Multiplication-addition modulaire: algorithmes itératifs et implantations sur FPGA

Jean-Luc Beuchat

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Multiplication-addition modulaire: algorithmes itératifs et implantations sur FPGA

Jean-Luc Beuchat et Jean-Michel Muller

No 4840

Juin 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4840--FR+ENG

INRIA
RHÔNE-ALPES

Multiplication-addition modulaire: algorithmes itératifs et implantations sur FPGA

Jean-Luc Beuchat et Jean-Michel Muller

Thème 2 — Génie logiciel
et calcul symbolique
Projet Arénaire

Rapport de recherche n ° 4840 — Juin 2003 — 10 pages

Abstract: This paper describes several improvements of an iterative algorithm for modular multiplication originally proposed by Jeong and Burleson. A first modification of the recurrence relation allows us to implement a fused multiply and add unit. Then, we show how to reduce the circuit area by a factor two when the operator offers the possibility to choose the modulo among a set $m_1, m_2, \ldots, m_q$. A new iterative algorithm making the implementation of modular exponentiation easier is eventually discussed. For 16-bit numbers, our operators perform for instance 6 millions of operations per second on a Virtex-E device while only requiring 17 slices.

Key-words: Computer arithmetic, modular multiplication, FPGA

(Résumé : tsvp)

This text is also available as a research report of the Laboratoire de l'Informatique du Parallélisme `http://www.ens-lyon.fr/LIP`.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International : +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International : +33 4 76 61 52 52

Multiplication-addition modulaire: algorithmes itératifs et implantations sur FPGA

Résumé : Cet article présente tout d'abord diverses améliorations d'un algorithme itératif de multiplication modulaire proposé en 1997 par Jeong et Burleson. Une simple modification de la récurrence permet l'implantation d'une multiplication-addition modulaire. Nous montrons ensuite comment réduire d'un facteur deux la taille du circuit lorsque l'opérateur offre le choix du modulo parmi un ensemble $m_1, m_2, \ldots, m_q$. Nous proposons finalement un nouvel algorithme facilitant la réalisation de l'exponentiation modulaire et présentons quelques résultats de synthèse pour des circuits FPGA de la famille Virtex de Xilinx. Pour des nombres de 16 bits, les opérateurs développés permettent par exemple d'effectuer 6 millions d'opérations à la seconde en utilisant uniquement 17 tranches.

Mots-clé : Arithmétique des ordinateurs, multiplication modulaire, FPGA

1 Introduction

La multiplication modulaire intervient dans les applications de traitement du signal tirant parti de la représentation modulaire des nombres (*Residue Number System*) et dans plusieurs algorithmes de cryptographie [5]. Cette opération arithmétique a par conséquent fait l'objet de nombreuses études dont voici quelques exemples :

- Montgomery a proposé un algorithme dont le principe consiste à remplacer les opérations modulo m par des calculs modulo une puissance de la base dans laquelle sont représentés les opérandes [6]. La référence [7] fournit une bonne bibliographie relative aux implantations matérielles de cet algorithme.
- Plusieurs chercheurs ont étudié des moduli particuliers, comme $2^n - 1$ et $2^n + 1$, facilitant la réalisation de la multiplication modulaire (voir par exemple [4, 9]).
- Lorsque le modulo m est premier, le groupe multiplicatif $(\mathbb{Z}/m\mathbb{Z})^*$ est isomorphe au groupe additif $\mathbb{Z}/(m-1)\mathbb{Z}$ [1, 8]. Il est ainsi possible de remplacer la multiplication modulo m par l'addition modulo $m-1$. L'inconvénient de cette méthode réside dans les blocs de mémoire indispensables au calcul de l'isomorphisme.

Dans cet article, nous étudions des algorithmes basés sur le schéma de Horner et caclulant itérativement le produit modulaire de deux opérandes. Nous proposons tout d'abord diverses améliorations d'une itération proposée par Jeong et Burleson [2] permettant d'effectuer une multiplication-addition modulaire tout en réduisant la surface de l'opérateur initial (section 2.1). Nous décrivons ensuite une nouvelle itération facilitant l'implantation d'opérations comme l'exponentiation modulaire (section 2.2) et présentons quelques résultats obtenus pour des circuits FPGA de la famille Virtex de Xilinx (section 3).

2 Multiplication-addition modulaire

L'appproche adoptée par Jeong et Burleson consiste à écrire la multiplication modulo m de deux opérandes entiers x et y sous la forme d'un schéma de Horner [2] :

$$xy \bmod m = ((\ldots(((x_{n-1}y)\cdot 2 + x_{n-2}y)\cdot 2 + x_{n-3}y)\cdot 2 + \ldots)\cdot 2 + x_0 y) \bmod m$$

où $0 \leq x, y < m$ et m est un nombre de n bits tel que $2^{n-1} + 1 \leq m \leq 2^n - 1$. La multiplication modulo m peut ainsi s'effectuer de manière récursive en définissant $P[n] = 0$ et en calculant :

$$P[n-i] = (2P[n-i+1] + x_{n-i}y) \bmod m \tag{1}$$

pour i allant de 1 à n. Il est facile de vérifier que $P[0]$ est bien égal à $xy \bmod m$. Cette méthode implique toutefois une addition modulo m lors de chaque itération. Si l'opérande y est strictement inférieur à m, nous déduisons de la relation de récurrence (1) que :

$$2P[n-i+1] + x_{n-i}y \leq 2\cdot(m-1) + m - 1 = 3m - 3$$

L'addition modulo m requiert par conséquent des comparaisons afin de déterminer s'il faut retrancher 0, m ou $2m$ à $2P[n-i+1] + x_{n-i}y$. Afin de réduire la surface et le temps de cycle du circuit, Jeong et Burleson proposent le calcul d'un nombre de $n+1$ bits congru à $2P[n-i+1] + x_{n-i}y$ modulo m. Considérons un entier z de $n+\alpha$ bit et définissons sa partie haute et sa partie basse telles que $z_H = z \operatorname{div} 2^n$ et $z_L = z \bmod 2^n$. Par conséquent, $z = 2^n z_H + z_L$ et le nombre de $n+1$ bits $z' = 2^n z_H \bmod m + z_L$ est congru à z modulo m. Lorsque le nombre de bits α de z_H est petit, l'expression $2^n z_H \bmod m$ se calcule efficacement à l'aide d'une table. En se basant sur ces constatations, Jeong et Burleson ont proposé l'itération suivante dans [2] :

$$\begin{cases} T[n-i] &= 2P[n-i+1] \bmod 2^n + \varphi(P[n-i+1] \operatorname{div} 2^n) \\ T^*[n-i] &= T[n-i] + x_{n-i}y \\ P[n-i] &= T^*[n-i] \bmod 2^n + \varphi(T^*[n-i] \operatorname{div} 2^n) \end{cases} \tag{2}$$

avec $P[n] = 0$ et $\varphi(k) = (k \cdot 2^n) \bmod m$. La figure 1a décrit l'architecture d'un étage d'itération d'un multiplieur modulaire conçu d'après la relation de récurrence (2). Jeong et Burleson ont développé leur algorithme pour des

implantations VLSI et ont par conséquent opté pour une représentation des nombres en *carry-save* afin de restreindre le temps de cycle de l'opérateur. La majorité des FPGA actuellement disponibles disposant de logique rapide destinée à la gestion des retenues, nous utilisons des additionneurs à retenue propagée pour la réalisation des circuits (opérateur + du langage VHDL).

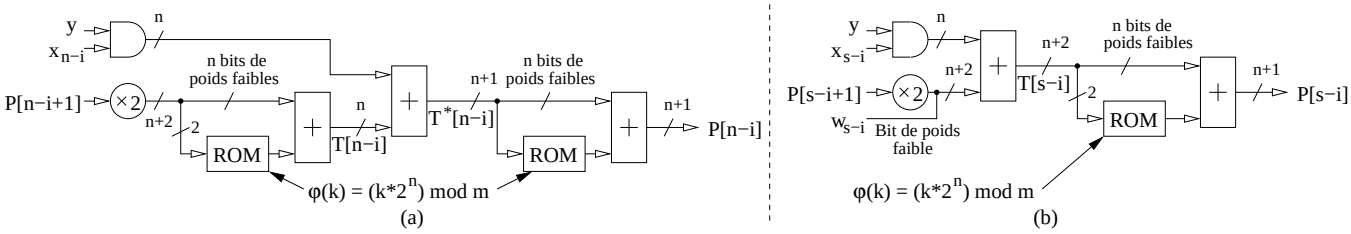

FIG. 1 – Un étage d'itération d'un multiplieur modulaire pour FPGA. (a) Algorithme de Jeong et Burleson. (b) Algorithme de Kim et Sobelman modifié.

Kim et Sobelman ont proposé une première amélioration de cet algorithme [3] en définissant l'itération suivante :

$$\begin{cases} T[n-i] = 2P[n-i+1] + x_{n-i}y \\ P[n-i] = T[n-i] \bmod 2^n + \varphi(T[n-i] \operatorname{div} 2^n) \end{cases}$$

où $P[n] = 0$. Cette approche permet la suppression d'un additionneur par rapport à la solution de Jeong et Burleson. Il n'est toutefois pas nécessaire que l'opérande x soit un nombre de n bits strictement inférieur à m. Dans la suite, nous considérerons que x est un entier de s bits. Notons de plus qu'il est trivial de modifier cette nouvelle relation de récurrence afin de calculer $(xy + w) \bmod m$:

$$\begin{cases} T[s-i] = 2P[s-i+1] + x_{s-i}y + w_{s-i} \\ P[s-i] = T[s-i] \bmod 2^n + \varphi(T[s-i] \operatorname{div} 2^n) \end{cases} \tag{3}$$

où $w \in \mathbb{N}$ (figure 1b). La principale difficulté consiste à déterminer le nombre de bits de $T[s-i]$ afin de dimensionner correctement la table implantant la fonction $\varphi(k)$. Puisque $0 \leq T[s-i] \bmod 2^n \leq 2^n - 1$ et $0 \leq \varphi(T[s-i] \operatorname{div} 2^n) \leq m - 1$, nous déduisons que $P[s-i]$ est un nombre de $n+1$ bits et que $0 \leq P[s-i] \leq 2^n + m - 2$. La relation de récurrence (3) nous permet alors de calculer une borne supérieure pour $T[s-i]$:

$$0 \leq T[s-i] \leq 2^{n+1} + 3m - 4$$

En substituant la valeur maximale de m dans l'équation ci-dessus, nous obtenons finalement :

$$0 \leq T[s-i] \leq 2^{n+2} + 2^n - 7$$

Kim et Sobelman utilisent une telle borne indiquant que $T[s-i]$ est un nombre de $n+3$ bits. La figure 2a illustre l'implantation de l'algorithme ainsi obtenu sur un circuit de la famille Virtex de Xilinx. Un premier additionneur à retenue propagée établit la somme de $(w_{s-i} + 2P[s-i+1])$ et de $x_{s-i}y$. Chacun des bits de $\varphi(T[s-i] \operatorname{div} 2^n)$ se détermine à l'aide d'un multiplexeur commandé par les trois bits de poids forts de $T[s-i]$. Les *look-up tables* (LUT) des FPGA considérés disposant de quatre entrées, il est possible de combiner le calcul de $\varphi(k)$ et la seconde addition à l'aide d'un unique étage de logique.

2.1 Amélioration de l'itération de Kim et Sobelman

Supposons maintenant que nous souhaitions concevoir un opérateur offrant la possibilité de sélectionner le modulo m_i désiré parmi un ensemble de q nombres $m_1, m_2, \ldots, m_q$. Seul le calcul de la fonction φ dépendant de m_i, il suffit

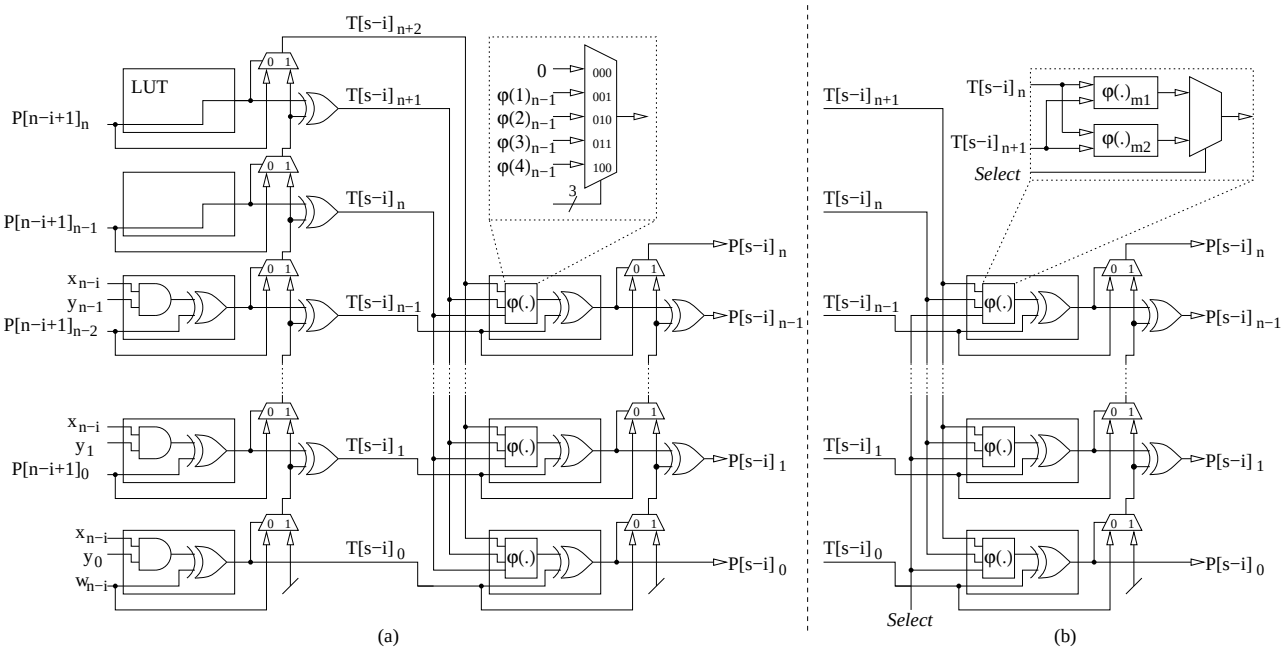

FIG. 2 – Implantation d'un étage d'itération sur un FPGA de la famille Virtex. (a) Algorithme de Kim et Sobelman. (b) Impact du théorème 1 sur l'architecture de l'opérateur.

de construire une table contenant les valeurs de $\varphi(k)$ pour chacun des moduli considérés. Il est dès lors important de disposer d'une borne précise sur $T[s-i]$ afin de réduire la taille de la table.

Théorème 1
Supposons que $x \in \mathbb{N}$, $y \in \{0, \ldots, m-1\}$ *et* $w \in \mathbb{N}$. *Les termes* $P[s-i]$ *et* $T[s-i]$ *satisfont alors les inégalités suivantes :*

$$0 \leq P[s-i] \leq \begin{cases} 2^{n+1} + 2^n - 2m - 1 & si\ \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ 2^{n+1} - m - 1 & si\ 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases} \tag{4}$$

et

$$0 \leq T[s-i] \leq \begin{cases} 2^{n+2} + 2^{n+1} - 3m - 2 & si\ \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ 2^{n+2} - m - 2 & si\ 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases} \tag{5}$$

De plus, les bornes supérieures P_{max} *et* T_{max}, *atteintes lorsque* $m = \lfloor 2^{n+1}/3 \rfloor + 1$, *sont définies par :*

$$P_{max} = \begin{cases} (2^{n+2} + 2^n - 5)/3 & si\ n\ est\ pair \\ (2^{n+2} + 2^n - 7)/3 & si\ n\ est\ impair \end{cases} \qquad et \qquad T_{max} = \begin{cases} 2^{n+2} - 3 & si\ n\ est\ pair \\ 2^{n+2} - 4 & si\ n\ est\ impair \end{cases}$$

Ce théorème, démontré dans l'annexe A, garantit ainsi que $T[s-i]$ est un nombre de $n+2$ bits. Considérons à nouveau le circuit de la figure 2a. Puisque le calcul de $\varphi(k)$ ne nécessite plus que deux bits, chacune des LUT du second additionneur à retenue propagée dispose maintenant d'une entrée libre. Nous pouvons y connecter un signal *Select* (figure 2b) et calculer

$$\varphi(T[s-i] \operatorname{div} 2^n) = \begin{cases} (2^n \cdot T[s-i] \operatorname{div} 2^n) \bmod m_1 & \text{si } Select = 0 \\ (2^n \cdot T[s-i] \operatorname{div} 2^n) \bmod m_2 & \text{si } Select = 1 \end{cases}$$

Notre opérateur est dès lors capable de déterminer $(xy+w) \bmod m_1$ et $(xy+w) \bmod m_2$ à l'aide des mêmes ressources matérielles que le circuit de la figure 2a. Lorsque le nombre de moduli devient plus important, nous pouvons stocker les diverses valeurs de $\varphi(k)$ dans les blocs de mémoire du FPGA. Le théorème 1 permet ainsi l'économie d'un bit d'adresse par rapport à la solution de Kim et Sobelman, réduisant de moitié la taille de la mémoire nécessaire.

L'opérateur fournissant un nombre de $n+1$ bits congru à $xy+w$ modulo m, une correction est nécessaire afin d'obtenir le résultat escompté. Le théorème suivant, démontré dans l'annexe B, fournit les informations nécessaires à la conception du circuit responsable de cette opération.

Théorème 2
Supposons que $x \in \mathbb{N}$, $y \in \{0, \ldots, m-1\}$ et $w \in \mathbb{N}$. L'itération définie par l'équation (3) génère un nombre $P[0]$ de $n+1$ bits tel que $P[0] = (xy+w) \bmod m + km$, où k est un entier satisfaisant la relation suivante :

$$0 \leq k \leq \begin{cases} 2 & si\ 2^{n-1}+1 \leq m \leq 2^{n-1}+2^{n-2}-1 \\ 1 & si\ 2^{n-1}+2^{n-2} \leq m \leq 2^n-1 \end{cases}$$

Lorsque k appartient à $\{0,1,2\}$, deux comparateurs nous permettent de déterminer la correction requise. Un soustracteur la retranche ensuite de $P[0]$ et nous obtenons ainsi le résultat escompté (figure 3a). Dans le second cas, nous pouvons nous inspirer de l'architecture traditionnelle de l'additionneur modulo m [1]. Nous déduisons du théorème 2 que :

$$(xy+w) \bmod m = \begin{cases} P[0] & \text{si } 0 \leq P[0] < m, \\ P[0]-m & \text{si } m \leq P[0] < 2m \end{cases}$$

En constatant que $P[0]-m$ est strictement inférieur à 2^n, nous obtenons :

$$P[0]-m = (P[0]-m) \bmod 2^n = (P[0] \bmod 2^n + 2^n - m) \bmod 2^n$$

La condition $m \leq P[0] < 2m$ peut également s'écrire :

$$\begin{aligned}(m \leq P[0] < 2^n) \vee (2^n \leq P[0] < 2m) &\Leftrightarrow (m \leq P[0] \bmod 2^n < 2^n) \vee (P[0] \operatorname{div} 2^n = 1) \\ &\Leftrightarrow (2^n \leq P[0] \bmod 2^n + 2^n - m) \vee (P[0] \operatorname{div} 2^n = 1)\end{aligned}$$

Par conséquent,

$$(xy+w) \bmod m = \begin{cases} (P[0] \bmod 2^n + 2^n - m) \bmod 2^n & \text{si } (P[0] \operatorname{div} 2^n = 1) \text{ ou } (P[0] \bmod 2^n + 2^n - m \geq 2^n) \\ P[0] \bmod 2^n & \text{sinon} \end{cases}$$

La figure 3b décrit le circuit correspondant. Un additionneur à retenue propagée détermine tout d'abord la somme $P[0] \bmod 2^n + 2^n - m$. Un multiplexeur commandé par le bit de poids fort de $P[0]$ et la retenue sortante de l'additionneur sélectionne ensuite le résultat correct.

2.2 Nouvelle itération

L'opérateur de multiplication-addition étudié ci-dessus nécessite que l'opérande y soit inférieur à m. Cette contrainte s'avère par exemple problématique dans le cas de l'exponentiation modulaire nécessitant des élévations au carré successives [5]. Il est en effet impossible de calculer $P[0]^2 \bmod m$ à l'aide du circuit de la figure 1b. Bien qu'il soit envisageable d'effectuer des corrections après chaque multiplication, nous préférons étudier une nouvelle itération. Une première solution consiste à calculer le terme $T[s-i]$ de la relation de récurrence (3) sur $n+3$ bits et d'implanter $\varphi(k)$ à l'aide d'une table à trois entrées. L'itération définie par :

$$\begin{cases} T[s-i] = 2P[s-i+1] + x_{s-i}y + w_{s-i} \\ P[s-i] = T[s-i] \bmod 2^{n-1} + \psi(T[s-i] \operatorname{div} 2^{n-1}) \end{cases} \tag{6}$$

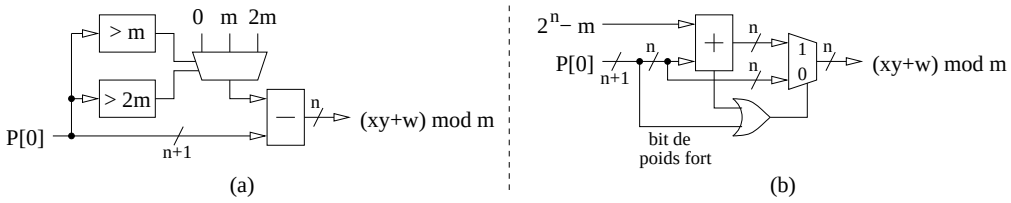

FIG. 3 – Correction de $P[0]$ afin d'obtenir $(xy+w) \bmod m$. (a) $P[0]$ est égal à $(xy+w) \bmod m$, à $(xy+w) \bmod m + m$ ou à $(xy+w) \bmod m + 2m$. (b) $P[0]$ est égal à $(xy+w) \bmod m$ ou à $(xy+w) \bmod m + m$.

avec $P[s] = 0$ et $\psi(k) = (2^{n-1} \cdot k) \bmod m$ permet cependant le calcul de $T[s-i]$ sur $n+2$ bits et facilite la correction de $P[0]$ nécessaire à l'obtention de $(xy+w) \bmod m$.

Théorème 3
Supposons que x et w appartiennent à $\mathbb{N}$ et que y est un entier de $n+1$ bits tel que

$$0 \leq y < \begin{cases} (2^{n+2}+11)/3 & \textit{si } n \textit{ est pair} \\ (2^{n+2}+7)/3 & \textit{si } n \textit{ est impair} \end{cases}$$

Les bornes supérieures de $P[s-i]$ et $T[s-i]$ sont alors définies par :

$$P_{max} = \begin{cases} (2^{n+2}-7)/3 & \textit{si } n \textit{ est pair} \\ (2^{n+2}-5)/3 & \textit{si } n \textit{ est impair} \end{cases} \qquad \textit{et} \qquad T_{max} = 2^{n+2}-1$$

Les valeurs P_{max} et T_{max} sont atteintes lorsque $m = \lfloor (2^{n+1}+2^{n-1})/3 \rfloor + 1$. De plus, le nombre $P[0]$ retourné par l'opérateur est égal à $(xy+w)\ mod m$ ou à $(xy+w)\ mod m + m$.

La démonstration, analogue à celle du théorème 1, n'est pas donnée dans cet article. Ce théorème garantit que le terme $T[s-i]$ de la nouvelle itération est un nombre de $n+2$ bits. Les conditions que l'opérande y doit satisfaire étant moins contraignantes que celles imposées par le théorème 1, il est maintenant possible d'utiliser la sortie $P[0]$ comme opérande y d'une nouvelle multiplication-addition sans effectuer de correction ni modifier l'architecture du circuit. Il suffit en effet de constater que :

$$P_{\max} = \begin{cases} (2^{n+2}-7)/3 < (2^{n+2}+11)/3 & \text{si } n \text{ est pair} \\ (2^{n+2}-5)/3 < (2^{n+2}+7)/3 & \text{si } n \text{ est impair} \end{cases}$$

Comme $P[0]$ est strictement inférieur à $2m$, la correction nécessaire à l'obtention de $(xy+w) \bmod m$ s'effectue à l'aide du circuit de la figure 3b, quel que soit $m \in \{2^{n-1}+1, \ldots, 2^n-1\}$.

3 Opérateurs de multiplication-addition sur FPGA

Afin de comparer les algorithmes décrits dans cet article, nous avons développé un programme C générant une description VHDL synthétisable d'un étage d'itération en fonction d'un ensemble de moduli $\{m_1, \ldots, m_q\}$ spécifié. Notre première expérience consiste à étudier l'impact du théorème 1 sur la surface du circuit initialement proposé par Kim et Sobelman (figure 4). Les résultats obtenus indiquent que le calcul de la borne supérieure exacte de $T[s-i]$ permet de réduire approximativement d'un facteur deux la taille du circuit dès que le nombre de moduli est supérieur à quatre.

Considérons maintenant un étage de calcul de la nouvelle itération définie par l'équation (6). En plaçant un registre en sortie de ce composant et en utilisant un mécanisme de rebouclage, nous pouvons calculer un nombre

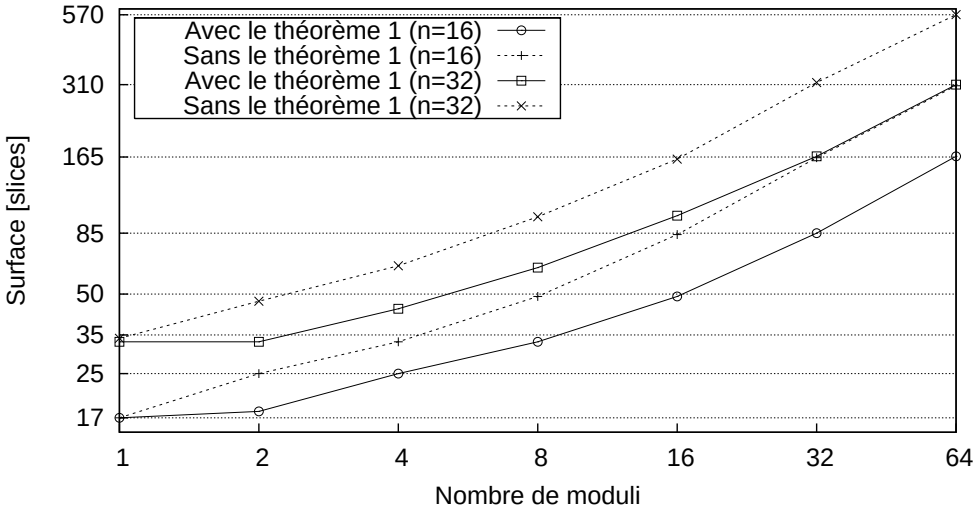

FIG. 4 – Surface d'un étage de l'itération de Kim et Sobelman en fonction du nombre de moduli sur un circuit Virtex XCV100E-6 (moduli générés aléatoirement ; synthèse, placement et routage effectués avec ISE 5.1.03i).

congru à $xy + w$ modulo m en s cycles d'horloge. Le nombre de multiplications-additions déterminées par seconde est alors de f/s, où f est la fréquence de fonctionnement du circuit. Sur un circuit Virtex-E, nous obtenons par exemple $f \approx 100$ MHz pour des opérandes de $s = 16$ bits. Par conséquent, nous effectuons environ six millions d'opérations à la seconde en utilisant uniquement 17 tranches.

4 Conclusion

Dans cet article, nous avons étudié des algorithmes itératifs de multiplication-addition modulaire retournant un nombre congru à $(xy + w) \bmod m$. En calculant la borne supérieure exacte des termes $T[s - i]$ de l'algorithme de Kim et Sobelman, nous avons pu réduire approximativement d'un facteur deux la surface des opérateurs traitant un ensemble de moduli $m_1, m_2, \ldots, m_q$. Nous avons également montré que l'architecture du circuit effectuant la réduction modulaire nécessaire à l'obtention de $(xy + w) \bmod m$ dépend du choix de m.

L'opérateur inspiré de l'algorithme de Kim et Sobelman nécessite toutefois que l'opérande y soit inférieur à m. Il est donc nécessaire d'effectuer une réduction modulaire du résultat si celui-ci est utilisé comme opérande y d'une autre multiplication. Afin de résoudre ce problème, nous avons proposé une nouvelle relation de récurrence. Nous envisageons la conception d'un opérateur d'exponentiation modulaire exploitant notre opérateur ainsi que l'étude d'algorithmes dans de plus grandes bases afin d'étudier les compromis entre le nombre d'itérations, la surface du circuit et le débit des calculs.

A Preuve du théorème 1

Remarquons tout d'abord que la fonction φ dépend du modulo m choisi. Il est facile de vérifier que $\varphi(0) = 0$, $\varphi(1) = 2^n \bmod m = 2^n - m$,

$$\varphi(2) = 2^{n+1} \bmod m = \begin{cases} 2^{n+1} - 2m & \text{si } \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ 2^{n+1} - 3m & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases}$$

et

$$\varphi(3) = (3 \cdot 2^n) \bmod m = \begin{cases} 3 \cdot 2^n - 3m & \text{si } 2^{n-1} + 2^{n-2} + 1 \leq m \leq 2^n - 1 \\ 3 \cdot 2^n - 4m & \text{si } \lfloor (2^{n+1} + 2^n)/5 \rfloor + 1 \leq m \leq 2^{n-1} + 2^{n-2} \\ 3 \cdot 2^n - 5m & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor (2^{n+1} + 2^n)/5 \rfloor \end{cases}$$

Le théorème se démontre par récurrence sur la borne supérieure de $T[s-i]$. Notons que les hypothèses du théorème garantissent que $0 \leq x_{s-i}y + w_{s-i} \leq m$. Par conséquent, $2P[s-i] + x_{s-i}y + w_{s-i} \leq 2P[s-i] + m$. Comme $P[n] = 0$, il est évident que $0 \leq T[n-1] \leq m$. Supposons maintenant que $T[s-i]$ satisfasse l'inégalité (5) et prouvons que $T[s-i-1]$ la vérifie également en utilisant la relation de récurrence (3) et la définition de la fonction φ donnée ci-dessus. Nous distinguons quatre cas en fonction de la valeur de $T[s-i]$:

1. Si $0 \leq T[s-i] \leq 2^n - 1$, nous déduisons de la relation de récurrence (3) que $0 \leq P[s-i] \leq 2^n - 1$ et que $0 \leq T[s-i-1] \leq 2^{n+1} + m - 2$. Comme m appartient à $\{2^{n-1}+1, \ldots, 2^n - 1\}$, $P[s-i]$ et $T[s-i-1]$ satisfont respectivement les inégalités (4) et (5).
2. Lorsque $2^n \leq T[s-i] \leq 2^{n+1} - 1$, nous obtenons $2^n - m \leq P[s-i] \leq 2^{n+1} - m - 1$ et $2^{n+1} - m \leq T[s-i-1] \leq 2^{n+2} - m - 2$. Les inégalités (4) et (5) sont à nouveau vérifiées.
3. En appliquant le même principe que ci-dessus, nous déduisons que

   $$\begin{cases} 2^{n+1} - 2m \leq P[s-i] \leq 2^{n+1} + 2^n - 2m - 1 & \text{si } \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ 2^{n+1} - 3m \leq P[s-i] \leq 2^{n+1} + 2^n - 3m - 1 & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases}$$

   et

   $$\begin{cases} 2^{n+2} - 3m \leq T[s-i-1] \leq 2^{n+2} + 2^{n+1} - 3m - 2 & \text{si } \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ 2^{n+2} - 5m \leq T[s-i-1] \leq 2^{n+2} + 2^{n+1} - 5m - 2 & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases}$$

   lorsque $2^{n+1} \leq T[s-i] \leq 2^{n+1} + 2^n - 1$. Si $2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor$, il est facile de vérifier que :

   $$2^{n+2} + 2^{n+1} - 5m - 2 \leq 2^{n+2} - m - 2$$

4. Le dernier cas à étudier dépend de la valeur de m. Nous devons en effet considérer les deux alternatives suivantes :

   $$2^{n+1} + 2^n \leq T[s-i] \leq \begin{cases} 2^{n+2} + 2^{n+1} - 3m - 2 & \text{si } \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ 2^{n+2} - m - 2 & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases}$$

   La relation de récurrence (3) ainsi que la définition de la fonction φ nous permettent de déduire que :

   $$\begin{cases} 2^{n+1} + 2^n - 3m \leq P[s-i] \leq 2^{n+2} + 2^{n+1} - 6m - 2 & \text{si } 2^{n-1} + 2^{n-2} + 1 \leq m \leq 2^n - 1 \\ 2^{n+1} + 2^n - 4m \leq P[s-i] \leq 2^{n+2} + 2^{n+1} - 7m - 2 & \text{si } \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^{n-1} + 2^{n-2} \\ 2^{n+1} + 2^n - 4m \leq P[s-i] \leq 2^{n+2} - 5m - 2 & \text{si } \lfloor (2^{n+1} + 2^n)/5 \rfloor + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \\ 2^{n+1} + 2^n - 5m \leq P[s-i] \leq 2^{n+2} - 6m - 2 & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor (2^{n+1} + 2^n)/5 \rfloor \end{cases}$$

   Nous laissons au lecteur le soin de vérifier que $P[s-i]$ satisfait l'inégalité (4). $T[s-i-1]$ vérifie par conséquent l'inégalité (5) et le théorème est démontré.

Le modulo m appartenant à $\{2^{n-1}+1, \ldots, 2^n - 1\}$, $2^{n+1} + 2^n - 2m - 1$ est strictement supérieur à $2^{n+1} - m - 1$ et la borne supérieure de $T[s-i]$ est atteinte pour $m = \lfloor 2^{n+1}/3 \rfloor + 1$. Comme

$$\lfloor 2^{n+1}/3 \rfloor = \begin{cases} (2^{n+1} - 2)/3 & \text{si } n \text{ est pair} \\ (2^{n+1} - 1)/3 & \text{si } n \text{ est impair} \end{cases}$$

on obtient finalement

$$P_{\max} = \begin{cases} (2^{n+2} + 2^n - 5)/3 & \text{si } n \text{ est pair} \\ (2^{n+2} + 2^n - 7)/3 & \text{si } n \text{ est impair} \end{cases} \quad \text{et} \quad T_{\max} = \begin{cases} 2^{n+2} - 3 & \text{si } n \text{ est pair} \\ 2^{n+2} - 4 & \text{si } n \text{ est impair} \end{cases}$$

B Preuve du théorème 2

Le théorème 1 fournit une borne supérieure pour $P[s-i]$ en fonction de m. Il suffit donc de déterminer le plus grand entier k tel que $(xy+w) \bmod m + km$ soit inférieur à $P[s-i]$ quels que soient x, y et w. Il faut par conséquent résoudre l'inéquation suivante :

$$\underbrace{\min((xy+w) \bmod m)}_{=0} + km \leq P_{\max}$$

Nous obtenons ainsi :

$$k = \begin{cases} \lfloor (2^{n+1} + 2^n - 2m - 1)/m \rfloor & \text{si } \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ \lfloor (2^{n+1} - m - 1)/m \rfloor & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases}$$

En effectuant la division entière, nous établissons finalement que :

$$k = \begin{cases} 2 & \text{si } 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \text{ ou } \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^{n-1} + 2^{n-2} - 1 \\ 1 & \text{si } 2^{n-1} + 2^{n-2} \leq m \leq 2^n - 1 \end{cases}$$

Le théorème est ainsi prouvé.

References

[1] Andreas V. Curiger. *VLSI Architectures for Computations in Finite Rings and Fields*, volume 26 of *Series in Microelectronics*. Hartung-Gorre Verlag, 1993.

[2] Yong-Jin Jeong and Wayne P. Burleson. VLSI array algorithms and architectures for RSA modular multiplication. *IEEE Transactions on Very Large Scale Integration Systems*, 5(2):211–217, June 1997.

[3] Sungwook Kim and Gerald E. Sobelman. Digit-serial modular multiplication using skew-tolerant domino CMOS. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1173–1176. IEEE Computer Society, 2001.

[4] Yutai Ma. A simplified architecture for modulo $(2^n + 1)$ multiplication. *IEEE Transactions on Computers*, 47(3):333–337, March 1998.

[5] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[6] Peter L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, 1985.

[7] Siddika Berna Örs, Lejla Batina, Bart Preneel, and Joos Vandewalle. Hardware implementation of a Montgomery modular multiplier in a systolic array. In *Proceedings of the 17th International Parallel & Distributed Processing Symposium*. IEEE Computer Society, 2003.

[8] Damu Radhakrishnan and Yong Yuan. Novel approaches to the design of VLSI RNS multipliers. *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing*, 39(1):52–57, January 1992.

[9] Reto Zimmermann. Efficient VLSI implementation of modulo $(2^n \pm 1)$ addition and multiplication. In *Proceedings of the 14th IEEE Symposium on Computer Arithmetic*, pages 158–167. IEEE Computer Society, 1999.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399