

Exploration of Tradeoffs in the Design of Integer Cosine Transforms for Image Compression

Levent Aksoy
INESC-ID
Lisboa, Portugal

Eduardo Costa
Universidade Catolica de Pelotas
Pelotas, Brazil

Paulo Flores
INESC-ID/IST TU Lisbon
Lisboa, Portugal

José Monteiro
INESC-ID/IST TU Lisbon
Lisboa, Portugal

Abstract—In image data compression, integer cosine transforms (ICTs) have been preferred to discrete cosine transforms (DCTs) due to their similar transform efficiency and lower implementation cost. However, there exist many alternative ICTs with different performance measures and implementation costs. In this work, we explore all possible ICTs, compute their performance measures, and find their implementation costs in terms of the number of adders/subtractors, where a state-of-art technique is used to realize ICTs under a shift-adds architecture. We also investigate the tradeoff between performance and implementation cost, present the pareto-optimal points of this tradeoff, and introduce promising ICTs that were not considered before.

I. INTRODUCTION

DCTs are widely used in image data compression due to their high efficiency. However, the elements of a DCT matrix are real numbers, leading to floating-point addition and multiplication operations that increase the complexity of a DCT design significantly. Although the floating-point numbers can be quantized to integers, for an order-16 (16×16) DCT, its elements have to be defined in at least 8 bits in order to guarantee that the image reconstruction errors are negligible [1]. Hence, the Walsh-Hadamard transform (WHT) [2] and C-matrix transform (CMT) [3], whose matrices consist of integer constants, have been used to reduce the hardware complexity of an image compression circuit. However, they were found to have significantly poor transform efficiency when compared to DCTs [4], [5]. Thus, the ICTs, that require integer operations as WHT and CMT, and that have similar properties to DCTs, were introduced in [4]–[6]. It was shown in [4] that there are 8×8 ICTs which have better performance measures than those of the 8×8 DCT. It was also indicated in [5] that the order-16 ICTs, which are represented with 6 bits, have a performance comparable to the order-16 DCT and yield a much simpler implementation. ICTs have also found applications in video coding [7].

The performance of ICTs in image data compression is clearly defined in terms of the transform efficiency (TE), maximum reducible bits (MRB), basis restriction error (BRE), and mean square errors (MSE) in images [4]. However, the implementation cost of an ICT is generally given in terms of the maximum bitwidth of integer constants in the ICT matrix, assuming that smaller integers lead to less complex designs [4], [5]. But, this metric does not provide an accurate information on the hardware complexity of an ICT design, and also, there exist many possible ICTs with the same bitwidth of constants. Hence, in this work, we consider the multiplierless realization of ICTs, where a recently proposed state-of-art algorithm [8] is used to find the fewest number

of adders/subtractors required to realize linear transforms. We explore the tradeoff between implementation cost in terms of the number of adders/subtractors used in the design of ICTs and performance in terms of TE and MRB. Furthermore, we introduce the pareto-optimal points related to the implementation cost-performance tradeoff and present a list of promising order-8 and -16 ICTs which were also synthesized at gate-level. It is shown that the availability of many ICTs leads to alternative circuits with different values of transformation performance and of gate-level area, delay, and power dissipation, and enables a designer to choose the one that fits best in a target application.

II. ORDER-8 AND -16 ICTS

A. Generation of Order-8 and -16 ICTs

The $n \times n$ DCT matrix D is an orthonormal matrix, i.e., $DD^T = I$, where I is the identity matrix, and its entries d_{ij} with $0 \leq j \leq n-1$ are determined as follows:

$$d_{ij} = \begin{cases} \sqrt{1/n} & i = 0 \\ (\sqrt{2/n}) \cdot \cos(\pi(j + 0.5)i/n) & 1 \leq i \leq n-1 \end{cases} \quad (1)$$

Fast algorithms for computing DCTs are given in [9], [10].

The $n \times n$ ICT matrix C is an orthogonal matrix, i.e., $CC^T = K$, where K is a diagonal matrix, and has the following properties [6]:

- 1) *Integer property*: Its entries c_{ij} , with $0 \leq i, j \leq n-1$, are integers.
- 2) *Orthogonality property*: Rows (or columns) of C are orthogonal.
- 3) *Relationship with DCT*:
 - a) $\text{sgn}(d_{ij}) = \text{sgn}(c_{ij})$ for $0 \leq i, j \leq n-1$.
 - b) If $d_{ij} = d_{rs}$, $c_{ij} = c_{rs}$ for $0 \leq i, j, r, s \leq n-1$

While the integer property eliminates the floating-point multiplication and addition operations, the orthogonality property ensures that the inverse of the ICT matrix has the same transform structure as the ICT. The relationships with DCT guarantee that the DCT structure is maintained in ICTs. Figure 1 presents the general forms of order-8 and -16 ICTs obtained based on the third property. The order-8 and -16 ICTs will respectively be denoted as $\text{ICT}(a, b, c, d, e, f)$ and $\text{ICT}(g, h, i, j, k, m, n, o, p, r, s, t, u)$ in this paper.

For the order-8 ICT matrix C_8 of Fig. 1, the following equality must be satisfied to ensure the orthogonality:

$$ab = ac + bd + cd \quad (2)$$

where $a \geq b \geq c \geq d$ and $e \geq f$ are the boundary conditions to keep a similar structure to the DCT [4], [6]. The elements e

$$C_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a & b & c & d & -d & -c & -b & -a \\ e & f & -f & -e & -e & -f & f & e \\ b & -d & -a & -c & c & a & d & -b \\ l & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ c & -a & d & b & -b & -d & a & -c \\ f & -e & e & -f & -f & e & -e & f \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix} \quad C_{16} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ g & h & i & j & k & m & n & o & -o & -n & -m & -k & -j & -i & -h & -g \\ p & q & r & s & -s & -r & -q & -p & -p & -q & -r & -s & s & r & q & p \\ h & k & o & -m & -i & -g & -j & -n & n & j & g & i & m & -o & -k & -h \\ t & u & -u & -t & -t & -u & u & t & t & u & -u & -t & -t & -u & u & t \\ i & o & -j & -h & -n & k & g & m & -m & -g & -k & n & h & j & -o & -i \\ q & -s & -p & -r & r & p & s & -q & -q & s & p & r & -r & -p & -s & q \\ j & -m & -h & o & g & n & -i & -k & k & i & -n & -g & -o & h & m & -j \\ l & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ k & -i & -n & g & -o & -h & m & j & -j & -m & h & o & -g & n & i & -k \\ r & -p & s & q & -q & -s & p & -r & -r & p & -s & -q & q & s & -p & r \\ m & -g & k & n & -h & j & o & -i & i & -o & -j & h & -n & -k & g & -m \\ u & -t & t & -u & -u & t & -t & u & u & -t & t & -u & -u & t & -t & u \\ n & -j & g & -i & m & o & -k & -h & h & k & -o & -m & i & -g & j & -n \\ s & -r & q & -p & p & -q & r & -s & -s & r & -q & p & -p & q & -r & s \\ o & -n & m & -k & j & -i & h & -g & g & -h & i & -j & k & -m & n & -o \end{bmatrix}$$

Fig. 1. General forms of 8×8 and 16×16 ICTs.

and f of C_8 are not present in its orthogonality condition and are respectively set to 3 and 1 due to higher TE values [4].

For the order-16 ICT matrix C_{16} of Fig. 1, the equalities, that must be satisfied for the orthogonality, are as follows:

$$gh + hk + io = jm + ik + gm + jn + no \quad (3)$$

$$gi + ho + km + gn + mo = ij + hj + kn \quad (4)$$

$$gj + gk + jo + mn = hm + hi + in + ko \quad (5)$$

$$pq = pr + qs + rs \quad (6)$$

where $g > h > i > j > k > m > n > o$, $p > q > r > s$, and $t > u$ are the boundary conditions [5]. Note that Eqn. 3-5 are related to the elements from g to o and Eqn. 6 includes only the elements p, q, r , and s . Similar to the order-8 ICTs, the elements t and u of C_{16} do not enter into the orthogonality conditions and are assigned to 3 and 1, respectively [5].

Table I presents the number of all valid order-8 and -16 ICTs when the maximum bitwidth of integer constants (bw) ranges from 4 to 8 and the elements of e and f of C_8 and the elements t and u of C_{16} are set to 3 and 1, respectively. As bw increases, the number of valid ICTs increases dramatically.

TABLE I. NUMBER OF ALL VALID ORDER-8 AND -16 ICTS.

bw	$\#C_8$	$\#C_{16}$
4	35	0
5	182	0
6	968	1,792
7	4,839	88,654
8	23,523	1,294,216

B. Performance Measures

In transform coding of images, TE is defined as the ability of a transform to decorrelate an input vector [5]. Let $n \times 1$ input vector x contains elements of samples from a one dimensional, zero mean, unit variance first order Markov process with correlation coefficient ρ . The elements of the $n \times n$ covariance matrix ϕ are determined as $\rho^{|i-j|}$ with $0 \leq i, j \leq n-1$. Let the $n \times n$ transform matrix be C and the transformed vector be y , $y = Cx$. Then, $E\{yy^T\} = CE\{xx^T\}C^T = C\phi C^T = B$ and TE is computed as follows:

$$TE = \frac{\sum_{i=0}^{n-1} |b_{ii}|}{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} |b_{ij}|} \quad (7)$$

where b_{ij} are the entries of the $n \times n$ matrix B .

Also, MRB [5], that is used to quantify the compression ability of a transform, is computed as follows:

$$MRB = -\frac{1}{2n} \sum_{i=0}^{n-1} \log_2 b_{ii} \quad (8)$$

III. MULTIPLIERLESS DESIGN OF LINEAR TRANSFORMS

Since the elements of an ICT matrix are determined beforehand and the realization of a multiplier in hardware is expensive in terms of area, delay, and power dissipation, the linear transforms, resulting from the multiplication of the constant matrix by a variable vector, can be implemented using only addition, subtraction, and shift operations. A straightforward approach for the multiplierless realization of constant multiplications, generally known as the digit-based recoding (DBR) method [11], is to define the constants in binary and for each 1 in the binary representation of the constant, to shift the variable and add up the shifted variables. As a simple example, consider the multiplication of a constant matrix by an input variable vector given in Fig. 2(a). The decomposed forms of linear transforms are given as follows:

$$\begin{aligned} y_0 &= 3x_0 + 11x_1 = (11)_{bin}x_0 + (1011)_{bin}x_1 \\ &= x_0 + x_0 \ll 1 + x_1 + x_1 \ll 1 + x_1 \ll 3 \\ y_1 &= 5x_0 + 13x_1 = (101)_{bin}x_0 + (1101)_{bin}x_1 \\ &= x_0 + x_0 \ll 2 + x_1 + x_1 \ll 2 + x_1 \ll 3 \end{aligned}$$

where a total of 8 operations are required (Fig. 2b).

However, the DBR technique does not exploit the sharing of partial products that can potentially reduce the number of operations. The prominent algorithms [8], [12] aim to maximize the sharing of partial products. The hybrid algorithm [8] combines an efficient common subexpression elimination (CSE) heuristic and the graph-based (GB) difference method. It iteratively finds alternative realizations of linear transforms using the GB difference method and applies a CSE heuristic to further reduce the complexity by sharing the common subexpressions. Hence, in the hybrid algorithm, the main drawback of a CSE algorithm, *i.e.*, its limitation to a number representation, is partially eliminated by using a GB algorithm and the main drawback of a GB algorithm, *i.e.*, its time-consuming search process, is partially decreased by using a CSE heuristic. Moreover, it is equipped with some hardware optimization techniques that take into account the type of the operation (addition or subtraction) and the size of input operands. Although it obtains good solutions in terms of the number of operations, leading to low-complexity designs at gate-level, its solutions can be realized with a large number of operations in series (generally known as the number of adder-steps), yielding designs with a large delay. To overcome this disadvantage, its modified version, that can find a solution with the fewest number of operations under a delay constraint, was also introduced in [8]. It was shown that they can

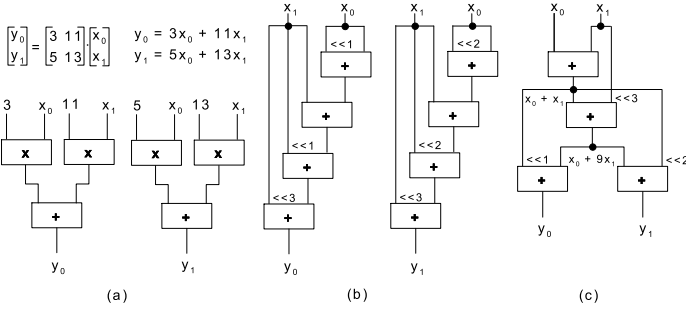


Fig. 2. (a) Direct realization of linear transforms $y_0 = 3x_0 + 11x_1$ and $y_1 = 5x_0 + 13x_1$; Their shift-adds implementations: (b) without partial product sharing; (c) with partial product sharing.

find significantly better solutions at both high-level and gate-level than previously proposed algorithms. Returning to our example in Fig. 2(a), the hybrid algorithm obtains a solution with 4 operations and 3 adder-steps by finding the common subexpressions $x_0 + x_1$ and $x_0 + 9x_1$ when constants are defined under binary and there is no delay constraint (Fig. 2c).

In [7], a 16×16 ICT was designed using two 8×8 integer transforms, where constant multiplications can be realized using a single adder/subtractor. However, the sharing of partial products and gate-level area optimization as used in the algorithm of [8] were not considered in [7].

IV. EXPERIMENTAL RESULTS

We developed an exhaustive search method, that determines the elements of all valid order-8 and -16 ICT matrices, satisfying the conditions given in Section II when the maximum bitwidth of constants (bw) was set to 6. This is simply due to a large number of possible ICTs when bw is greater than 6 as shown in Table I and there exist 6-bit ICTs with performance values comparable to DCTs. After a valid ICT matrix was obtained, we computed its TE and MRB values as given in Eqn. 7-8 when ρ was 0.95 as used in [4], [5], [7], which indicates that the input data is very highly correlated. We found the implementation cost of an ICT using the hybrid algorithm of [8] when constants were defined under the canonical signed digit representation [11]. Furthermore, among all valid ICTs, we chose the promising ones for further analysis. In this selection, we favored the ones with the maximum TE and MRB values and with the minimum number of operations. When there exist more than one ICT with the best value on one metric, their values on other metrics are considered. These promising ICTs were also implemented under a delay constraint set to the minimum adder-steps of ICTs, as described in [8], if this minimum value was less than that of the solution found by the hybrid algorithm [8] without a delay constraint. These ICTs were described in VHDL and synthesized using Synopsys Design Compiler with UMCLogic 180nm Generic II library when the bitwidths of variables in the input vector were 16. Their VHDL codes and additional information on the implementation cost-performance tradeoff are available at <http://algorithms.inesc-id.pt/~levent/icts.html>.

Over all valid 968 order-8 ICTs, the minimum values of TE, MRB, and the number of operations were determined as 0.8372, 1.2812, and 30, respectively. The maximum values of these metrics were found as 0.9417, 1.4642, and 50, respectively. Table II presents the parameters of the selected order-8 ICTs, where ICT_{81} has the only maximum TE value, ICT_{82} has the maximum MRB value when the number of

TABLE II. PARAMETERS OF SELECTED ORDER-8 ICTS.

Index	ICT($a, b, c, d, 3, 1$)
ICT_{81}	45, 39, 26, 9
ICT_{82}	35, 30, 20, 7
ICT_{83}	12, 10, 6, 3
ICT_{84}	10, 9, 6, 2

TABLE III. SUMMARY OF HIGH-LEVEL AND GATE-LEVEL RESULTS OF ORDER-8 DCT, SELECTED ICTS, CMT, AND WHT.

Linear Transform	High-level				Gate-level		
	TE	MRB	op	as	area	delay	power
DCT	0.9399	1.4660	54 54	6 5	27.5 28.2	6.7 6.8	84.4 92.0
ICT_{81}	0.9417	1.4642	49 49	7 5	24.2 24.6	6.8 6.7	72.9 74.5
ICT_{82}	0.9414	1.4642	42 42	6 5	20.3 20.8	6.3 6.4	58.3 60.7
ICT_{83}	0.9298	1.4588	30 30	4 4	14.4 14.4	5.4 5.4	40.6 40.6
ICT_{84}	0.9409	1.4640	38 38	5 4	17.8 18.1	5.6 5.6	49.2 50.4
CMT	0.9185	1.4447	44 44	7 5	22.4 22.4	6.2 6.2	74.3 73.5
WHT	0.8531	1.3198	24 24	3 3	11.2 11.2	4.2 4.2	26.4 26.4

operations is minimum and TE is maximum, and ICT_{83} has the fewest number of operations with the maximum TE value. ICT_{84} was also chosen due to its good tradeoff between TE, MRB, and the number of operations.

The high-level results on order-8 DCT, selected ICTs, CMT, and WHT, are given in Table III, where op and as denote the number of operations and the number of adder-steps, respectively. The italic results were obtained by the hybrid algorithm [8] under a delay constraint that was set to the minimum adder-steps of ICTs. Their gate-level results are also presented in this table, where $area$, $delay$, and $power$ stand for the area in mm^2 , the critical path delay in ns , and the power dissipation estimation in mW , respectively. Note that the TE and MRB values of DCT were computed based on its floating-point elements and its op , as , and gate-level results were found after its elements were converted to 8-bit fixed-point values to respect the quantization requirements [1].

First, consider the results of linear transforms obtained based on the solutions of the algorithm [8] without a delay constraint. It is observed that although ICT_{81} and ICT_{82} have similar TE and MRB values, the difference of the number of operations between these ICTs is 7, which yields a 16.9% area reduction at gate-level. Also, the area reduction between the designs of ICT_{81} and ICT_{83} is obtained as 40.7% at a cost of a slight decrease in the performance of ICT_{83} . Moreover, it is clearly shown that the reduction on the number of operations in ICTs has also a significant impact on delay and power dissipation of the ICT design due to the reduction of area at gate-level. In addition to ICT_{81} , ICT_{82} , and ICT_{83} , that present the pareto-optimal points in TE, MRB, and the number of operations, respectively, ICT_{84} has promising performance values close to those of ICT_{81} and ICT_{82} with significant reductions at gate-level results and has better performance values than those of ICT_{83} with a slight increase at gate-level complexity. This observation implies that using an exhaustive search method, which incorporates an algorithm for the multiplierless realization of ICTs, a designer can find an ICT design that best fits in an application, taking into account the performance and implementation cost of an ICT. On the other hand, the order-8 DCT has a TE value worse than that of ICT_{81} and ICT_{82} and the best MRB value among all

TABLE IV. PARAMETERS OF SELECTED ORDER-16 ICTs.

Index	ICT($g, h, i, j, k, m, n, o, p, q, r, s, 3, 1$)
ICT_{16_1}	62, 61, 49, 47, 37, 31, 21, 5, 60, 51, 34, 12
ICT_{16_2}	42, 38, 37, 32, 22, 19, 10, 4, 60, 51, 34, 12
ICT_{16_3}	42, 38, 37, 32, 22, 19, 10, 4, 12, 10, 6, 3
ICT_{16_4}	42, 38, 37, 32, 22, 19, 10, 4, 15, 12, 8, 3

TABLE V. SUMMARY OF HIGH-LEVEL AND GATE-LEVEL RESULTS OF ORDER-16 DCT, SELECTED ICTs, AND WHT.

Linear Transform	High-level				Gate-level		
	TE	MRB	op	as	area	delay	power
DCT	0.8845	1.5705	178 178	11 6	92.4 94.4	8.7 8.1	358.3 368.7
ICT_{16_1}	0.8656	1.5619	157 157	9 6	80.3 81.3	8.7 8.2	289.5 303.8
ICT_{16_2}	0.8642	1.5623	145 145	9 6	74.0 75.5	8.3 7.9	263.9 274.4
ICT_{16_3}	0.8586	1.5595	133 133	9 6	67.6 68.4	8.3 7.9	237.0 241.4
ICT_{16_4}	0.8640	1.5618	139 139	9 6	70.9 71.7	8.3 7.9	249.3 253.6
WHT	0.7065	1.3610	64	4	30.6	4.9	94.2

linear transforms. However, its design has the worst gate-level area and power dissipation results due to a large number of operations. The CMT has worse performance results than the selected ICTs and its design occupies larger area than those of ICTs, except ICT_{8_1} . While the WHT design has the least gate-level complexity, it has the poorest performance.

Second, when the number of adder-steps is reduced in these linear transforms, the number of operations is not changed with respect to those of solutions found without a delay constraint. In this case, the area of designs is increased, except on CMT, but their delay is not changed significantly due to a small decrease in the number of adder-steps.

Over all valid 1792 order-16 ICTs, the minimum values of TE, MRB, and the number of operations were 0.7873, 1.4666, and 133, respectively and the maximum values of these metrics were 0.8656, 1.5623, and 165, respectively. Compared to the order-8 ICTs, the ranges between the maximum and minimum values in TE, MRB, and the number of operations increase in the order-16 ICTs. Table IV presents the selected ICTs based on their TE, MRB, and the number of operations. Note that ICT_{16_1} has the maximum TE value with the minimum number of operations, ICT_{16_2} has the maximum MRB value when the number of operations is minimum and TE is maximum, and ICT_{16_3} has the fewest number of operations with the maximum TE and MRB values. ICT_{16_4} was chosen due to its good tradeoff between these metrics.

Table V presents the high-level and gate-level results of the selected order-16 ICTs given in Table IV and of the order-16 DCT and WHT. Again, the *op*, *as*, and gate-level results of DCT were obtained after its floating-point elements were converted to 8-bit integers [1].

As can be observed from Table V, ICT_{16_3} , which requires the minimum number of operations and has TE and MRB values close to their maximum values, has the least design complexity among the ICTs. Note that the area reduction between the designs of ICT_{16_1} and ICT_{16_3} obtained by the hybrid algorithm without the delay constraint is 15.8% and this value between ICT_{16_2} and ICT_{16_3} is 8.6%. Also, ICT_{16_4} presents performance values similar to those of ICT_{16_1} and ICT_{16_2} and has better results at gate-level than these ICTs. In turn, the order-16 DCT shows the best performance, but its

design occupies the largest area. The WHT has the worst TE and MRB values that are far away from those of the selected ICTs. However, its design has the least hardware complexity.

Furthermore, observe from Table V that the reduction of the number of adder-steps increases the gate-level area slightly, but reduces the delay significantly, where the maximum delay reduction is obtained as 6.8% on DCT, which is due to a large decrease in the number of adder-steps.

V. CONCLUSIONS

This paper explored the implementation cost-performance tradeoff in the multiplierless design of ICTs for image data compression. The implementation cost of an ICT design was defined in terms of the number of adders/subtractors and was determined by a recently proposed state-of-art algorithm. It also introduced promising order-8 and -16 ICTs with high performance measures and less gate-level complexity that were not considered before. It was observed that there are many possible ICTs with similar performances, but requiring different number of operations and gate-level area.

VI. ACKNOWLEDGMENTS

This work was partially supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under the project "ParSat: Parallel Satisfiability Algorithms and its Applications" PTDC/EIA-EIA/103532/2008 and the project PEst-OE/EEI/LA0021/2013.

REFERENCES

- [1] M. Guglielmo, "An Analysis of Error Behavior in the Implementation of 2-D Orthogonal Transformations," *IEEE Tran. on Communications*, vol. 34, no. 9, pp. 973–975, 1986.
- [2] B. J. Fino and V. R. Algazi, "Unified Matrix Treatment of the Fast Walsh-Hadamard Transform," *IEEE Tran. on Computers*, vol. 25, no. 11, pp. 1142–1146, 1976.
- [3] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. New York: Springer, 1976.
- [4] W.-K. Cham, "Development of Integer Cosine Transforms by the Principle of Dyadic Symmetry," *IEE Proc. I Communications, Speech and Vision*, vol. 136, no. 4, pp. 276–282, 1989.
- [5] W.-K. Cham and Y.-T. Chan, "An Order-16 Integer Cosine Transform," *IEEE Tran. on Signal Processing*, vol. 39, no. 5, pp. 1205–1208, 1991.
- [6] K.-M. Cheung, F. Pollara, and M. Shahshahani, "Integer Cosine Transform for Image Compression," Jet Propulsion Laboratory, California, Pasadena, The Telecommunications and Data Acquisition Progress Report 42-105, 1991.
- [7] J. Dong and K. N. Ngan, "16 × 16 Integer Cosine Transform for HD Video Coding," in *Proc. of the 7th Pacific Rim Conference on Advances in Multimedia Information Processing*, 2006, pp. 114–121.
- [8] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Multiplierless Design of Linear DSP Transforms," in *VLSI-SoC: Advanced Research for Systems on Chip*, S. Mir, C.-Y. Tsui, R. Reis, and O. Choy, Eds. Springer, 2012, ch. 5, pp. 73–93.
- [9] W. Chen, C. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. on Communications*, vol. 25, no. 9, pp. 1004–1009, 1977.
- [10] E. Feig and S. Winograd, "Fast Algorithms for the Discrete Cosine Transform," *IEEE Trans. on Signal Processing*, vol. 40, no. 9, pp. 2174–2193, 1992.
- [11] M. Ercegovac and T. Lang, *Digital Arithmetic*. Morgan Kaufmann, 2003.
- [12] N. Boullis and A. Tisserand, "Some Optimizations of Hardware Multiplication by Constant Matrices," *IEEE Trans. on Computers*, vol. 54, no. 10, pp. 1271–1282, 2005.