2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools

Arithmetic units for RNS moduli $\{2^n - 3\}$ and $\{2^n + 3\}$ operations

Pedro Miguens Matutino
Department of Electronics, Telecommunications and Computer Engineering
ISEL / INESC-ID / IST
Lisbon, Portugal
pmiguens@deetc.isel.pt

Ricardo Chaves
Department of Electrical and Computer Engineering
IST / INESC-ID
Lisbon, Portugal
ricardo.chaves@inesc-id.pt

Leonel Sousa
Department of Electrical and Computer Engineering
IST / INESC-ID
Lisbon, Portugal
las@inesc-id.pt

***Abstract*—A new moduli set $\{2^n - 1, 2^n + 3, 2^n + 1, 2^n - 3\}$ has recently been proposed to represent numbers in Residue Number Systems (RNS), increasing the number of channels. With this, the processing time can be reduced by simultaneously exploiting the carry-free characteristic of the modular arithmetic and improving the parallelism. In this paper, hardware structures for addition and multiplication operation in RNS for the moduli $\{2^n - 3\}$ and $\{2^n + 3\}$ are proposed and analyzed. In order to evaluate the performance of the proposed units they were implemented on an ASIC technology. The obtained experimental results suggest that the performance of the moduli $\{2^n \pm 3\}$ are acceptable but demand more area resource and impose a larger delay than the typically used $\{2^n \pm 1\}$ arithmetic units. Addition units require at least 42% more area for a performance identical to the $\{2^n + 1\}$ modulo adder. The multiplication units require up to 37% more area and impose a delay 25% higher. This paper also suggests that more balanced moduli sets should be developed in order to achieve more efficient RNS.**

I. Introduction

Residue Number Systems (RNS) are a good alternative to the conventional arithmetic, based on a weighted number system. One of the key advantages of RNS is the fact that it is a carry free propagation scheme, the main cause for performance degradation in arithmetic circuits. In applications requiring intensive computation, such as digital signal processing [1], [2], this carry free characteristics allow for concurrent computation in each of the RNS moduli channels. Over the last years many authors have proposed several moduli sets and arithmetic structures, in order to achieve better performance. The main focus of the RNS community is being given to the development of novel conversion units, somewhat disregarding the arithmetic units. However, another approach should also be considered, in order to take into account the complexity of the basic operations on each channel, in particular the multiplication units, usually the most critical units of the RNS implementations. Depending on the moduli set different computational delays can be observed, for the same channel bit length. In order to achive more efficient and balanced systems the length of each channel should be adjusted, to achieve identical delays on each moduli set channels.

The most common moduli sets used in RNS applications is the traditional $\{2^n - 1, 2^n, 2^n + 1\}$ set [3]. Also, the novel $\{2^{2n} - 1, 2^n, 2^{2n} + 1\}$[4] moduli set, with a dynamic range of $5n$ bit. Recently, a 4-modulus base set $\{2^n - 1, 2^n + 3, 2^n + 1, 2^n - 3\}$ has been proposed [5] as a hierarchical base composed by two pairs of moduli [6], [7].

Even though several moduli sets have been proposed using $\{2^n \pm 3\}$ modulo operations, no significance advancement has been made in the design of the required arithmetic units. In this paper, improved units for the $\{2^n-3\}$ and $\{2^n+3\}$ moduli sets are proposed. A comparative analysis of area and delay for the $\{2^n\}$, $\{2^n \pm 1\}$, and $\{2^n \pm 3\}$ is also performed. As it is further discussed in this paper, these results will show that to achieve better RNS performance and area results, the length of each moduli set should not be equal. Results also suggest that the proposed $\{2^n - 3\}$ and $\{2^n + 3\}$ moduli addition and multiplication units improve the current state of the art by 40% in terms of area and 10% in terms of performance delay.

The rest of the paper is organized as follows. The next section formulates the problem of computing the addition and multiplication for the moduli $\{2^n - 3\}$ and $\{2^n + 3\}$. Section III describe the proposed modular arithmetic units (addition and multiplication). Section IV describes the implementations of the proposed structures in ASIC, and evaluates their relative performance against existing units. Section V, concludes this paper with some final remarks.

II. Modular arithmetic

Addition and multiplication are the two basic operations intensively required for digital signal processing. This section analysis the modular arithmetic operations required for addition and multiplication on each channel of the considered moduli sets.

A. Addition

The addition modulo m of the values A and B can be implemented by performing a binary addition of the two operands and executing the reduction when the result is larger or equal then m. If the result is larger or equal then m, the reduction can be performed by subtracting the value m. In a similar fashion to the $\{2^n \pm 1\}$ modular addition [8], the following presents the approach for modulo $\{2^n - 3\}$ and $\{2^n + 3\}$.

978-0-7695-4171-6/10 $26.00 © 2010 IEEE
DOI 10.1109/DSD.2010.77

Modulo $\{2^n-3\}$: Applying the generic approach to this modulo, it is necessary to detect when the addition of A and B is greater or equal to 2^n-3. This is equivalent to detecting if $A+B+3$ is greater or equal than 2^n. Furthermore, when the result is greater or equal than 2^n the modular result is given by $A+B-(2^n-3)=A+B+3-2^n$. Thus, considering two input values, as integers A and B in the range $[0,2^n-3[$, the addition modulo 2^n-3 can be computed by:

$$\langle A+B\rangle_{2^n-3}=\begin{cases} A+B+3 & ,A+B+3\geq 2^n \\ A+B & ,A+B+3<2^n \end{cases} \quad (1)$$

Modulo $\{2^n+3\}$: Identically to modulo $\{2^n-3\}$, one needs to detect when the addition of A and B is greater or equal than 2^n+3.

This is the same as detecting if $A+B+2^{n+1}-(2^n+3)=A+B+2^n-3$ is greater or equal than 2^n+1. Furthermore, when the result is greater or equal to 2^n+1 the modular result is given by $A+B+2^n-3$. These formulations are valid for an integer A and B in the range $[0,2^n+3[$ and is represented by equation (2).

$$\langle A+B\rangle_{2^n+3}=\begin{cases} A+B+2^n-3 & ,A+B+2^n-3\geq 2^{n+1} \\ A+B & ,A+B+2^n-3<2^{n+1} \end{cases} \quad (2)$$

B. Multiplication

To compute the modular multiplication of two intgeres A and B, we can consider the binary product of the input operands followed by a reduction of the product, modulo m.

Considering $P=A\times B$, and $P_{[n-1:0]}$, the n least significantly bits of product, the following equations describes the arithmetic operations involved in the computation of the multiplications modulo $\{2^n\pm 1\}$ and $\{2^n\pm 3\}$.

Modulo $\{2^n-1\}$: Considering P as the binary product of A and B, multiplication modulo $\{2^n-1\}$ can be described by:

$$\begin{aligned}\langle A\times B\rangle_{2^n-1} &= \left\langle 2^n\cdot P_{[2n-1:n]}+2^0\cdot P_{[n-1:0]}\right\rangle_{2^n-1} \\ &= \left\langle 2^n\cdot P_1+2^0\cdot P_0\right\rangle_{2^n-1} \\ &= \langle P_1+P_0\rangle_{2^n-1}\ . \end{aligned} \quad (3)$$

Modulo $\{2^n+1\}$: For this modulo, the modular multiplication can be described by:

$$\begin{aligned}\langle A\times B\rangle_{2^n+1} &= \left\langle 2^{2n}\cdot P_{[2n+1:2n]}+2^n\cdot P_{[2n-1:n]}+\right. \\ &\quad \left.+2^0\cdot P_{[n-1:0]}\right\rangle_{2^n+1} \\ &= \langle P_2-P_1+P_0\rangle_{2^n+1}\ . \end{aligned} \quad (4)$$

Modulo $\{2^n-3\}$: Considering P as the binary product of A and B, multiplication modulo $\{2^n-3\}$ can be described by:

$$\begin{aligned}\langle A\times B\rangle_{2^n-3} &= \left\langle 2^n\cdot P_{[2n-1:n]}+2^0\cdot P_{[n-1:0]}\right\rangle_{2^n-3} \\ &= \left\langle 2^n\cdot P_1+2^0\cdot P_0\right\rangle_{2^n-3} \\ &= \left\langle (2^n-3+3)\cdot P_1+2^0\cdot P_0\right\rangle_{2^n-3} \\ &= \langle 3\cdot P_1+P_0\rangle_{2^n-3}\ . \end{aligned} \quad (5)$$

Modulo $\{2^n+3\}$: The multiplication modulo $\{2^n+3\}$ of A and B, can be calculated by:

$$\begin{aligned}\langle A\times B\rangle_{2^n+3} &= \left\langle 2^{2n}\cdot P_{[2n+1:2n]}+2^n\cdot P_{[2n-1:n]}+\right. \\ &\quad \left.+2^0\cdot P_{[n-1:0]}\right\rangle_{2^n+3} \\ &= \left\langle 2^{2n}\cdot P_2+2^n\cdot P_1+2^0\cdot P_0\right\rangle_{2^n+3} \\ &= \left\langle (-3)^2\cdot P_2-3\cdot P_1+P_0\right\rangle_{2^n+3} \\ &= \langle 9\cdot P_2-3\cdot P_1+P_0\rangle_{2^n+3}\ . \end{aligned} \quad (6)$$

III. Hardware structure

In this section addition and multiplications structures are presented, based on the equations presented in the previous section.

A. Addition

Herein the addition units structures for the moduli $\{2^n\}$ and $\{2^n\pm 1\}$ are summarily described, and several structures for the moduli $\{2^n\pm 3\}$ are proposed and described. Two initially addition units are presented for a generic moduli $\{2^n\pm m\}$, a more straightforward structure with smaller area requirements, designated as *basic*, and the other one with the delay improved, that requires additional area resources [9], designated as *speed*.

Modulo $\{2^n-1\}$ and $\{2^n+1\}$: These two moduli have been extensively studied by several authors [8], [10], [11]. One of the most efficient in terms of area and delay, and widely used, it is based on a modified Slansky addition structure with an end around carry scheme to perform the modulo reduction [8].

For modulo $\{2^n-1\}$ the final result is computed in two steps: first step computes $A+B$; if the result exceeds 2^n the correct output is given by $A+B+1$, otherwise the result is already given by $A+B$ and no further operations need to be performed. The described computation can be efficiently implemented with a modified Slansky addition structure with fast increment [8].

For modulo $\{2^n+1\}$ addition an identical structure can be used. However, the operands have $(n+1)$ bits in order to represent the value 2^n. Note that if the $(n+1)^{th}$ bit is equal to 1 all remaining bit are zero [8]. Therefore, it is necessary to implement a pre-compensation block, for these cases where any of the $(n+1)^{th}$ bit assume the logical value one. This block performs a pre-addition of $A_{[n-1:0]}+B_{[n-1:0]}+comp$, where the compensation value ($comp$) takes into account the $(n+1)^{th}$ bit of the inputs. The compensation takes the values $\left\langle -A_{[n]}-B_{[n]}-2\right\rangle_{2^n+1}$.

Modulo $\{2^n-3\}$: In order to implement a structure to compute the addition of A and B and test if is greater or equal to 2^n-3, as described in (1), two structures can be used, namely *basic* and *speed*. The structure designated by *speed* calculate $A+B$ and $A+B+3$ in parallel minimizing the critical path from two full additions, to one bit Full-Adder (CSA structure) and one full addition. This performance

improvement is achieved with an area cost of one CSA structure of n bits, in comparison with the $basic$ structure.

Modulo $\{2^n + 3\}$**:** For this modulo, the computation of $A + B$ and $(A + B) + 2^n - 3$ is performed in parallel, with a critical path of one Full-Adder and one full addition of $n + 1$ bits. The $speed$ implementation is the best known structure considering the delay metric.

B. Multiplication

Herein, existing multiplication structures are presented and novel structures described for the moduli $\{2^n - 3\}$ and $\{2^n + 3\}$.

Modulo $\{2^n - 1\}$**:** Note that, as seen in (3), this multiplication structure has a straightforward implementation. This structure require a standard 2^{2n} multiplier and a compressor 2:1 [8].

Modulo $\{2^n + 1\}$**:** To perform the computation of (4), it is required to compute $-P_1$:

$$\begin{aligned}\langle -P_1\rangle_{2^n+1} &= \langle 2^n + 1 - P_1\rangle_{2^n+1} \\ &= \langle 2^n - 1 - P_1 + 2\rangle_{2^n+1} \\ &= \left\langle \overline{P_1} + 2\right\rangle_{2^n+1} . \end{aligned} \tag{7}$$

Applying this equation to (4) we obtain:

$$\begin{aligned}\langle P_2 - P_1 + P_0\rangle_{2^n+1} = \ &= \left\langle P_2 + 2 + \overline{P_1} + P_0\right\rangle_{2^n+1} \\ &= \left\langle comp + \overline{P_1} + P_0\right\rangle_{2^n+1} . \end{aligned} \tag{8}$$

The $comp$ value is given by $\left\langle P_{2_{[0]}} + 2\right\rangle_{2^n+1}$. In order, to compute (8) a Carry-Save-Adder (CSA) and a modular full addition are required.

Modulo $\{2^n - 3\}$**:** To accomplish the computation of (5), a modification to the calculation described in (9) must be performed, in order to efficiently compute the $3P_1$ value. This can be performed by adding P_1 and $2 \cdot P_1$. This last value is calculated by left shifting P_1 one position. It is also, necessary to perform modular reduction of $P_{1_{[n-1]}}$ bit, since it has been weighted by 2^n after shifting. This reduction is accomplished with the same method; (9) depicts all modular reduction applied to (5). To compute the final addition a modular compressor 4:1 moduli $\{2^n - 3\}$ is necessary.

$$\begin{aligned}&\langle P_0 + 3 \cdot P_1\rangle_{2^n-3} = \\ &= \langle P_0 + (2 \cdot P_1) + P_1\rangle_{2^n-3} \\ &= \left\langle P_0 + \left(P_{1_{[n-2:0]}}0 + 3 \cdot P_{1_{[n-1]}}\right) + P_1\right\rangle_{2^n-3} \\ &= \left\langle P_0 + \left(P_{1_{[n-2:0]}}0 + 0_{[n-3:0]}P_{1_{[n-1]}}P_{1_{[n-1]}}\right) + P_1\right\rangle_{2^n-3} \\ &= \langle P_0 + (P_{1,1} + P_{1,2}) + P_1\rangle_{2^n-3} \end{aligned} \tag{9}$$

The $0_{[n-3:0]}$ expression represents a vector of zeros with $n - 2$ bits of length.

Modulo $\{2^n + 3\}$**:** In order to calculate (6), $\langle -P_1\rangle_{2^n+3}$ needs to be computed. This computation is given by:

$$\begin{aligned}\langle -P_1\rangle_{2^n+3} &= \langle 2^n + 3 - P_1\rangle_{2^n+3} \\ &= \langle 2^n - 1 - P_1 + 4\rangle_{2^n+3} \\ &= \left\langle \overline{P_1} + 4\right\rangle_{2^n+3} . \end{aligned} \tag{10}$$

By using constant multiplication method used in (9) for modulo $\{2^n - 3\}$, the computation of (6) can be performed as:

$$\begin{aligned}&\langle P_0 - 3 \cdot P_1 + 9 \cdot P_2\rangle_{2^n+3} = \\ &= \left\langle P_0 + 3 \cdot (\overline{P_1} + 4) + 9 \cdot P_2\right\rangle_{2^n+3} = \\ &= \left\langle P_0 + \left(2 \cdot \overline{P_1}\right) + \overline{P_1} + 12 + 9 \cdot P_2\right\rangle_{2^n+3} = \\ &= \left\langle P_0 + \left(\overline{P_1}_{[n-2:0]}0 + 0_{[n-3:0]}\overline{P_1}_{[n-1]}\overline{P_1}_{[n-1]}\right) + \right. \\ &\quad \left. + \overline{P_1} + 12 + 9P_2\right\rangle_{2^n+3} = \\ &= \left\langle P_0 + \left(\overline{P_1}_{[n-2:0]}0\right) + \overline{P_1} + comp\right\rangle_{2^n+3} = \\ &= \left\langle P_0 + P'_{1,2} + P'_1 + comp\right\rangle_{2^n+3} . \end{aligned} \tag{11}$$

Taking into account that P_2 is a single bit vector, it can be added in the compensation factor. This compensation value is a function of P_2 and $\overline{P_1}_{[n-1]}$.

IV. Experimental Results

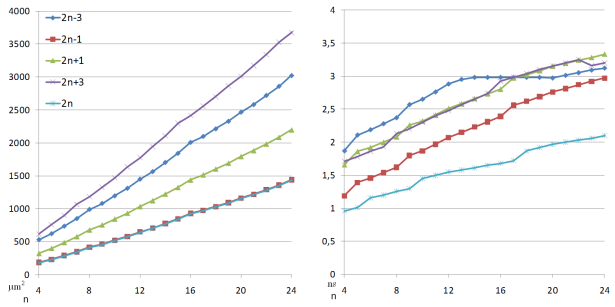
In order to fully analyze the proposed arithmetic units modulo $\{2^n - 3\}$ and $\{2^n + 3\}$ and relatively evaluate them regarding the arithmetic units for the $\{2^n \pm 1\}$ and $\{2^n\}$ moduli channels, all the structures were described in VHSIC Hardware Description Language (VHDL) and mapped to a specific technology. Application Specific Integrated Circuit (ASIC) technology was selected, in particular the UMC 0.13 μm CMOS technology [12], as the target technology. Both synthesis and mapping were performed using Design Vision Version A-2007.12-SP5 from Synopsys.

A. Adders

The first step to evaluate the RNS adders is to compare the obtained results for the considered modulo $\{2^n - 3\}$ and $\{2^n - 3\}$ adder structures, namely the *i)* $basic$, and *ii)* $speed$. As expected from theoretical analysis, the $basic$ structure is the most area efficient realization, at the expense of a significantly higher delay. On the other hand the $speed$ structure is the one with the smallest delay.

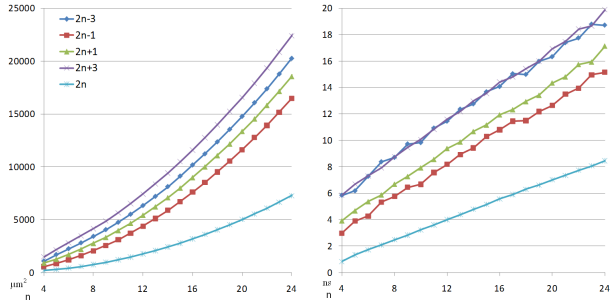
Units for moduli $\{2^n - 1\}$, $\{2^n\}$ and $\{2^n + 1\}$ have also been synthesized, in order to compare circuit area, and delay. These results are depicted in Figure 1 *(a)*, and *(b)*, respectively. Experimental results indicate that moduli $\{2^n\}$ and $\{2^n - 1\}$ have identical area metrics. However, the modulo $\{2^n - 1\}$ addition has a significantly higher delay. When comparing $\{2^n + 1\}$ with $\{2^n\}$ modulo, this has more 62% area and a 65% higher delay, this result is expected due to the additional CSA used for implementing this unit.

The implementation results for the $\{2^n - 3\}$ and $\{2^n + 3\}$ moduli additions, suggest area requirements 42% to 74% higher than the $\{2^n+1\}$ modulo adder and 131% to 182% then the $\{2^n\}$ modulo adder. Furthermore, the critical path results



(a) Area (b) Delay

Fig. 1. Results for addition units on ASIC for moduli 2^n, $2^n \pm 1$ and $2^n \pm 3$



(a) Area (b) Delay

Fig. 2. Results for multiplication units on ASIC for moduli 2^n, $2^n \pm 1$ and $2^n \pm 3$

for the $\{2^n - 3\}$ and $\{2^n + 3\}$ moduli additions suggest delay up to 6% higher than the $\{2^n + 1\}$ modulo adder and 75% to 64% then the $\{2^n\}$ modulo adder.

B. Multipliers

Identically to the addition units, the relative performance of the several moduli multipliers has also been analysed. As depicted in Figure 2 these units require significantly more area and impose an much higher delay, when compared with the adder structures. Three distinct delay patterns can be observed in Figure 2(b): the $\{2^n\}$ channel is 130% faster than $\{2^n \pm 1\}$ channels and 213% faster than $\{2^n \pm 3\}$ channels, as already suggested by the literature [13].

The most important result is obtained when comparing the $\{2^n \pm 1\}$ channels with the $\{2^n \pm 3\}$ channels. This analysis suggests that the $\{2^n \pm 1\}$ modular operation is 25% faster than the $\{2^n \pm 3\}$ channels. Higher area requirements are also imposed for the $\{2^n \pm 3\}$ channels for the same bit length. Note that the implemented $\{2^n \pm 1\}$ moduli multiplication structures are the most straightforward ones. Using more optimized structures proposed in the literature [14], [15], an even more significant difference in the performance is registered between the several moduli channels.

It can be conclude from the obtained results that more balanced moduli sets should be used. These results reinforce the need of moduli channels with different bit lengths, in order to obtain more balanced results between the moduli channels and overall improved performance for the RNS.

V. Conclusions

In this paper, addition and multiplication units for Residue Number Systems (RNS) using moduli $\{2^n - 3\}$ and $\{2^n + 3\}$ channels are proposed. Performance results for proposed modular units for moduli $\{2^n \pm 3\}$ and for moduli $\{2^n \pm 1\}$ and $\{2^n\}$ were obtained, using from an 0.13 μm ASIC technology. The obtained results suggest that the proposed modular addition units require between 42% and 74% more area and impose a critical path up to 6% slower than the $\{2^n + 1\}$ modulo addition. For multiplication, the obtained results suggest that the proposed units require a 16% to 37% more area and impose a delay increase of about 25%, in comparison with the most critical units of the remaining moduli set elements, in particular the $\{2^n+1\}$ channel. This allows us to conclude that the $\{2^n \pm 3\}$ units cause an imbalance in the RNS systems, so more balacend moduli set $\{2^{n-k} - 3, 2^n - 1, 2^n + 1, 2^{n-k} + 3\}$ should be used. In conclusion, this paper can be used by the RNS community to start considering the use of different bit length moduli channels, in order to devise more efficient and better balanced RNS moduli sets.

References

[1] M. Soderstrand, W.Jenkins, G. Jullien, and F. Taylor, Eds., *Residue number system arithmetic: modern applications in digital signal processing*. Piscataway, NJ, USA: IEEE Press, 1986.

[2] P. M. Matutino and L. Sousa, "An rns based specific processor for computing the minimum sad," in *11th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools - DSD2008*, 2008.

[3] F. E. P. Dale Gallaher and P. Srinivasan, "The digit paralell method for fast rns to weighted number system conversion for specific moduli $(2^n - 1, 2^n, 2^n + 1)$," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, 1997.

[4] A. Hariri, K. Navi, and R. Rastegar, "A new high dynamic range moduli set with efficient reverse converter," *Computers and Mathematics with Applications*, vol. 55, no. 4, pp. 660 – 668, 2008.

[5] M.-H. Sheu, S.-H. Lin, C. Chen, and S.-W. Yang, "An efficient vlsi design for a residue to binary converter for general balance moduli $(2^n - 3, 2^n + 1, 2^n - 1, 2^n + 3)$," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 51, no. 3, pp. 152 – 155, march 2004.

[6] L.-S. Didier and P.-Y. Rivaille, "A generalization of a fast rns conversion for a new 4-modulus base," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 56, no. 1, pp. 46 –50, jan. 2009.

[7] P. Ananda Mohan, "Reverse converters for the moduli sets $2^{2N} - 1$, 2^N, $2^{2N} + 1$ and $2^N - 3$, $2^N + 1$, $2^N - 1$, $2^N + 3$," in *SPCOM '04*, 11-14 2004, pp. 188 – 192.

[8] R. Zimmermann, "Efficient vlsi implementation of modulo $(2^n \pm 1)$ addition and multiplication," in *14th IEEE Symposium on Computer Arithmetic*, 1999.

[9] A. Omondi and B. Premkumar, Eds., *Residue Number Systems: Theory and Implementation*. London, UK: Imperial College Press, 2007.

[10] S. B. R.A. Patel, M. Benaissa and N. Powell, "Power-delay-area efficient modulo $2^n + 1$ adder architecture for rns," *Electronics Letters*, 2005.

[11] M. B. R.A. Patel and S. Boussakta, "Efficient new approach for modulo $2^n - 1$ addition in rns," in *IEE Proceedings on Computers and Digital Techniques*, 2006.

[12] Virtual Silicon Technology Inc., "v2.4 esimroute/11^{TM}," High Performance Standard Cell Library (UMC 0.13 μm), Tech. Rep., 2004.

[13] L. S. Ricardo Chaves, "$\{2^n + 1, 2^{n+k}, 2^n - 1\}$: A new rns moduli set extension," in *EUROMICRO Systems on Digital System Design*, 2004.

[14] L. Sousa and R. Chaves, "A universal architecture for designing efficient modulo 2 n+ 1 multipliers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 6, pp. 1166–1178, 2005.

[15] H. Vergos and C. Efstathiou, "Design of efficient modulo 2 n+ 1 multipliers," *Computers & Digital Techniques, IET*, vol. 1, no. 1, pp. 49–57, 2007.