Handel-C による暗号処理ボード SEBSW-2 への暗号回路の実装

古市　洋希[†]　山内　剛[‡]　梶崎　浩嗣[†]　岩井　啓輔[†]　黒川　恭一[†]
[†]防衛大学校情報工学科　[‡]陸上自衛隊

1．はじめに

大量のデータを高速に暗号化するために，多くのセキュリティシステムにて共通鍵暗号方式が用いられている。しかし，近年の計算機性能の向上や暗号解読技術の進歩により，同じ暗号方式を継続して使用することは安全でない。そこで，我々は暗号方式と鍵をダイナミックに変更可能な FPGA を用いた暗号処理ボード（SEBSW-2：SEcret-key Block cipher SWitcher）を開発し[1]，Verilog-HDL を用いて CRYPTREC において次世代共通鍵暗号の候補となっている暗号アルゴリズム（3-DES，MISTY1，Hierocrypt-L1，Camellia，AES）の実装を行い，その性能を評価した。[2]

SEBSW-2 はコアとして大規模 FPGA を持つ。これにより，ハードウェア実装された暗号アルゴリズムをダイナミックに変更することが可能であるとともに、高速動作が可能である。また、新しい暗号アルゴリズムの導入も容易である。しかしながら，その暗号アルゴリズムに対応した回路自体は HDL によって設計されており、多大な労力が必要となっている。

これに対して、近年 C ベース設計技術の向上により，HDL を用いることなく，C 言語からハードウェアへダイレクトなソリューションを提供できるツールが開発されている[3]。この場合，システム開発者は，既存の C 言語のプログラムを使用し，容易にハードウェアを開発することが可能である。したがって，現在の暗号アルゴリズム等に何らかの理由で変更しなければならない必要性が生じた場合，このような言語及びツールを使用することで，より迅速かつ容易に変更可能になる。SEBSW-2 にこれを用いることで，新しい暗号アルゴリズムの導入期間の短縮が期待できる。

本稿では，このような C 言語を拡張した言語である Handel-C 言語を用いて SEBSW-2 において利用可能な暗号回路を実装してゆく開発環境の構築について提案する。米国標準暗号として使われてきた DES 暗号を例に，その環境について行っているチェックの過程も合わせて示す。

2．暗号処理ボード SEBSW-2 の概要

2.1　仕様

SEBSW-2 は，汎用の IBM PC 互換機同士が，頻繁に暗号方式や鍵を交換しながら通信することによって，データ等の保障を得ることを目的とした DCCS（Dynamically Changeable Cipher System）の中核を成す PCI ベースの暗号処理ボードである。

“Implementation of cryptographic circuits for SEBSW-2 with Handel-C”
Hiroki Furuichi[†], Tsuyoshi Yamauchi[‡], Hirotsugu Kajisaki[†], Keisuke Iwai[†], Takakazu Kurokawa[†]
[†] Dept. of Computer Science of National Defense Academy
[‡]Japan Ground Self Defense Force

このシステムは，自衛隊における暗号通信，特に航空機等に搭載することを想定したものであり，そのために求められる仕様は以下のとおりである。

(1) 航空機搭載を想定し，軽量かつ低消費電力であること
(2) 暗号方式の追加時には，ボードレベルでのハードウェアの変更が少ないこと
(3) 公開もしくは非公開の共通鍵ブロック暗号の暗号化及び復号ができること

これらは，再構成可能素子である FPGA を使用することにより解決できる。

2.2　SEBSW-2 の概要

暗号処理ボード SEBSW-2 は，PCI コントローラ，SRAM，FPGA，ネットワークコントローラ，コンフィグレーションコントローラの 5 つのブロックで構成されており，それぞれ 32bit データバスと 32bit アドレスバスとで接続されている。さらに，様々な機器との接続のため，インターフェイスとして PCI や USB インターフェイスを想定している。PCI インターフェイスを持つ SEBSW-2 のブロック図を図 1 に示す。

図 1 中，下段左は，PCI bus とのインターフェイスを行う PCI コントローラを表しており，PCI bus と Local bus の信号の送受信を行う。その右の SRAM は FPGA の構成データを一時的に蓄えるためのものである。右から 2 番目は FPGA を表し，構成データを変更することによって異なる暗号アルゴリズムの処理を行う。下段右端のネットワークコントローラは，Ethernet 等の外部ネットワークへデータを送受信するためのコントローラである。上段のコンフィグレーションコントローラは，一時的に SRAM に蓄えられた FPGA の構成データを FPGA へ転送するための機能を果たし，CPLD を用いて実装している。

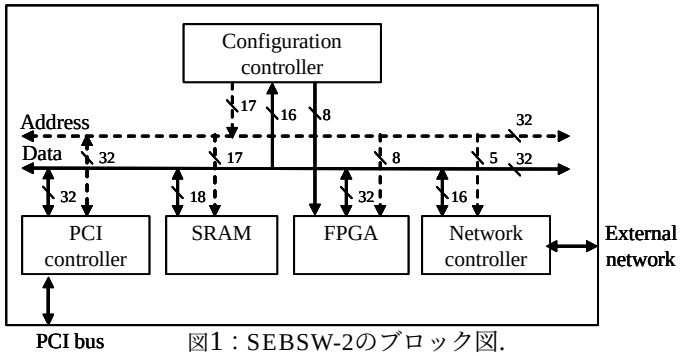


図1：SEBSW-2のブロック図.

3．システム構成

3.1　開発環境

今回構築した開発環境を以下にまとめる。まず，開発言語として Handel-C 言語を用い，実装するターゲットデバイスは SEBSW-2 に搭載されている Xilinx 社 XCV300-PQ240-4 とした。また，論理合成，配置配線及び性能評価は Celoxica 社の DK1 デザインスイートを利用した。

3.2 Handel-C

Handel-C 言語は，Celoxica 社の DK1 デザインスウィートで使用する C 言語を拡張したプログラミング言語である。C 言語をベースとしているため，C 言語によるアルゴリズム記述からの変換が容易となっている。また，DK1 のデザインフロー（図 2）は，Handel-C 言語からダイレクトに FPGA 用のネットリストを生成するため，HDL で回路を記述してからネットリストを生成する場合に比べ，開発にかかる手間が少なくなる。Handel-C 言語は，そのソースコードの各命令を 1 クロックサイクルで実行するシンプルタイミングモデルをベースとした言語であるため，設計の予測がつきやすい。

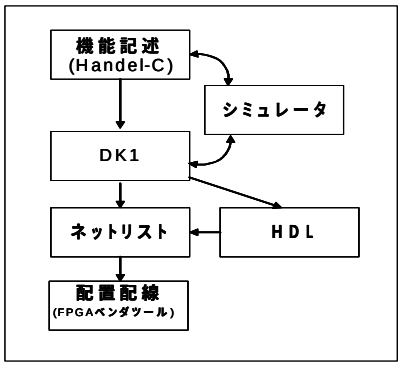


図2：DK1デザインフロー.

```
set clock = external “Clock”;

typedef rom unsigned 4 SBox[64];
macro expr SBox1 = { 14, 4, 13, 1, 2, 15, 11, …}
macro permutation ( in, list, length) = {  … }
…
unsigned 64 data;
unsigned 56 key;
…
void main( )
{
   signal unsigned 32 Lin, Rin;
   signal unsigned 48 ExRin;

   par
   {
      Lin = data[63:32];     Rin = data[31:0];
      ExRin = permutation{ Rin, pbox, 32};
      …
```

図3：Handel-Cの一例.

図 3 に Handel-C のソースの一例として DES を記述したものの一部を示す。Handel-C 特有の記述として，“ set clock ”でクロックを表し，“ macro ”によって SBox や permutation（置換）を記述している。また，“ unsigned ”以降の数値でデータ幅及び鍵長を表す。メインルーチンでは，入力データを 2 分し，“ par ”以下によって 2 分されたデータを並列に処理させている。

4．暗号アルゴリズムの実装

DES はブロック暗号に属し，基本的な関数を繰り返す逐次的なアルゴリズムである。このような暗号回路の実装方法としては，以下の 3 種類のアーキテクチャが考えられる。

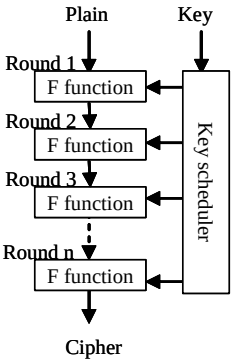
(1) Fully loop unrolled
(2) Pipeline
(3) Loop

(1)は，F 関数全てを実現するアーキテクチャであり，回路規模が大きくなるが，処理速度は比較的高い（図 4）。
(2)は，(1)をパイプライン化したアーキテクチャであり，スループットは最も大きくなる。
(3)は，F 関数 1 ラウンド分のみを実装して，それをループさせることにより実現するアーキテクチャであり，スループットは小さくなるが，回路規模が小さくなる。(図 5)。

SEBSW-2 への実装には，ターゲットデバイスが Xilinx 社の XCV300-PQ240-4 であるため，回路規模としてスライス数 3,072 以下に収める必要がある。従って，ループアーキテクチャを選択して回路規模を抑えるとともに，できる限り高速化を図ることとした。

なお，Virtex におけるメモリの実現方法には，Block Memory 及び分散 Memory という 2 方式が用意されている。後者の場合，クロックに同期しないでデータの読み出しができるため，高速化が可能となる。一方前者の場合，データの読み出しに 1 クロックサイクル必要となるため，サイクル数が増加してしまう問題点がある。ここでは，可能な限り後者を使用し，リソース内に収まらないものについては，前者も使用することを基本方針とした。

現在，DES アルゴリズムの実装については，シミュレーションまで完了している。



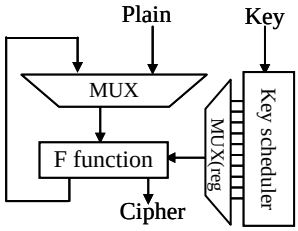


図4：Fully loop unrolled architecture.　　図5：Loop architecture.

5. おわりに

本研究では，Handel-C 言語を用いて，SEBSW-2 において利用可能な暗号回路を実装してゆく開発環境の構築について提案した。今後は，Verilog-HDL を用いて実装した結果と比較し，その性能等について検討する。また，様々な暗号アルゴリズムを実装し回路削減や高速化の手法を導いてゆく予定である。

なお、本研究は東京大学大規模集積システム設計教育研究センターを通し、Celoxica 株式会社の協力で行われたものである。

参考文献

[1] 梶崎浩嗣，黒川恭一：“ 暗号処理ボード SEBSW-2 の設計と性能評価 ”,信学技報 VLD2002-123,CPSY2002-76,2002.
[2]山内剛，梶崎浩嗣，黒川恭一：“ 暗号処理ボード SEBSW-2 への暗号回路の実装 ”,FIT2003，C-034，2003.
[3] http://www.celoxica.co.jp
[4] Bruce Schneier：“ Applied Cryptography ” SoftBank Publishing，2003.