MULTIPLIERLESS DCT ALGORITHM FOR IMAGE COMPRESSION APPLICATIONS

Vassil Dimitrov and Khan Wahid

Abstract: This paper presents a novel error-free (infinite-precision) architecture for the fast implementation of 8x8 2-D Discrete Cosine Transform. The architecture uses a new algebraic integer encoding of a 1-D radix-8 DCT that allows the separable computation of a 2-D 8x8 DCT without any intermediate number representation conversions. This is a considerable improvement on previously introduced algebraic integer encoding techniques to compute both DCT and IDCT which eliminates the requirements to approximate the transformation matrix elements by obtaining their exact representations and hence mapping the transcendental functions without any errors. Apart from the multiplication-free nature, this new mapping scheme fits to this algorithm, eliminating any computational or quantization errors and resulting short-word-length and high-speed-design.

Keywords: DCT, Image Compression, Algebraic Integers, Multiplier-less Architecture

Introduction

The Discrete Cosine Transform (DCT) is the core transform of many image processing applications for reduced bandwidth image and video transmission including JPEG and MPEG standards. Several algorithms and architectures have been proposed to optimize DCT implementations using 1-D and 2-D algebraic integer (AI) encoding of the DCT basis functions, where both single and multidimensional AI schemes have been used which allow low-complexity and parallel architectures [Dimitrov, 1998][Dimitrov, 2003]. In all of these previous encoding techniques, conversion from the output of the 1-D DCT algorithm has been required, even if the DCT is being used in a separable 2-D DCT computation. Recently, we have introduced a new algebraic integer encoding technique which, along with a previously published scalar quantization algorithm [Arai, 1988], removes the need for conversion to binary at the end of the first 1-D DCT [Wahid, 2004]. But here in this paper, we present an extensive analysis of that idea of 2-D error-free algebraic integer encoding, in terms of computational complexity and mathematical precision required to implement the algorithm. Here, we also show that this new algorithm provides a considerable reduction in hardware for the separable 2-D DCT computation, and also a reduction in hardware for a stand-alone 1-D DCT computation.

The final conversion step, where we convert the algebraic integer numbers to fixed-precision (FP) binary, may generate some rounding errors but these errors are only introduced at the very end of the transformation process, not distributed throughout the calculation, as is the case for a finite-precision binary implementation. This 2-D algebraic integer quantization not only reduces the number of arithmetic operations, but also reduces the dynamic range of the computations. This scheme can also be extended for the error-free computation of the Scaled Inverse-DCT [Arai, 1988].

Algebraic Integer Quantization (AIQ)

Algebraic integers are defined by real numbers that are roots of monic polynomials with integer coefficients [Dedekind, 1996]. As an example, let $\omega = e^{\frac{2\pi j}{16}}$ denote a primitive 16th root of unity over the ring of complex numbers. Then ω is a root of the equation $x^8 + 1 = 0$. If ω is adjoined to the rational numbers, then the associated ring of algebraic integers is denoted by $Z[\omega]$. The ring $Z[\omega]$ can be regarded as consisting of polynomials in ω of degree 7 with integer coefficients. The elements of $Z[\omega]$ are added and multiplied as polynomials, except that the rule $\omega^8 = -1$ is used in the product to reduce the degree of powers of ω to below 8. For an integer, *M*, $Z[\omega]_M$ is used to denote the elements of with coefficients between $-\frac{M}{2}$ and $\frac{M}{2}$.

The idea of using algebraic integers in DSP applications was first explored by Cozzens and Finkelstein [Cozzens, 1985]. In their work, the algebraic integer number representation, in which the signal sample is represented by a set of (typically four to eight) small integers, combines, if necessary, with the Residue Number System (RNS) to produce processors composed of simple parallel channels [Games, 1989]. In their procedure, AI representation was used to approximate complex input signals. People have found various applications of algebraic integer in Coding theory such as, algebraic integers can produce exact pole zero cancellation pairs that are used in recursive complex finite-impulse response, frequency sampling filter designs [Meyer, 2001].

Apart from the low-complexity error-free computation of DCT and IDCT, our group has also introduced algebraic integer coding to compute the 'cas' function of the Discrete Hartley Transform [Baghaie, 2001] and the basis functions of the Discrete Wavelet Transform [Wahid, 2003]. In case of DWT, using 2-D AI encoding technique, not only have we achieved significant improvement in quality of reconstructed image but the hardware is also greatly reduced. Like the DWT, the application of the DCT and the IDCT is also in the field of image and video compression for low bandwidth transmission, and so the enhancement of 2-D AI encoding technique for these transforms is quite necessary and timely in this regard. Another advantage of using AI scheme to these discrete-valued transforms is that greater accuracy can be achieved using fewer bits than necessary with a conventional two's complement approach.

Discrete Cosine Transform (DCT)

For a real data sequence $x(n)$ of length *N*, the DCT is defined as follows:

$$F(k) = 2\sum_{n=0}^{N-1} x(n)\cos\left[\frac{(2n+1)k}{2N}\pi\right];\ 0 \le k \le N-1 \tag{1}$$

The Inverse DCT (IDCT) is also defined as:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} \overline{F}(k)\cos\left[\frac{(2n+1)k}{2N}\pi\right];\ 0 \le n \le N-1 \tag{2}$$

Where, $\overline{F}(k) = \begin{cases} \dfrac{F(0)}{2} & k=0 \\ F(k) & otherwise \end{cases}$

The 2-D DCT and IDCT can also be found by extending the above 1-D equations.

AI Encoding of the Classical DCT: Both 1-D and 2-D AI encoding have been applied to classical DCT by our group [Dimitrov, 1998][Dimitrov, 2003]. In order to better understand the concept of 2-D algebraic integer quantization to DCT-SQ algorithm, here we will provide a quick review of AI encoding to classical DCT. Taking $z_1 = 2\cos\frac{\pi}{16}$ and $z_2 = 2\cos\frac{\pi}{4}$ and considering the 2-D polynomial expansion, $f(z_1, z_2) = \sum_{i=0}^{K}\sum_{j=0}^{L} a_{ij} z_1^i z_2^j$, we can exactly represent all cosine angles without error as shown in Table 1.

Table 1: 2-D AI representation of the cosine functions for 8-point DCT

2*cos*(0.π/16)	$\begin{bmatrix} 2&0&0&0 \\ 0&0&0&0 \end{bmatrix}$	2*cos*(4.π/16)	$\begin{bmatrix} 0&0&0&0 \\ 1&0&0&0 \end{bmatrix}$
2*cos*(1.π/16)	$\begin{bmatrix} 0&1&0&0 \\ 0&0&0&0 \end{bmatrix}$	2*cos*(5.π/16)	$\begin{bmatrix} 0&3&0&-1 \\ 0&1&0&0 \end{bmatrix}$
2*cos*(2.π/16)	$\begin{bmatrix} -2&0&1&0 \\ 0&0&0&0 \end{bmatrix}$	2*cos*(6.π/16)	$\begin{bmatrix} 2&0&-1&0 \\ -2&0&1&0 \end{bmatrix}$
2*cos*(3.π/16)	$\begin{bmatrix} 0&-3&0&1 \\ 0&0&0&0 \end{bmatrix}$	2*cos*(7.π/16)	$\begin{bmatrix} 0&-1&0&0 \\ 0&-3&0&1 \end{bmatrix}$

The use of multidimensional AI provides us with a variety of advantages. First of all, we are in a position to choose the best representation from a computational viewpoint, such that the equivalent representation scheme is as sparse as possible. Secondly, there are 24 possible combinations of applicable pairs of parameters, which makes this encoding scheme extremely flexible. Thirdly, the final reconstruction can be accomplished by making use of systolic architectures for polynomial evaluations. This technique allows reduction of the degree of polynomial expansion by a factor of two (compared to 1-D encoding where the degree of polynomial is 7 [Dimitrov, 1998]) and consequently speeds up the final reconstruction step by a factor of 2. A 2-D 8x8 DCT IP core based on this technique has recently been designed and fabricated. The core size of this chip is 1.8mmX1.2mm, the latency is 80 clock cycles, the power consumption is 4.8mW and the overall throughput is 75 mega-pixels/seconds [Jullien, 2003]. A micrograph of the chip is shown in Figure 1.

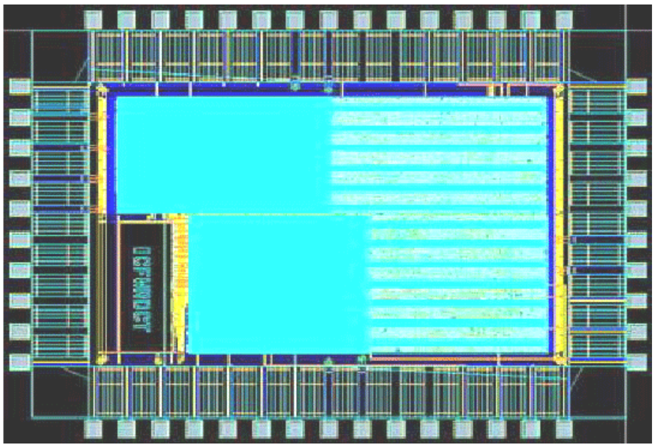

Figure 1: Micro-graph of AI-based DCT chip

Scaled DCT Algorithm

The DCT-SQ (sequential quantization) algorithm proposed by Arai et. al. [Arai, 1988] is presented as follows:

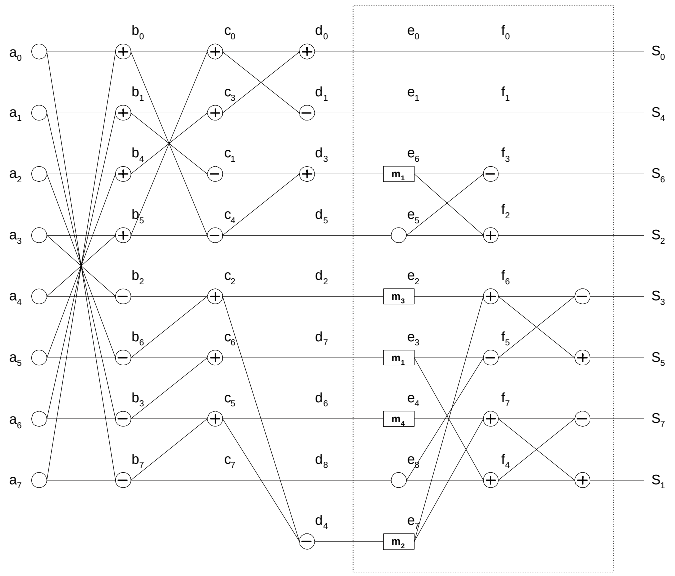

Figure 2: Signal Flow Graph of Arai Algorithm

where $\{a_i\}$ are input elements, $\{S_i\}$ are scaled DCT coefficients, and fixed multipliers are given by eqn. (3).

$$\{m_1, m_2, m_3, m_4\} = \left\{\cos\frac{4\pi}{16}, \cos\frac{6\pi}{16}, (\cos\frac{2\pi}{16} - \cos\frac{6\pi}{16}), (\cos\frac{2\pi}{16} + \cos\frac{6\pi}{16})\right\} \quad (3)$$

Proposed AI Encoding

The outlined area in Figure 2, (with a hardware cost of 5 multiplications and 10 additions) is where our new algebraic integer mapping will be used. Not only will we reduce the hardware count but we will also produce error-free results based on the exact representation of the basis function multipliers.

1-D Algebraic Integer Encoding: Let $z = \sqrt{2+\sqrt{2}}$ and consider the polynomial expansion:

$$f(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 \tag{4}$$

Since, $\cos\frac{2\pi}{16} = \frac{\sqrt{2+\sqrt{2}}}{2}$, $\cos\frac{4\pi}{16} = \frac{\sqrt{2}}{2}$ and $\cos\frac{6\pi}{16} = \frac{\sqrt{2-\sqrt{2}}}{2}$, we can represent $\{m_1, m_2, m_3, m_4\}$ (from eqn. (3)) exactly (infinite precision) with the integer coefficients (scaled by 2) as shown in Table 2.

Table 2: 1-D error-free multiply encoding

	a_0	a_1	a_2	a_3
m_1	-2	0	1	0
m_2	0	-3	0	1
m_3	0	4	0	-1
m_4	0	-2	0	1

Note that the multiplication between any real number and these coefficients can now be implemented with at most 2 shifts and 1 addition. This reduces the 5 multiplications and the 10 subsequent additions to only 9 AI additions. So, the total number of addition required to perform 1-D DCT is 30. We also note that there is no longer a precision problem since the AI encoding provides an exact representation. The flow graph of Figure 2 can now be implemented as shown in Figure 3.

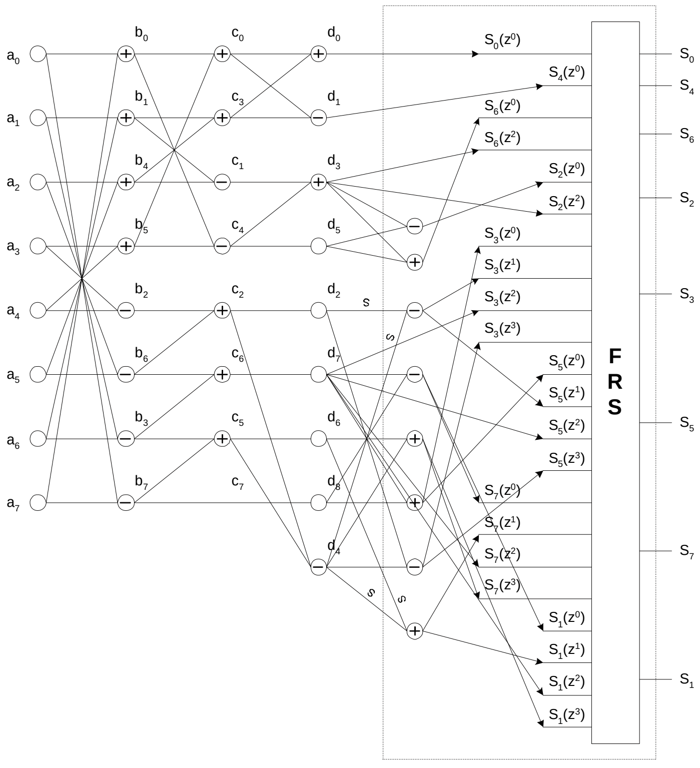

Figure 3: 1-D DCT (1-D error-free encoding)

The real numbers of $f(z)$ form a ring which may be denoted as $Z[\sqrt{2+\sqrt{2}}]$. Addition in this ring is component-wise and multiplication is equivalent to a polynomial multiplication modulo $z^4 - 4z^2 + 2 = 0$.

2-D Algebraic Integer Encoding: Applying a 2-D algebraic integer scheme to this algorithm results in a more sparse representation and more flexible encoding compared to previous techniques [Dimitrov, 2003]. For this encoding, the polynomial is expanded into 2 variables:

$$f(z_1, z_2) = \sum_{i=0}^{K} \sum_{j=0}^{L} a_{ij} z_1^i z_2^j \qquad (5)$$

Here we choose *K*=1 and *L*=1 to guarantee error-free encoding. For the most efficient encoding (i.e., to obtain the most sparse matrix), we have found the following: $z_1 = \sqrt{2+\sqrt{2}} + \sqrt{2-\sqrt{2}}$ and $z_2 = \sqrt{2+\sqrt{2}} - \sqrt{2-\sqrt{2}}$. The corresponding coefficients (scaled by 4) are encoded in the form of $\begin{bmatrix} a_{00} & a_{10} \\ a_{01} & a_{11} \end{bmatrix}$ as shown in Table 3.

Table 3: 2-D error-free multiply encoding

m 1	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	*m 3*	$\begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}$
m 2	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	*m 4*	$\begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$

We have therefore, mapped the multiplier transcendental functions without any error and with very low complexity. Note that in the encoding of all four multipliers, a_{00} requiring only 3 independent parallel channels, as a result, the flow graph in Figure 2 can now be implemented as in Figure 4.

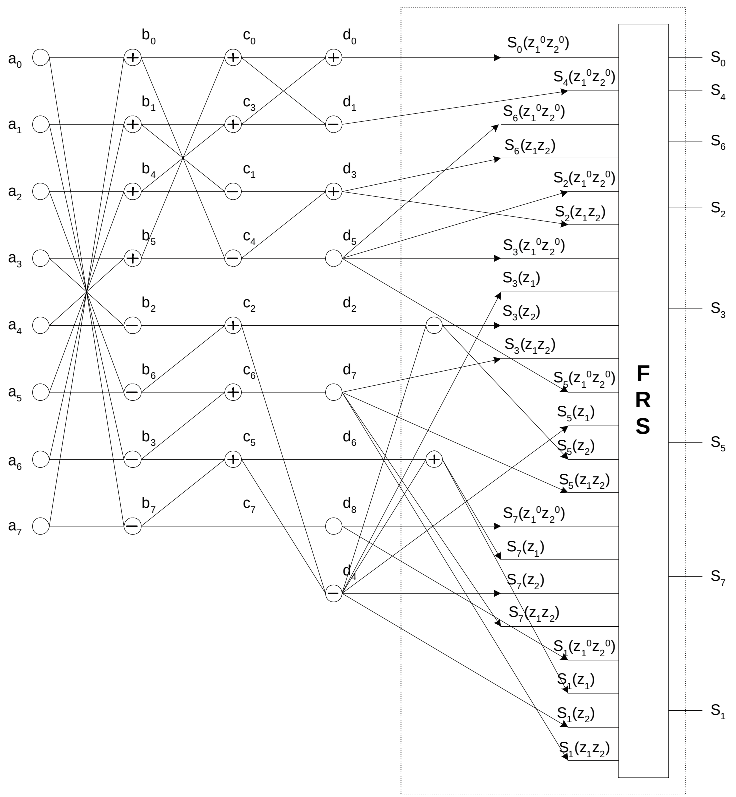

Figure 4: 1-D DCT (2-D error-free encoding)

The outlined area in Figure 4 contains only 2 adders and a Final Reconstruction Stage (FRS), where we finally map to a binary output, is performed as shown in Figure 5 and 6.

Final Reconstruction Step: For the computation of 2-D DCT, we need to recover the integer part of the result and the most significant bit of the fractional part, to allow correct rounding. By applying the 10-bit mappings of Table 4 to the 2-D AI representations of the inputs to the FRS stage, we reduce the hardware cost of the entire outlined area to 5 adders. A considerable reduction from the 5 multiplications and 10 adders of the architecture of Figure 2. Hence a total of only 24 adders is required to perform 8-point 1-D DCT.

For the final reconstruction, we can use Horner's rule [Knuth, 1981]. In that case, eqn. (4) and eqn. (5) can be re-written as:

$$f(z) = ((a_3 z + a_2)z + a_1)z + a_0 \tag{6}$$

$$f(z_1, z_2) = (a_{11} z_1 + a_{01}) z_2 + a_{10} z_1 \tag{7}$$

Now, taking different bit-lengths, and using Booth encoding, we can easily find the errors for different substitution precision. The signed-digit encoding errors (%) for different word lengths are provided in Table 5.

Table 4: FRS for different encoding scheme

Scheme	*Parameter*		*FRS*
1-D	z	10 bits	$2 - 2^{-2} - 2^{-5} + 2^{-8}$
		12 bits	$2 - 2^{-2} - 2^{-5} + 2^{-8}$
2-D	z_1	10 bits	$2 + 2^{-1} + 2^{-3} - 2^{-6}$
		12 bits	
	z_2	10 bits	$1 + 2^{-4} + 2^{-6} + 2^{-8}$
		12 bits	$1 + 2^{-4} + 2^{-6} + 2^{-8}$

Table 5: Bit encoding errors (%)

No of bits	*1-D*	*2-D*	
	z	z_1	z_2
8	2.16×10^{-3}	1.40×10^{-3}	3.94×10^{-3}
10	5.57×10^{-5}	1.40×10^{-3}	3.33×10^{-4}
12	5.57×10^{-5}	3.10×10^{-4}	3.33×10^{-4}
14	5.57×10^{-5}	3.36×10^{-5}	4.86×10^{-6}

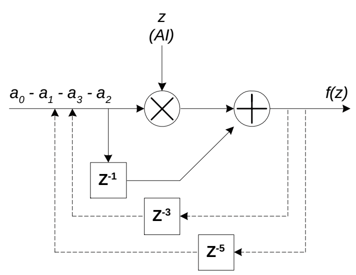

Figure 5: Final reconstruction step (1-D encoding)

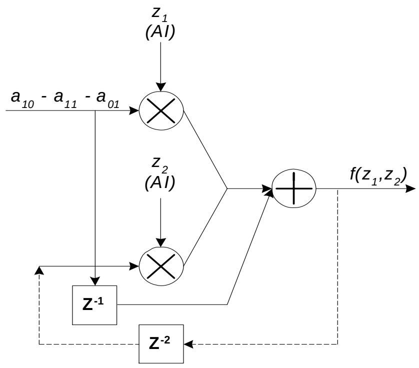

Figure 6: Final reconstruction step (2-D encoding)

Comparisons

In Table 6, we compare the computational complexity of previously published AI-based DCT encoding with the proposed scheme. In all cases, the new 2-D AI encoding scheme has the least number of computations. In Table 7, we present a comparison between some other published 2-D DCT architectures and the proposed algebraic integer approach. Taking the additions as the main computational block, the new multidimensional algebraic integer-quantization based architecture clearly has the lowest hardware count. Also remember the fact that all AI computations are performed without any error.

Table 6: Hardware complexity for different AIQ schemes

Algorithm	Degree of Polynomial	Additions	Shifts	Multiplications	Total Additions
1-D AI-based Chen DCT [Dimitrov, 1998]	7	6	9	0	156
2-D AI-based Chen DCT [Dimitrov, 2003]	7	3	4	0	132
Proposed 1-D AIQ	3	1	2	0	30
Proposed 2-D AIQ	2	0	1	0	24

Table 7: Comparison between different 8-point 2-D DCT

Algorithm	Multiplications	Additions
DCT-SQ [Arai, 1988]	80	464
Chen DCT [Chen, 1977]	256	448
Distributed DCT [Shams, 2002]	0	672
Proposed 1-D AIQ	0	480
Proposed 2-D AIQ	0	384

Conclusions

In this paper, we have introduced a new encoding scheme to compute both 1-D and 2-D DCT and IDCT which effectively reduces the overall arithmetic operations and allows multiplication-free, parallel, and very fast hardware implementation. Except for the final reconstruction stage, the complete 2-D DCT and IDCT can be implemented without error. The use of integers in the encoding scheme also results exact reconstruction. This idea of using algebraic integer scheme can be easily generalized to other algorithms when it is necessary to use real algebraic numbers of special form. The future work is directed towards the VLSI implementation of this approach for 2-D DCT and IDCT.

Acknowledgements

The authors are very thankful to Mr. Minyi Fu for his work on the VLSI implementation of the the 2-D AI algorithm shown in Figure 1. The authors also wish to thank the Alberta Informatics Circle of Research Excellence (iCORE), the Natural Sciences and Engineering Council (NSERC) of Canada, the Micronet and Gennum Corporation for their financial support for this research. The authors are also indebted to the Canadian Microelectronics Corporation (CMC) for providing the hardware and software infrastructure used in the development of this design.

Bibliography

[Arai, 1988] Y. Arai, T. Agui and M. Nakajima, "A Fast DCT-SQ Scheme for Images", Transactions of Institute of Electronics, Information and Communication Engineers, vol. E71, no. 11, pp. 1095-1097, 1988.

[Baghaie, 2001] R.Baghaie and V.Dimitrov, "Systolic Implementation of Real-valued Discrete transforms via Algebraic Integer Quantization", International Journal on Computers and Mathematics with Applications, vol. 41, pp. 1403-1416, 2001.

[Cozzens, 1985] J. H. Cozzens and L. A. Finkelstein, "Computing the Discrete Fourier Transform using Residue Number Systems in a Ring of Algebraic Integers", IEEE Transactions on Information Theory, vol. 31, pp. 580-588, 1985.

[Chen, 1977] W. Chen, C. Smith and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Transactions on Communications, vol. COM-25, no. 9, pp. 1004-1009, 1977.

[Dedekind, 1996] Richard Dedekind, "Theory of Algebraic Integers", Translated and introduced by John Stillwell, 1996.

[Dimitrov, 1998] V. S. Dimitrov, G. A. Jullien and W. C. Miller, "A New DCT Algorithm Based on Encoding Algebraic Integers", IEEE International Conference on Acoustics, Speech and Signal processing, pp. 1377-1380, 1998.

[Dimitrov, 2003] V. Dimitrov and G. A. Jullien, "Multidimensional Algebraic Integer Encoding for High Performance Implementation of the DCT and IDCT", IEE Electronics Letters, vol. 29, no. 7, pp. 602-603, 2003.

[Games, 1989] R.A.Games, D.Moulin, S.D.O'Neil and J.Rushanan, "Algebraic Integer Quantization and Residue Number System Processing", IEEE International Conference on Acoustics, Speech and Signal processing, pp. 948-951, May 1989.

[Jullien, 2003] M. Fu, G. A. Jullien, V. S. Dimitrov, M. Ahmadi and W. C. Miller, "The Application of 2D Algebraic Integer Encoding to a DCT IP Core", Proceedings of the 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, vol. 1, pp. 66-69, 2003.

[Knuth, 1981] D. Knuth, "The Art of Computer Programming", vol. 2 - Seminumerical Algorithms, 3rd edition, Addison Wesley, 1981.

[Meyer, 2001] U. Meyer-Base and F. Taylor, "Optimal Algebraic Integer Implementation with Application to Complex Frequency Sampling Filters", IEEE Transactions on Circuits and Systems -II: Analog and Digital Signal processing, vol. 48, no. 11, pp. 1078-82, 2001.

[Shams, 2002] A. Shams, W. Pan, A. Chidanandan and M. Bayoumi, "A Low Power High Performance Distributed DCT Architecture", Proceedings of IEEE Annual Symposium on VLSI, pp. 21-27, 2002.

[Wahid, 2003] K. Wahid, V. Dimitrov, G. Jullien and W. Badawy, "Error-Free Computation of Daubechies Wavelets for Image Compression Applications", IEE Electronics Letters, vol. 39, no. 5, pp. 428-429, March 2003.

[Wahid, 2004] Vassil Dimitrov, Khan Wahid and Graham Jullien, "Multiplication-Free 8x8 2D DCT Architecture using Algebraic Integer Encoding", IEE Electronics Letters, vol. 40, no. 20, pp. 1310-1311, 2004.

Authors' Information

Vassil Dimitrov – Dept. of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, Canada; e-mail: dimitrov@atips.ca

Khan Wahid – Dept. of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, Canada; e-mail: wahid@atips.ca