

——Design Wave設計コンテスト2008総評

RSA暗号化器設計のポイント

和田知久, 林 輝彦

本誌2008年5月号において、Design Wave 設計コンテスト2008の入賞者を発表した。ここでは、課題に対する回路設計上のポイントを解説する。Professional部門の入賞者を決定するに当たっては、技術的な評価のために、エンジニアの方々に協力いただいた。その審査委員である林 輝彦氏に、Professional部門の総評をお願いした。もう一方のStudent部門については、琉球大学にご協力いただき、同大学共催の「LSI デザインコンテスト2008」を同部門とさせていただいている。主催者でもある和田知久氏に、Student部門の総評と入賞設計の概要の解説をお願いした。

(編集部)

1. コンテスト課題の概要

和田知久

今回の設計テーマは、電子メールや電子ファイルの暗号化に用いられるRSA暗号の暗号化・復号化回路の設計です(図1)。

RSA暗号は、1977年に発明されました。発明者であるRon Rivest, Adi Shamir, Len Adlemanの頭文字をつなげてこのように呼ばれています。

RSA暗号は、けた数が大きい整数の素因数分解が困難であることを利用して、安全性を保障している暗号です。従って、大きなけた数の整数演算が必要となります。

RSAの暗号化と復号化の計算は対称(同じ演算)です。暗号鍵と復号鍵は異なるものですが、同一の指数演算とmod(剰余)演算の組み合わせの演算となります。式(1)は明文Pを暗号化鍵E、Mを用いて暗号Cを計算する式です。復

号の場合は、まったく同じ式で、Pの部分と暗号Cの部分が入れ替わり、別の鍵を用いる計算となります。

$$C = P^E \bmod M \quad \dots\dots\dots(1)$$

演算として一見単純に見えますが、指数の計算のために、計算機では簡単に扱えない大きな数があり、計算には特別な工夫が必要となります。

2. Professional部門総評

林 輝彦

Design Wave 設計コンテストでは、参加を希望される方には例年、できるかぎり参加の登録をさせていただいています(登録しなくてもコンテストへの参加は可能)。また希望される方には、設計に使用するFPGA評価ボードやFPGA開発ツールの貸し出しを行っています。今回も従来と同様にコンテストへの参加の登録、評価ボードや設計ツールの貸し出しの希望は活発でした。しかし、最終的に実際に設計を完了し、コンテストの応募レポートを作成できた参加者は限られたようでした。

最終的に審査に残った応募は例年に比べてかなり少なく、2件の応募レポートを優勝候補としてコンテスト審査委員会で評価することになりました。いずれの応募も以前に設計コンテストに複数回応募されている方からの応募で、完成度、実際のアプリケーションに使用できるだけの拡張性、柔軟性を備えた優れた設計に仕上がっていて、優劣を付け難く、コンテスト審査委員会は大変難しい選択を迫られる

Keyword

RSA暗号, モンゴメリ乗剰余演算, ビット・シリアル, シストリック・アレイ, CSA

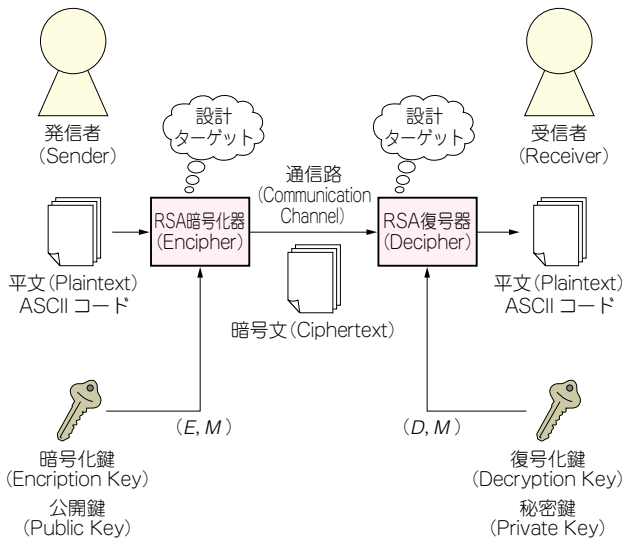


図1 設計課題のシステム概要

ことになりました。

● 審査結果

「Design Wave 設計コンテスト 2008」の入賞者は以下の方に決定しました。

- 優勝 上原昭宏 (冗長表現ビット・シリアル演算回路によるRSA暗号回路の設計)
- 準優勝 石井康雄 (Scalable RSA Architecture)
(敬称略)

賞品として、優勝の上原氏には、沖縄における発表会講演を兼ねた2泊3日の沖縄旅行のほか、副賞のBlu-rayビデオ・レコーダ(パナソニック DMR-BW800-K)が、準優勝の石井氏には32V型液晶テレビ(シャープ LC-32D10B)が贈られました。

● RSA暗号の暗号化と復号化の手順

今年のDesign Wave設計コンテストの設計課題であるRSA暗号化器に実装する計算のアルゴリズムは、式で表現すると大変簡素です。平文 P 、暗号文 C 、暗号化鍵 E 、復号鍵 D 、法 M のそれぞれが正の整数で表されているとき、以下の式で示される計算によって暗号化を行います。

$$C = P^E \bmod M \quad \dots\dots\dots (2)$$

また、RSA暗号を復号化する計算は以下になります。

```
Input : N, 0 < A < N, 0 < B < N
PREP : V = - N-1 mod R
Output : S = A · B · R-1 mod N

Q := (A · B) V mod R;
S := A · B + Q · N;
S := S / R;
If (N < S) then S := S - N;
Out(S)
```

図2 モンゴメリ法による乗剰余演算 ($N < R = 2^r$)

$$P = C^D \bmod M \quad \dots\dots\dots (3)$$

これらの式から分かるように、暗号化器を設計することで同じ設計を復号化器に使うことができるわけです。

● モンゴメリ乗剰余演算法の採用

式(2)や式(3)は、数式としては単純です。しかし、累乗の計算を含んでいるため、実際にハードウェアに実装しようとすると、そのまま計算したのでは計算量が大変多くなり、計算の途中のデータのけた数が膨大になってしまいます。またmodの計算で必要になる除算は、計算の途中で何らかの判断を行う必要が出てくるので、乗算のようにストレートに効率良く処理することが難しいものです。

そこで、上原氏、石井氏ともにモンゴメリ乗剰余演算法(Montgomery Multiplication; 以下モンゴメリ法)に着目し、設計に適用していました。

モンゴメリ法は二つの値を乗算し剰余を取る、乗剰余演算を行うアルゴリズムで、例えば図2のように表すことができます。

少し調べてみると、乗剰余演算のアルゴリズムの研究は現在も盛んです。いかに小さな回路で高速に処理するかという観点から、今日でもいろいろな提案、報告が継続して行われています。また、実際にIP(Intellectual Property)コア製品として市場に出回っているRSA暗号の暗号器、復号器でも、このモンゴメリ法を中心的なアルゴリズムとして利用しているものも多いようです。

● アーキテクチャの比較

最終的に審査に残った上原氏と石井氏の設計は、乗剰余演算にモンゴメリ法を採用した点では似通っていました。両氏とも課題で与えられたビット長32にとどまらず、1024～4096ビットの現実的なビット長にまで拡張可能な実用的な暗号器に仕上がっています。そこで設計のアルゴリズムや