IMPLEMENTATION OF TWO-DIMENSIONAL DISCRETE COSINE TRANSFORM AND ITS INVERSE

Jari Nikara, Rami Rosendahl, Konsta Punkka, and Jarmo Takala

Institute of Digital and Computer Systems, Tampere University of Technology,
P.O.Box 553, FIN-33101 Tampere, Finland
Email: {jari.nikara, rami.rosendahl, konsta.punkka, jarmo.takala}@tut.fi

ABSTRACT

In this paper, the implementation of a unified 8×8 discrete cosine transform (DCT) and its inverse is described. First, the accuracy of the structure that has been reported earlier is analyzed with Matlab in order to have internal word length requirements for the implementation. Then, the structure is modeled as a data path structure with Synopsys Module Compiler. When synthesizing the model with 19-bit internal word length onto 0.11 μm CMOS technology, the resulting pipeline exhibits an operation frequency of 253 MHz and uses 40 000 equivalent gates. The latency for both transforms is 94 cycles. Finally, the comparison to another unified pipeline structure reveals up to 15% smaller estimated area.

1. INTRODUCTION

The discrete cosine transform (DCT) has been regarded as one of the best tools in digital signal processing and thus, it has many applications in the area of multimedia. In image and video processing, especially two-dimensional (2-D) DCT and its inverse have gained popularity due to good energy compaction properties. Another essential property of the DCT is separability which allows the decomposition of the multidimensional transform into successive one-dimensional (1-D) transforms. Applying such a decomposition for the 2-D DCT is known as a row-column approach. An alternative approach is to compute DCT directly over 2-D data, which is referred as a direct method.

In general, the direct method is considered to produce algorithms with lower arithmetic complexity and especially with lower number of multiplications than the row-column approach. However, the resulting control complexity with the row-column approach is lower implying regularity and modularity. Hence, the row-column approach has gained popularity in VLSI realizations. Nevertheless, it is possible to obtain regular algorithms with the direct method but at the expense of arithmetic complexity. However, the regularity allowing the efficient exploitation of linear mapping methods may be preferred for area-efficient implementations. Moreover, the noise behaviour of the implementation reflects area efficiency, i.e., better noise behaviour implies shorter word length resulting in savings in area.

In principle, a unified structure can be constructed by providing additional data path to reverse the data flow of the DCT for IDCT computation as described, e.g., in [1]. Such an approach will, however, introduce high routing costs and complicated control. Therefore, it is desirable that both the DCT and IDCT share a common data path. In addition, the pre- or post-processing is disadvantageous when targeting at unified VLSI realizations; the pre-processing in forward transform results in post-processing in inverse transform and vice versa. Therefore, in the unified approach both pre- and post-processing needs to be realized implying additional hardware cost, e.g., in [2].

In this paper, the VLSI implementation of the unified DCT/IDCT is described. The design is based on the structure that has been reported in [3]. To summarize, the main contributions of the paper are the following:

- *Accuracy analysis* of the structure proposed in [3] with respect to IEEE Std. 1180-1990 [4], which shows that the structure requires the internal word length of 19 bits with rounding.
- *Implementation* of the DCT/IDCT pipeline, where the structure is modeled as a 19-bit data path with Synopsys Module Compiler and synthesized onto 0.11 μm standard cell CMOS technology. The resulting implementation requires 40 000 gates operating at 253 MHz and possess the latency of 94 cycles.
- *Comparison*, which reveals that in terms of gate count, the structure provides up to 15% better area efficiency than the reference.

The remaining of the discussion is organized as follows. First, the unified DCT/IDCT structure in [3] is introduced briefly in Section 2. In Section 3, the required internal word length is determined for the implementation which is described in Section 4. In Section 5, the structure is compared to another pipeline structure. Finally, the discussion is concluded in Section 6.

2. STRUCTURE

The pipeline implementation is based on the perfect shuffle topology algorithms and structures proposed earlier in [3]. However, let us next outline briefly the structural derivation. The 1-D DCT and IDCT algorithms are depicted in Fig. 1(a) and (b), respectively. The algorithms contain processing columns represented with the aid of matrices $Q_8^{(s)}$ of order eight. Correspondingly, depending on the stage s, processing columns are comprised of stages of butterflies F_8, multiplications $N_8^{(s)}$, local subtractions $M_8^{(s)}$, and local exchanges $H_8^{(s)}$. The processing columns are interconnected with perfect shuffle permutations [5]. The corresponding 2-D algorithms resemble 1-D algorithms; the 2-D algorithms have the same topology but the computation is performed directly over 2-D data in an array form. Consequently, each column is followed by another column with the same operation but with the operands eight times more apart. In general,

This work is supported by the Academy of Finland (project no. 50554), Nokia Foundation, and Jenny and Antti Wihuri Foundation.

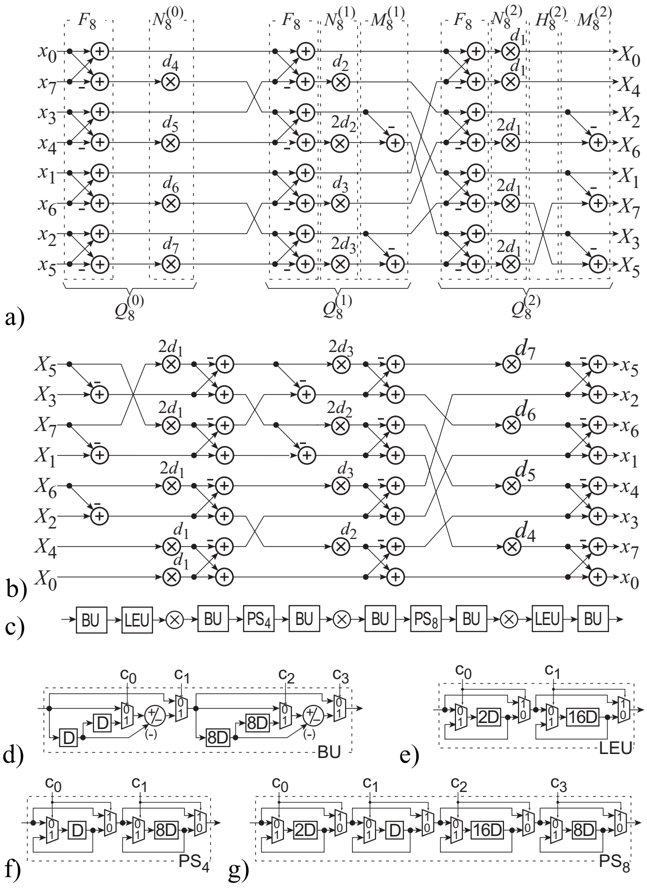

Figure 1: Signal flow graphs of 8-point (a) DCT and (b) IDCT and block diagrams of (c) 8 × 8 DCT/IDCT pipeline, (d) butterfly unit (BU), (e) local exchange unit (LEU), (f) 4-point perfect shuffle permutation (PS$_4$), and (g) 8-point perfect shuffle permutation (PS$_8$). $d_1 = \sqrt{0.5}, d_{2i} = \sqrt{0.5(1+d_i)}, d_{2i+1} = \sqrt{0.5(1-d_i)}$. KD: shift register of size K. c$_k$: control signal.

e.g., the butterfly stage in the 2-D algorithm can be formulated with the aid of Kronecker's product as

$$F_N \otimes F_N = (F_N \otimes I_N)(I_N \otimes F_N). \qquad (1)$$

The 2-D algorithms are mapped vertically onto a fully sequential pipeline structure that supports both transforms. The block diagram of the resulting structure without pipeline registers is illustrated in Fig. 1(c). In short, each stage is mapped onto corresponding unit and the final structure is a fully sequential data path constructed out of butterfly units (BU) in Fig. 1(d), multipliers and sequential permutation networks, i.e., local exchange unit (LEU) in Fig. 1(e), 4-point perfect shuffle-based interconnection (PS4) in Fig. 1(f), and 8-point perfect shuffle-based interconnection (PS8) in Fig. 1(g). The BU is capable of computing two successive radix-2 butterflies, first over consecutive samples and then operands eight samples apart. In addition, the BU can be bypassed with the aid of multiplexer. Therefore, the BU is capable of performing irregular subtractions in the algorithm, which can be interpreted as halves of a butterfly. The multiplier manages the column of the multiplications. All the required permutations are performed with sequential permutation networks, which are based on shift-exchange units [6]. Finally, by providing appropriate control for the pipeline, it computes either 8 × 8 DCT or IDCT.

3. ACCURACY ANALYSIS

In general, the fixed-point arithmetic is employed for speed and area efficiency of the design. However, a fixed-point representation introduces an accuracy problem due to the finite word length. Consequently, the signals must be quantized to the given word length and can be represented only in finite precision. Another issue related to finite word length is to keep the signal values within numeric range during the arithmetic operations, i.e., to avoid overflows. The overflows can be avoided with the aid of scaling, which means adjusting the signal level with a proper scaling factors. However, if the number of bits for representing the coefficients is not increased the scaling causes lost in precision, i.e., overflows are avoided with the expense of signal-to-noise-ratio (SNR). From the implementation point of view, the truncation of two's complement, i.e., neglecting the lowest bits, is the cheapest quantization method. On the other hand, although being slightly more complex and area demanding method, the rounding is associated with the smallest errors, thus requiring shorter word length. In any case, the required internal word length for fulfilling the specified accuracy standard has to be determined before the implementation.

3.1 Simulation Model

For the accuracy analysis, the parameterizable simulation model of the pipeline structure is modeled with C-language. While the structure has been achieved by mapping the operations of the 2-D signal flow graph onto 1-D sequential data path, the simulation model is formed vice versa. In other words, since every sample is fed through the units, e.g., multiplier, the corresponding capabilities are available for every sample in simulation model. On the other hand, the units that do not affect to accuracy, e.g., permutations, can be modeled trivially by indexing. The coefficients at the same multiplication stage are given in the same format, i.e., the same number of digit and fractional bits. With such an arrangement, the significant bits are always selected from the same location at the multiplier on hardware, thus simplifying the control.

The result of each arithmetic operation is checked for possible overflow. If the overflow is detected, the signal level is decreased by scaling, which can be performed either by shifting or multiplying. Consequently, the scaling factors are limited to be powers of two with shifting while by multiplying the scaling factors can be selected with finer resolution. In the simulation model, the shifting corresponding rewiring on hardware can be utilized between every stage. For not introducing any extra hardware, the signals are always quantized with truncation in scaling by shifting. On the other hand, there is no reason to scale by shifting if there is multiplier just before the overflowing stage. The proper scaling factor to maximize the signal level is determined from the relative peak value at the overflow location.

In order to maintain the correct signal level at the output, the internal scalings performed have to be compensated. The compensation can be performed at the last multiplication stage and/or at the output depending on the used scaling factors. Let us remark that in fixed-point notation, the loc-

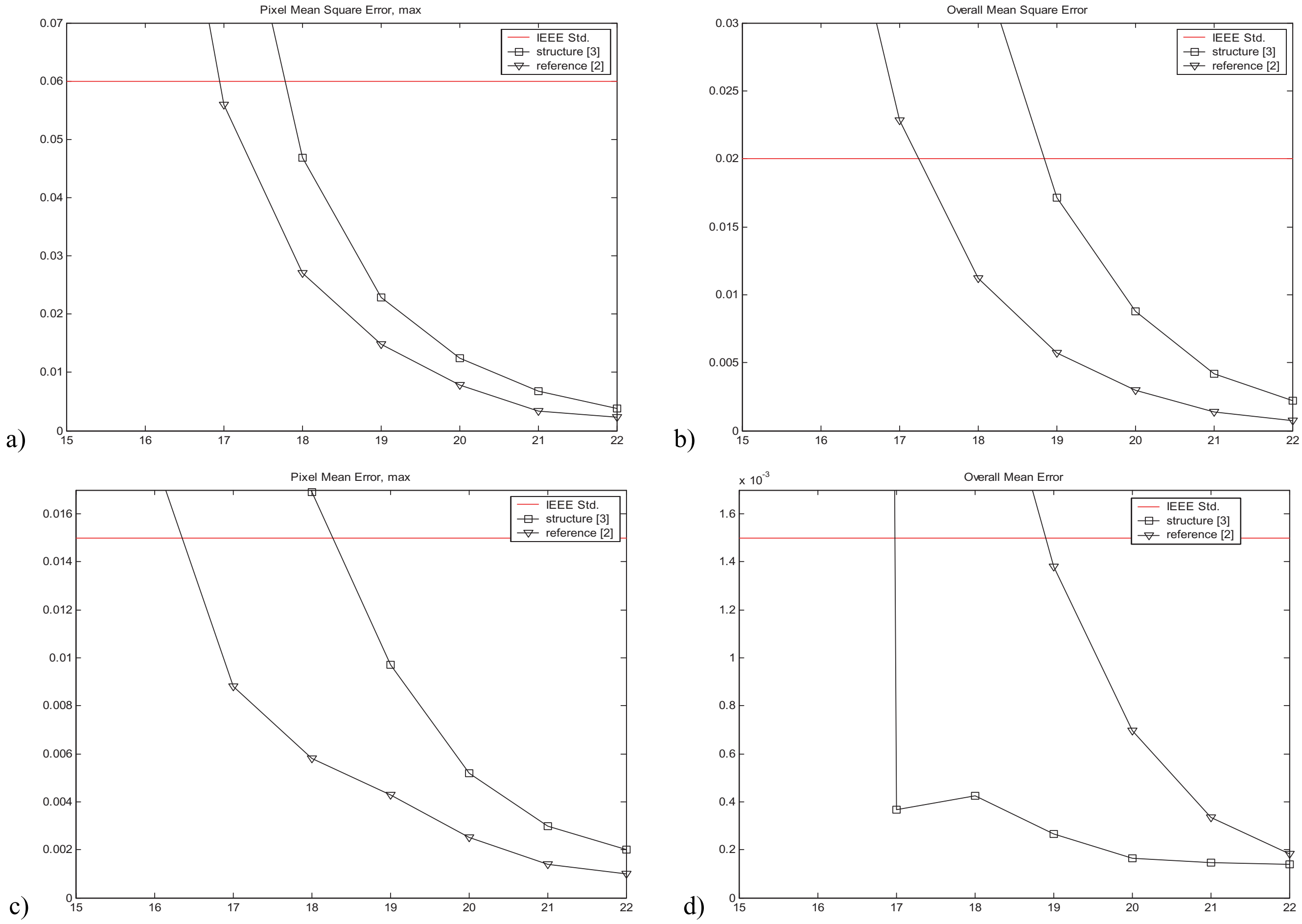

Figure 2: Results of the accuracy analysis: (a) pixel mean square error, (b) overall mean square error, (c) pixel mean error, and (d) overall mean error.

ation of the radix point does not affect the functionality of arithmetic unit but it is for correct interpretation of binary-valued vector. Therefore, the powers of two scalings can be compensated at the output as shifting the radix point. Otherwise the compensation is performed at the last multiplier stage.

3.2 Experimental Results

Before going into discussion on experiments, let us remark that in general, fully sequential unified DCT/IDCT pipeline structures based on direct computation over two dimensional data are really rarely reported. However, based on the complexity estimates presented in [3], the pipeline structure represents state-of-the-art from the hardware complexity point of view. Therefore, in this paper, the structure [3] in Fig. 1(c) is compared only to the most relevant reference design presented in [2]. In order to be fair later in comparison, the accuracy analysis for the chosen reference structure is analyzed with the exactly similar procedure.

The accuracy analyses for the structure [3] and the reference [2] are performed against IEEE specification [4] with Matlab. The simulations are performed with different word lengths and using rounding as a quantization method. The coefficients are given as long as internal word length but rounded down in magnitude. The results of the accuracy analysis are shown in Fig. 2. Briefly, the pixel mean square error (pmse), overall mean square error (omse), pixel mean error (pme), and overall mean error (ome) are represented as a function of the internal word length. According to simulations, the internal word length of 19 bits is required for both structures in order to fulfill the standard. For the structure [3], the limiting error values are omse and pme while the word length of the reference [2] is defined by ome. Furthermore, the requirements for pixel peak error (ppe) and all-zero input are met with 19 bits in both structures. Finally, let us remark that although the pmse, omse, and pme are slightly larger in the structure [3] than in the reference [2], ome is smaller in structure [3] due to fact that the errors are not biased, i.e., the negative and positive errors cancel each others. Instead, in the reference [2], the errors are positive biased.

4. IMPLEMENTATION

The structure in Fig. 1(a) has been described as a data path with Module Compiler Language and synthesized with Synopsys Module Compiler onto a 0.11 μm standard cell CMOS technology. The adder/subtracters in BU have been implemented as carry-look-ahead (CLA) adders, because the data width being 19 bits CLA type of adders are the fastest and the

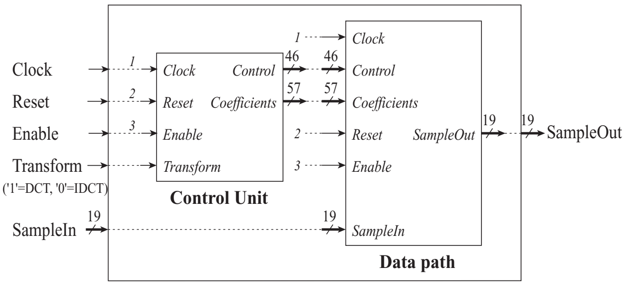

Figure 3: Block diagram of implementation.

area of CLA-adders is only slightly larger than the area of 19-bit ripple-carry adders. Multipliers have been implemented as shift add multipliers since in the target technology, Booth multiplier wouldn't provide any advantage with 19-bit word length. With the aid of the multiplexer, which is in parallel with the multiplier, the multiplier can be bypassed. Therefore, the multiplication with one can be performed despite of the fractional number representation. The data permutations correspond exactly to block diagrams depicted in Fig. 1(e)-(g). The characteristics of the implementation are summarized into Table 1.

Table 1: Characteristics of 8 × 8 DCT/IDCT pipeline.

Technology	0.11 μm CMOS
Function	8×8 DCT or IDCT
Internal word length	19 bits
Frequency	253 MHz
Latency	94 cycles
Gate count	39 424

The functionality of the data path has been verified at a register transfer level and gate level with the aid of structure illustrated in Fig. 3. Since Module Compiler enables optimization of high performance data path captured at the structural level of abstraction but is not suitable for synthesizing general logic, e.g., random Boolean logic or state machines, the control for the DCT/IDCT data-path is described with VHDL. The control is realized trivially as 64-state state machine, which generates all the control signal and coefficients to the data path.

5. COMPARISON

In order to be fair in comparison, the complexity of the structures is first determined in terms of the number of arithmetic units, multiplexers, and delay registers. All the multiplexers are estimated as equivalent 2-to-1 multiplexers and pipeline registers are not included. In addition, all resources required for implementation are assumed to be similar from the same standard cell technology. The assumed gate counts of area and speed optimized resources with different word lengths are depicted in Table 2. Altogether, the presented implementation exhibits improvement in estimated gate count as summarized in Table 3. Let us remark that both structures can be designed for the same throughput rates due to arithmetic units are not located in feedback loops.

Table 2: Gate count estimates for the basic units.

	Area optimized / Speed optimized			
W	×	+/−	R	M
19	2088 / 3288	290 / 1279	39 / 151	95 / 109

×: Multiplier. +/−: Add/subtract unit. R: Register. M: 2-to-1 multiplexer.

Table 3: Comparison of unified DCT/IDCT structures.

Structure	×	+/−	R	M	W	Gate Count
[2]	4	14	216	18	19	33 634 / 57 320
proposed	3	12	180	44	19	28 560 / 51 476

×: Number of multipliers. +/−: Number of add/subtract units. R: Number of registers. M: Number of 2-to-1 multiplexers. W: Required word length

6. CONCLUSIONS

In this paper, a pipeline structure supporting both 8 × 8 DCT and IDCT has been analyzed and implemented. In order to fulfill IEEE accuracy standard specified for the IDCT, 19-bit internal word length is required. By synthesizing the structure with Synopsys Module Compiler onto 0.11 μm standard cell CMOS technology, the clock frequency of 253 MHz has been achieved. When comparing the implementation to the most relevant reference, it provides 10-15% better area efficiency in terms of estimated gate count.

REFERENCES

[1] K.-H. Cheng, C.-S. Huang, and C.-P. Lin, "The design and implementation of DCT/IDCT chip with novel architecture," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, May 28–31 2000, pp. 741–744.

[2] S.-F. Hsiao and J.-M. Tseng, "New matrix formulation for two-dimensional DCT/IDCT computation and its distributed-memory VLSI implementation," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 149, no. 2, pp. 97–107, Apr. 2002.

[3] J. Takala, J. Nikara, and K. Punkka, "Pipeline architecture for two-dimensional discrete cosine transform and its inverse," in *Proceedings of the 9th IEEE International Conference on Electronics, Circuits and Systems*, vol. 3, Dubrovnik, Croatia, Sept. 15–18 2002, pp. 947 – 950.

[4] IEEE Std 1180-1990, "IEEE standard specification for the implementations of 8x8 inverse discrete cosine transform," Institute of Electrical and Electronics Engineers, New York, USA, International Standard, Dec. 1990.

[5] H. Stone, "Parallel processing with perfect shuffle," *IEEE Trans. Comput.*, vol. 20, no. 2, pp. 153–161, Feb. 1971.

[6] C. B. Shung, H.-D. Lin, R. Cypher, P. H. Siegel, and H. K. Thapar, "Area-efficient architectures for the Viterbi algorithm I. Theory," *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 636–644, Apr. 1993.