DESIGN OF RNS-BASED DISTRIBUTED ARITHMETIC DWT FILTERBANKS

J. Ramírez [1], A. García [2], U. Meyer Bäse [3], F. Taylor [4], P. G. Fernández [5] and A. Lloris [1]
[1] *University of Granada* [2] *Universidad Autónoma de Madrid* [3] *Darmstadt University of Technology*
[4] *University of Florida* [5] *University of Jaén*

ABSTRACT

The need for both speed and increased precision in modern digital signal processing (DSP) applications represents a serious implementation obstacle. This paper explores the arithmetic benefits provided by the residue number system (RNS) for the design of such systems. Concretely, the fusion of the RNS with the popular distributed arithmetic (DA) is considered for the implementation of a discrete wavelet transform (DWT) filter bank. An exhaustive comparison of the advantages of RNS-DA over the traditional two's complement design, 2C-DA, is carried out for field-programmable logic (FPL), and cell-based ASIC technologies. The results show that the reported RNS-DA methodology, compared to a traditional 2C-DA design, enjoys a significant performance advantage that increases with precision.

1. INTRODUCTION

Digital signal processing (DSP) is arithmetic-intensive. DSP facilitating technologies include general-purpose DSP μps, application-specific integrated circuits (ASIC), application specific standard products (ASSP), and field programmable logic (FPL) devices that include field programmable gate arrays (FPGA). Within this mix, ASIC are becoming the dominant technology compared to DSP μps. The FPL ASIC market is expected to expand at a rate of 20% *per annum* rate, with DSP applications leading the way. While FPL houses champion their technology as a provider of system-on-a-chip (SOC) DSP solutions, engineers have historically viewed FPLs as a prototyping technology. In order for FPL to begin to compete in areas currently controlled by low-end standard cell, a means must be found to more efficiently implement DSP objects. A review of FPL vendor supplied application notes, establishes that FPGAs have intrinsically weak arithmetic capabilities. A general-purpose $n\times n$-bit multiplier or multiply-accumulate (MAC) unit, for example, is inferior to a well designed ASIC ALU in both speed and area [1]. In addition, FPL deficiencies increase geometrically with precision (wordlength). It is the FPGA's arithmetic limitations that have caused solution developers to consider alternative structures. The most popular technique, found in common practice, is called distributed arithmetic [2-5], or DA. The DA method is routinely used to implement linear DSP algorithms with fixed known *a priori* coefficients. The DA technique reduces an algorithm to a set of sequential lookup table (LUT) calls, and two's-complement (2C) shift-adds. A survey of the contemporary FPL DA art indicates that most solutions are of low-order and low-precision. An alternative design paradigm is advocated in this paper that is based on fusion of DA and the residue number system or RNS [6, 7]. The RNS advantage is gained by reducing arithmetic to a set of concurrent operations that reside in small wordlength non-communicating channels. This attribute makes the RNS potentially attractive for implementing DSP objects with FPL devices and standard cell CMOS technologies. The advantages of RNS-DA over 2C-DA were extensively studied through the synthesis of high-level definition models. The impact of the synthesis process on 2C-DA and RNS-DA architectures was considered under a study case of a discrete wavelet transform (DWT) filter bank implementation and will build upon previous RNS-DA [7-9] and RNS-FPL design studies [10].

2. BACKGROUND

Computer arithmeticians have long held that the RNS offers the best MAC speed-area advantage [6]. In the RNS, numbers are represented in terms of a relatively prime basis set (moduli set) $P=\{m_1, \ldots m_L\}$. Any number $X\in Z_M=\{0 ,..., M-1\}$, where $M=\prod m_i$, has a unique RNS representation $X\leftrightarrow\{X_1,...,X_L\}$, where $X_i= X \bmod(m_i)$. RNS arithmetic is defined by pair-wise modular operations:

$$\begin{aligned} Z = X \pm Y &\leftrightarrow \left\{ \langle X_1 \pm Y_1 \rangle_1, \ldots, \langle X_L \pm Y_L \rangle_L \right\} \\ Z = X \times Y &\leftrightarrow \left\{ \langle X_1 \times Y_1 \rangle_1, \ldots, \langle X_L \times Y_L \rangle_L \right\} \end{aligned} \quad (1)$$

where $<Q>_j$ denotes $Q \bmod(m_j)$. The individual modular arithmetic operations are typically performed as LUT calls to small memories. The RNS differs from traditional weighted numbering systems in that arithmetic is *carry-free* and can operate at a constant speed over a wide range of wordwidths. Like the 2C system, the RNS arithmetic is exact as long as the final result is bounded within the system's dynamic range Z_M. Mapping from the RNS back to the integer domain is defined by the Chinese Remainder Theorem (CRT) [6].

3. DISCRETE WAVELET TRANSFORM

Wavelets [11] are gaining in importance, especially for image coding applications. The discrete wavelet transform decomposes a signal at increasing resolution levels. An attractive feature of the wavelet series expansion is that the underlying multiresolution structure leads to an efficient discrete-time algorithm based on a filter bank implementation. An N^{th}-order 1-D DWT decomposition of a sequence x_n is defined by:

$$\begin{aligned} a_n^{(i)} &= \sum_{k=0}^{N-1} g_k a_{2n-k}^{(i-1)} & i &= 1, 2, \ldots, J \\ d_n^{(i)} &= \sum_{k=0}^{N-1} h_k a_{2n-k}^{(i-1)} & a_n^{(0)} &\equiv x_n \end{aligned} \quad (2)$$

where $a_n^{(i)}$ and $d_n^{(i)}$ are level-i approximation and detail sequences, respectively, and g_k and h_k (k= 0, 1, …, N-1) correspond to the low-pass and high-pass analysis filter coefficients. The signal x_n can be perfectly recovered through its multiresolution decomposition by iteration:

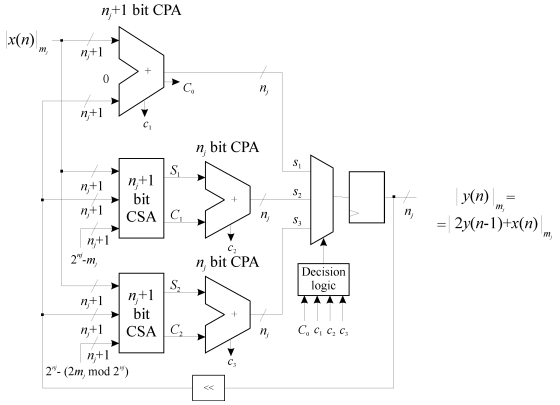

Fig. 1. RNS-DA scaled accumulator design.

$$\hat{a}_m^{(i-1)} = \begin{cases} \sum_{k=0}^{N/2-1} \bar{g}_{2k} \hat{a}_{\frac{m}{2}-k}^{(i)} + \sum_{k=0}^{N/2-1} \bar{h}_{2k} \hat{d}_{\frac{m}{2}-k}^{(i)} & m \text{ even} \\ \sum_{k=0}^{N/2-1} \bar{g}_{2k+1} \hat{a}_{\frac{m-1}{2}-k}^{(i)} + \sum_{k=0}^{N/2-1} \bar{h}_{2k+1} \hat{d}_{\frac{m-1}{2}-k}^{(i)} & m \text{ odd} \end{cases} \quad (3)$$

where $\bar{g}_k$ and $\bar{h}_k$ represent low-pass and high-pass synthesis filter coefficients.

4. 2C-DA DWT FILTER BANK

Conventional 2C-DA has been successfully applied to the implementation of FIR filters and other linear discrete transforms. DA designs replace general multiplication with scaling operations implemented using LUT calls. DA architectures are bit-serial and are often reported to be faster than MAC-centric designs. In addition, compared to programmable MAC solutions, DA designs often have a lower roundoff error budget [12]. A 2C DA design assumes that the input to a DSP-object (*viz*; FIR filter) is the B_i-bit word:

$$a_n^{(i-1)} = -2^{B_i-1} a_{n,B_i-1}^{(i-1)} + \sum_{l=0}^{B_i-2} 2^l a_{n,l}^{(i-1)} \quad (4)$$

where $a_{n,l}^{(i-1)}$ is the l-th bit of the input sample $a_n^{(i-1)}$. Substituting Equation (4) in Equation (2) and interchanging the order of summation leads to:

$$\begin{aligned} a_n^{(i)} &= -2^{B_i-1} \Phi_g(B_i-1) + \sum_{l=0}^{B_i-2} 2^l \Phi_g(l) \\ d_n^{(i)} &= -2^{B_i-1} \Phi_h(B_i-1) + \sum_{l=0}^{B_i-2} 2^l \Phi_h(l) \end{aligned} \quad (5)$$

where the DA LUT functions $\Phi_g(l)$ and $\Phi_h(l)$ are given by:

$$\Phi_g(l) = \sum_{k=0}^{N-1} g_k a_{2n-k,l}^{(i-1)} \qquad \Phi_h(l) = \sum_{k=0}^{N-1} h_k a_{2n-k,l}^{(i-1)} \quad (6)$$

The computation of the i^{th}-octave-filters is carried out using two $2^N \times W$ LUTs representing the functions $\Phi_g(l)$ and $\Phi_h(l)$, that are addressed by the N-bit vector $\{a_{2n,l}^{(i-1)}, a_{2n-1,l}^{(i-1)}, \ldots, a_{2n-N+1,l}^{(i-1)}\}$, $W \le b+\lceil \log_2(N) \rceil$, where b represents the filter coefficient precision. Observe that computing Equation (5) requires repeated calls to the tables $\Phi_g(l)$ and $\Phi_h(l)$, followed by a shift-add (scaled accumulation).

The DA LUT size increase exponentially with the filter length. To palliate this drawback the polyphase decomposition can be considered leading to a reduction in the DA LUT address space. Defining the polyphase filters to be $g_0(k)= g_{2k}$, $h_0(k)= h_{2k}$, $g_1(k)= g_{2k+1}$, and $h_1(k)= h_{2k+1}$ (k= 0, 1, …, N/2-1), the resulting DA architecture consists of four independent filters operating on even- and odd-indexed input samples:

$$\begin{aligned} a_n^{(i)} &= \sum_{k=0}^{N/2-1} g_0(k) a_{2n-2k}^{(i-1)} + \sum_{k=0}^{N/2-1} g_1(k) a_{2n-2k-1}^{(i-1)} \\ d_n^{(i)} &= \sum_{k=0}^{N/2-1} h_0(k) a_{2n-2k}^{(i-1)} + \sum_{k=0}^{N/2-1} h_1(k) a_{2n-2k-1}^{(i-1)} \end{aligned} \quad (7)$$

In this way, the defining DA relationships are:

$$\begin{aligned} \Phi_{g_0}(l) &= \sum_{k=0}^{N/2-1} g_0(k) a_{2n-2k,l}^{(i-1)} \quad \Phi_{g_1}(l) = \sum_{k=0}^{N/2-1} g_1(k) a_{2n-2k-1,l}^{(i-1)} \\ \Phi_{h_0}(l) &= \sum_{k=0}^{N/2-1} h_0(k) a_{2n-2k,l}^{(i-1)} \quad \Phi_{h_1}(l) = \sum_{k=0}^{N/2-1} h_1(k) a_{2n-2k-1,l}^{(i-1)} \end{aligned} \quad (8)$$

and the architecture consists of four $2^{N/2} \times W'$ LUTs, where $W' \le b' + \lceil \log_2(N/2) \rceil$, and b' represents the filter bank coefficient precision.

5. RNS-DA DWT FILTER BANK

The RNS represents a potentially efficient means of implementing a DA-based DSP solution [7] within small wordlength channels. The RNS, developed in Section 2, is defined in terms of a moduli set $P=\{m_1, m_2, \ldots, m_L\}$ where typically $m_i \le 2^8$. Therefore the solution can be defined to reside within 8-bit channels. An RNS solution will, however, require a modulo m_j scaled accumulator be developed to realize the mapping $|y(n)|_{m_j} = |2y(n-1)+x(n)|_{m_j}$ according to the following selection rules:

$$|y(n)|_{m_j} = \begin{cases} 2|y(n-1)|_{m_j} + |x(n)|_{m_j} & A < m_j \\ 2|y(n-1)|_{m_j} + |x(n)|_{m_j} - m_j & m_j \le A < 2m_j \\ 2|y(n-1)|_{m_j} + |x(n)|_{m_j} - 2m_j & 2m_j \le A < 3m_j \end{cases} \quad (9)$$

where A denotes $2|y(n-1)|_{m_j} + |x(n)|_{m_j}$. The design of a modulo m_j scaled accumulator, for RNS-DA applications, was presented in [7]. An improved architecture can be innovated by using CSAs (carry save adders) to realize the 2nd and 3rd terms in Equation (9), as shown in Figure 1. The new RNS-DA accumulator has only one carry propagation stage in the critical path, thus improving the performance of an RNS-DA based system. The final result is selected on the basis of the carries generated in the summation stages. Thus, s1 is selected when C0c1c2c3= 0000 or 0001, s2 when C0c1c2c3= 0011 or 1011, and s3 when C0c1c2c3= 1010, 1000 or 0100.

A modulo m_j path of the direct RNS-DA implementation of the i^{th}-octave filter bank is defined in terms of an n_j-bit unsigned representation:

$$\left|a_n^{(i-1)}\right|_{m_j} = \sum_{l=0}^{n_j-1} 2^l \left|a_{n,l}^{(i-1)}\right|_{m_j} \quad (10)$$

where $n_j = \lceil \log_2(m_j) \rceil$, and l denotes the l-th bit of the indicated residue. Substituting Equation (11) into Equation (2), interpreted in a modulo m_j sense, the i^{th}-octave-approximation and detail sequences can be written as:

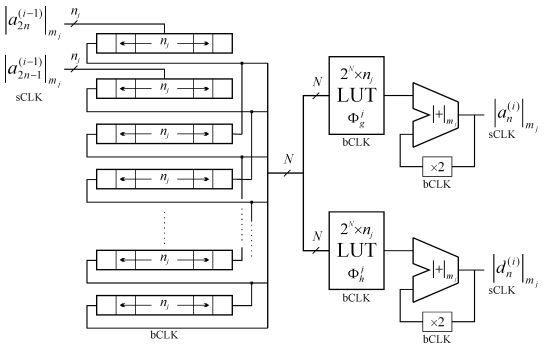

Fig. 2. RNS-DA 1-D DWT architecture.

$$\left|a_n^{(i)}\right|_{m_j} = \left|\sum_{k=0}^{N-1} g_k \sum_{l=0}^{n_j-1} 2^l \left|a_{2n-k,l}^{(i-1)}\right|_{m_j}\right|_{m_j} \qquad i = 1, 2, ..., J$$
$$\left|d_n^{(i)}\right|_{m_j} = \left|\sum_{k=0}^{N-1} h_k \sum_{l=0}^{n_j-1} 2^l \left|a_{2n-k,l}^{(i-1)}\right|_{m_j}\right|_{m_j} \qquad \left|a_n^{(0)}\right|_{m_j} \equiv \left|x_n\right|_{m_j} \tag{11}$$

Finally, by defining the DA functions:

$$\Phi_g^j(l) = \left|\sum_{k=0}^{N-1} g_k a_{2n-k,l}^{(i-1)}\right|_{m_j} \qquad \Phi_h^j(l) = \left|\sum_{k=0}^{N-1} h_k a_{2n-k,l}^{(i-1)}\right|_{m_j} \tag{12}$$

and interchanging the order of summations in Equation (12), the RNS encoded i^{th}-octave filter bank is computed by:

$$\left|a_n^{(i)}\right|_{m_j} = \left|\sum_{l=0}^{n_j-1} 2^l \Phi_g^j(l)\right|_{m_j} \qquad \left|d_n^{(i)}\right|_{m_j} = \left|\sum_{l=0}^{n_j-1} 2^l \Phi_h^j(l)\right|_{m_j} \tag{13}$$

Figure 2 shows the modulo m_j RNS-DA architecture for the i^{th}-octave analysis filter bank. N registers are used to left shift the input samples. The two inputs are sampled at a rate sCLK, the two LUTs storing Φ_g^j and Φ_h^j are accessed in each bit clock (bCLK) cycle to generate the terms of the relationship given in Equation (14). The clock sCLK is easily generated dividing bCLK by the modulus width, n_j. Finally, two modified modulo m_j accumulators compute recursively and in parallel the sequences shown in Equation (13). The polyphase RNS-DA architecture, suitable for high-order filter banks, is shown in Figure 3. The filter is computed using the polyphase filter bank decomposition represented by:

$$\left|a_n^{(i)}\right|_{m_j} = \left|\left|\sum_{l=0}^{n_j-1} 2^l \Phi_{g_0}^j(l)\right|_{m_j} + \left|\sum_{l=0}^{n_j-1} 2^l \Phi_{g_1}^j(l)\right|_{m_j}\right|_{m_j}$$
$$\left|d_n^{(i)}\right|_{m_j} = \left|\left|\sum_{l=0}^{n_j-1} 2^l \Phi_{h_0}^j(l)\right|_{m_j} + \left|\sum_{l=0}^{n_j-1} 2^l \Phi_{h_1}^j(l)\right|_{m_j}\right|_{m_j} \tag{14}$$

where the contents of the four $2^{N/2} \times n_j$ LUTs are given by:

$$\Phi_{g_0}^j(l) = \left|\sum_{k=0}^{N/2-1} g_0(k) a_{2n-2k,l}^{(i-1)}\right|_{m_j} \qquad \Phi_{g_1}^j(l) = \left|\sum_{k=0}^{N/2-1} g_1(k) a_{2n-2k-1,l}^{(i-1)}\right|_{m_j}$$
$$\Phi_{h_0}^j(l) = \left|\sum_{k=0}^{N/2-1} h_0(k) a_{2n-2k,l}^{(i-1)}\right|_{m_j} \qquad \Phi_{h_1}^j(l) = \left|\sum_{k=0}^{N/2-1} h_1(k) a_{2n-2k-1,l}^{(i-1)}\right|_{m_j} \tag{15}$$

In this way, four $2^{N/2} \times n_j$ are necessary to compute Equation (15) instead of two $2^N \times n_j$ LUTs as previous required. The four LUT outcomes, read in each bCLK cycle, are processed by means of four modified modulo m_j accumulator, and finally pair-wise added to obtain the final outputs.

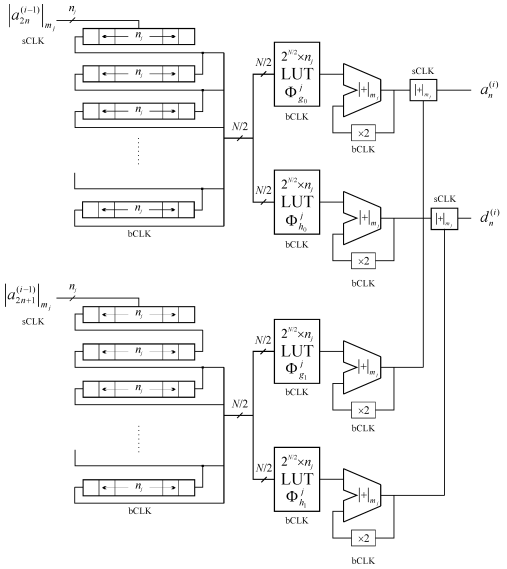

Fig. 3. Polyphase RNS-DA 1-D DWT architecture.

The presented RNS-DA architectures take advantage of short wordlength computations to increase the system bandwidth. Processing within an RNS channel is accomplished in n_j accumulation cycles, where n_j is small. RNS-DA designs can exceed the performance of a 2C-DA system that requires more accumulation cycles to implement a typical filter. The sampling frequency of the RNS-DA scheme, $f_{\text{RNS-DA}}^{\text{sCLK}}$, is related to that of the 2C-DA rate $f_{\text{2C-DA}}^{\text{sCLK}}$, by:

$$f_{\text{RNS-DA}}^{\text{sCLK}} = \left(B/n_j\right)\left(f_{\text{RNS-DA}}^{\text{bCLK}} / f_{\text{2C-DA}}^{\text{bCLK}}\right) f_{\text{2C-DA}}^{\text{sCLK}} \tag{16}$$

where $f_{\text{RNS-DA}}^{\text{bCLK}}$ and $f_{\text{2C-DA}}^{\text{bCLK}}$ are the accumulation clock frequency of RNS-DA and 2C-DA schemes, respectively. In this way, the RNS-DA improvements are proportional to the wordlength ratio (*i.e.*, B:n_j), and the bit-clock ratio.

6. SIMULATION RESULTS

An 8-tap DWT filter bank was used to illustrate the design of 2C-DA and RNS-DA systems. The comparison was carried out using FPL and two standard cell ASIC technologies. The selected FPL devices were the 0.5µm Altera FLEX10K [13], while the ASIC libraries were for the 0.8µm MSU SCMOS and the Chip Express 0.35µm triple-level metal CX3003 CMOS technology. FLEX10K devices consist of an array of logic elements (LEs) and a number of built-in tables called embedded array blocks (EABs). EABs provides 2K-bit memory blocks and are the target to map DA LUTs.

Different input, coefficient, and output precisions were considered. The use of 6-bit wide RNS channels was found to be an attractive choice for FPL devices since performance is divided by the modulus width. The dynamic range is covered with 6-bit moduli, say {64, 63, 61, 59} and {64, 63, 61, 59, 55}, cover a range from 23 to 29-bits respectively. Tables I and II audit the number of LEs and EABs and area requirements of a 2C-DA and RNS-DA design. Figure 4 displays the sample rate as a function of the input precision using 2C-DA and RNS-DA. The results show that the reported RNS-DA methodology,

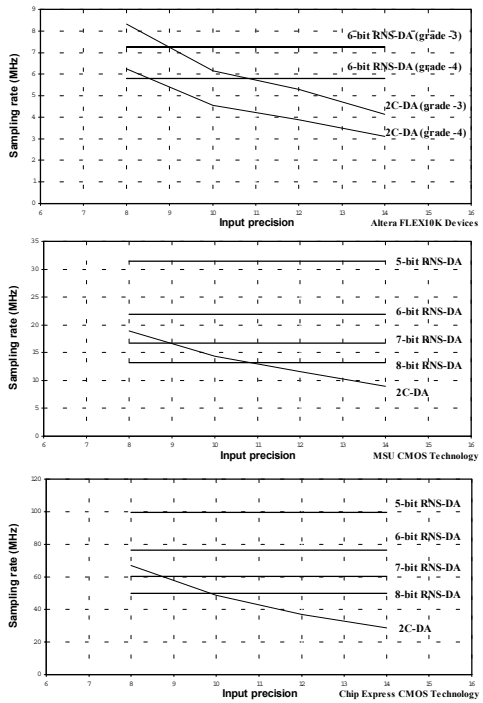

Fig. 4. Maximum sample rate for RNS-DA and 2C-DA architectures as a function of the input precision.

compared to a traditional 2C-DA design, enjoys a performance advantage that increases with precision. For FPL devices, RNS-DA increases the throughput by 84.72% and 109.13%, for grade –3 and –4 devices respectively, over an equivalent 2C-DA system having a 14-bit input. For a cell-based design and when timing constrains were imposed to the accumulation paths looking for an area-delay optimum design, the mean increase in the total area for the 2C-DA designs was 30% and 47% for the 0.8µm and 0.35µm ASIC technologies, respectively while the consequent reduction in the critical path delay was of 65% and 52%. Even with these delay optimisations, RNS-DA was found to be considerably fastest than a 2C-DA design, especially for high precision applications as shown in Figure 4, and Tables I and II.

	RNS-DA		2C-DA			
	6	64	[8,10,23]	[10,10,23]	[12,12,27]	[14,12,29]
#LEs	189	100	220	248	292	320
#EABs	2	2	4	4	4	4

Table I. #LEs and #EABs for 6-bit RNS-DA channels and for different precision 2C-DA architectures. [*x*,*y*,*z*] represents *x*-bit input, *y*-bit coefficients and *z*-bit output.

7. CONCLUSION

This paper considers the design and implementation of digital filters using an RNS-DA paradigm using FPL and ASIC technologies. To achieve a high level of performance a new CSA-based scaled modulo accumulator was developed. With this, and other innovations, the RNS-DA architecture was shown to be well suited for integrating DSP objects. To test the voracity of the proposed methodology, a DWT filter bank was used as a standard. An exhaustive comparison of 2C-DA and RNS-DA was carried out using commercially available FPL and standard-cell technologies with RNS-DA shown to be advantageous, especially for high-precision applications.

n_j $m_j \le 2^{n_j}$	MSU 0.8µm CA/NCA (µm^2)		Chip Express 0.35µm CA/NCA (#modules)	
	RNS-DA			
	$m_j < 2^{nj}$	$m_j = 2^{nj}$	$m_j < 2^{nj}$	$m_j = 2^{nj}$
5	48368/20320	11328/19680	1104/302	231/320
6	51952/24000	13824/23360	1506/378	306/378
7	59864/27680	16024/27040	1741/437	332/437
8	65960/31360	17664/30720	1897/497	364/497
wordwidths	**2C-DA**			
[8,12,23]	71864/57920		1674/885	
[10,12,25]	78008/64160		1677/988	
[12,12,27]	84256/70240		1891/1091	
[14,12,29]	90240/76320		1948/1192	

Table II. Combinational Area (CA) and non-CA (NCA) required by an 8-tap polyphase design. [*x*,*y*,*z*] represents *x*-bit input, *y*-bit coefficients and *z*-bit output.

ACKNOWLEDGEMENTS

The authors were supported by the Comisión Interministerial de Ciencia y Tecnología (Spain) under project PB98-1354.

REFERENCES

[1] C. Dick, "FPGAs: The High-End Alternative for DSP Applications," *DSP Engineering*, Spring 2000.

[2] F. Taylor and J. Mellot, *Hands-On Digital Signal Processing*, McGraw Hill, 1998.

[3] F. Taylor, "An Analysis of the Distributed Arithmetic Digital Filter," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 34, no. 5, pp. 1165-1170, 1986.

[4] S. A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review," *IEEE Acoustics, Speech and Signal Processing Magazine*, pp. 4-19, 1989.

[5] A. Peled, B. Liu, "A New Hardware Realization of Digital Filters," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-22, no. 6, pp. 456-462, 1974.

[6] M. Soderstrand, W. Jenkins, G. A. Jullien, F. J. Taylor, *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*. IEEE Press, 1986.

[7] A. García, U. Meyer-Bäse, A. Lloris, F. Taylor, "RNS Implementation of FIR Filters based on Distributed Arithmetic Using Field-Programmable Logic," *Proc. of the 1999 IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 486-489, 1999.

[8] H. Safiri, H. Ahamadi, G. Jullien, V. Dimitrov, "Design of FPGA Implementation of Systolic FIR Filters Using Fermat Number ALU," *Proc. of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, 1997.

[9] E. DiClaudio, F. Piazza, and G. Orlandi, "Fast Combinational RNS Processors for DSP Applications, *IEEE Trans. on Computers*, pp. 624-633, May 1995.

[10] J. Ramírez, A. García, P. G. Fernández, L. Parrilla, A. Lloris, "RNS-FPL Merged Architectures for the orthogonal DWT," *Electron. Lett.*, no. 14, pp. 1198-1199, 2000.

[11] G. Strang, T. Nguyen, *Wavelets and Filter Banks*. Wellesly-Cambridge Press, 1997.

[12] F. Taylor, *Digital Filter Design Handbook*, 1983.

[13] Altera Corp., *Altera Digital Library*, Jun. 2000.