Implementation of RNS Analysis and Synthesis Filter Banks for the Orthogonal Discrete Wavelet Transform over FPL devices

J. Ramirez, A. Garcia, L. Parrilla, A. Lloris
Dept. of Electronics and Computer Technology
University of Granada
Campus Universitario Fuentenueva
Granada, Spain 18071

P. G. Fernández
Dept. of Electrical Engineering
University of Jaén
Escuela Politécnica Superior
Jaén, Spain 23071

Abstract—RNS architectures to compute the orthogonal DWT and its inverse are shown. The relation between the coefficients of the analysis and synthesis filters allows to halve the number of required LUTs and modular adders. Simulations of one- and two- octave implementations using VHDL and FPL devices show a performance advantage of up to 23.45 and 96.58% when compared to the 2’s complement arithmetic versions, respectively.

I. INTRODUCTION

The number of applications requiring high-precision and high-bandwidth signal processing is constantly increasing and the development of structures for these systems is of growing interest. The RNS [1, 2] (Residue Number System) is gaining relevance in the DSP field due to its intrinsic parallelism and its throughput advantage over traditional arithmetic systems. The RNS palliates the drawback of two’s complement DSP systems by not suffering from long propagation delays. The RNS provides a fundamental methodology for partitioning a large dynamic range system into a number of smaller but independent channels over which computations are performed in parallel. Each channel consists of a replica of the original system being all the arithmetic operations defined as modulo operations. With this scheme, multiplication by fixed coefficients is carried out efficiently by pipelined Look-Up Tables (LUTs) and the throughput of the global system is not prejudiced by the increasing precision since it is only necessary to include more channels to cover the increasing dynamic range.

Field-Programmable Logic (FPL) devices have recently generated interest for real time DSP systems due to their ability to implement custom solutions while maintaining flexibility through device reprogramming. These devices represent an alternative to ASICs, since a significant performance improvement over microprocessors is obtained with moderate development cost, while maintaining system programmability. The new programmable device families, such as Altera FLEX10K or APEX20K [3], with embedded memory blocks, provide RNS arithmetic support by means of small built-in embedded memories and logic supporting fast carry and cascade operations. These systems are gaining relevance and a number of modern DSP applications were accelerated by using efficient RNS arithmetic units [4-7].

Interest in wavelet transform [8, 9] has grown dramatically during the last decade in many areas, including speech, image and video processing, numerical analysis, statistics, physics, etc. Thus, different VLSI architectures have been reported: single chip parallel filter implementations, bit-serial, systolic arrays, distributed memory, and SIMD arrays [10-16]. This paper shows the implementation of RNS analysis and synthesis filter banks for the orthogonal 1-D DWT (One-Dimensional Discrete Wavelet Transform) over Altera FLEX10K FPL devices. These architectures are modelled and synthesized using structural VHDL. Hardware requirements are assessed in terms of the number of logic and memory blocks. These RNS architectures are compared to the equivalent systems based on two’s complement arithmetic showing a considerable increase in performance.

II. RESIDUE NUMBER SYSTEM

An RNS [1, 2] is defined by a set of positive pairwise relatively prime integers \( m_1, m_2, ..., m_L \) called moduli. Any integer \( X \in [0, M-1] \) is represented by the \( L \)-tuple \( [x_1, x_2, ..., x_L] \), where \( x_i = x \mod m_i (i=1, 2, ..., L) \) is the residue of \( X \) modulo \( m_i \) while \( M \), the dynamic range of the RNS system, is given by:

\[
M = \prod_{i=1}^{L} m_i
\]

(1)

Arithmetic in the RNS is defined over the ring of integers modulo \( M \). For \( 0 \leq X, Y, Z < M \):

\[
Z = \left[ X \circ Y \right]_M = \left[ z_1 \equiv [x_1 \oplus y_1]_m, ..., z_L \equiv [x_L \oplus y_L]_m \right]
\]

(2)

where \( \oplus \) represents either addition, subtraction or multiplication.

As derived from (2), the RNS provides a fundamental methodology for the partitioning of a entire system in a number of parallel channels based on modular arithmetic. In this way, the dynamic range of the system is handled by these small wordwidth channels and, consequently, throughput is increased because of the efficiency of modulo addition and modulo multiplication using high-performance synchronous LUTs.

III. DISCRETE WAVELET TRANSFORM

Many new transform techniques have arisen in the last years that are specifically oriented to image coding. These techniques frequently appear as multiresolution analysis, time-frequency analysis, pyramid algorithms and wavelet transforms [8, 9]. They offer better compression ratios than DCT-based coding techniques and do not suffer from the effects of blocking.

A signal can be expressed as a series expansion using a set of basis functions generated from a basic wavelet function by dilations and translations. The wavelet series expansion represents signals with a sequence of wavelet approximation
and detail coefficients corresponding to sampling in the time-scale space. The successive discrete approximation sequences are lower and lower resolution versions of the original, each sampled twice as sparsely as its predecessor. The DWT can be implemented by a tree-structured algorithm in the multiresolution analysis framework. Thus, the level J 1-D DWT decomposition of a sequence \( x(n) \) is defined by the recurrent equations:

\[
\begin{align*}
\alpha_i^{(l)} &= \sum_{k=0}^{N/2^l - 1} g_k a_{2^l - k}^{(l-1)} & i = 1, 2, ..., J \\
\beta_i^{(l)} &= \sum_{k=0}^{N/2^l - 1} h_k d_{2^l - k}^{(l-1)} & \beta_i^{(0)} = x(n) 
\end{align*}
\]  

(3)

where the sequences \( \alpha_i^{(l)} \) and \( \beta_i^{(l)} \) are the approximation and detail sequences at level \( i \), and the \( g_k \) and \( h_k \) coefficients identify the wavelet family used. These equations correspond to an identical structure which consists of a pair of convolvers and decimators that is repeated at each decomposition level. The signal \( x(n) \) can be perfectly recovered through its multiresolution decomposition \( \{ \alpha_n^{(0)}, \alpha_n^{(1)}, \alpha_n^{(2)}, \ldots, \alpha_n^{(J)} \} \) by:

\[
\begin{align*}
\alpha_{m-i}^{(l-1)} &= \sum_{k=0}^{N/2^{l-1} - 1} h_{2^l - k} \alpha_{m-k}^{(l)} + \sum_{k=0}^{N/2^{l-1} - 1} g_{2^l - k} \beta_{m-k}^{(l)} & m \text{ even} \\
\beta_{m-i}^{(l-1)} &= \sum_{k=0}^{N/2^{l-1} - 1} h_{2^l - k} \beta_{m-k}^{(l)} + \sum_{k=0}^{N/2^{l-1} - 1} g_{2^l - k} \beta_{m-k}^{(l)} & m \text{ odd}
\end{align*}
\]  

(4)

Fig. 1 shows the architecture based on filter banks to compute the three octave decomposition \( \{ \alpha_n^{(0)}, \alpha_n^{(1)}, \alpha_n^{(2)}, \alpha_n^{(3)} \} \) of the input sequence, \( x(n) \), and the corresponding perfect reconstruction structure.

While there are different kinds of wavelet families, most applications make use almost exclusively of orthogonal wavelets. The perfect reconstruction property and the orthogonality conditions imposed on the wavelets imply the following relation between the high-pass decomposition and reconstruction filter coefficients, \( h_k \) and \( h_k \), and low-pass decomposition and reconstruction filters, \( g_k \) and \( g_k \), respectively:

\[
\begin{align*}
h_k &= (-1)^{k+1} g_{N-k-1} & k = 0, 1, ..., N - 1 \\
\overline{h_k} &= (-1)^{k+1} g_{N-K-1} & \overline{h_k} \end{align*}
\]  

(5)

where \( N \) is the even number of taps of the analysis and synthesis FIR filters. Thus, the orthogonality condition given in (5) can be exploited to halve the number of multipliers and adders in each octave of the analysis and synthesis multiresolution wavelet filters.

IV. ORTHOGONAL 1-D DWT RNS ARCHITECTURE

By substituting the orthogonality condition (5) in (3), the octave-\( i \) detail sequence \( \beta_i^{(0)} \) can be expressed as:

\[
\begin{align*}
\alpha_i^{(l)} &= \sum_{k=0}^{N/2^l - 1} h_k a_{2^l - k}^{(l-1)} & i = 1, 2, ..., J \\
\beta_i^{(l)} &= \sum_{k=0}^{N/2^l - 1} \overline{h_k} d_{2^l - k}^{(l-1)} & \beta_i^{(0)} = x(n)
\end{align*}
\]  

(6)

In this way, the octave-\( i \) approximation, \( \alpha_i^{(0)} \), and detail, \( \beta_i^{(0)} \), sequences can be computed sharing \( N \) multipliers of the low-pass filter with coefficients \( \{ g_0, g_1, \ldots, g_{N-1} \} \) in alternate cycles and in accordance with:

\[
\begin{align*}
\alpha_i^{(0)} &= \sum_{k=0}^{N/2^l - 1} g_k a_{2^l - k}^{(l-1)} & i = 1, 2, ..., J \\
\beta_i^{(0)} &= \sum_{k=0}^{N/2^l - 1} \overline{g_k} a_{2^l - k}^{(l-1)} & \beta_i^{(0)} = x(n)
\end{align*}
\]  

(7)

Likewise, the reconstruction algorithm involved in the 1-D IDWT (One-Dimensional Inverse Discrete Wavelet Transform) can benefit from the relations between the synthesis filter coefficients, \( h_k \) and \( g_k \), given in (5) by sharing the \( N \) LUTs of only one filter. Thus, each of the reconstruction stages consists of a pair of interpolators and convolvers that compute:

\[
\begin{align*}
\alpha_{m-i}^{(l-1)} &= \sum_{k=0}^{N/2^{l-1} - 1} h_{2^l - k} \alpha_{m-k}^{(l)} + \sum_{k=0}^{N/2^{l-1} - 1} g_{2^l - k} \beta_{m-k}^{(l)} & m \text{ even} \\
\beta_{m-i}^{(l-1)} &= \sum_{k=0}^{N/2^{l-1} - 1} h_{2^l - k} \beta_{m-k}^{(l)} + \sum_{k=0}^{N/2^{l-1} - 1} g_{2^l - k} \beta_{m-k}^{(l)} & m \text{ odd}
\end{align*}
\]  

(8)

and that can be implemented sharing the multipliers of the low-pass synthesis filter \( \{ \overline{g}_0, \overline{g}_1, \ldots, \overline{g}_{N-1} \} \) to generate the low-pass and high-pass filter outputs.

Fig. 2 and 3. They implement the analysis and synthesis filters, respectively, halving the number of LUTs due to the orthogonality relation between the filter coefficients. The architecture for the octave-\( i \) modulo \( m_i \) decomposition
consists of \( N \) shared LUTs and two modular adder trees that compute the approximation and detail sequences. The LUTs are clocked at the input frequency and are used in alternate clock cycles to compute the products of the low-pass and high-pass filters. These products are added by two modular adder trees controlled by the half of the input frequency out-of-phase clocks CLK1 and CLK2. On the other hand, the operation of the architecture given in Fig. 3 for the 1-D IDWT is similar. In even and odd cycles only \( N/2 \) multipliers of each of the low-pass and high-pass synthesis filters are used. Thus, the output can be computed with only \( N \) products per cycle. Finally, the octave-\( i \) modulo \( m \), reconstruction sequence is computed with two modular adder trees clocked by the half the input frequency out-of-phase clocks CLK1 and CLK2.

Both architectures allow substituting the two modular adder trees clocked by CLK1 and CLK2 for only one modular adder tree clocked at the input frequency. This alternative reduces the resources required, but the performance of the system can be degraded if long wordwidth channels are used, since the modular adder tree must operate at the input frequency. On the other hand, by using fixed coefficients multipliers mapped on fast LUTs, a considerable increase in performance can be obtained when these RNS-based solutions are compared to fixed point two's complement arithmetic architectures.

V. IMPLEMENTATION ON FPL DEVICES

FPL devices providing embedded LUTs and dedicated logic blocks are potential solutions for RNS-based MAC-intensive algorithms. These devices consist of LEs (Logic Elements) and LUTs. Depending on the family, each LE includes one or more variable input size LUTs (typical are 2\( \times \)1, 2\( ^2 \times \)1, 2\( ^3 \times \)1 or 2\( ^4 \times \)1), fast carry propagation logic and one or more flip-flops. These LUTs allow us to build specialized memory functions such as ROM or RAM.

Specifically, each LE included in the Altera FLEX10K [3] device consists of a 2\( ^{\times} \)1 LUT, an output register and dedicated logic for fast carry and cascade chains in arithmetic mode. A number of Embedded Array Blocks (EABs), providing a 2K-bit RAM or ROM and configurable as 2\( ^{\times} \)8, 2\( ^{\times} \)4, 2\( ^{\times} \)2 or 2\( ^{\times} \)1, are the cores for the implementation of RNS LUT-based multipliers.

Two's complement and RNS versions of the proposed 1-D DWT and IDWT architectures were implemented over Altera FLEX10K devices to compare hardware complexity and performance. Structural VHDL was used in the synthesis and parameters such as area and performance were assessed. 8-bit input samples and 10-bit filter coefficients were assumed, so that one octave and two octaves required 21- and 34-bit dynamic ranges, respectively. Table I shows the results obtained for each 6-, 7- and 8-bit RNS channel and for traditional arithmetic over grade -4 speed FLEX10K devices. Hardware requirements were assessed in terms of the number of LEs and EABs while performance, measured in MegaSamples Per Second (MSPS), was evaluated in terms of the register-to-register maximum delay path. In this way, assuming an RNS consisting of 6-bit channels, performance was up to 23.45% and 96.58% better for one octave and two octaves, respectively, when compared to equally pipelined traditional arithmetic implementations. This increase in throughput is achieved by the efficiency of fixed coefficients multiplication using fast LUTs.

VI. CONCLUSION

Regular and compact RNS architectures to compute the orthogonal 1-D DWT and 1-D IDWT were proposed. They are scalable, highly flexible (regarding the number of taps of the filters) and suitable for being considered as part of larger processors. The relation between the filter coefficients allows halve the LUTs required by each octave and their performance is independent of the precision of the input and the filter coefficients. One- and two-octave RNS and traditional arithmetic implementations using VHDL and Altera FLEX10K FPL devices were carried out to quantify complexity and performance parameters. The performance advantage of the proposed RNS-based solutions when compared to fixed point architectures was up to 23.45% and 96.58% for one and two octaves, respectively.
TABLE I
Hardware requirements and performance of FPL architectures based on
the RNS and binary two's complement arithmetic for the orthogonal 1-D DWT and 1-D IDWT

<table>
<thead>
<tr>
<th></th>
<th>1-D DWT #LEs</th>
<th>1-D DWT #EABs (Memory bits)</th>
<th>1-D IDWT #LEs</th>
<th>1-D IDWT #EABs (Memory bits)</th>
<th>Throughput (MSPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-bit RNS channel</td>
<td>316</td>
<td>8 (8x6/64)</td>
<td>321</td>
<td>8 (8x6/64)</td>
<td>68.96</td>
</tr>
<tr>
<td>7-bit RNS channel</td>
<td>362</td>
<td>8 (8x7/128)</td>
<td>367</td>
<td>8 (8x7/128)</td>
<td>67.11</td>
</tr>
<tr>
<td>8-bit RNS channel</td>
<td>403</td>
<td>8 (8x8/256)</td>
<td>409</td>
<td>8 (8x8/256)</td>
<td>65.78</td>
</tr>
<tr>
<td>One-octave 21-bit Dynamic Range (Binary two’s complement)</td>
<td>1164</td>
<td>-</td>
<td>1173</td>
<td>-</td>
<td>55.86</td>
</tr>
<tr>
<td>Two-octave 34-bit Dynamic Range (Binary two’s complement)</td>
<td>3689</td>
<td>-</td>
<td>3705</td>
<td>-</td>
<td>35.08</td>
</tr>
</tbody>
</table>

ACKNOWLEDGEMENT
The authors were supported by the Dirección General de Enseñanza Superior (Spain) under project PB98-1354.
CAD tools and supporting material were provided by Altera Corp. under the Altera University Program.

REFERENCES