方法が必要となる.

ここで, DES と AES のコンフィギュレーションが連続して発生するケースを考える. このとき, DES の実行時間内に AES のコンフィギュレーションが完了すれば, ストールする必要はなくなる. 表 1 の DES の性能と表 2 の AES の Config.Time から, AES のコンフィギュレーションを完全に隠蔽するために必要な, DES の処理データ量は約 408 バイトである. 同様に, 各暗号処理に対して, コンフィギュレーション時間を完全に隠蔽するために必要となる処理データのサイズを計測すると, 約数 100 から 1500 バイト程度のデータを処理すれば, 次に実行する暗号処理のコンフィギュレーションを隠蔽することができる. この結果より, IPsec 通信を行うにあたって実用的な時間でコンフィギュレーションを行うことが可能であると考えられる.

6. 関 連 研 究

FPGA の大規模化に伴ない, 様々な応用分野で FPGA を用いることによって, 高速性, 柔軟性の面で効率的な暗号処理アクセラレータが開発されている.

University of Southern California's Information Sciences Institute (USC-ISI) の SLAAC プロジェクトでは, Xilinx 社の Virtex XCV1000 を 3 つ搭載した SLAAC-1V ボードで, Triple-DES や AES の高速化を実現している [8]. AES では 571Mbit/sec, Triple-DES では, 116 Mbit/sec のスループットを達成している.

Dandails ら [9] は, FPGA をベースにした IPsec アクセラレータである Adaptive Cryptographic Engine (ACE) を提案している. ACE は, FPGA 上に 5 つの共通鍵暗号アルゴリズムを実装し, 実行時に動的に再構成することで暗号処理を切り替える. 各暗号処理は Xilinx 社の Virtex 上に実装し, ソフトウェアよりも約 4-20 倍の性能を達成している.

これらのシステムは, FPGA を利用することで性能と柔軟性の両面で優れているが, 複数の FPGA を搭載したり, PCI バス経由でホスト PC に接続することを前提としていることから, 組み込み機器への応用は難しいといえる.

7. 結　　論

本稿では, NEC エレクトロニクス社の動的再構成可能プロセッサ DRP-1 を用いた複数暗号処理エンジンの実装を行った. 複数の暗号処理を DRP-1 でアクセラレートし, これらの暗号処理をオンデマンドで動的に切り替えることにより, 少ないハードウェア資源で仮想ハードウェアを実現する.

評価結果より, 各暗号処理は DRP-1 の 1 Tile から 2 Tile 程度の小規模のリソースで実現可能で, DSP や MIPS と比較して高いスループットを達成した. また, MIPS と DRP-1 との協調動作を想定した評価では, 認証値生成処理においてソフトウェアのみで実行した場合と比較して, ソフトウェアの処理を約 90% を削減し, 約 1.6 から 3.5 倍の性能向上を達成した.

しかしながら, 仮想ハードウェアの実現に際して, 動的にコンフィギュレーションを行うことによるオーバヘッドが大きいという問題があり, コンフィギュレーションの高速化手法の検討が重要な課題であるといえる.

文　　献

[1] G. J. M. Smit, P. J. M. Havinga, L. T. Smit, and P. M. Heysters. Dynamic Reconfiguration in Mobile Systems. In *Proceedings of International Conference on Field Programmable Logic and Application (FPL2002)*, pp. 162–170, 2002.

[2] 片山, 甲斐, 吉田, 山田, 塩本, 山中. リコンフィギャラブルプロセッサを用いた 10Gb/s ファイアウォール装置の実現. 第 4 回リコンフィギャラブルシステム研究会 論文集, pp. 67–72, 2004.

[3] M. Motomura. A Dynamically Reconfigurable Processor Architecture. *Microprocessor Forum*, October 2002.

[4] IPFlex. http://www.ipflex.com/.

[5] QuichSilver Technology, Inc. http://www.qstech.com/.

[6] 粟島, 戸井, 中村, 紙, 加藤, 若林, 宮澤, 李. 動的再構成可能チップ DRP の C コンパイラ. 電子情報通信学会技術研究報告 VLD2003-118, Vol. 103, No. 578, pp. 23–28, 2004.

[7] Z. Ye, A. Moshovos, S. Hauck, and P. Banerjee. CHIMAERA: A High-Performance Architecture with a Tightly-Coupled Reconfigurable Functional Unit. In *Proceedings of the International Symposium on Computer Architecture*, pp. 225–235, 2000.

[8] P. Chodowiec, K. Gaj, P. Bellows, and B. Schott. Experimental Testing of the Gigabit IPSec-Compliant Implementations of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board. In *Proceedings of the 4th International Conference on Information Security (ISC2001)*, pp. 220–234, October 2001.

[9] A. Dandails, V. K. Prasanna, and J. D. P. Rolim. An Adaptive Cryptographic Engine for IPsec Architectures. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM2000)*, pp. 132–141, 2000.