

©2002 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

This copyright notice is taken from the IEEE PSPB Operations Manual, section 8.1.10 entitled "Electronic Information Dissemination". At the time of this notice, this section is posted at

http://www.ieee.org/portal/index.jsp?pageID=corp_level1&path=about/documentation/copyright&file=policies.xml&xsl=generic.xsl

A modulus replication complex-adaptive-filter IP core

Un filtre adaptatif complexe de réplication de module pour un noyau de PI

A. Garg, G.A. Jullien, G.H. McGibney, and J.W. Haslett*

In this paper the modulus replication residue number system (MRRNS) is used to design a high-throughput multirate equalizer for an asymmetrical wireless LAN, where all of the equalization functions are carried out in the base station. The MRRNS mapping technique allows the implementation of parallel and independent channels computing complex arithmetic over finite fields with moduli of 17 and 257. The channels are built as linear systolic arrays using a basic finite-field index calculus and conversion processor; the same processor is also used for the input and output mapping. The processor is defined as a hard intellectual property (IP) block in the target technology, with module generators to build the maps and filter channels. In this paper the authors briefly discuss the mathematical technique used, the basic structure of the processor block and the firm IP generation techniques, and the architectural advantages that the system provides for system-on-chip applications.

Cet article utilise un système de résidus de nombres de réplication de module (MRRNS) pour concevoir un égalisateur à cadences multiples à haute performance pour un réseau sans fil asymétrique dans lequel toutes les fonctions d'égalisation s'exécutent sur la station de base. La technique de mappage MRRNS permet l'implémentation de canaux parallèles indépendants calculant en arithmétique complexe sur des champs finis avec des modules de 17 et 257. Les canaux sont construits comme des matrices systoliques linéaires utilisant un calcul avec index à champ fini et un processeur de conversion; le même processeur sert aussi au mappage d'entrée-sortie. Le processeur est défini comme un block dur de propriété intellectuelle (PI) dans la technologie cible avec des générateurs de modules pour construire les cartes et les canaux de filtrage. Cet article décrit l'approche mathématique employée, la structure de base du block de traitement et les techniques de génération de PI, de même que les avantages que le système procure aux applications de systèmes sur une puce.

Keywords: adaptive FIR filters, computer arithmetic, modulus replication, system-on-chip, IP cores, wireless networks

I. Introduction

With the increasing push towards the creation of systems on chip (SoCs) for real-time applications, there is a demand for high-performance, efficient data-stream processing blocks. These blocks must be suitable for embedding in an SoC platform, and must allow for a wide range of applications. In this paper we specifically target inner-product processors, which are used in a variety of systems, such as our target application of finite-impulse response (FIR) filters for channel equalization. The ability to create a firm intellectual property (IP) core for an adaptive FIR filter, in a way that satisfies a large range of applications and can be integrated into a high-performance system, is the topic of this paper.

The design of an application-specified integrated circuit (ASIC) has different goals than the design of a firm IP core. The implementation of an IP core not only involves interfacing standards, often the topic of discussion within SoC block authoring, but also requires that we evaluate whether the generic solution is competitive for a custom block. To generate such an IP, which is competitive and versatile, careful consideration of the architecture of the core is required. The authors propose the use of modulus replication theory [1]–[6], which allows completely independent computation over parallel channels, introducing a variety of SoC advantages including the ability to purposely skew clocks in order to mitigate system noise; lower connectivity spread allowing easier testing and lower power; and easily implemented fault tolerance [7].

In Section II, a gigabit wireless local area network (LAN) is intro-

duced, representing our target application for this general filter. This is the application that requires the general FIR filter core described in this paper. An overall system perspective is reviewed with a brief introduction to the proposed core of the adaptive equalization filter. Section III provides a brief introduction to the quadratic residue number system (QRNS) and describes its advantages in implementing complex digital filters. Section IV provides an introduction to the modulus replication residue number system (MRRNS) and describes its advantages for creating both real and complex filters. Section V shows the architectural implementation of a firm IP and explains how the numerical representations allow for the core to be completely configurable and reusable. Section VI displays previous and current performance results for modulo digital filters, while discussing the advantages they reveal for creating a competitive-FIR-filter firm IP core.

II. A gigabit wireless LAN

TRLabs in Calgary, Alberta, has created a next-generation wireless LAN that is targeted to transmit over a gigabit of data per second. When transmitting at these high data rates through a mobile radio channel, intersymbol interference (ISI) caused by frequency-selective fading [8] is of concern. The adaptive equalization filters, particularly the adaptive-filter core, used to correct these distortions become increasingly difficult to implement due to the high data rate. The pipelined implementation must have a high clock frequency to match the symbol rate, and the number of filter taps is significantly increased in order to suppress the added noise in a high-bandwidth system [8]. This gigabit wireless LAN requires a complex adaptive high-speed filter core with a built-in decimator. It also requires a 400 MHz clock frequency, programmable coefficients, complex arithmetic processing,

*A. Garg, G.A. Jullien, and J.W. Haslett are with ATIPS Laboratory, University of Calgary, Calgary, Alberta. G.H. McGibney is with TRILabs, Calgary, Alberta.

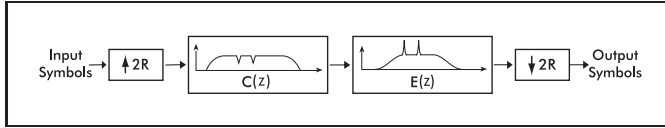


Figure 1: Post-equalization operation.

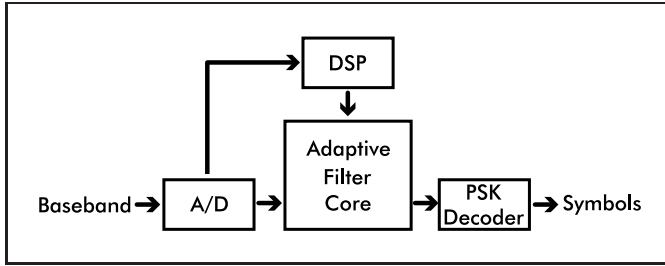


Figure 2: Block-level adaptive equalization filter.

128 filter taps and at least 22-bit internal accuracy. These conditions mean that the core must perform 409.6 billion arithmetic operations per second. Given that the system is an asymmetric wireless LAN, where the entire signal processing is conducted in the base station while the terminals are simple RF transceivers, the filter is used as a post equalizer as shown in Fig. 1. Here $E(z)$ is the equalization-filter transfer function and $C(z)$ is the channel transfer function, which includes the RF transmitter and receiver as well as the wireless channel. Also, $\uparrow 2R$ is a multirate expander, and $\downarrow 2R$ is a multirate decimator that is used in conjunction with other methods [8] to equalize the channel.

The adaptive equalization filter is composed of a number of components, which interface together as shown in Fig. 2. The baseband signal is an analogue signal that contains the encoded and distorted data transmitted from the terminal. The analogue-to-digital converter (A/D) converts the baseband signal and feeds it into the adaptive-filter core and the digital signal processor (DSP). The DSP uses the information from the A/D, and computes the coefficients for the filter core and programs the values during the training cycle. The adaptive core filters the data in order to counter the distortion elements within the channel. The phase-shift keying (PSK) decoder uses the filtered data to recover the encoded symbols.

Traditionally, this system would be created with several components connected at the board level; however, the advent of SoC densities allows the implementation of the entire system on a single die. This provides numerous advantages, including significant power reduction and increase in throughput rate and reliability. The creation of an SoC alleviates many of the board-level problems a high-performance system would face. All the components of the SoC solution will be commercial IP cores with a custom FIR-filter IP core. From previous work [1]–[2], [6]–[7], [9], the use of modulo arithmetic has been shown to produce an efficient hardware implementation of the filter core.

Modulo arithmetic has been used and extensively researched for signal processing applications for the last three and a half decades [10]–[11]. The most popular mapping, the residue number system (RNS), provides carry-free computations over small-dynamic-range orthogonal channels. A complete background on RNS can be found in [10]. Using a combination of modulus replication RNS (MRRNS) and quadratic RNS (QRNS) [1]–[2], [6], a complex digital FIR filter can be built using identical, replicated channels. It has been shown that this approach provides substantial power savings, reduced clock requirements, reduced current inrush, and the ability to easily add fault tolerance and fault correction, while being able to compute independent complex calculations using a simplified multiply-accumulate (MAC) cell [1]–[2], [6]–[7], [9]. This project is being targeted for the TSMC 0.18 μm CMOS process, provided by the Canadian Microelectronics Corporation (CMC).

III. Quadratic residue number system

QRNS [4] is a number system designed to handle complex data, so that the real and imaginary channels data are processed independently [12]–[13]. This method maps the real and imaginary data to normal and conjugate channels that compute over finite fields.

For moduli, m_i , of the form $4K - 1$, there exists a solution for the monic quadratic $x^2 + 1 = 0$ in a modulo finite quadratic ring, $QR(m_i) = \{S : \oplus, \otimes\}$, where \oplus and \otimes are addition and multiplication modulo m_i . In other words, the solutions of the monic quadratic, j_i and $-j_i$, are elements of the ring, $QR(m_i)$. Although an extension field cannot be built based on these solutions, an extension ring can be generated. The extension element can be written as $AQ_i = (A_i^\circ, A_i^*)$, where $A_i^\circ = r_i \oplus j_i \otimes i_i$ (normal) and $A_i^* = r_i \oplus (-j_i) \otimes i_i$ (conjugate); r_i and i_i are the real and imaginary values respectively; and $r_i, i_i, A_i^\circ, A_i^* \in GF(m_i)$.

The two operations of addition and multiplication, over the quadratic ring, are computed as follows:

Addition:

$$AQ_i \oplus BQ_i = (A_i^\circ \oplus B_i^\circ, A_i^* \oplus B_i^*). \quad (1)$$

Multiplication:

$$AQ_i \otimes BQ_i = (A_i^\circ \otimes B_i^\circ, A_i^* \otimes B_i^*). \quad (2)$$

Here the real and imaginary results of the computation can be recovered from the normal and conjugate parts of the result, C_i° and C_i^* respectively, as

$$\begin{bmatrix} Y_{iR} \\ Y_{iI} \end{bmatrix} = \begin{bmatrix} 1 & j_i \\ 1 & -j_i \end{bmatrix}^{-1} \begin{bmatrix} C_i^\circ \\ C_i^* \end{bmatrix}. \quad (3)$$

This forms a commutative ring with identity. It should be noted that the ring is isomorphic to a finite ring of Gaussian integers, which will be denoted as $R(m_i)$, and that both arithmetic operations involve only two base field operations (\oplus, \otimes).

The advantage of the QRNS mapping is that the normal and conjugate computations occur without any interaction between the two independent channels. This allows both multiplication and addition to occur without any crossover connections, unlike the case with a conventional complex multiplication, which requires interaction between the real and imaginary parts for every multiplication. The lack of interaction also provides a savings on the mathematical operational costs within the MAC. The multiplication for a complex number normally requires four real multiplications and two real additions; however, with QRNS, multiplication is conducted independently over the normal and conjugate channels, requiring only two finite ring multiplications. This difference provides hardware and power savings. A decrease in the design cost of the tap is extremely important since it is extensively replicated.

IV. Modulus replication residue number system

MRRNS representation [2] also provides some useful computational advantages, which are briefly introduced here. If a binary representation of an integer is given by $s = \sum_{i=0}^{\beta-1} s_i 2^i$, where $s_i \in (0, 1)$, the binary representation can be rewritten using a polynomial representation of the following form:

$$s = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \cdots \sum_{i_n=0}^{d_n} s_{i_1 i_2 \cdots i_n} 2^{(i_1 \beta_1 + i_2 \beta_2 + \cdots + i_n \beta_n)}, \quad (4)$$

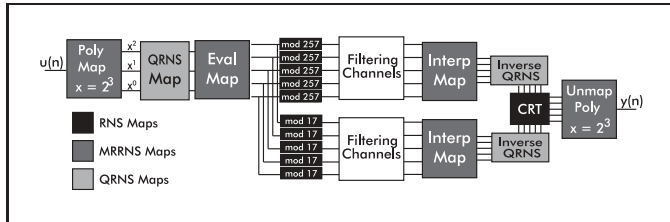


Figure 3: MRRNS and QRNS filter structure.

where $\beta = \beta_0 > \beta_1 > \dots > \beta_n$ are integers. The number is thus represented as a series of polynomials with known indeterminates representing weights of powers of 2. In this case we are representing the binary number in hexadecimal and octal notation. For example, the number 120 with an 8-bit binary representation of 01111000 can be decomposed into a polynomial representation of $1x^2 + 7x + 0$, where $x = 8$.

The polynomial representation is now mapped to a finite polynomial ring (modulo M), where $M = \prod m_i$, and then to a direct product ring, $Z_M \times Z_M \times \dots \times Z_M$, using an evaluation map. The evaluation map is the computation of the polynomial at $n + 1$ arbitrary values for each indeterminate, where n is the order of the resulting polynomial from the series of mathematical operations within the system. This operation is performed by multiplying the vector of polynomial coefficients with the Vandermonde matrix of all the possible roots of the mapping ideal [5]. The ideal should have a greater degree than any resulting polynomial after computations so that no actual reduction takes place when inverting the direct product map. The independent calculations can be performed over each of the copies of Z_M . Conditions for reversing the mapping procedure are discussed in [5] and [14]. An important restriction is that the resulting finite polynomial coefficients do not exceed the modulus, M .

V. Firm IP architecture

An IP core must be configurable and reusable in order to be considered a firm IP. This type of IP core provides an attractive option, since it allows the end user to change the design specifications for specific applications. Also, the reusability of the architecture allows the same core to be considered for a variety of other applications. The numerical representation allows for architectural advantages for the creation of a firm IP.

A. Reusable architecture

Using the numerical representations in Sections III and IV, the digital filter core can be implemented using a regular structure with optimized standard cells for the taps that are small and extensively optimized. The block-level view of the FIR filter architecture is shown in Fig. 3. There is a clear partition between each of the mapping blocks. The input, $u(n)$, is first represented as a polynomial in the *Poly Map* with $x = 2^3$, after which the *QRNS Map* converts the real and imaginary numbers to normal and conjugate representation. The *Eval Map* evaluates the polynomial, for both the normal and conjugate channels, and creates a direct product ring where the modulus computations are computed independently. The *Interp Map* recovers the coefficients of the polynomial from the direct product ring, and *Inverse QRNS* converts the normal and conjugate values to the real and imaginary representations. The *CRT* (Chinese Remainder Theorem) block converts the RNS representation (mod 257 and 17 channels) into polynomial coefficients. The final step is evaluating the polynomial at the selected indeterminate, $x = 2^3$. The input and output maps function independently and can be removed without affecting the system. For example, if a real filter is to be designed, the QRNS maps can be removed without affecting the channel structure or the MRRNS map. This modular structure

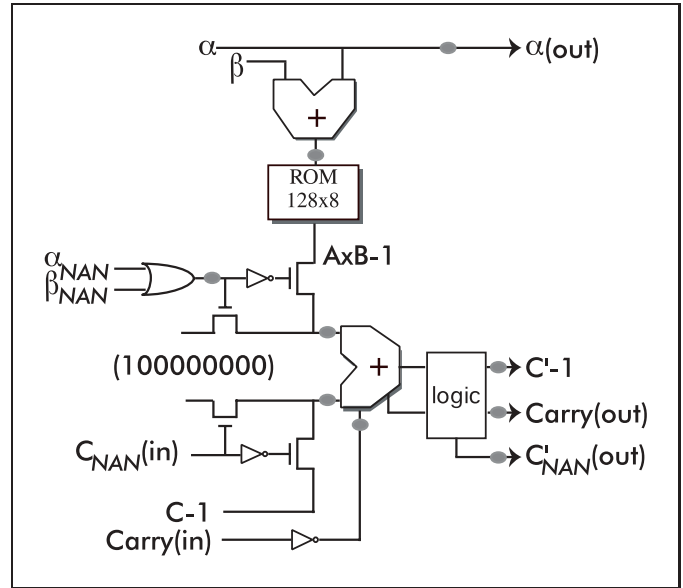


Figure 4: Fermat ALU.

allows for rapid redesign and reuse for numerous filter applications.

In order to optimize the taps, which perform the function $A \times B + C$, the moduli for the above system are chosen based on their mathematical properties. Two such moduli are 17 and 257, which are Fermat primes of the form $Ft = 2^{2^t} + 1$, where t is an integer. Previous work on using Fermat fields, $GF(Ft)$, has demonstrated that an efficient tap structure can be created; this has been referred to as a half index domain MAC, or a Fermat ALU, and can exploit the properties of $GF(Ft)$ using an index calculus for the multiplier and diminished-1's representation [15] for the modulo accumulator. A complete discussion of the MAC architecture can be found in [2] and [14]; a mod 257 MAC is shown in Fig. 4.

Since index calculus is being used within the Fermat ALU, the index addition of α and β , where α and β are the index forms of A and B , requires a binary adder (8 bits for 257 and 4 bits for 17). This, in conjunction with the ROM block, which maps the result of the index addition into a diminished-1's representation, serves as the generic modulo multiplier. The diminished-1's accumulator also uses a binary adder with zero mapped to the "overflow" value. Signals of type *NAN* are used to identify whether the value on the corresponding bus is a valid number or whether the number should be zero. The code for zero, in diminished-1's representation, is input to the accumulator whenever a zero value (*NAN*) is detected. Thus, we can handle the "overflow" case with a small amount of extra logic. These tap structures are cascaded and repeated independently throughout the entire set of channels without interleaved interconnect between the channels. Fig. 4 shows the pipeline latches (ovals) that are used to ensure a clock rate of 400 MHz. Our simulations show a worst-case critical path of 2.18 ns for the accumulating adder stage. This provides a greater than 10% safety margin.

Due to the nature of the input and output maps, which are a network of multiply and accumulates, the optimized taps can also be used to create the mapping stages. For example, the architecture of the evaluation and interpolation maps for the MRRNS architecture reuses the optimized tap structure, in conjunction with a series of delays, to perform the matrix multiplications. The architecture of the evaluation map is shown in Fig. 5, where r are the roots of the ideal, n is the order of the input polynomial, and $2n$ is the expected order of the output polynomial. The index maps (I) to the Fermat ALUs (FA) are delayed into the cells depending on the delay stages within the FA block. The delay stages synchronize the FAs, so that they perform the matrix multiplication.

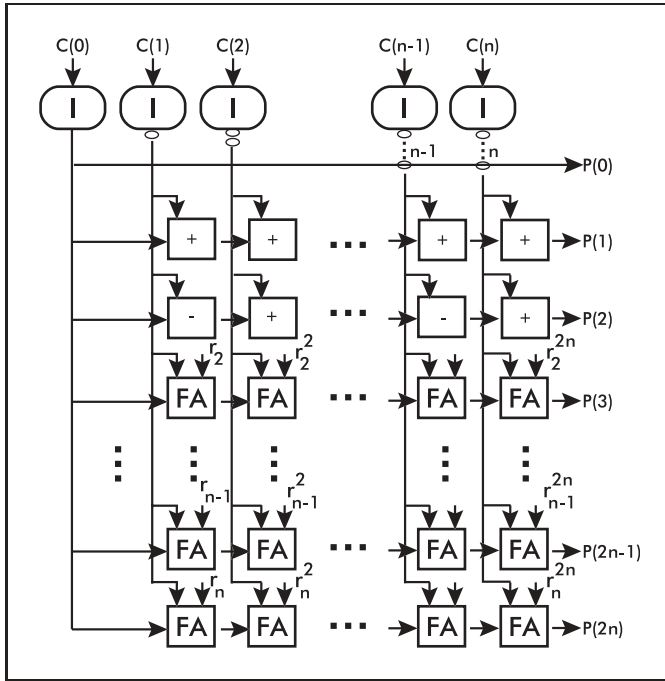


Figure 5: Evaluation MAP.

B. Configurable architecture

The ability of the user to configure the IP core is extremely important in digital filters since they vary greatly from application to application. However, the goal is to create a generic solution that is still competitive with a customized block. For an MRRNS architecture, varying the evaluation (interpolation) map changes the parameters of the core. Since these maps are not part of the tap structure, their cost is spread over the entire filter core. Thus, for a desired set of filter specifications, a combination of input and output maps can be generated to easily create the system for a required number of taps. When generating an MRRNS processor, the only parameter to determine is the indeterminate, x , which is a function of the input bits and number of taps. Determining the indeterminate is an iterative process in which one is trying to solve the following equation [14]:

$$\left\lceil \frac{B}{\beta} \right\rceil (2^{\beta-1}) N \leq M, \quad (5)$$

given that B is the number of input bits (excluding the sign bit), β is the integer that determines the indeterminate such that $x = 2^\beta$, N is the number of taps, and $M = \prod m_i$. The values of the indeterminate and moduli are varied until the system has reduced its number of channels.

The design of the Vandermonde (evaluation) matrix is trivial since the roots of the ideal are arbitrary and optimization of these blocks is not critical. From a parameterization point of view, the evaluation and interpolation maps are easily scalable and are also functions of the same parameters. Let us consider the evaluation matrix shown in (6):

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & -1 & \cdots & (-1)^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & r_i & \cdots & (r_i)^n \\ 1 & -r_i & \cdots & (-r_i)^n \end{bmatrix}, \quad (6)$$

where n is the number of channels and r_i are the nonzero unique roots of the ideal given by

$$r_i = \left\{ 1, -1, \dots, -\left(\frac{B-1}{\beta} - 1\right), \frac{B-1}{\beta} - 1 \right\}. \quad (7)$$

Table 1
Power comparison of MAC cells

| MAC cells | Power (mW/100 MHz) | |
|------------------------------|--------------------|--------------------|
| | 0.5 μm | 0.35 μm |
| Fermat IPSP(257 \times 17) | 38.5 | 32.0 |
| Fermat IPSP(257) | 22.85 | 18.7 |
| Booth algorithm MAC | 41 | 34 |

Table 2
System-noise comparison between a single buffered clock and a skewed clocking scheme

| Clock frequency | Single buffered clock | | Skewed clocks | |
|---------------------|-----------------------|---------|---------------|---------|
| | 125 MHz | 300 MHz | 125 MHz | 300 MHz |
| Max. spike (mA) | 64.4 | 93.8 | 23.4 | 32.8 |
| Max. di/dt (A/ns) | 16.3 | 36.6 | 2.2 | 11.9 |
| Power (mW) | 19.5 | 35.8 | 23.5 | 57.3 |

Given that the indeterminate has been computed and the input number of bits is specified, an evaluation and corresponding interpolation matrix can be generated. Thus, the architecture allows us to create a completely generic solution, for complex and real filters, without the need for a generic MAC cell.

VI. Performance analysis

The next two sections illustrate the performance advantages provided by the use of this architecture over conventional cores.

A. Power savings

Previous work on MRRNS systems has demonstrated considerable power savings within the filter core. Results extracted from previous publications [1]–[3], [5], [14] provide power comparisons for equivalent dynamic ranges. Table 1 shows the comparison of a real 10×10 Booth multiply, 24-bit accumulate MAC cell to a five-channel Fermat ALU system referred to as the Fermat inner-product step processor (Fermat IPSP), in two mature CMOS technologies, with computations over either $GF(257)$ or $G(17 \times 257)$. The table demonstrates that there are power savings to be obtained even with the larger dynamic range calculations. Using $GF(257)$, the power savings are considerable.

Extrapolating the results from a real MAC cell to a complex MAC cell, we note that there is a large increase in power consumption for the Booth algorithm MAC versus the Fermat IPSP. A complex MAC will normally perform four multiplications and four additions for a multiply-accumulate. In terms of power dissipation, this is approximately equivalent to four times the power required for a real MAC cell. For an MRRNS and QRNS system, the number of channels is doubled (from Table 1, this would be an increase from five to 10 channels) for the independent normal and conjugate channels, a doubling of the power dissipation. Since many of our complex-filter MRRNS designs will require computing over $\bar{R}(257 \times 17)$, the Fermat IPSP yields an estimated power consumption of 64 mW/100 MHz per tap compared to 136 mW/100 MHz per tap required by the Booth-encoded MAC. This represents a power savings of over 50% for the 0.35 μm technology. We expect similar savings with the use of the targeted 0.18 μm CMOS technology.

B. System noise

The ability to clock the MRRNS filter channels independently allows the skewing of the clock edges between channels. This provides a re-

duction in overall system noise (i.e., current spikes) as recent results have shown [9]. Extracted results, presented in Table 2, show that there is improvement by a factor of 3 in overall switching noise. This is particularly attractive since the density of SoC chips is expected to exacerbate the already growing problem of system noise. Skewing the clocks across the independent channels causes the current inrush to be spread across time instead of requiring synchronized clock edges for the entire system. Timing issues that typically arise with these self-timed channels cause little problem since they need only be synchronized before and after the large filter channels. With the option to skew the internal clocks, the filter core will produce less switching noise, thus decreasing its negative effect on the surrounding cores.

VII. Conclusion

This paper has discussed the design of an adaptive FIR filter for a gigabit wireless LAN. The filter uses modulus replication theory to allow independent processor channel implementation of the complex pipelined processor. The processor is targeted for a $0.18\ \mu\text{m}$ CMOS technology using SoC design techniques. The SoC concept is a perfect fit for the entire system, since it encompasses all the major components of an embedded system, including the interfacing of the peripherals, microprocessor, on-chip memory and IP cores. The architecture of the IP core was chosen to allow the core to be reusable and portable while maintaining performance and system specifications. Not only does it meet the need for a firm IP block, but the architectural scheme also provides advantages specifically useful for creation of high-density SoC chips. The creation of this IP core for practical use in a next-generation system also allows us to explore the potential of SoC and our transition toward this evolving design paradigm.

VIII. Acknowledgements

This work has been funded by iCORE, the Micronet Network of Centres of Excellence, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and TRILabs, Calgary. The authors also acknowledge support for the ATIPS laboratory design tools and workstations from the Canadian Microelectronics Corporation. We are also indebted to the work of Mr. Ian Steiner, an intern student in the ATIPS Laboratory, who performed many of the simulations associated with the MRRNS filter structure. The authors also acknowledge the many

comments and suggestions from the anonymous reviewers of the original manuscript.

References

- [1] N. Wigley and G.A. Jullien, "On modulus replication for residue arithmetic computations of complex inner products," *IEEE Trans. Comput.* vol. 39, no. 8, Aug. 1990, pp. 1065–1076.
- [2] M. Shahkarami, G.A. Jullien, W.C. Miller, and R. Muscedere, "General purpose FIR filter arrays using optimized redundancy over direct product polynomial rings," in *Proc. Asilomar Conf. Circuits, Systems, and Computers*, Monterey, Calif., 1998, pp. 1209–1213.
- [3] G.A. Jullien, S. Bizzan, and N.M. Wigley, "Using redundant finite rings for fault tolerant signal processors," *SPIE, Advanced Signal Processing: Algorithms, Architectures and Implementations V*, 1994, pp. 2296–2275.
- [4] S. Bizzan, *Architecture and Implementations for Polynomial Ring Engine Over Small Residue Rings*, Ph.D. dissertation, Dept. of Electrical Engineering, University of Windsor, Windsor, Ont., 1997.
- [5] G.A. Jullien, W. Luo, and N.M. Wigley, "High throughput VLSI DSP using replicated finite rings," *J. VLSI Signal Processing*, vol. 14, 1996, pp. 207–220.
- [6] N.M. Wigley, G.A. Jullien, and D. Reaume, "Large dynamic range computations over small finite rings," *IEEE Trans. Comput.*, vol. 43, no. 43, 1994, pp. 78–86.
- [7] L. Imbert and G.A. Jullien, "Fault-tolerant computation of large inner products," *IEE Electron. Lett.*, vol. 37, Apr. 2001, pp. 551–552.
- [8] G.H. McGibney, *Wireless Networking with Simple Terminals*, Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Calgary, Calgary, Alta., 2000.
- [9] Daniel Gonzalez et al., "A new methodology for efficient synchronization of RNS-based VLSI systems," presented at 12th Int. Workshop on Power And Timing Modeling, Optimization and Simulation, Sept. 2002.
- [10] N.S. Szabo and R.I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*, McGraw-Hill Publishing Co., 1967.
- [11] M.A. Soderstrand, W.K. Jenkins, G.A. Jullien, and F.J. Taylor, *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, New York: IEEE Press, 1986.
- [12] W.K. Jenkins and J.V. Krogmier, "The design of dual-complex signal processors based on quadratic modular number codes," *IEEE Trans. Circ. Syst.*, vol. 34, Apr. 1987.
- [13] S.H. Leung, "Application of residue number systems to complex digital filters," in *Proc. Fifteenth Asilomar Conf. on Circuits, Systems, and Computers*, Monterey, Calif., 1981, pp. 70–74.
- [14] M. Shahkarami, *Exploiting Redundancy in Modulus Replication Inner Product Processors*, Ph.D. dissertation, Dept. of Electrical Engineering, University of Windsor, Windsor, Ont., 1999.
- [15] L.M. Leibowitz, "A simplified binary arithmetic for the Fermat number transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 24, no. 5, Oct. 1976.
- [16] N.M. Wigley, G.A. Jullien, and D. Reaume, "Large dynamic range computations over small finite rings," VLSI Research Group, University of Windsor, Windsor, Ont., Tech. Rep., 1994.