Opérateurs itératifs de multiplication-addition modulaire pour FPGA

Jean-Luc Beuchat, Jean-Michel Muller

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Opérateurs itératifs de multiplication-addition modulaire pour FPGA

Jean-Luc Beuchat et Jean-Michel Muller

No 4937

Septembre 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4937--FR

INRIA
RHÔNE-ALPES

Opérateurs itératifs de multiplication-addition modulaire pour FPGA

Jean-Luc Beuchat et Jean-Michel Muller

Thème 2 — Génie logiciel
et calcul symbolique
Projet Arénaire

Rapport de recherche n ° 4937 — Septembre 2003 — 18 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. These operators involve small tables depending on the required set of moduli. A straightforward approach is to automatically generate a VHDL description of the modular multiplier according to $m_1, m_2, \ldots, m_q$. That design methodology assumes that the moduli are known a priori and works only for small sets of moduli. In order to avoid these drawbacks, we propose here an algorithm that allows to build the table while we perform the first steps of the modular multiplication.

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

Opérateurs itératifs de multiplication-addition modulaire pour 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 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. Ces différents opérateurs nécessitent toutefois de petites tables dépendant du modulo m choisi au moment de l'écriture du code VHDL. Afin de remédier à cet inconvénient, nous proposons une méthode récursive du calcul des tables s'effectuant parallèlement aux premières itérations de la multiplication modulaire.

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

Ce document constitue une version étendue du rapport de recherche LIP n° 2003–31. Nous proposons une démonstration du théorème 3 et décrivons un nouvel algorithme supprimant la génération de tables dépendant du modulo m lors de la génération du code VHDL.

1 Introduction

La multiplication modulaire intervient, entre autres, 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 calculant 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-E de Xilinx (section 3).

Ces premiers opérateurs nécessitent toutefois de petites tables (de 3 à 6 mots de la taille des opérandes) dépendant du modulo m choisi. Ce paramètre doit par conséquent être connu au moment de la description VHDL du multiplieur qui ne fonctionnera que pour une unique valeur de m. Cette caractéristique peut s'avérer trop restrictive dans certaines applications. S'il est possible de générer les tables correspondant à un ensemble $\{m_1, \ldots, m_q\}$ de moduli, il nous semble plus intéressant de développer un nouvel opérateur dont le modulo m est l'une des entrées. Nous proposons dans la section 4 une méthode récursive de construction des tables s'effectuant parallèlement aux premières itérations de l'algorithme de multiplication modulaire.

Rappelons que les FPGA considérés dans ce travail sont essentiellement constitués de blocs logiques programmables (CLB ou *Configurable Logic Block*), d'une matrice de routage configurable, de blocs de mémoire et de blocs d'entrée/sortie. Chaque CLB est divisé en deux tranches contenant chacune deux générateurs de fonctions de quatre variables (LUT ou *look-up table*), deux éléments de mémorisation et de la logique additionnelle permettant une implantation efficace d'additionneurs à retenue propagée. Ce dernier point est important : en pratique, il n'est d'aucun intérêt d'utiliser sur FPGA des techniques d'addition redondante comme l'addition *carry-save*.

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 $n+q$ 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 q 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] :

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

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).

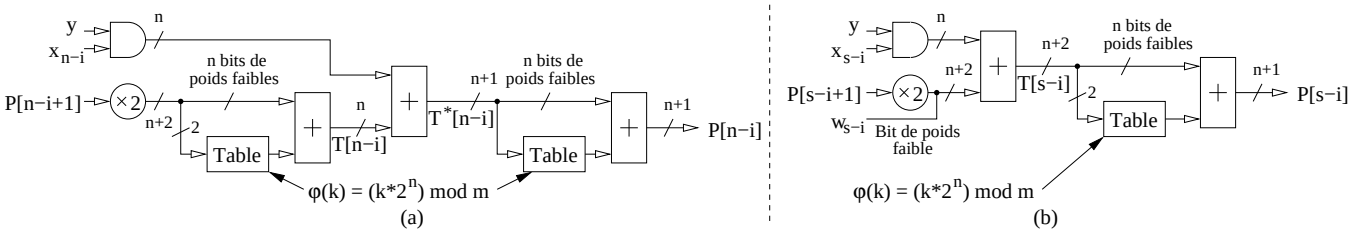

FIG. 1 – Un étage d'itération d'un multiplieur modulaire pour FPGA. (a) Algorithme de Jeong et Burleson. (b) Algorithme de Kim et Sobelman modifié.

Kim et Sobelman ont proposé une première amélioration de cet algorithme dans la référence [3]. Nous donnons ici l'itération en base deux traditionnelle qui se déduit facilement de la version *carry-save* définie dans [3] :

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

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 :

$$0 \leq T[s-i] \leq 2^{n+2} + 2^n - 7 \tag{4}$$

Cette borne indique 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-E 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* 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.

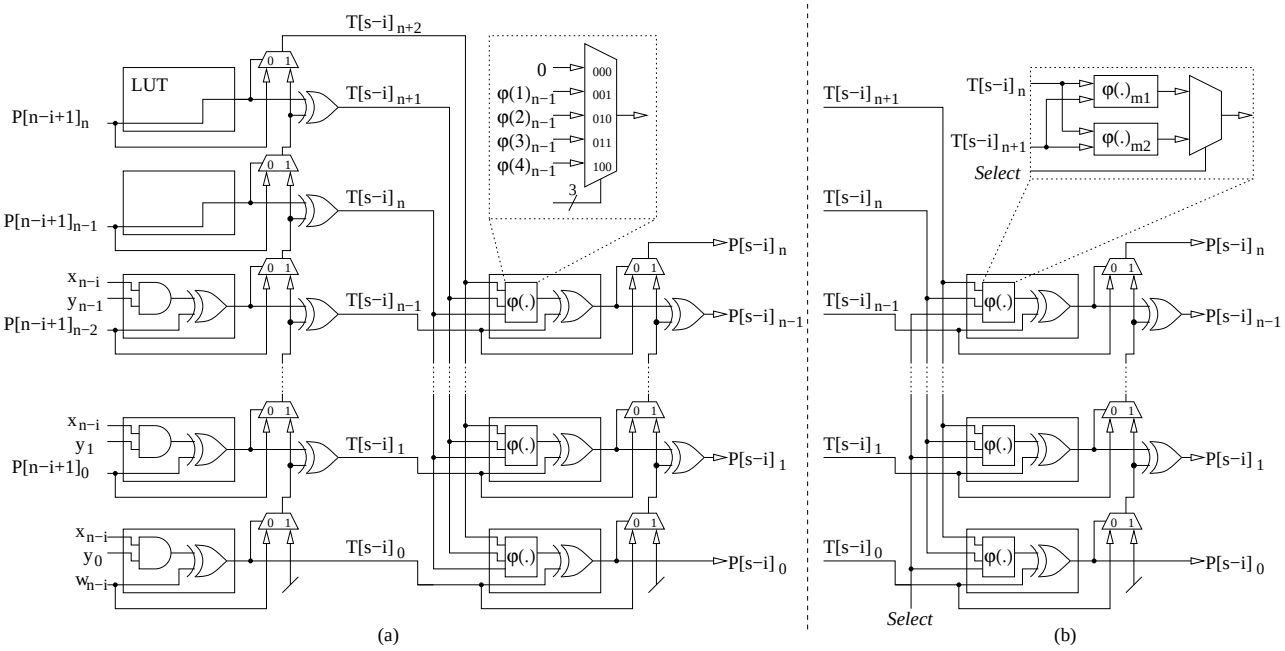

FIG. 2 – Implantation d'un étage d'itération sur un FPGA de la famille Virtex-E. (a) Utilisation de la borne indiquant que $T[s-i]$ est un nombre de $n+3$ bits. (b) Impact du théorème 1 sur l'architecture de l'opérateur.

2.1 Calcul de la borne supérieure de $T[s-i]$

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 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 :*

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

et

$$0 \leq T[s-i] \leq \begin{cases} 2^{n+2} + 2^{n+1} - 3m - 2 & si\ \lfloor 2^{n+1}/3 \rfloor + 1 \leq m \leq 2^n - 1 \\ 2^{n+2} - m - 2 & si\ 2^{n-1} + 1 \leq m \leq \lfloor 2^{n+1}/3 \rfloor \end{cases} \tag{6}$$

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

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

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

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

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 présentée, 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* $P[0] = (xy+w)\ mod m + km$, *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.

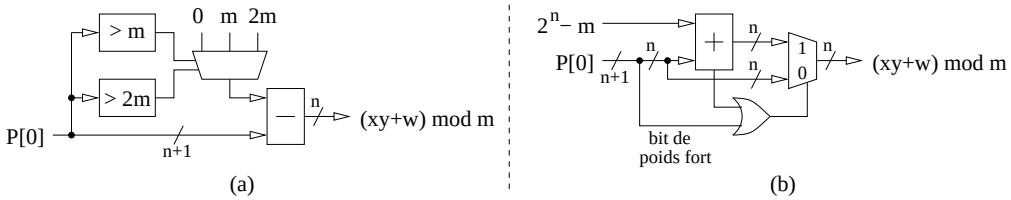

FIG. 3 – Correction de $P[0]$ afin d'obtenir $(xy+w) \bmod m$. (a) $P[0]$ est égal à $(xy+w) \bmod m$, à $(xy+w) \bmod m + m$ ou à $(xy+w) \bmod m + 2m$. (b) $P[0]$ est égal à $(xy+w) \bmod m$ ou à $(xy+w) \bmod m + m$.

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 :

$$\begin{cases} T[s-i] = 2P[s-i+1] + x_{s-i}y + w_{s-i} \\ P[s-i] = T[s-i] \bmod 2^{n-1} + \psi(T[s-i] \operatorname{div} 2^{n-1}) \end{cases} \tag{7}$$

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 soit un entier de $n+1$ bits tel que

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

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 $(xy + w) \bmod m$ ou à $(xy + w) \bmod m + m$.

La démonstration, analogue à celle du théorème 1, est proposée dans l'annexe C. 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 comparer la surface d'un circuit utilisant la borne définie par l'inégalité (4) à celle d'un circuit exploitant le théorème 1. 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 (figure 4).

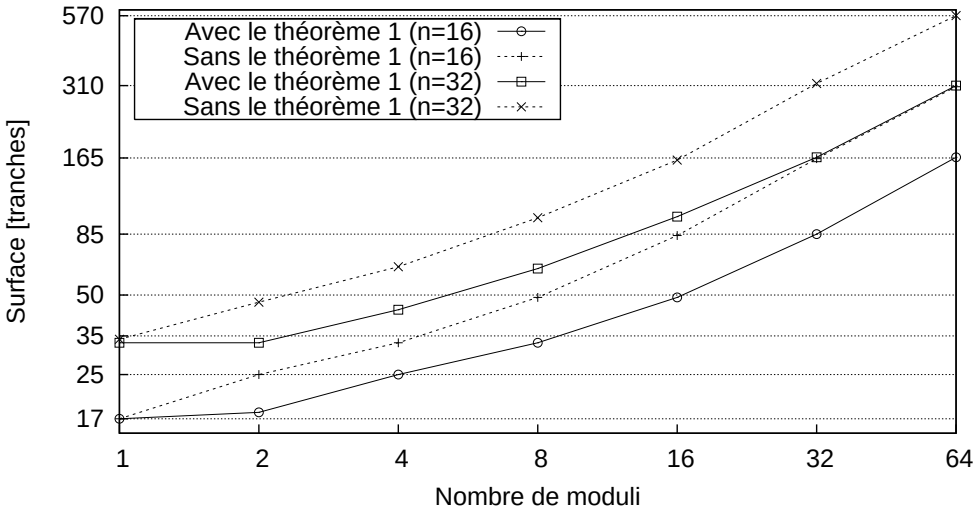

FIG. 4 – Surface d'un étage de l'itération de Kim et Sobelman en fonction du nombre de moduli sur un circuit XCV100E-6 (moduli générés aléatoirement ; synthèse, placement et routage effectués avec ISE 5.1.03i). L'échelle utilisée en ordonnée est logarithmique.

Considérons maintenant un étage de calcul de la nouvelle itération définie par l'équation (7). En plaçant un registre en sortie de ce composant et en utilisant un mécanisme de rebouclage, nous pouvons calculer un nombre 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 Calcul des fonctions φ et ψ

Les opérateurs décrits jusqu'ici fonctionnent uniquement pour un ensemble particulier de moduli et nécessitent des tables précalculées au moment de la génération du code VHDL. Il serait toutefois intéressant de disposer d'un opérateur capable de calculer le produit xy modulo m pour n'importe quelle valeur de m strictement comprise entre 2^{n-1} et 2^n. Dans la conclusion de leur article, Jeong et Burlseon suggèrent de construire les tables contenant les valeurs de $\varphi(k)$ avant de débuter la multiplication [2]. Ils considèrent que le coût de ce prétraitement est acceptable si le même modulo m intervient lors de plusieurs opérations successives, comme, par exemple, lors d'une exponentiation modulaire. Ils ne proposent cependant aucune méthode d'évaluation des coefficients $\varphi(k)$. En constatant 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

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

nous pouvons construire la table implantant la fonction φ à l'aide de comparateurs, de multiplexeurs, de soustracteurs et de multiplieurs par des constantes calculant les termes $3m$ et $5m$. Le circuit résultant de cette approche s'avérerait toutefois plus complexe que l'étage de calcul illustré par la figure 1b et la fréquence maximale de fonctionnement de l'opérateur dépendrait de l'étape de prétraitement. Nous proposons ici une méthode de calcul récursive des coefficients $\varphi(k)$ pouvant s'effectuer parallèlement aux premières itérations des algorithmes de multiplication modulaire précédemment décrits. Considérons une fonction α paramétrée par m et définie par

$$\alpha(m) = 2m - 2^n = \sum_{i=0}^{n-2} m_i 2^{i+1}$$

L'évaluation de $\alpha(m)$ en matériel est triviale : il suffit en effet de décaler m d'une position vers la gauche tout en négligeant le bit de poids fort m_{n-1}. Cette fonction permet toutefois le calcul itératif de φ, comme l'indique le théorème suivant démontré dans l'annexe D.

Théorème 4
Les valeurs successives de $\varphi(k)$, $k \geq 1$, se calculent récursivement à l'aide de la relation suivante

$$\varphi(k) = \begin{cases} \varphi(k-1) - \alpha(m) & \textit{si } \varphi(k-1) - \alpha(m) \geq 0 \\ \varphi(k-1) - \alpha(m) + m & \textit{sinon} \end{cases}$$

où $\varphi(0) = 0$.

La figure 5a illustre l'architecture du circuit générant la table contenant les coefficients $\varphi(1)$, $\varphi(2)$ et $\varphi(3)$. Initialement, les trois registres contiennent la valeur 0. Lors du premier cycle d'horloge, ce dispositif évalue $\varphi(1) = \varphi(0) - \alpha(m) + m$ et mémorise le résultat dans les registres 1 et 3. Ce dernier est utilisé pour implanter le mécanisme de rétroaction. Lors des deuxième et troisième cycles d'horloge, les valeurs $\varphi(2)$ et $\varphi(3)$ sont respectivement calculées. Une petite unité de contrôle génère un signal d'initialisation ainsi que les signaux de chargement des registres

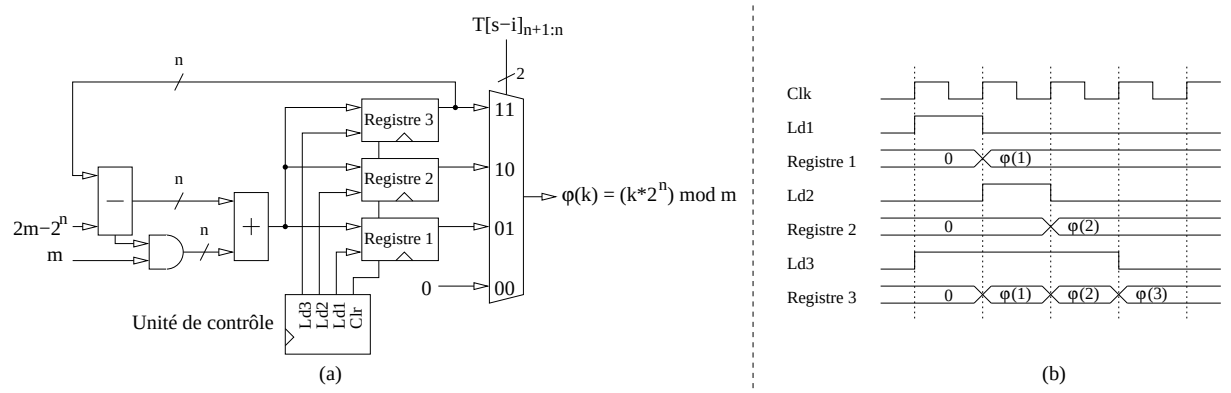

FIG. 5 – Calcul itératif de la fonction φ. (a) Architecture générale du circuit. (b) Génération des signaux de contrôle.

(figure 5b). Finalement, un multiplexeur commandé par les deux bits de poids forts de $T[s-i]$ sélectionne la valeur désirée dans la table. La complexité du circuit ainsi obtenu est inférieure à celle d'un étage de calcul de la relation de récurrence et la construction de la table n'a plus d'impact sur la fréquence de fonctionnement de l'opérateur.

Etudions maintenant l'évolution de $T[s-i]$ lors des premières itérations de l'algorithme de multiplication modulaire défini par la relation de récurrence (3). Comme l'opérande y est supposé strictement inférieur à m, nous savons que $0 \leq T[s-1] \leq m$. Par conséquent, seule la valeur $\varphi(0) = 0$ intervient lors de la première étape et $0 \leq P[s-1] \leq m$. Lors de la seconde itération, nous obtenons $0 \leq T[s-2] \leq 3m < 2^{n+1} + 2^n - 1$. Il faut donc disposer de $\varphi(0)$, $\varphi(1)$ et $\varphi(2)$ afin de calculer $P[s-2]$. La solution consiste à retarder le début de la multiplication d'un cycle d'horloge afin de calculer $\varphi(1)$ lors d'une étape de prétraitement. Nous évaluons ensuite en parallèle $P[s-1]$ et $\varphi(2)$, puis $P[s-2]$ et $\varphi(3)$.

La méthode de calcul de $\varphi(k)$ établie dans ce paragraphe s'applique également à la fonction ψ intervenant dans l'algorithme défini par la relation de récurrence (7). Définissons encore une fonction $\beta(m)$ telle que $\beta(m) = m - 2^{n-1}$. Le théorème suivant, dont la démonstration est analogue à celle du théorème 4, permet le calcul itératif de la fonction ψ.

Théorème 5
Les valeurs successives de $\psi(k)$, $k \geq 2$, se calculent récursivement à l'aide des relations suivantes

$$\psi(k) = \begin{cases} \psi(k-1) - \beta(m) & \textit{si } \psi(k-1) - \beta(m) \geq 0 \\ \psi(k-1) - \beta(m) + m & \textit{sinon} \end{cases}$$
$$= \begin{cases} \psi(k-2) - \alpha(m) & \textit{si } \psi(k-2) - \alpha(m) \geq 0 \\ \psi(k-2) - \alpha(m) + m & \textit{sinon} \end{cases}$$

où $\psi(0) = 0$ et $\psi(1) = 2^{n-1}$.

Il faut à nouveau déterminer quelles valeurs de $\psi(k)$ sont nécessaires lors des premières itérations afin de synchroniser correctement la génération de la table et la multiplication. Nous déduisons de la relation de récurrence (7) et du théorème 3 que

$$0 \leq T[s-1] \leq \begin{cases} (2^{n+2} + 11)/3 & \text{si } n \text{ est pair} \\ (2^{n+2} + 7)/3 & \text{si } n \text{ est impair} \end{cases}$$

Si n est supérieur ou égal à cinq, il est facile de montrer que $T[s-1] \leq 3 \cdot 2^{n-1} - 1$. Nous déduisons respectivement de

$$(2^{n+2} + 11)/3 \leq 3 \cdot 2^{n-1} - 1$$

et

$$(2^{n+2}+7)/3 \leq 3 \cdot 2^{n-1} - 1$$

que $2^{n-1} \geq 14$ si n est pair et $2^{n-1} \geq 10$ si n est impair. Ces inégalités sont satisfaites lorsque $n \geq 5$ et seules les valeurs de $\psi(0)$, $\psi(1)$ et $\psi(2)$ interviennent lors de la première itération de l'algorithme. Il faut maintenant distinguer deux cas en fonction de la valeur de $x_{n-1}y + w_{n-1}$:

- Si $0 \leq y \leq 2^n - 2$, alors $0 \leq x_{n-1}y + w_{n-1} \leq 2^n - 1$ et nous déduisons de la relation de récurrence (7) ainsi que de la définition de la fonction ψ que $0 \leq P[s-1] \leq 2^n - 1$. Par conséquent,

$$0 \leq T[s-2] \leq \underbrace{2^{n+1} - 2}_{2P[s-1]} + \underbrace{2^n - 2 + 1}_{\max(x_{n-2}y + w_{n-2})} = 5 \cdot 2^{n-1} + 2^{n-1} - 3$$

  et le calcul de $P[s-2]$ nécessite les valeurs de $\psi(k)$ pour $k \in \{0, \ldots, 5\}$.
- Lorsque $2^n - 1 \leq y \leq y_{\max}$, nous établissons que $2^n - m \leq P[s-1] \leq y_{\max} + 1 - m$ et que

$$\begin{aligned} T[s-2] &\leq 3(y_{\max}+1) - 2m \\ &\leq \begin{cases} 2^{n+2} + 11 - 2(2^{n-1}+1) & \text{si } n \text{ est pair} \\ 2^{n+2} + 7 - 2(2^{n-1}+1) & \text{si } n \text{ est impair} \end{cases} \\ &= \begin{cases} 6 \cdot 2^{n-1} + 9 & \text{si } n \text{ est pair} \\ 6 \cdot 2^{n-1} + 5 & \text{si } n \text{ est impair} \end{cases} \end{aligned}$$

  La valeur de $\psi(6)$ est donc nécessaire lors de la seconde itération de l'algorithme. Définissons $y'_{\max}$ tel que

$$0 \leq y'_{\max} < (2^{n+2}+1)/3$$

  Il est évident que, si $2^n - 1 \leq y \leq y'_{\max}$, la valeur de $T[s-2]$ est inférieure ou égale à $6 \cdot 2^{n-1} - 1$ et $\psi(6)$ n'intervient plus lors du calcul de $P[s-2]$.

Ces diverses considérations nous permettent de formuler le théorème suivant :

Théorème 6
Considérons la relation de récurrence (7) avec $n \geq 5$ et $0 \leq y \leq y'_{max} < (2^{n+2}+1)/3$. La construction de la table définissant la fonction $\psi(k)$ peut s'effectuer parallèlement aux premières étapes de la multiplication-addition modulaire :

- *les valeurs de $\psi(2)$ et $\psi(3)$ sont évaluées à partir de $\psi(0)$ et $\psi(1)$ lors d'une étape de prétraitement ;*
- *les valeurs de $\psi(4)$ et $\psi(5)$ sont ensuite calculées parallèlement à $P[s-1]$;*
- *finalement, le calcul de $\psi(6)$ et $\psi(7)$ s'effectue simultanément à celui de $P[s-2]$.*

La figure 6 illustre l'architecture du circuit évaluant la fonction ψ conformément au théorème 6. Remarquons encore que $P_{\max} < y'_{\max}$. Les nouvelles conditions imposées à y permettent toujours d'effectuer une exponentiation modulaire sans réduction modulo m entre deux multiplications successives.

Le tableau 1 présente quelques résultats de synthèse sur un circuit XCV400E-6. Ces expériences permettent de vérifier que le temps de cycle ne dépend pas du calcul des fonctions $\varphi(k)$ ou $\psi(k)$. Comme la table n'est plus intégrée dans les LUT de l'additionneur établissant $P[s-i]$, la fréquence maximale de fonctionnement s'avère toutefois inférieure à celle de l'opérateur de la figure 2a.

5 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 adapté à la représentation binaire traditionnelle, nous avons pu réduire approximativement d'un

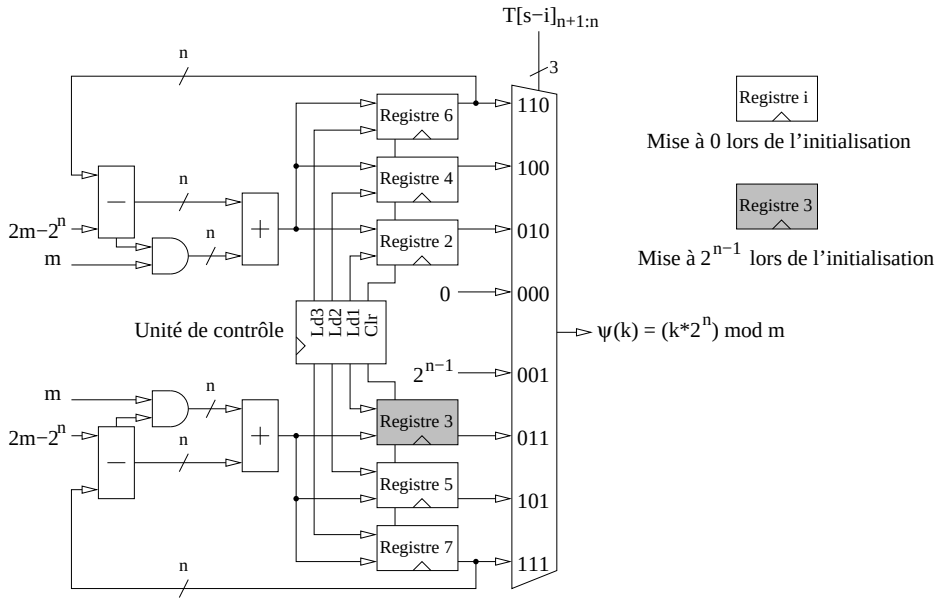

FIG. 6 – Calcul itératif de la fonction ψ.

	8 bits	16 bits	24 bits	32 bits
Itération (3)	44 tranches	80 tranches	117 tranches	155 tranches
(calcul de $\varphi(k)$)	13 ns	15.6 ns	17.2 ns	19.5 ns
Itération (7)	69 tranches	130 tranches	192 tranches	254 tranches
(calcul de $\psi(k)$)	14.5 ns	15.5 ns	17.4 ns	20 ns

TAB. 1 – Surface et délai d'un opérateur de multiplication-addition avec calcul des fonctions $\varphi(k)$ ou $\psi(k)$ sur un circuit XCV400E-6.

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 indispensable 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. Finalement, les méthodes de calcul itératives des fonctions φ et ψ permettent respectivement la suppression des tables contenant les valeurs précalculées des termes $(k \cdot 2^n) \bmod m$ et $(k \cdot 2^{n-1}) \bmod m$. Le code VHDL de notre multiplieur-additionneur ne dépend par conséquent plus de m.

Nous souhaitons adapter notre nouvel algorithme aux circuits ASIC. Il s'agit essentiellement d'étendre notre travail à l'itération en *carry-save* propsée par Kim et Sobelman [3]. Nous envisageons également 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 (6) 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 (5) et (6).
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 (5) et (6) 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 (5). $T[s-i-1]$ vérifie par conséquent l'inégalité (6) 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

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

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 : $\min((xy+w)\bmod m)+km\leq P_{\max}$. Comme $\min((xy+w)\bmod m)=0$, nous obtenons :

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

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.

C Preuve du théorème 3

La fonction ψ dépend évidemment du modulo m choisi. Il est facile de vérifier que $\psi(0)=0$, $\psi(1)=2^{n-1}$, $\psi(2)=2^n-m$,

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

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

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

$$\psi(6)=\begin{cases}2^{n+1}+2^n-3m & \text{si } 2^{n-1}+2^{n-2}+1\leq m\leq 2^n-1\\ 2^{n+1}+2^n-4m & \text{si } \lfloor(2^{n+1}+2^n)/5\rfloor+1\leq m\leq 2^{n-1}+2^{n-2}+1\\ 2^{n+1}+2^n-5m & \text{si } 2^{n-1}+1\leq m\leq\lfloor(2^{n+1}+2^n)/5\rfloor\end{cases}$$

$$\psi(7)=\begin{cases}2^{n+1}+2^n+2^{n-1}-3m & \text{si } 2^{n-1}+2^{n-2}+2^{n-3}+1\leq m\leq 2^n-1\\ 2^{n+1}+2^n+2^{n-1}-4m & \text{si } \lfloor 7\cdot 2^{n-1}/5\rfloor+1\leq m\leq 2^{n-1}+2^{n-2}+2^{n-3}\\ 2^{n+1}+2^n+2^{n-1}-5m & \text{si } \lfloor 7\cdot 2^{n-1}/6\rfloor+1\leq m\leq\lfloor 7\cdot 2^{n-1}/5\rfloor\\ 2^{n+1}+2^n+2^{n-1}-6m & \text{si } 2^{n-1}+1\leq m\leq\lfloor 7\cdot 2^{n-1}/6\rfloor\end{cases}$$

Le théorème se démontre par récurrence sur les bornes supérieures de $P[s-i+1]$ et $T[s-i]$. Lors de la première itération, les hypothèses du théorème nous garantissent que $P[s]=0$ et

$$0 \leq T[s-1] \leq \begin{cases} (2^{n+2}+11)/3 & \text{si } n \text{ est pair} \\ (2^{n+2}+7)/3 & \text{si } n \text{ est impair} \end{cases}$$

$P[s]$ et $T[s-1]$ sont ainsi respectivement inférieurs à $P_{\max}$ et $T_{\max}$. Supposons maintenant que le théorème soit vérifié pour $P[s-i+1]$ et $T[s-i]$ et montrons qu'il l'est également pour $P[s-i]$ et $T[s-i-1]$. Nous distinguons huit cas en fonction de la valeur de $T[s-i]$:

1. Si $0 \leq T[s-i] \leq 2^{n-1}-1$, la relation de récurrence (7) indique que $0 \leq P[s-i] \leq 2^{n-1}-1$ et que

$$0 \leq T[s-i-1] \leq 2^n - 2 + \begin{cases} (2^{n+2}+11)/3 & \text{si } n \text{ est pair} \\ (2^{n+2}+7)/3 & \text{si } n \text{ est impair} \end{cases}$$

   $P[s-i]$ et $T[s-i-1]$ sont donc bien respectivement inférieurs à $P_{\max}$ et $T_{\max}$.

2. Supposons que $2^{n-1} \leq T[s-i] \leq 2^n-1$. En appliquant le même raisonnement que ci-dessus, nous déduisons que $2^{n-1} \leq P[s-i] \leq 2^n-1$ et que

$$2^n \leq T[s-i-1] \leq 2^{n+1} - 2 + \begin{cases} (2^{n+2}+11)/3 & \text{si } n \text{ est pair} \\ (2^{n+2}+7)/3 & \text{si } n \text{ est impair} \end{cases}$$

   Les valeurs de $P[s-i]$ et $T[s-i]$ sont à nouveau inférieures aux bornes $P_{\max}$ et $T_{\max}$.

3. Lorsque $2^n \leq T[s-i] \leq 2^n + 2^{n-1} - 1$, la définition de la fonction ψ nous indique que $2^n - m \leq P[s-i] \leq 2^n + 2^{n-1} - m - 1$ et que

$$2^{n+1} - 2m \leq T[s-i-1] \leq 2^{n+1} + 2^n - 2m - 2 + \begin{cases} (2^{n+2}+11)/3 & \text{si } n \text{ est pair} \\ (2^{n+2}+7)/3 & \text{si } n \text{ est impair} \end{cases}$$

   Il est facile de vérifier que les valeurs de $P[s-i]$ et $T[s-i]$ sont inférieures aux bornes $P_{\max}$ et $T_{\max}$.

4. En appliquant le même principe que ci-dessus, nous déduisons que

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

   et que

$$\begin{cases} 2^{n+1} + 2^n - 2m \leq T[s-i-1] \leq 2^{n+2} - 2m - 1 + y_{\max} & \text{si } 2^{n-1} + 2^{n-2} + 1 \leq m \leq 2^n - 1 \\ 2^{n+1} + 2^n - 4m \leq T[s-i-1] \leq 2^{n+2} - 4m - 1 + y_{\max} & \text{si } 2^{n-1} + 1 \leq m \leq 2^{n-1} + 2^{n-2} \end{cases}$$

   lorsque $2^n + 2^{n-1} \leq T[s-i] \leq 2^{n+1} - 1$. La valeur maximale de $T[s-i-1]$ s'obtient pour $m = 2^{n-1} + 2^{n-2} + 1$. Nous laissons au lecteur le soin de vérifier que les bornes $P_{\max}$ et $T_{\max}$ sont à nouveau respectées.

5. Si $2^{n+1} \leq T[s-i] \leq 2^{n+1} + 2^{n-1} - 1$, la relation de récurrence et la définition de la fonction ψ indiquent que

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

   et

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

   La vérification du respect des bornes $P_{\max}$ et $T_{\max}$ s'effectue en substituant $m = \lfloor 2^{n+1}/3 \rfloor + 1$ dans les inégalités ci-dessus.

6. Lorsque $2^{n+1}+2^{n-1} \le T[s-i] \le 2^{n+1}+2^n-1$, nous obtenons :

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

et, par conséquent,

$$\begin{cases} 2^{n+2}+2^n-4m \le T[s-i-1] \le 2^{n+2}+2^{n+1}-4m-1+y_{\max} & \text{si } \lfloor (2^{n+1}+2n-1)/3 \rfloor + 1 \le m \le 2^n-1 \\ 2^{n+2}+2^n-6m \le T[s-i-1] \le 2^{n+2}+2^{n+1}-6m-1+y_{\max} & \text{si } 2^{n-1}+2^{n-2}+1 \le m \le \lfloor (2^{n+1}+2n-1)/3 \rfloor \\ 2^{n+2}+2^n-8m \le T[s-i-1] \le 2^{n+2}+2^{n+1}-8m-1+y_{\max} & \text{si } 2^{n-1}+1 \le m \le 2^{n-1}+2^{n-2} \end{cases}$$

Les conditions imposées sur m dans les inéquations ci-dessus garantissent que les valeurs maximales de de $P[s-i]$ et $T[s-i-1]$ s'obtiennent lorsque $m = \lfloor (2^{n+1}+2n-1)/3 \rfloor + 1$. Ainsi,

$$\begin{aligned} P[s-i] &\le 2^{n+1}+2^n-1-\begin{cases} 2\cdot((2^{n+1}+2^{n-1}-1)/3+1) & \text{si } n \text{ est pair} \\ 2\cdot((2^{n+1}+2^{n-1}-2)/3+1) & \text{si } n \text{ est impair} \end{cases} \\ &= \begin{cases} (2^{n+2}-7)/3 & \text{si } n \text{ est pair} \\ (2^{n+2}-5)/3 & \text{si } n \text{ est impair} \end{cases} \end{aligned}$$

et

$$\begin{aligned} T[s-i-1] &\le 2^{n+2}+2^{n+1}-2+\begin{cases} (2^{n+2}+11)/3-4\cdot((2^{n+1}+2^{n-1}-1)/3+1) & \text{si } n \text{ est pair} \\ (2^{n+2}+7)/3-4\cdot((2^{n+1}+2^{n-1}-2)/3+1) & \text{si } n \text{ est impair} \end{cases} \\ &= 2^{n+2}-1 \end{aligned}$$

Les bornes $P_{\max}$ et $T_{\max}$ sont donc atteintes pour $m = \lfloor (2^{n+1}+2n-1)/3 \rfloor + 1$. Il reste à terminer la preuve en montrant que ces bornes sont également respectées lorsque $T[s-i] \operatorname{div} 2^{n-1} = 6$ et $\operatorname{div} 2^{n-1} = 7$.

7. Si $2^{n+1}+2^n \le T[s-i] \le 2^{n+1}+2^n+2^{n-1}-1$, nous obtenons les inégalités suivantes pour $P[s-i]$ et $T[s-i-1]$:

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

et

$$\begin{cases} 2^{n+2}+2^{n+1}-6m \le T[s-i-1] \\ \quad \le 2^{n+2}+2^{n+1}+2^n-6m-1+y_{\max} & \text{si } 2^{n-1}+2^{n-2}+1 \le m \le 2^n-1 \\ 2^{n+2}+2^{n+1}-8m \le T[s-i-1] \\ \quad \le 2^{n+2}+2^{n+1}+2^n-8m-1+y_{\max} & \text{si } \lfloor (2^{n+1}+2^n)/5 \rfloor + 1 \le m \le 2^{n-1}+2^{n-2}+1 \\ 2^{n+2}+2^{n+1}-10m \le T[s-i-1] \\ \quad \le 2^{n+2}+2^{n+1}+2^n-10m-1+y_{\max} & \text{si } 2^{n-1}+1 \le m \le \lfloor (2^{n+1}+2^n)/5 \rfloor \end{cases}$$

La substitution de $m = 2^{n-1}+2^{n-2}+1$ et de $y_{\max}$ dans les inégalités définissant $P[s-i]$ et $T[s-i-1]$ permet de vérifier que $P[s-i] < P_{\max}$ et $T[s-i-1] < T_{\max}$.

8. Finalement, lorsque $2^{n+1}+2^n+2^{n-1} \le T[s-i] \le 2^{n+2}-1$, la définition de ψ et la relation de récurrence nous permettent de borner respectivement $P[s-i]$ et $T[s-i-1]$ par

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

et

$$\begin{cases} 2^{n+2}+2^{n+1}+2^n-6m \le T[s-i-1] \le 2^{n+3}-1-6m+y_{\max} & \text{si } 2^{n-1}+2^{n-2}+2^{n-3}+1 \le m \le 2^n-1 \\ 2^{n+2}+2^{n+1}+2^n-8m \le T[s-i-1] \le 2^{n+3}-1-8m+y_{\max} & \text{si } \lfloor 7\cdot 2^{n-1}/5 \rfloor + 1 \le m \le 2^{n-1}+2^{n-2}+2^{n-3} \\ 2^{n+2}+2^{n+1}+2^n-10m \le T[s-i-1] \le 2^{n+3}-1-10m+y_{\max} & \text{si } \lfloor 7\cdot 2^{n-1}/6 \rfloor + 1 \le m \le \lfloor 7\cdot 2^{n-1}/5 \rfloor \\ 2^{n+2}+2^{n+1}+2^n-12m \le T[s-i-1] \le 2^{n+3}-1-12m+y_{\max} & \text{si } 2^{n-1}+1 \le m \le \lfloor 7\cdot 2^{n-1}/6 \rfloor \end{cases}$$

La substitution de $m = 2^{n-1}+2^{n-2}+2^{n-3}+1$ et de $y_{\max}$ dans les inégalités définissant $P[s-i]$ et $T[s-i-1]$ indique à nouveau que $P[s-i] < P_{\max}$ et $T[s-i-1] < T_{\max}$ et le théorème est ainsi démontré.

D Preuve du théorème 4

Montrons tout d'abord que les coefficients $\varphi(k)$ peuvent se calculer itérativement grâce à la relation suivante :

$$\varphi(k) = \begin{cases} 0 & \text{si } k = 0 \\ (\varphi(k-1) - \alpha(m)) \bmod m & \text{sinon} \end{cases}$$

Il est facile de vérifier que, pour $k \ge 1$,

$$\begin{aligned} \varphi(k) &= (\varphi(k-1) - \alpha(m)) \bmod m = (((k-1)2^n) \bmod m - 2m + 2^n) \bmod m \\ &= ((k-1)2^n - 2m + 2^n) \bmod m = (k \cdot 2^n) \bmod m \end{aligned}$$

Comme $0 \le \varphi(k-1) \le m-1$ et $2^{n-1}+1 \le m \le 2^n-1$, nous déduisons de plus que :

$$-m+1 \le -2m+2^n \le \varphi(k-1) - 2m + 2^n \le 2^n - m - 1 \le m-1$$

Par conséquent,

$$\varphi(k) = (\varphi(k-1) - \alpha(m)) \bmod m = \begin{cases} \varphi(k-1) - \alpha(m) & \text{si } \varphi(k-1) - \alpha(m) \ge 0 \\ \varphi(k-1) - \alpha(m) + m & \text{sinon} \end{cases}$$

et le théorème est établi.

Bibliographie

[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