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.

動的リコンフィギャラブルプロセッサを用いた IPsec アクセラレータの設計と実装

長谷川揚平† 阿部 昌平† 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, 続々と登場する動的リコンフィギャラブルデバイスは, 複数の回路構成情報をチップ内部に保持するマルチコンテキスト機能や, 外部より構成情報を動的にロードする機能をもつ. これらの機能を有効に利用することにより, 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や, 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実現が可能となる. 本稿では, NEC の動的リコンフィギャラブルプロセッサである Dynamically Reconfigurable Processor(DRP) を用いて, 複数の暗号処理を動的にスイッチすることが可能な IPsec アクセラレータを設計, 実装し, 評価を行った. また仮想ハードウェア実現に向けた設計上の問題点を明らかにし, その解決策についての検討を行った.
キーワード リコンフィギャラブルプロセッサ, DRP, 動的再構成, 仮想ハードウェア, マルチアプリケーション, IPsec, 暗号処理, ネットワークセキュリティ

The Design and Implementation of The IPsec Accelerator with The Dynamically Reconfigurable Processor

Yohei HASEGAWA†, Shohei ABE†, 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 with rapid reconfiguration can hold the multiple configurations inside and load the alternative configuration from the external memory. These functions enable the practical virtual hardware such that implements the huge or mlutiple applications. In this paper, the scalable IPsec accelerator with multiple cryptographic applications has implemented with the NEC's Dynamically Reconfigurable Processor(DRP). Also, we discuss the problems and solutions of its design.
Key words reconfigurable processor, DRP, dynamic reconfiguration, virtual hardware, multi-applications, IPsec, cryptography, network security

1. はじめに

近年, 動的リコンフィギャラブルプロセッサの進歩と普及により, 広い応用分野で利用されることが予想されている. 動的リコンフィギャラブルプロセッサは, 複数のコンテキストと呼ばれる回路構成情報をチップ内部に保持し, これを動作時に切り替えながら利用することのできるプログラマブルデバイスである. これらのデバイスは, 粗粒度のプロセッシングエレメントの接続と機能を動作中に変更することで, ハードウェアの処理能力と, ソフトウェアの柔軟性を優れた面積効率で実現する. 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 [1] [2] [3].

これらの特徴から, 動的リコンフィギャラブルプロセッサの応用分野として, 携帯端末や通信制御機器などでの利用が期待されている [4] [5]. 特に近年では, インターネットなどの通信技術の進歩により, ネットワークに接続することのできる家電製品も登場しており, これらの機器に求められる機能は高機能化がますます進んでいる. 例えば, 音声や動画などのマルチメディア処理は必須の機能となり, さらにはセキュリティの面から通信の暗号化や, 電子認証などのセキュリティ機能も必要となる. しかしながら, これらの処理は大量のデータを扱うため, 非常に高負荷であると同時に, 画像圧縮技術である JPEG2000 や暗号化規格である AES などに代表されるように, 次々と新しい強力なアルゴリズムが登場するという傾向にある.

性能面では, 汎用の CPU のみでは必要とされる性能を満足することは難しく, 専用ハードウェアの利用が現実的である. 一方で専用ハードウェアでは, 様々な通信規格やアルゴリズムに柔軟に対処することは難しく, さらに今後これらの機能が情報家電や民生機器にまで及ぶと, 容易にハードウェアの機能を変更したりすることはできなくなってしまうという問題がある.

このような背景から, 本稿では計算資源の不足する携帯端末や情報家電を対象として, 動的リコンフィギャラブルプロセッサをネットワークセキュリティ技術に応用することを考える. 具体的には, NEC 社の動的リコンフィギャラブルプロセッサである DRP-1 を用いて, IPsec アクセラレータの設計および実装を行った. IPsec は, 現在のインターネットの標準となっている IP に, 様々な暗号アルゴリズムを用いてセキュリティ機能を付加する枠組みである. 次世代の IPv6 では必須の機能となっており, 今後携帯端末や家電機器への需要が高まっていくことが予想される.

本稿は以下の構成をとる. まず 2. 節で動的リコンフィギャラブルプロセッサと, 実際のデバイスである NEC 社の DRP のアーキテクチャについて紹介する. 続いて, 3. 節で今回実装した IPsec の概要について説明する. そして 4. 節で DRP を用いた IPsec アクセラレータの設計および実装について述べ, 5. 節でその評価結果について報告する.

2. Dynamically Reconfigurable Processor

2.1 動的再構成と仮想ハードウェア

近年の動的リコンフィギャラブルプロセッサの進歩はめざましいものである. これらのデバイスは, 従来の FPGA の課題であったプログラマブル配線の増大を, 複数のコンフィギュレーションデータ (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. このような動的再構成の技術は, 面積効率を改善することに加え, プログラマブルデバイスに対してソフトウェア的な概念に基づいた新しい計算システムの構築を可能にした. 特筆すべきは以下の 2 点である.

(1) 複数のアプリケーションを状況に応じて切り替えながら実行する. プロセススイッチングに類似した概念を導入し, 実行しないアプリケーションをメモリに退避させておいたり, 使用頻度の低いアプリケーションはチップ外部に退避させることが可能となる.

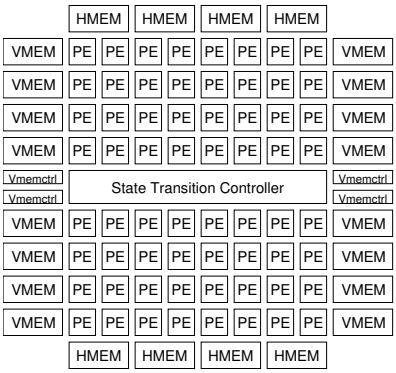


図 1 Tile の構造

(2) 利用できる計算資源の容量を越える巨大なアプリケーションを時分割多重実行する. OS によるメモリスワッピングに類似した概念を導入することが可能となる.

これらの機能を利用することで, 仮想的に膨大な量のハードウェアがあるように見せかけることができる. これを仮想メモリにならって仮想ハードウェアとよぶ [6]. このように, 仮想ハードウェアの実現が可能となると, ハードウェアサイズの制限はなくなり, ソフトウェア的な柔軟性をもったシステムを構築することが可能となる. このような特徴は, 限られた資源で多くの機能を必要とする情報家電での要求によく適合するものである.

本稿では, (1) の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応用例として, 実際の動的リコンフィギャラブルプロセッサを用いて, 複数の暗号処理を必要に応じて切り替えることを可能にする IPsec アクセラレータの実装を行った. (2) のケースについては, 文献 [7] での検討を参照されたい.

以下で, 利用した動的リコンフィギャラブルプロセッサである NEC 社の DRP のアーキテクチャについて説明する.

2.2 DRP アーキテクチャ

DRP [1] は, 粗粒度のリコンフィギャラブルデバイスであり, Tile と呼ばれるリコンフィギャラブルユニットを構成単位とする. 各 Tile は図 1 に示す通り, 8×8 の Processing Element(PE) アレイ, 状態制御を行なうシーケンサである State Transition Controller(STC) から構成される. また, 8bit×256 エントリのメモリ (VMEM) 8 セットとメモリコントローラ 2 セットを Tile の左右にもち, 8bit×8192 エントリのメモリ (HMEM)4 セットとコントローラを Tile の上下にもつ.

各 PE は図 2 に示すように, 8bit の ALU, シフトや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. 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 DRP-1 は, 4×2 個の Tile から構成される Core の周辺部に 32 ビットの乗算器を 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 となっており, 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.

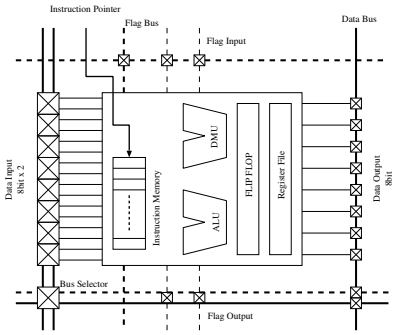


図 2　PE の構造

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.

3. IP security(IPsec)

本稿での実装対象である IPsec の概略を述べる.

IPsec [8] とは, 現在のインターネットの標準プロトコルである IP に, セキュリティ機能を付加するための枠組みである. ネットワーク層のプロトコルである IP にセキュリティ機能を付加することにより, 上位層のサービスに透過的にそのセキュリティ機能を提供することができる. IPsec は IPv4 ではオプション扱いだが, IPv6 では必須の機能である.

IPsec は, いくつかのセキュリティプロトコルを組み合わせることで実現する. IPsec において, 利用される代表的なセキュリティプロトコルは, 以下の Authentication Header(AH) と Encapsulating Security Payload(ESP) である. これらのプロトコルを利用することで, データの完全性や機密性の確保や, 送信元の正当性を検証することが可能となる.

- **AH(認証ヘッダ)**: AH は, Message Authentication Code(MAC) を利用してデータを認証するためのプロトコルである. IPsec では, 一方向ハッシュ関数を使用して MAC を生成する Hased-based MAC(HMAC) アルゴリズムを使用する. AH では送信者側で認証データ (MAC) を生成し送信することで, 受信者側でデータの完全性と送信元の正当性を検証する. AH で必須の認証アルゴリズムとして, HMAC-MD5-96 と HMAC-SHA-1-96 がある.
- **ESP(暗号ペイロード)**: ESP は, 共通鍵暗号を利用してデータの暗号化を行うためのプロトコルである. IP パケットのペイロード部を暗号化することで, データの機密性を保証する. ESP の暗号アルゴリズムとして, DES-CBC をサポートすることが必須となる. これ以外の暗号アルゴリズムはオプションである. DES 自体は脆弱性が報告されており, 2001 年には次世代暗号標準化規格 (AES) [9] として Rijndael が選定されたことから, AES やそれに準ずる暗号処理の実装が必要である.

IPsec 通信を行うためには, 通信相手との間で Security Association(SA) と呼ばれる論理的なコネクションを確立する. SA は通信を行うトラフィック毎に確立され, トラフィック情報, 暗号アルゴリズム, 認証アルゴリズム等のトラフィックに適用するセキュリティ情報を含む. 自動鍵管理プロトコルを使用した場合, 対象パケットの受信を契機に, 自動的に通信相手とネゴシエーションを行い, 鍵を交換して SA を確立する. IPsec では, このような自動鍵管理プロトコルとして Internet Key Exchange(IKE) を規定している.

IPsec において, セキュリティ機能とともにトンネリング機能を利用することで, 公衆網を使用して拠点間を安全に接続する Virtual Private Network(VPN) を構築することが可能である.

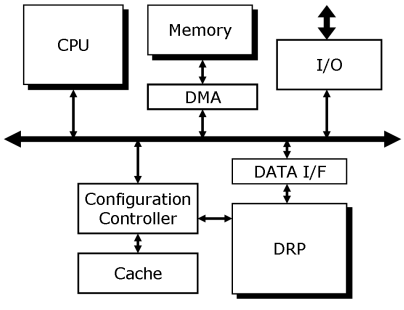


図 3　想定するシステムの構成

4. 設計および実装

4.1 想定するシステム

本稿で想定するシステムを図 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 を複数個結合した協調動作システムである. 従来の専用ハードウェアと 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の開発コストが増大するという問題があった. 本システムでは, 専用ハードウェアにかわり, DRP のもつ高い柔軟性を活かすことで, 従来の SoC の課題に対処する.

図 3 のシステムでは, DRP のコンフィギュレーションデータは隣接する Configuration Controller を経由してロードされる. DRP の再構成は Configuration Controller によって制御され, 仮想ハードウェア機能によって複数のアプリケーションを状況に応じて実行したり, 巨大なアプリケーションをシステムの動作を妨げることなく実行することが可能となる. また, Configuration Controller はコンフィギュレーションデータを高速転送するためのキャッシュを保持し, これを制御する.

また, メモリと CPU, DRP 間のデータ通信はバスを介して行われ, DMA 転送をサポートするものとする.

4.2 実験環境

本稿では, 上記のシステムでの IPsec アクセラレータの動作を想定し, DRP 評価ボードを用いてエミュレーションによる動作検証を行った. 図 4 に DRP 評価ボードの構成を示す.

本システムにおいて, 暗号化処理のアクセラレーションを行う DRP-1 コアは, 入出力を制御する FPGA と 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 コアにコンフィギュレーションコードをロードする際などに使用される. また, DRP-1 コアと 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

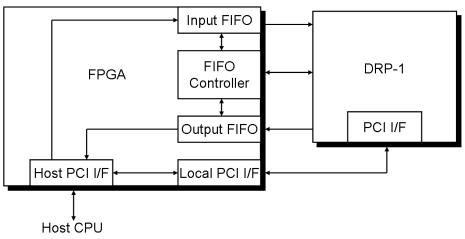


図 4　DRP 評価ボードの構成

して, ホストプロセッサと接続される. DRP-1 の再構成の制御や, 入力 FIFO への値の書き込み, 出力 FIFO からの読み出しは, ホスト CPU から PCI バスを介して行われる.

なお, 今回の実装では, DRP 評価ボードを用いて, 想定するシステムの動作検証を行なった. IPsec の処理のうち負荷の高い暗号化処理および認証値の生成処理を DRP-1 上に実装し, スループットを計測した. その他のパケット処理および DRP-1 のコンフィギュレーションの制御はホスト CPU 上で処理を行う. 今回は, システムとしての定量評価には至らず, システムのボトルネック部分の解析および改善策の提案に留めた. 5. 節において, 評価結果を示す.

4.3　データフロー

本システム上での主な処理の流れは以下の通りである. ホスト CPU は, ネットワークインタフェースよりパケットを受け取り, 適用するセキュリティポリシー (IPsec を適用するか) を判別する. IPsec を適用する場合には, ホスト CPU は SA のデータベースの探索を行い, 対象パケットに適用する SA を探し出す. 適用する SA が見つかった場合には, その SA で示される処理を DRP-1 で実行するための準備を行う. 例えば, ESP における暗号化アルゴリズムとして DES-CBC が指定された場合, ホスト CPU は DRP-1 のコンフィギュレーションライブラリより DES-CBC のコンフィギュレーションデータを取りだし, PCI バス経由でコンフィギュレーションデータを書き込む. DRP-1 自体は, 動作時に部分再構成をすることが可能だが, 今回は実装上の制限により, DRP-1 の再構成の間はすべての処理を停止するものとした.

続いて, 処理するデータを DRP-1 の入出力を制御する FPGA の FIFO に書き込み, DES の処理を開始する. 処理結果は FPGA の出力 FIFO に書きこまれ, ホスト CPU はこれを取り出し IPsec パケット処理を行う. IPsec パケット処理後のパケットは, ネットワークインタフェースないし上位層のアプリケーションに渡される.

4.4　暗号アルゴリズムの実装と開発環境

本システムでは, 複数の暗号アルゴリズムや認証アルゴリズムをサポートする. これらのアルゴリズムを, C ベースの動作記述言語を用いて記述し, DRP コンパイラおよびその統合開発環境 Musketeer [10] を用いてコンパイルを行った. DRP コンパイラの生成した DRP-1 のコンフィギュレーションコードをデータベースに登録し, この中の任意の暗号アプリケーションを選択し DRP-1 上で実行できるようにした. また, 新たに暗号アプリケーションのコンフィギュレーションコードを用意し,

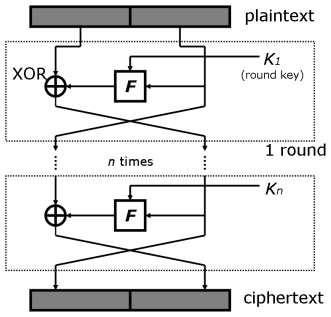


図 5　Feistel ネットワーク構造

データベースを更新することで, 容易に対応アルゴリズムを追加することができる. さらに, 本システムでは, 暗号化処理を状況に応じて DRP-1 で実行するか, ソフトウェアでそのまま実行するかを選択することができようにもなっている.

今回の実装では, DRP コンパイラのサポートする手動スケジューリングモードでの実装を行った. DRP コンパイラは状態に基づくコンテキスト分割 [10] を行うが, このモードではクロックの挿入およびコンテキストの切り替えを手動で設定することが可能である. また, DRP コンパイラはコンテキストサイズの不一致や, リソース制約を改善するため, 面内多状態という概念を導入し, 複数の状態を 1 コンテキストにマッピングする.

4.5　暗号化アルゴリズムの典型的な実装

上記の開発環境を利用して, DRP-1 上に実装した暗号化アルゴリズムは, DES-CBC, AES-CBC, CAST128-CBC, CAST256-CBC, RC6-CBC である. また, 認証アルゴリズムとして HMAC-MD5-96, HMAC-SHA-1-96 の実装を行った.

DES をはじめとする多くの対称鍵ブロック暗号は, 64 ビットないしは 128 ビットのまとまったデータを, 1 ブロックとして暗号化を行う. また, 多くのブロック暗号は, 図 5 に示す Feistel ネットワーク構造を採用している. Feistel ネットワーク構造をもつブロック暗号では, 入力された平文ブロックデータを左右 2 個のサブブロックに分解する. 片方のサブブロックにユーザ鍵から生成したラウンド鍵を使って攪乱処理を行った後, F 関数によって変換を行い, 他方のサブブロックとの排他的論理和をとる. ここまでの処理を 1 ラウンドとし, これを反復することで Feistel 構造を形成する. DES は 16 回のラウンド処理を行うアルゴリズムであり, CAST256 や RC6 など, 最近の 128 ビットブロック暗号も, 32 ビットの 4 つのブロックに分割し, 同様の処理を繰り返す. 復号はこの反復を逆順に繰り返すことで達成される.

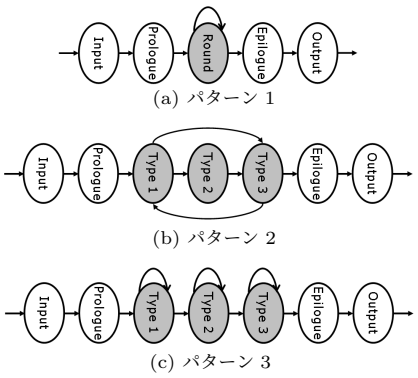
本システムでサポートする暗号アルゴリズムは, 以下の基本方針に従って実装した.

(1)　1 ラウンドの処理を 1 クロックで実装

各ラウンド間にはデータ依存性が存在することから, 1 つのラウンドを 1 クロック (1 コンテキスト) で実装した.

(2)　各ラウンドは繰り返し (iteration) 構造を採用

多くの暗号アルゴリズムで採用される S-box とよばれるテーブルを, DRP-1 上では分散メモリである VMEM で実現している. このため, VMEM はメモリ参照に 1 クロックの遅延を要する. また, クリティカルパスの増大も問題となるため, ラウン



(a) パターン 1

(b) パターン 2

(c) パターン 3

図 6　暗号アルゴリズムの実装

ド間のループ展開は行わない.

以上の基本方針に従い, DES の場合には, 1 ラウンドが 1 コンテキストにマッピングされ, そのコンテキスト内で 16 回処理を繰り返すこととなる (図 6(a)).

CAST128 や CAST256 のケースでは, 複数のパターンのラウンド処理を交互に繰り替えすことから, それらのラウンドをそれぞれ単独のコンテキストにマッピングし, 複数のコンテキストでループを形成する構成をとった (図 6(b)).

また, MD5 や SHA-1 などの一方向ハッシュアルゴリズムも, 同様の反復構造をもち, 複数のラウンド処理のパターンが存在する. これらは, 1 つのラウンド処理を一定回数繰り替えすため, ラウンドの種類毎にループを形成する構造をとった (図 6(c)).

RC6 のように, 1 ラウンドに要する処理が多く, 1 ラウンドを 1 クロックで実行するには難がある場合には, クリティカルパスとラウンド数とのバランスを考慮し, 2 クロック実行に分割を行った.

5. 評　　価

DRP-1 上に実装した共通鍵暗号, および認証値計算で用いられる一方向ハッシュアルゴリズムの実装結果を表 1 に示す.

表 1　各アルゴリズムの DRP-1 上への実装結果 (暗号化処理のみ)

	Required Resources				Throughput[Mbps]	
name	Context	Max PE	All PE	Vmem	DRP-1	DSP
AES	12	129	452	32	344.0	16.9
DES	6	66	236	10	113.7	84.2
CAST128	6	31	137	20	104.7	95.4
CAST256	11	87	341	21	38.4	36.1
RC6	6	71	188	8	149.9	96.6
MD5	13	52	370	14	173.8	135.5
SHA-1	10	63	446	24	193.4	44.0

5.1　使用リソースの評価

まず, 各アルゴリズムで使用した使用リソースについて考察する. 表 1 の結果は, DRP コンパイルフロー [10] における配置配線後の使用コンテキスト数, 最大使用 PE 数, 総使用 PE 数, 最大使用 VMEM 数を示す. 最大使用 PE 数 (Max PE) 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 (All PE) 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 の結果は暗号化処理モジュールのみの評価であり, 鍵スケジューラおよび復号化モジュールの結果は含まれていない.

この結果より, 多くのアルゴリズムで DRP-1 の 1 Tile (最大使用 PE 数 $\leqq$ 64) 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, 今回の実装は, 処理するブロック間でフィードバックをとる CBC モードでの実装を対象としている. セキュリティ強度は低下するが, 処理速度を重視するなどの理由で, フィードバックのない ECB モードなどを利用する場合には, 並列に複数のブロックを処理することが可能である. 今回実装した暗号化モジュールを複数個マッピングすることで, 並列に処理することができる.

5.2　DSP との性能比較

実装したアプリケーションの性能の評価を行うため, DRP-1 上に実装したコードと同様の C のソースコードを記述し, DSP 上で動作検証を行った.

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 225 MHz 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 によりコンパイルした. すべてのコードを最適化オプションを付けてコンパイルを行っている.

評価方法として, ブロック暗号の場合には 10000 個のブロックを処理した際の最少クロック数からスループットを計測し, MD5, SHA-1 の一方向ハッシュアルゴリズムの場合には, 512 ビットの処理を 10000 回行い, その最小クロック数よりスループットを計算した. 表 1 に示す評価結果より, 各アルゴリズムにおいて, DRP-1 上に実装した暗号化処理は, DSP を上回る性能を達成することができた.

5.3　仮想ハードウェア実現に向けた検討事項

以上のように, DRP-1 上での暗号アプリケーションの実装結果を示したが, IPsec ではこれらのアプリケーションを状況やユーザの要求に応じて切り替える必要がある. したがって, 仮想ハードウェアの機構を備えることが必要不可欠となる.

しかしながら, 今回の実装では, 実装上の制約から, DRP 評価ボード (図 4) において PCI バスを経由したデータ転送を行うため, データ転送のバンド幅に制約された設計となっている. 特に, 現状ではコンフィギュレーションデータを外部から読み込み設定するまでの時間がボトルネックとなり, システムの性能を大きく制限することは明らかである.

以下で, 図 3 に示す DRP を用いた仮想計算システムにおいて必要な機能を検討し, 今後の課題とする.

最もボトルネックとなる部分は, 明らかにコンフィギュレーションデータのロード時間である. 今回実装したシステムでは, 暗号処理を切り替える際に, DRP-1 を一度リセットしてからコンフィギュレーションデータをロードし, 再構成を行う必要がある. DRP-1 は粗粒度のアーキテクチャを採用しているた

め, 一般的に通常の FPGA よりもコンフィギュレーションデータの量は少ないといわれている. 今回実装した暗号アプリケーションのコンフィギュレーションデータのサイズを表 2 に示す. なお, AES, DES に関しては, 鍵スケジューラを含むものである. しかしながら, 今回の実装の前提では, DRP-1 におけるコンフィギュレーションデータのロードは, 32bit PCI バス経由で行われるため, それでも数 1000 クロックを必要とする. このため, 必然的にコンフィギュレーションデータのスイッチ時間を短縮するための機構が必要となる.

表 2　コンフィギュレーションデータのサイズ [bytes]

AES	29004	DES	39052
CAST128	14188	CAST256	22920
RC6	10352		
MD5	12520	SHA-1	16472

また, コンフィギュレーションのバンド幅を確保することが最も重要であるが, さらに以下のような工夫が必要となる.

- 構成情報の圧縮による高速化

われわれは既に DRP-1 におけるコンフィギュレーションデータの圧縮手法についての検討を行っている [11]. この手法を用いる場合, DRP-1 の数% ほどの資源を利用して復号回路を付加することにより, 構成情報を 50% から 60% 程に削減できることが示されている.

- DMA 転送機能の付加

DRP-1 に DMA 転送機構を付加することで, 構成情報のロード時間を短縮することが可能である. DRP 評価ボードでは, ホスト CPU から PCI バス経由で構成情報を DRP-1 に書き込んでいる. ここで, DMA 転送機構を設けることで, データの入出力と構成情報のロードの高速化を達成する. さらに DMA 転送が可能となった場合, 再構成を行っている間に CPU のソフトウェア側で処理を行うことも可能となり, アプリケーションスイッチ時間を隠蔽することが可能となる.

- コンフィギュレーションキャッシュの付加

チップ内部に大規模なバックアップメモリを置くことで, 高速大容量転送を行う. DRP-1 では, 文献 [12] において, コンテキストキャッシュによるコンテキストの仮想化が検討されている.

- コンテキストのプリフェッチ機構の付加

ある暗号アプリケーションが動作中に, 使用していないコンテキストや使用が終了しているコンテキストに対して, 次に実行するコンテキストをプリフェッチしてロードすることで, 効率的な再構成を実現する.

このように, 再構成の高速化手法を組み合わせることによって, 再構成のオーバヘッドを削減し, 仮想ハードウェアの実現可能性を広げることが可能である. しかしながら, 頻繁な構成情報のロードは, やはり性能を損ねることになってしまうため, ホスト CPU 側で再構成のスケジューリングを行い, なるべく構成情報のロードの頻度を抑える枠組みが必要である.

6. 結　　論

本稿では, NEC 社の動的リコンフィギャラブルプロセッサ DRP のプロトタイプチップである DRP-1 を用いて, 複数の暗号アプリケーションをサポートする IPsec アクセラレータの実装について報告した. DRP-1 上に実装した暗号アプリケーションは, DSP 上で動作させたアプリケーション以上の性能を達成することができた.

また, 動的に複数の暗号アプリケーションを状況に応じてスイッチさせることにより, 動的リコンフィギャラブルプロセッサを用いた仮想ハードウェア機構の実現について検証を行った. また, 仮想ハードウェアの実現に関しては, まだアプリケーションスイッチに要する時間がボトルネックになるという問題がある. 本稿では, その問題に対する対応策についての若干の検討を行った. これらの点については, 今後の課題として引き続き検討していく.

近年では, インターネットの発達により, その応用範囲は移動端末や民生機器にまで及んでいる. これらの情報家電に要求される機能は非常に多様化しかつ高性能なものである. こうした中で, ハードウェアの高速性と, ソフトウェアの柔軟性を併せもつ, 動的リコンフィギャラブルプロセッサの特徴を活かすことができると考えられる.

謝　　辞

DRP およびその開発環境を提供して頂き, 本研究に対する多くのアドバイスをしてくださった NEC エレクトロニクスおよび NEC システムデバイス研究所の皆様に深く感謝致します.

文　　献

[1] M.Motomura. A Dynamically Reconfigurable Processor Architecture. *Microprocessor Forum*, October 2002.
[2] IP FLEX DAP/DNA. http://www.ipflex.com/.
[3] QuickSilver Technology. http://www.quicksilvertech.com/.
[4] G. J. M. Smit, P. J. M. Havinga, L. T. Smit, P. M. Heysters. Dynamic Reconfiguration in Mobile Systems. *Proceedings of International Conference on Field Programmable Logic and Application(FPL2002)*, pp. 162–170, 2002.
[5] 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.
[6] X.-P. Ling, H. Amano. WASMII: A Data Driven Computer on a Virtual Hardware. *Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines(FCCM'93)*, pp. 33–42, 1993.
[7] H.Amano, M.Suzuki, T.Inuo, H.Kami, T.Fujii. Techniques for Virtual Hardware on a Dynamic Recongurable Processor -An approach to tough cases-. *Proceedings of International Conference on Field Programmable Logic and Application(FPL2004)*, pp. 464–473, 2004.
[8] S.Kent, R.Atkinson. Security Architecture for the Internet Protocol. *RFC 2401*, 1998.
[9] NIST. Advanced Encryption Standard(AES). *FIPS PUBS 197*, 2001.
[10] 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.
[11] T.Kitaoka, H.Amano, and K.Anjo. Reducing the Configuration Loading Time of a Coarse Grain Multicontext Reconfigurable Device. *Proceedings of International Conference on Field Programmable Logic and Application(FPL2003)*, pp. 171–180, 2003.
[12] 犬尾, 西野, 中谷, 梶原, 紙, 戸井, 粟島. 動的再構成プロセッサ DRP を用いた関数オフローダの試作. 電子情報通信学会技術研究報告 VLD2003-121, Vol. 103, No. 578, pp. 41–46, 2004.