

—Design Wave設計コンテスト2008 Professional部門 優勝

# 回路規模最小を目指した FPGA向けRSA暗号化器の設計

上原昭宏

Design Wave 設計コンテスト2008のProfessional部門で優勝した設計を紹介する。改良モンゴメリ乗算アルゴリズムを用いたべき乗剰余演算アルゴリズムによるビット・シリアル演算回路で構成したRSA暗号化器を元としている。けた上げが上位けたに伝播しないビット・シリアル加算および乗算/2乗回路をFPGAに最適化した。ビット長をパラメータ宣言しているので、任意のビット長に対応できる。 (編集部)

Design Wave 設計コンテスト2008課題の、ビット長32の平文をRSA暗号化する回路に、「原理的に最小規模でかつ高クロック動作の設計ができるか」を目標に取り組みました。このために、RSA暗号化の核であるべき乗剰余演算に正面から取り組み、FPGAに実装した際の回路規模と速度の設計限界を原理的に見極めようとしてしました。

設計検討のために文献を調査しても内容が高度で理解できず冷や汗をかいているところに、改良モンゴメリ乗算アルゴリズムを用いたべき乗剰余演算アルゴリズムによるビット・シリアル演算回路で構成したRSA暗号化器<sup>(1)</sup>に出会いました。

回路規模が平文のビット長に比例する、そして処理ビット長が長くなっても最大動作クロックが低下しないビット・シリアル演算回路は、回路規模と速度の見極めに適していると思い、これを採用しました。さらに、けた上げが上位けたに伝播しないビット・シリアル加算および乗算/2乗回路をFPGAに最適化して設計しました。これらを、ビット長をパラメータ宣言して設計することで、任意のビット長に対応した実用的なRSA暗号化器を設計しまし

た。さらにバッファを共有およびバッファが不要な信号フローをとって回路規模を最小にしました。

米国Xilinx社のSpartan-3(スピード・グレード4)をターゲットに合成したRSA暗号化器の回路規模と最大動作周波数は、それぞれ583スライスおよび78MHz(50入力XORに対して回路規模58倍、遅延時間1.1)でした。また、べき乗演算回路およびRSA暗号化回路の規模は、平文1ビット当たりそれぞれ10スライス、17スライスで設計できることが分かりました。

## 1. 設計目標を決める

### ● RSA暗号処理の中身

成果を上げるには目標が何かを明確にしなければなりません。設計目標を決めるには、処理内容を地に足がついた感覚でしっかりと把握することが必須です。そこで、まずRSA暗号化と復号化を見直しました。

平文、暗号文、暗号化鍵、復号鍵、法をそれぞれ $C_p$ 、 $C_e$ 、 $E$ 、 $D$ 、 $M$ と表します。ここで法 $M$ は十分大きな二つの素数 $p$ および $q$ の積、平文 $C_p$ は法 $M$ より小さい正整数、そして正整数 $L$ を $(p-1)$ と $(q-1)$ の最小公倍数数とすると式(1)が成り立ちます。

$$C_p^L \bmod M = 1 \quad \dots\dots\dots(1)$$

式(1)のmodは剰余を求める演算子です。 $A \bmod B$ は $A$ を $B$ で割った余りを表します。つまり式(1)は法 $M$ の剰余系のべき乗には周期 $L$ があることを示しています。

### Keyword

RSA暗号化器、改良モンゴメリ乗算アルゴリズム、ビット・シリアル演算回路、Spartan-3、べき乗剰余演算

ここで、鍵 $E$ および $D$ をある整数 $H$ に対して、

$$E \cdot D = H \cdot L + 1 \quad \dots\dots\dots(2)$$

を満たすようにとり、暗号化と復号化をそれぞれ、

$$C_e = C_p^E \bmod M \quad \dots\dots\dots(3)$$

$$C_p = C_e^D \bmod M \quad \dots\dots\dots(4)$$

のべき乗剰余演算で与えます。こうすると、

$$\begin{aligned} C_e^D \bmod M &= (C_p^E \bmod M)^D \bmod M = C_p^{E \cdot D} \bmod M \\ &= C_p^{(H \cdot L + 1)} \bmod M = C_p \quad \dots\dots\dots(5) \end{aligned}$$

のように、鍵 $E$ で暗号化した文を鍵 $D$ で復号化できます。

このように式自体は短くて理解しやすいのですが、平文、鍵、法のそれぞれの項が、通常はビット長1024以上とけた数が多いことが計算を難しくします。ビット長32の場合ですら関数電卓を使っても計算は大変です。

## ● 設計方針を決める

本設計の目的は優れた設計例を示すことであり、その評価ポイントは回路規模と速度、ユニークさの3点です。この3点のどこに重きを置くかで結果の8割が決まると考えました。

RSA暗号化回路は、数多くの設計例が既に報告されています。ユニークさを出すにも一朝一夕に新アイデアは出てきません。そこで今回のコンテスト特有の条件「平文のビット長が最大32」を活用できないかと考えました。

ところがビット長が短くても、剰余系演算という本質の困難さには違いがないために、手が浮かびませんでした。そこで規模と速度の2点に、正面から取り組むことにしました。また、せっかく1から設計するのですから、FPGAにおけるRSA暗号化回路の回路規模と最大動作クロックの設計限界はどこにあるのかを原理的に明らかにしたいと考えました。そしてコンテスト課題限定の設計では面白くないので、ビット長がより長い平文も処理できる実用的な設計を目指しました。

## ● 文献調査で巨人の肩に乗る

RSA暗号化は広く一般に使われており回路設計例の報告も数多くあります。そこで過去の成果を吸収すれば手早く巨人の肩に乗れるだろうと文献調査をしました。また文献

調査は過去の設計例との差異を押さえて自分のオリジナル点を主張するためにも重要です。

「べき乗剰余演算」や「RSA」などのキーワードを使い、Google Scholar、Google Patent、CiteSeer、IEEE Xploreなどで検索をかけて、引用数が多い論文を特に重点的に見ていきました。ところが内容がさっぱり理解できません。英文は読めて意味も分かる(分かっているはず)のですが、RSA暗号化回路の一部分、べき乗もしくは乗算に論点が集中するので、RSAの全体回路を知らない私には全体像がピンときません。そのうえ、何年にも渡る研究を基にしているために、議論も高度で理解が追いつかずに途方にくれてしまいました。

何とかモンゴメリ・アルゴリズムが乗法演算に重要だと分かり、原典<sup>(2)</sup>を参照したり初期の研究論文<sup>(3)</sup>を読んでみました。しかし、回路全体の姿が見通せずに苦しみました。そんなときに内容が理解できて自分の設計方針にぴったりくるChing-Chaoらの論文<sup>(1)</sup>に出会い、これをFPGAに最適化することにしました。

## 2. アルゴリズムをFPGAにはめこんでいく

### ● RSA暗号化回路の構成

Ching-Chaoらのべき乗アルゴリズム<sup>(1)</sup>をC言語風に表記したものを図1に示します。これだけでは分かりにくいので具体例で説明していきます。

べき乗は2乗と前処理した平文 $C_p'$ との積を繰り返すものです。前処理した平文 $C_p'$ については後で説明します。例として指数が6のときの処理ステップを図2に示します。2乗すれば指数を2倍(左に1ビット・シフト)、 $C_p'$ を掛ければ指数に+1することになります。従って、2進数表記した指数を最上位ビットから1ビットずつ見ていき、ビット・シフトと+1に対応した2乗または $C_p'$ との積を繰り返せばべき乗が求まります。

このべき乗剰余演算のほとんどの処理時間を占めるのが乗算です。例えば $n$ ビットの $A$ と $B$ の積を計算するには、まず $(A \times B)$ を求めてから法 $M$ の剰余を求めなければなりません。この剰余を求めるためのビット長 $2n$ の除算回路は設計が厄介です。ところが、この乗算をビット・シフトと加算だけで行うのがモンゴメリ・アルゴリズム<sup>(2)</sup>です。

これは $(A \times B \times 2^{-n})$ を求めるものです。まず $(A \times B)$