SDN/NFVにおける高位合成を用いたFPGAリソース活用の一検討

保米本徹[†1]　西山聡史[†1]　右近祐太[†2]　片山勝[†1]　行田克俊[†1]

昨今ネットワーク装置を汎用計算機等によって置き換えるNFV（Network Functions Virtualization）の進展がめざましい．筆者らは通信キャリア固有かつ，高スループット・低レイテンシが求められるネットワーク機能の仮想化のため，FPGA（Field Programmable Gate Array）を搭載したハードウェアアクセラレータと高位合成技術を用いたFPGAの構成書き換えにより，ソフトウェア技術者がNFVソフトウェアとアクセラレータを一括でメンテナンスするフレームワークに着目している．本稿ではそのユースケースとして，10Gイーサネット伝送路暗号化回路の回路規模とコード量を従来設計と高位合成設計で比較し，通信キャリア固有機能NFV化への本フレームワークの適用性を示した．

A Study for Application of FPGA Resources on SDN/NFV with High-level Synthesis Design

Toru Homemoto[†1]　Satoshi Nishiyama[†1]　Yuta Ukon[†2]　Masaru Katayama[†1]
Katsutoshi Koda[†1]

Network Functions Virtualization (NFV) has been proposed to make conventional network appliances replaced by general-purpose compute nodes, achieving network cost reduction. We consider designing NFV function circuit on Field Programmable Gate Array (FPGA) via high-level synthesis engine, which allows software developers co-design NFV software and NFV circuit configuration both with imperative programming language such as C. In this paper, we compare the circuit scale and the code size of 10G Ethernet encryption module between the conventional design method and with the high-level synthesis, indicating effectiveness of applying the framework into the network of Telecom service provider.

1. はじめに

SDN（Software Defined Networking）およびNFV（Network Function Virtualization）技術に代表される，ネットワーク仮想化の検討が進んでいる．これら技術にはネットワーク構造・経路を遠隔地から動的に変更することで維持管理を迅速かつ簡便に行えるようにすることのほか，役割別の専用機器群で構成されていたネットワーク機器を汎用装置上のソフトウェアで実現することで部品共通化によるコスト低減が行えることが期待されている．

SDN/NFV技術は仮想化されたサーバ資源を提供するクラウド事業者を中心に商用環境への導入が進められてきた．現在では，ネットワークの既存インフラを大量に保有し，それらを置き換えることの影響が大きい通信キャリアにおいても，厳しいコスト削減への要求に応える手段としてSDN/NFV技術の適用が盛んに検討されている[1].

2. 通信キャリアネットワークの仮想化方式

2.1 データセンタネットワークにおけるSDN

クラウド事業者などが持つデータセンタ（DC）はサーバ群とそれらをつなぐネットワーク機器が大量に配備された施設である．サーバリソースの仮想化が進むDCネットワークには，例えばその仮想サーバ間や，仮想サーバとクラウド利用者の拠点の間をつなぐ回線を同じく仮想的に，すなわち物理的な配線トポロジの上に論理的な配線構造を重畳する形で提供する役割などが求められており，これにはOpenFlow[2] などに代表されるSDNのアプローチが適用可能である．OpenFlowはトラヒックをヘッダ情報で区別されるフロー単位で取り扱い，フロー単位での操作を行うネットワークスイッチ（OpenFlow Switch）と，スイッチ群を共通化されたプロトコルでコントローラ（OpenFlow Controller）から制御するものであり，動的なネットワーク接続構造の制御やVPN（Virtualized Private Network）の構築に用いることで先述の役割を担うことが可能である．

これに加えて，スイッチ機器内で実際のパケット処理を行うASIC（Application Specific IC）の制御APIをオープン化し，ユーザが当APIを利用してネットワークスイッチを開発できるようにする取り組み [3] などが進んでいる．これはネットワークスイッチなどのハードをコモディティ化されたホワイトボックス（White Box Switch, WBS）として取り扱うことを可能とする．WBS上にOpenFlowなどのプロトコルに対応したOS（Operating System）を搭載した形で開発されるスイッチを用いることで部品の共通化によるコスト減の効果が期待できる．

2.2 通信キャリアネットワークにおけるNFV

通信キャリアのネットワーク基盤においては，DCネットワーク同様に仮想的なネットワーク構造を提供する機能のほかに，通信キャリア特有のサービスなどを提供するための固有機能の仮想化が検討されている．NFVは従来専用ハードウェア上に実装されて提供されてきたこれら機能を

†1 NTT ネットワークサービスシステム研究所
NTT Network Service System Laboratories
†2 NTT デバイスイノベーションセンタ
NTT Device Innovation Center

汎用サーバ上に実装し，動的に機能の変更や保守を行えるようにする取り組みであり，ETSI（the European Telecommunications Standards Institute）配下の NFV ISG（Industry Specification Group）にて要求条件の策定が進んでいるとともに，その仕様も文献[4]のような形で公開されている．個々の機能単位での検討も進行しており，モバイルネットワークにおいて認証や課金等を行う基盤となる EPC（Evolved Packet Core）の仮想化[5] や，FTTH などのブロードバンド回線における BRAS（Broadband Remote Access Server）のような利用者認証機能の仮想化[6] などが例として挙げられる．

2.3 汎用計算ノード構成の限界

しかしながら，ネットワーク機能には高スループット・低レイテンシが求められるケースがしばしばあり，前述の汎用の計算ノードとソフトウェアをベースとした NFV では性能が不足する場合がある．この課題は，侵入検知・防御システム（Intrusion Detection System (IDS) / Intrusion Protection System (IPS)）における内部処理の一例である DPI（Deep Packet Inspection）の他，長距離光伝送に必要な伝送路符号化，デジタル変調など，大きな演算量を必要とし，概ねスループットでは 10Gbps から 100Gbps を超える処理速度が要求される場合に顕著である．文献[7]は前者を Graphic Processing Unit (GPU) の援用によって高速化した検討例であり，文献[8]では後者を専用ハードウェア上に実装していることから，両者ともに CPU のみでの高速処理は困難な領域といえる．

2.4 ハードウェアアクセラレータの利用

これら通信キャリア特有かつ，高スループット・低レイテンシを要求する機能を，先述の WBS などの量産 ASIC に搭載し，共通化された API から呼び出せるようにすることは，通信キャリア以外のマーケットでの利用頻度が低いことから製造側にとっての利点が少ない．

文献[9] ではこの課題に対し， CPU / GPU / NPU (Network Processing Unit) など並列度が異なるプロセッサ，FPGA (Field Programmable Gate Array) などの再構成可能デバイス，ASIC (Application Specific IC) ベースの専用ハードなどで構成された HWA（HardWare Accelerator）群を定義し，NFV に求められる速度や処理の性質に応じて，処理に適した HWA を割り当てることで性能を担保するアーキテクチャを提案している．

2.5 再構成可能アクセラレータを用いた NFV

文献[9] の示す構成においても，専用ハードによってしか達成されない機能については，予め必要な種類の専用 HWA を，同時に使用される最大数まで機器に装備しておくことが必要であり，装置のコスト高の原因となる．

筆者らは高速伝送に適した並列化アルゴリズムを FPGA に実装することによって，符号化機能を内蔵した汎用光インタフェースを利用して仮想的に長距離高速光伝送を構成する方式を提案・実装評価しており[10]，これまで専用ハードが必要であった部分を FPGA などの再構成可能なハードウェア上で実現できる場合が増えているといえる．

ゆえに，通信キャリアのネットワークにおける固有機能についても，汎用の計算ノードと FPGA などの再構成可能デバイスを用いたハードウェアアクセラレータ（仮に再構成可能アクセラレータと呼ぶ）を組み合わせた装置によれば，その多くを仮想化することができ，その結果装置の集約，専用部品を削減することによるネットワーク全体の低コスト化を達成できるものと考えられる．この考え方を用いて，PE ルータ（Provider Edge Router, 通信キャリアと顧客のネットワークの間に設置された特殊なルータ）と長距離光伝送装置をひとつの汎用装置上に組み込んだ集約装置の構成イメージを図 1 に示す．

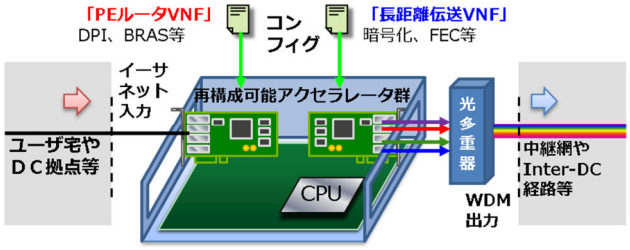

図 1　再構成可能アクセラレータを用いた PE ルータ VNF と長距離伝送 VNF の一体構成

3 章では，この汎用の CPU，その上で動作するソフトウェア（ホストソフトウェアと呼ぶ），再構成可能アクセラレータとそのコンフィグレーション（アクセラレータソフトウェアと呼ぶ）の 4 つの組み合わせを用いて，通信キャリア固有機能のうち高スループット・低レイテンシが求められる機能の NFV 化を実現する領域について取り扱う．

3. 再構成可能アクセラレータによる通信キャリア固有機能の NFV

3.1 ソフトとアクセラレータのライフサイクルの差

通信キャリア固有のネットワーク機能仕様や伝送規格はこれまで長期間（10 年以上）保守されることが多かった．一方，汎用のサーバ機器の保守期間は数年程度が一般的であるため，汎用サーバ機器上に VNF の形で通信キャリア固有機能を実装した場合，ハードウェア更新の際に VNF ソフトウェア群を移植する必要性が生じる．

この時，ホストソフトウェアを例えば C 言語などの命令型プログラミング言語で記述されているものと仮定する．OS や言語から利用するライブラリ群が新しいハードウェアに対応していれば，ソースコードから新しいハードウェア向けのバイナリを再コンパイルすることで同様に動作する場合が多く，更新時の移植は容易と考えられる．

一方，現在利用できる再構成可能アクセラレータ向けのアクセラレータソフトウェアは，例えばその上に搭載され

た FPGA のコンフィグレーションを VHDL, Verilog HDL などのハードウェア記述言語で記述することが一般的である．この記述方法はソフトウェアプログラマにはまだ一般的ではないほか，アクセラレータ上の配線構造や搭載部品を意識（ピンアサイン）した設計や，FPGA 特有のハードマクロ（メモリブロック，DSP，高速 I/O）の呼び出し等が含まれることから，FPGA を用いた再構成可能アクセラレータ上が新規格に更新される際のコンフィグレーションの移植は，コードレベルでの変更が多数発生し，図 2 に示すようにハードウェア更新時のコストが大きくなる恐れがある．

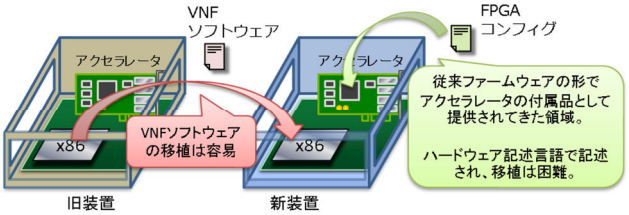

図 2　再構成可能アクセラレータを用いた VNF におけるハードウェア更新時の課題

3.2 ソフト・アクセラレーター括管理のフレームワーク

前節の課題を解くためには，例えば接続されるハードウェアの詳細を隠蔽して抽象化したレイヤで，ホストソフトウェアとアクセラレータソフトウェアを一括してキャリア固有機能の維持期間メンテナンスしていくフレームワークが必要と考えられる．そのフレームワークの候補として以下のようなものが挙げられる．

(1) 統一言語方式

文献 [11] で述べられている P4（Programming Protocol-independent Packet Processors）言語はプロトコルやハードウェア・ソフトウェアの違いを意識することなく，ネットワーク処理をプログラミングできる言語であり，図 3 のように前述の CPU，FPGA の区別をつけずに記述を行えるとしている．

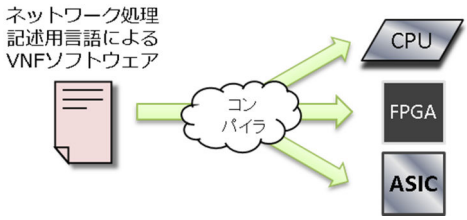

図 3　統一言語方式による管理イメージ

さらに，P4 言語で動作を記述できるデバイスとして[12]に挙げられる PISA（Protocol Independent Switch Architecture）構造を採用したプログラマブルなスイッチチップが開発されている．これは現在の Ethernet スイッチチップと同等かそれを凌駕する性能とプロトコルを選ばない再プログラム性を同時に確保できるとしており，暗号化や伝送路符号化などの演算量の高い処理も効率良くサポートできる可能性がある．

(2) 高位合成方式

もう一つの方式として，図 4 のように，高位合成エンジンを介することで FPGA のコンフィグレーションをホスト側ソフトウェアと同様の命令型プログラムの形で記述したアクセラレータソフトウェアとしてメンテナンスする方法が考えられる．

当方式によれば，両ソフトウェアの境界は明示的に分かれているものの，ハードウェア記述言語に比してアクセラレータソフトウェアの移植をより容易にできることが期待される．また，既に FPGA 用の高位合成ツールが市販されていることから実現性も高い．

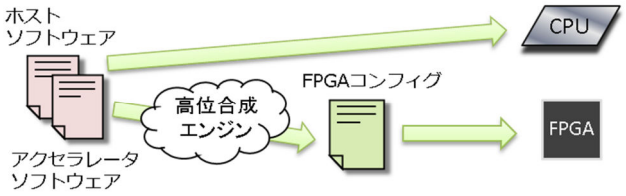

図 4　高位合成方式による管理イメージ

3.3 フレームワークの比較

3.2 節の方式群を比較すると，性能面では両者とも 10G から 100Gbps を超える処理速度に対応可能だが，電力効率や面積効率などの面では ASIC ベースでのプログラマビリティを実現することを謳う統一言語方式が有利と考えられる．一方，FPGA を利用する高位合成方式はロジック単位での再構成が可能であり，FPGA の容量が許す限り未知の規格への対応が可能である点で柔軟性がやや高いといえる．最後に，実用可能時期の観点では高位合成方式であれば既に対応製品が存在する利点がある．これら比較を表 1 に示す．

表 1　NFV ソフトウェア記述フレームワークの比較

方式	性能と電力効率	柔軟性	実用可能時期
統一言語方式	速度・電力効率とも良い	良い	開発中
高位合成方式	速度は十分だが，電力効率に劣る	最も良い	製品が存在

3.4 検証すべき項目

前項の比較から，直近で適用性の評価が可能である高位合成方式に着目する．高位合成方式を実現するためには，FPGA のコンフィグに高位合成エンジンを用いることによる性能の低下や回路規模の増加が HDL による記述に対して現実的な範囲に収まることが重要である．

近年市販されている論理合成・配置配線までを一括で提供する EDA ツールを用いたときの性能例として，HDL による記述に対し多くの場合で 10%～数 10%の性能ペナルティのみで実装できる例[13]が示されている．しかしながら，この時のコード移植性がどの程度改善するかは未評価と考えられる．

4 章では特に通信キャリア向けの代表的なユースケースを一つ挙げ，性能と移植性の 2 軸で評価した結果を示すことで，高位合成エンジンを通して VNF ソフトウェアをメンテナンスする方式の有効性を見積もる．

4. ユースケース評価

4.1 伝送路暗号化

高位合成方式によってソフトウェア・アクセラレータを一括管理する通信キャリア向け機能の VNF において，その性能と移植性は対象とする VNF の内部処理の種類によって大きく異なることが考えられる．そこで，2.5 節で示した集約装置のうち，一番スループットへの要求が高い長距離伝送装置の機能に的を絞って評価を実施する．なぜなら，今後の汎用 CPU の性能向上に伴って，速度やレイテンシが要求されない機能から順にソフトウェア実装が可能となることが予想されるためである．

長距離伝送は光伝送によって行われることが多く，その媒体となる光ファイバは，電磁的なエネルギーを外部に放射しないことから盗聴が困難と思われていたが，昨今は量子暗号への期待の高まり[14]が示すように光ファイバ盗聴が現実の懸念となりつつある．これに対応するように文献[15] に示される政府機関のネットワークの要件は VPN 回線の暗号化を定めている．この要件を充足するネットワークを構築するためには，例えば図 5 のように長距離伝送の物理リンク単位で暗号化する方法があり，その製品も既に市販されている[16].

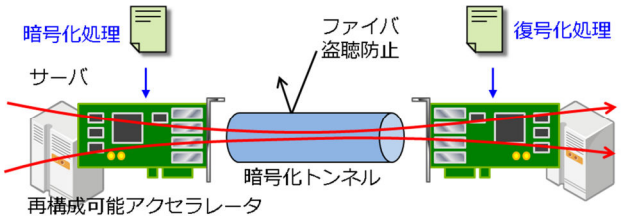

図 5　伝送路の包括的な暗号化を行う VNF

暗号化・復号化処理は文献[17]などに示されるように，適切な暗号利用モード（CTR もしくは GCM）等を選ぶことでパイプライン化・並列化が容易であり，クロック周波数の面で不利となる FPGA においても 10G から 100Gbps オーダの速度が実現できる．このことから長距離光伝送用 VNF のモデルケースとして選定し比較に用いた．

評価用の暗号化アルゴリズムとして，仕様がオープンであり C 言語の実装が提供されている Camellia [18] を用いた．Camellia（鍵長 128bit）の外部インタフェースは図 6 のように，鍵と同時に 128bit の平文を入力することで同じく 128bit の暗号文が得られるようになっている．またその内部処理は図 7 に示されるような Feistel 構造を繰り返し実施するものとなっており，Feistel 構造をレジスタを介して直列接続することで，パイプライン化による高速処理が容易に実装できる．また，暗号化モードと復号化モードは同一の Feistel 構造の計算回路を再利用し，入力データの入れ替えによって切り替えることができる．

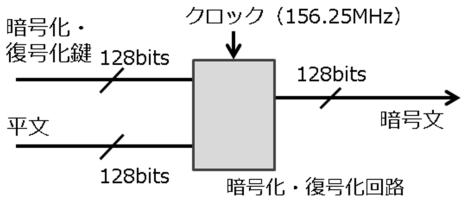

図 6　Camellia の外部インタフェース

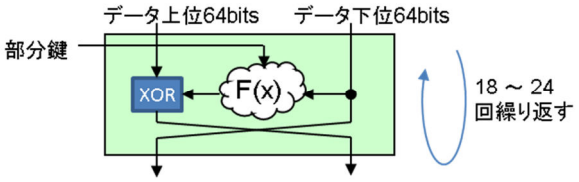

図 7　Feistel 構造の概略

4.2 ハードウェアによる暗号化が必要な領域

暗号化処理は個々のアプリケーションのレイヤでもしばしば実施される処理であり，昨今の汎用 CPU はこれをサポートする拡張命令を持つ場合が多い．伝送路暗号化がハードウェアアクセラレータを必須とする領域を決定するため，AES 拡張命令をサポートした OpenSSL と汎用サーバの組み合わせによる操作速度を測定した結果を表 2 に示す．

表 2　汎用 CPU 上での暗号化処理速度

OpenSSL (1.0.1k 8 Jan 2015) +Intel® Xeon™ E3-1220 V2 (3.1GHz 動作)
データサイズ:1024byte ブロック，暗号利用モード: CBC

暗号化アルゴリズム	スループット
AES 128bit（AES 命令なし）	1144 Mbps
Camellia 128bit	1408 Mbps
AES 128bit（AES 命令あり）	5520 Mbps

この結果から，汎用 CPU 上では AES 拡張命令を使用しても 10Gbps 超のサポートは難しいことがわかる．そこで，本検討でのターゲットを 10Gbps の伝送路暗号化とした．

4.3 暗号化通信プロトコル

実際の長距離伝送には伝送距離を延長するための誤り訂正符号の使用に適した，OTN などの伝送規格を用いることが常であるが，本検討では簡易化のためイーサネットの暗号化を対象とする．イーサネットフレームを暗号化する際のプロトコルとして通常は IEEE802.1AE MACSec [19] や IPSec ESP [20] などが用いられるが，最低限暗号化されたペイロードを受信機にて復元するためには，暗号化後のフレームから元のユーザデータの長さを復元することができればよい．これは，高速用途に用いられるブロック単位の暗号化（例えば，128bit 単位や 256bit 単位ブロック）によってブロックの境界までデータが伸張（パディング）されるため，復号後に伸張された部分のデータの削除を行う必

要があるためである．そこで本試験用アーキテクチャでは，単純にペイロード長（2bytes）をヘッダ直後に埋め込む簡易プロトコルによってこれを実現した．

また，FPGA ベンダが提供している 10Gbps イーサネット MAC コアのインタフェースが 64bit 幅であり，Camellia のデータ幅 128bit と 2 倍の差があることから，暗号化/復号化前にデータ幅を変換する処理を追加した．

暗号の利用モードとしては簡易のため ECB（Electronic Code Book）方式を用いている．この方式は平文を Camellia 暗号化器に直接導入するものであり，同じ鍵と同じ平文を用いた時に同じ暗号が生成されることから，データの傾向秘匿には使用できない．しかし，他の利用モードと同等の回路規模を持つと考えられるため，規模と移植性の評価は可能と思われる．図 8 にはイーサネット MAC から受け取ったデータ（64bit x 最大 190 rows）を上記の手順で暗号化する一連の流れを示す．

4.4 回路及び開発規模

前節のアーキテクチャを従来の RTL ベースの実装と C 言語を用いた高位合成ベースの両方で実装した．実装の際は，両者とも 10Gbps の最大レートの伝送をサポートした上で，できるだけ回路規模が小さくなるように設計を行った．

その後，両実装を FPGA 用の EDA ツールを用いて FPGA コンフィグ化し，FPGA の論理ブロックとメモリ使用量を性能の指標として，コード行数で表される開発規模を簡易的に移植性の指標として記録した．その結果を表 3 に示す．対象とする FPGA はここでは Xilinx Virtex-7 (XC7VX690T-3FFG1761)を用いた．測定対象としたのは暗号化モードに固定した回路 1 つである．

表 3　回路および開発規模比較

	回路規模（Logic Cell 数）	36kbit BlockRAM 消費量（個）	開発規模（コード行数）
RTL ベース実装	10600	19	約 11.1kLines
高位合成ベース実装	7584	47.5	約 1.2kLines

回路規模の面では，高位合成ベースの実装が RTL ベースの実装に対しておよそ 28%の規模削減を達成する結果となった．ただし，BlockRAM の使用タイル数が 2 倍以上に増加しているため，総合的なリソース消費の大小は明らかではない．

開発規模の面では，高位合成ベースの実装が RTL ベースの実装に比して約 9 分の 1 となるコード行数を達成した．

4.5 実機動作確認

また，本回路を汎用の FPGA ボード上に実装し動作を確認した．図 9 のような系を取り，暗号化モードと復号化モードに設定した回路一つずつを用い，イーサネットの規格として 10GBase-SR を片方向のみ用いて，トラヒックテスタから暗号化回路，復号化回路を通して戻る構成とした．

その結果，1518bytes ロングパケットに対してはヘッダを含めた速度 9.960 Gbps，ランダムパケット長に対しては 9.939 Gbps での動作を確認した．パケット長に依存して速度が変わるのは，小さいパケットほど暗号化時のパディングによるデータの伸張割合が相対的に大きく観測されることによる．

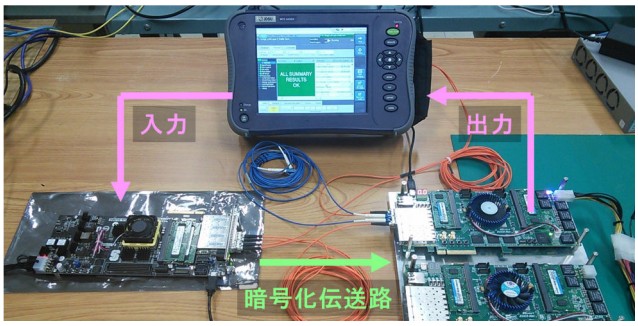

図 9　実機動作確認の様子

5. 得られた知見

5.1 リソース規模の比較

前章の回路規模の比較では，RTL ベースの実装と高位合成ベースの実装の間でロジック使用量と BlockRAM 使用量で傾向に差があり，単純な比較は困難であった．しかし，ロジック使用量，BlockRAM 使用量ともに使用デバイスの利用可能量の 2%～3%付近で極端な差はないことから，両アプローチのリソース消費に大きな差はないといえる．これは文献 [13]に述べられている傾向とある程度一致する．

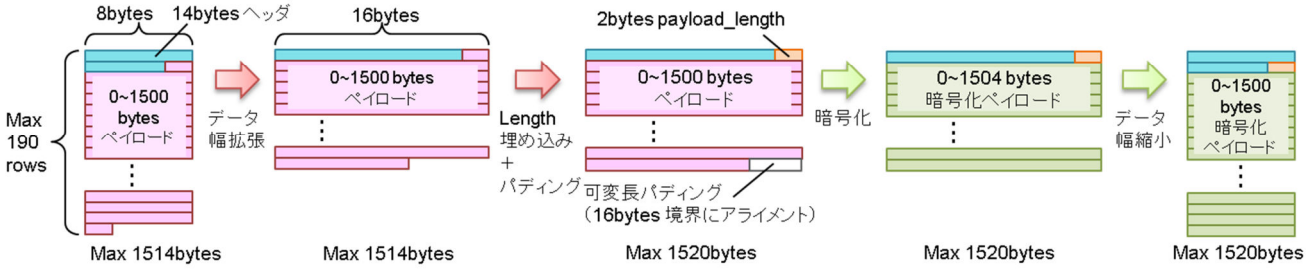

図 8　用いた暗号化プロトコルによる変換フロー

なお，BlockRAM は主に Camellia 内部の Feistel 構造における SBOX（8bit 全単射のルックアップテーブル）に用いられているため，その利用サイズは 2048bits 単位であり，36kbits（ないしは 18kbits）単位の BlockRAM にマッピングした時の利用率が低いことと，リードオンリーでの利用となるため，読み書き可能な SRAM による実現は非効率となる点を付け加える．

設計をモジュールごとに詳細に比較したところ，RTL ベースの設計においてはデバッグの容易性やモジュールの再利用性向上のため，図 8 のような変換フローのステップごとにモジュールを分けて設計しており，共通化されたインタフェースごとに小さな FIFO を装備してバースト的な入力を許容する設計としていたところ，高位合成による設計ではコンパイラが自動的に判断し処理フローの大部分を 1 モジュールにマージしていたことが読み取れた．RTL 版においても同様のモジュールのマージを行い，回路面積を削減する余地は存在するが，保守性とのトレードオフのため容易には実施できない．よって，掛けられる製作コストに制約がある場合は，高位合成による設計が RTL ベースの設計に対して性能面でも有利となる場合があると結論付けられる．

また，移植性の指標として用いたコード行数は，C 言語による記述が RTL の約 1/9 となり，大幅に実装の労力を削減できる結果を得た．さらには，RTL ベースの実装が仕様書を元に書き起こして得たものに対し，高位合成による実装はオープンとなっている C 言語のリファレンス実装を最大限生かした設計となっていることから，実際のコーディング作業，検証作業の労力の面では高位合成による実装がさらに有利となる可能性がある．

6. おわりに

本検討では，通信キャリアネットワークにおけるキャリア固有機能の仮想化方式について検討した．速度と少数生産への対応の面で有利と考えられる汎用 CPU と再構成可能なアクセラレータを用いた構成は，汎用化して短くなったハードウェアのライフサイクルに合わせてアクセラレータのコンフィグレーションをメンテナンスするコストが大きい．本稿ではこの課題に対し，高位合成エンジンを用いてホストソフトウェアとアクセラレータソフトウェアを同じ命令型プログラムによってメンテナンスする高位合成方式に着目し，性能と移植性の両面から従来の設計手法との比較を実施した．

その結果，ある条件においては高位合成による実装が同等の回路規模を実現しながら約 1/9 のコード量での実装が可能となることを確認できた．今回の試みはアクセラレータの初期導入に相当するステップの評価に相当するため，今後の課題として，ハードウェア更改時の移植作業，例えばネットワーク規格のアップグレードに対して十分な移植性を確保できるか検証する必要がある．

参考文献

1) “SDxCentral SDN and NFV Market Size Report 2015 Edition,” (https://www.sdxcentral.com/reports/sdn-nfv-market-size-forecast-report-2015/).
2) N. McKeown, et al., “OpenFlow: enabling innovation in campus networks,” ACM SIGCOMM Computer Communication Review, 38(2), 69-74.
3) “OpenNSL 2.0”, (http://www.broadcom.com/collateral/pb/OpenNSL-PB100-R.pdf), 2015.
4) ETSI GS NFV-INF 001 v. 1.1.1, “Network Functions Virtualization (NFV); Infrastructure Overview,” Section 7.1, Jan. 2015.
5) H. Hawilo, et al.,“NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC)",
6) 西山ら，“仮想化におけるアクセラレータを用いたサービスエッジ機能の実装検討”，信学技報 115(483), 335-340, 2016.
7) G. Vasiliadis, et al., "Gnort: High Performance Network Intrusion Detection using Graphics Processors," 11th International Workshop on Recent Advances in Intrusion Detection (RAID 2008). Springer Berlin Heidelberg, pp. 116-134, 2008.
8) 堀口ら，“100G パケットトランスポートシステム (100G-PTS) の実用化,” NTT 技術ジャーナル, 2010.
9) Z. Bronstein, et al., “Uniform Handling and Abstraction of NFV Hardware Accelerators,” IEEE Network Journal, pp. 22-29, May/June 2015.
10) T. Homemoto et al., “Frame Length Averaging Method for Multi-carrier Aggregation Transport,” 21st Optoelectronics and Communications Conference (OECC) 2016, ThA2-4.
11) P. Bosshart et al., “P4: Programming Protocol-independent Packet Processors,” ACM SIGCOMM Computer Communication Review, 44(3), 87-95.
12) “The World’s Fastest and Most Programmable Networks,” (https://barefootnetworks.com/media/white_papers/Barefoot-Worlds-Fastest-Most-Programmable-Networks.pdf), 2016.
13) E. Homsirikamol and K. Gaj, "Can high-level synthesis compete against a hand-written code in the cryptographic domain? A case study," 2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14), Cancun, 2014, pp. 1-8.
14) “世界初，ハードウェア処理で高速化した量子暗号鍵抽出システムを開発,” (http://www.nec.co.jp/press/ja/1004/1602.html), 2010.
15) 内閣サイバーセキュリティセンター, “政府機関の情報セキュリティ対策のための統一基準”, 2014.05.19.
16) PacketLight Networks,“光ネットワーク向け最大 80Gb レイヤ 1 暗号化 PL-1000TE-Crypto”, (http://www.packetlight.com/?CategoryID=166&ArticleID=482).
17) 片下ら,“暗号ハードウェアの実装性能と物理的安全性評価”, 電子情報通信学会論文誌 A Vol. J95-A No. 5 pp. 392-406, (2012).
18) NTT セキュアプラットフォーム研究所, “Camellia 新着情報,” (https://info.isl.ntt.co.jp/crypt/camellia/intro.html)
19) “IEEE Standard for Local and metropolitan area networks : Media Access Control (MAC) Security,” (https://standards.ieee.org/getieee802/download/802.1AE-2006.pdf), IEEE Std 802.1AE-2006.
20) Proposed Standard : IP Encapsulating Security Payload (https://tools.ietf.org/html/rfc4303), Dec. 2005.