Pipeline Architecture for 8×8 IDCT with Fixed-Point Error Analysis

Jari A. Nikara, Rami J. Rosendahl, and Jarmo H. Takala

Tampere University of Technology
Digital and Computer Systems Laboratory,
P.O.B. 553, FIN-33101 TAMPERE, FINLAND
E-mail: jari.nikara@tut.fi, rami.rosendahl@tut.fi, jarmo.takala@tut.fi

ABSTRACT

In this paper, a sequential architecture for 8×8 inverse discrete cosine transform (IDCT) based on row-column decomposition is described. The sequential one-dimensional IDCT kernel is derived by utilizing vertical projection to fast IDCT algorithm. The matrix transposition network is realized with a register-based sequential permutation network and the resulting modular two-dimensional architecture can be freely pipelined. Moreover, the accuracy of the proposed architecture is analyzed in order to fulfil the IEEE standard for 8×8 *IDCT.*

1. INTRODUCTION

Discrete cosine transform (DCT) and its inverse (IDCT) are widely used tools in digital signal processing. Several architectures for DCT and/or IDCT implementations have been proposed for multimedia purposes. Typically high speed operation is achieved with the aid of parallelism. In principle, parallel architectures can be developed by exploiting inherent spatial and/or temporal parallelism in fast algorithms for DCT and IDCT. However, such algorithms are often irregular, which may limit the exploitation level of the parallelism. In addition to high data rates, the accuracy of the implementation is important; e.g., IEEE Standard 1180-1990 [1] defines accuracy requirements for two-dimensional 8×8 IDCT implementations.

Direct mapping of algorithm will result in architecture with both spatial and temporal parallelism. In general, the cost of the implementation should be low, i.e., the resources, especially number of arithmetic units, in the architecture should be minimized. Exploitation of spatial parallelism results in column architectures where operands are fed into the architecture in parallel. The arithmetic units are recursively used to compute the entire transform. Exploitation of temporal parallelism, in turn, results in pipeline architectures (or systolic array) where data is fed into the architecture sequentially. For this purpose the linear array processor approach described in [2] can be used.

In this paper, a sequential two-dimensional IDCT architecture is presented utilizing the principles used in architectural derivation of fast Fourier transform [2]. Vertical projection is applied to signal flow graph of IDCT, which results in cascaded one-dimensional IDCT architecture. The row-column decomposition is used for constructing two-dimensional transform. The required matrix transposition network is sequential and register-based with optimal number of registers. Due to the loop free structure, the architecture can be freely pipelined for improving throughput. Furthermore, the internal word width requirements are determined and analyzed for reaching the IEEE standard [1].

2. ARCHITECTURE

Architectural derivation is based on rescheduled constant geometry DCT algorithm of type II presented earlier in [3]. Since the DCT is orthogonal transform, the corresponding signal flow graph of IDCT in Fig. 1 is achieved by transposing the signal flow graph of the DCT. In addition, the signal flow graph is flipped in order to have the operands for each operation available when the result is needed. Such an arrangement offers an advantage in serial realization; every sample is not delayed in implementation thus decreasing the latency. The coefficients d_i can be generated recursively as

$$d_1 = \sqrt{\frac{1}{2}}, \quad d_{2i} = \sqrt{\frac{(1+d_i)}{2}},$$
$$d_{2i+1} = \sqrt{\frac{(1-d_i)}{2}} \qquad (1)$$

In order to reduce the dimensionality of the signal flow graph, the vertical projection [4] is applied to the operational stages in Fig. 1; the stages are collapsed into a one dimension resulting in basic sequential blocks. In order to

The first author acknowledges Nokia Foundation for financial support.

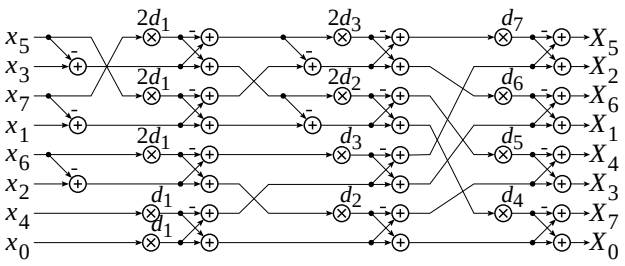

Fig. 1. Signal flow graph of fast algorithm for inverse 8-point DCT-II.

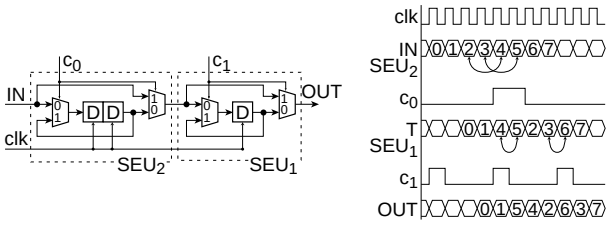

Fig. 3. Block diagram and timing diagram of sequential permutation network of 8-point perfect shuffle permutation. D: Delay register. c_k: control signal.

guarantee correct operation, the causality, i.e., the order of computation should be concerned. Furthermore, the resulting processing element realizing only one operation simultaneously introduces the requirement of unambiguity.

The operational stages of IDCT algorithm in Fig. 1 are actually similar to stages in DCT algorithm in [3]. Thus, the sequential basic blocks introduced in [3] can be utilized for realizing the sequential IDCT kernel. The basic data processing blocks needed in addition to multiplier are butterfly unit and local subtraction unit, which is capable of performing the first operational stage of IDCT. The block diagrams of the blocks are depicted in Fig. 2 (a) and (b).

The functionality of processing blocks in Fig. 2 can be explained as follows. In order to compute both operations of butterfly, subtraction and addition, each sample is stored for two sample periods introducing two storage elements into butterfly unit. The computation of operations requires one arithmetic unit that can be controlled to perform either subtraction or addition. The local subtraction unit in Fig. 2 (b) passes samples through but when subtraction is needed, it is computed between incoming and delayed value.

All the needed data reorderings in signal flow graph of IDCT in Fig. 1 can be performed with a sequential permutation network constructed of shift-exchange units (SEU) as proposed in [5]. A shift-exchange unit of size K (SEU_K) depicted in Fig. 2 (c) is capable of exchanging data elements K samples apart in sequential data stream. In general, perfect shuffle reorders elements of a sequence in such a way that the elements of the first half of a sequence are

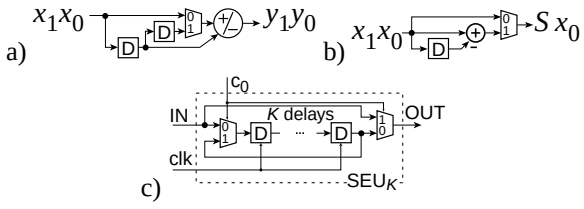

Fig. 2. Block diagrams of (a) butterfly unit, (b) local subtraction unit, and (c) shift-exchange unit of size K (SEU_K). D: Delay register

interlaced with the elements of the second half of the sequence. In other words, perfect shuffle permutation of a vector $\mathbf{x} = (x_0, x_1, \ldots, x_{K-1})^T$ results in a vector $\mathbf{y} = (x_0, x_{K/2}, x_1, x_{K/2+1}, x_2, \ldots, x_{K-1})^T$. Now, a 4-point perfect shuffle permutation can be realized with a single SEU_1 unit and an 8-point perfect shuffle with cascade of SEU_2 and SEU_1 units as illustrated with a timing diagram in Fig. 3. Apart from the global reorderings between the operational stage, there is also a single local reordering, i.e., exchanging of data elements two samples apart, before first multiplications in the signal flow graph in Fig. 1. Such a sample exchange can be realized with a single SEU_2 unit.

By cascading the basic data processing and data reordering blocks described previously, the sequential 8-point IDCT kernel can be constructed as illustrated in Fig. 4. Each unit in the 1-D IDCT architecture corresponds to a specific operational stage in Fig. 1. The loopfree structure enables the efficient pipelining. It should be noted that the pipeline registers are not included in Fig. 4. However, the degree of pipelining is a compromise between latency and throughput. Assuming that each arithmetic unit is followed by pipeline register, the latency of one-dimensional IDCT kernel equals to 17 cycles.

In two-dimensional IDCT architectures, which are based on row-column method, silicon area may be consumed into realization of the intermediate matrix transposition. The implementation efficiency is mainly dependent on interpretation of matrix transposition. The most straightforward way to realize the matrix transposition is its direct interpretation, i.e., rows in, columns out. However, such an approach will introduce double buffering with large silicon area and increased latency, since every sample is stored before reading [6].

The other difference is the way of storing the samples, i.e., the realization may be either memory-based or register-based. Here, the matrix transposition is realized with the register-based sequential permutation network presented in [7]. The corresponding structure and principal operation is illustrated in Fig. 5. It should be noted that the network is optimal from latency point of view since the maximum distance of the element to be moved in sequence equals to latency, which is 49 cycles.

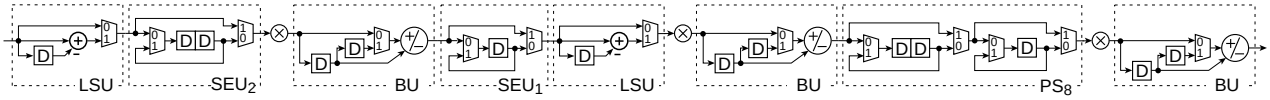

Fig. 4. Block diagram of sequential IDCT architecture. LSU: Local subtraction unit. D: Delay register. SEU$_k$: Shift-exchange unit of size k. BU: Butterfly unit. PS$_8$: 8-point perfect shuffle permutation. Clock and control signals are omitted for clarity.

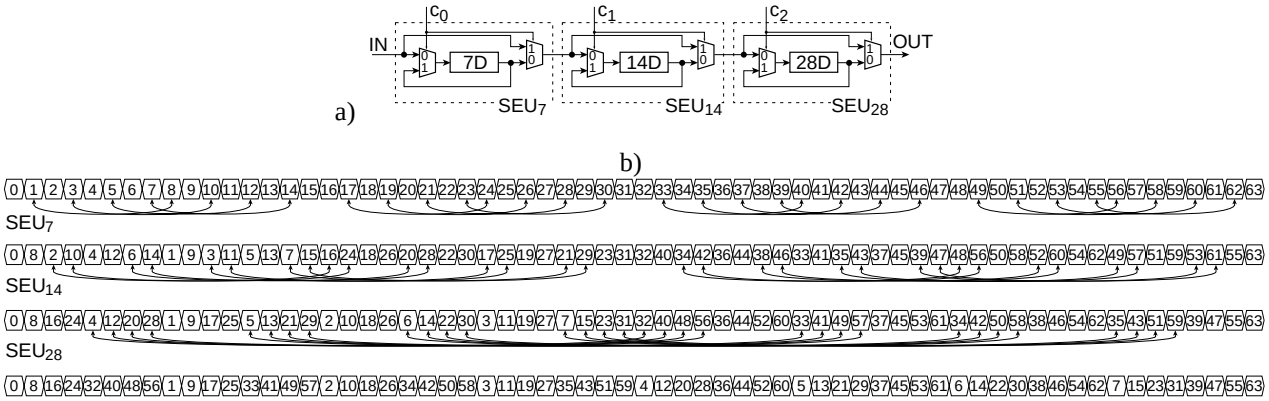

Fig. 5. Sequential 8×8 matrix transposition: (a) structure and (b) principal operation [7]. SEU$_k$: Shift-exchange unit of size k. KD: Shift register of size K.

3. ACCURACY ANALYSIS

The hardware implementations are often based on fixed-point, i.e., fractional, number representation due to the area friendly realization. This requires the scaling of intermediate signal levels for avoiding overflow during the computations. Typically scaling without additional hardware costs is done by rewiring, i.e., scaling factors are powers of two. Due to the fact, that all the intermediate data vectors are passed through multipliers in the proposed architecture, the signal levels can be adjusted at multipliers. This, on the contrary, allows scaling factors to be selected with finer resolution without additional hardware costs.

In the realizations of fixed-point number representation, the main error is caused by the finite word width in the intermediate arithmetic. This error is also known as quantization error. A test suite for the accuracy analysis of the proposed IDCT architecture is made according to the IEEE Standard 1180-1990 [1].

The performance of the pipeline architecture based on the IDCT algorithm shown in Fig. 1 is analyzed with simulations. First, six random test data sets are generated as specified in IEEE standard. Next, the proposed architecture is simulated with different word widths for estimating the error behaviour. Furthermore, two different quantization methods, rounding to the nearest integer and truncation of two's complement ("rounding towards minus infinity") are utilized. The coefficients d_i are rounded to the same word width as the internal data.

The obtained error values, mean error and mean square error per pixel and overall mean error and mean square error, are presented in Fig. 6 with different word widths. Overall mean square error reveals to be the limiting factor in simulations and, thus, 17 bits are required to fulfil the specification if rounding is used. It should be noted, however, that this method is more expensive from implementation area point of view.

If the hardware optimal quantization method, i.e., the truncation of two's complement is utilized, 22 bits are required for internal arithmetic. The quantized values are biased always towards minus infinity and, therefore, the sign of the error is negative at each pixel location. This removes the variance present in rounding method and makes the mean error value almost the same as the mean square error. Word width can be reduced if the error with opposite sign can be generated, i.e., introduce some variance to the error.

4. CONCLUSION

In this paper, a pipeline architecture for 8×8 IDCT is proposed. The architecture is based on row-column decomposition where the IDCT kernel is obtained by projecting the signal flow graph of fast IDCT algorithm vertically. The matrix transposition is realized with register-based sequential permutation network with optimal number of registers. The architecture can be freely pipelined for increasing throughput. The internal word width requirements in case of fixed-point realization was analysed with the aid of

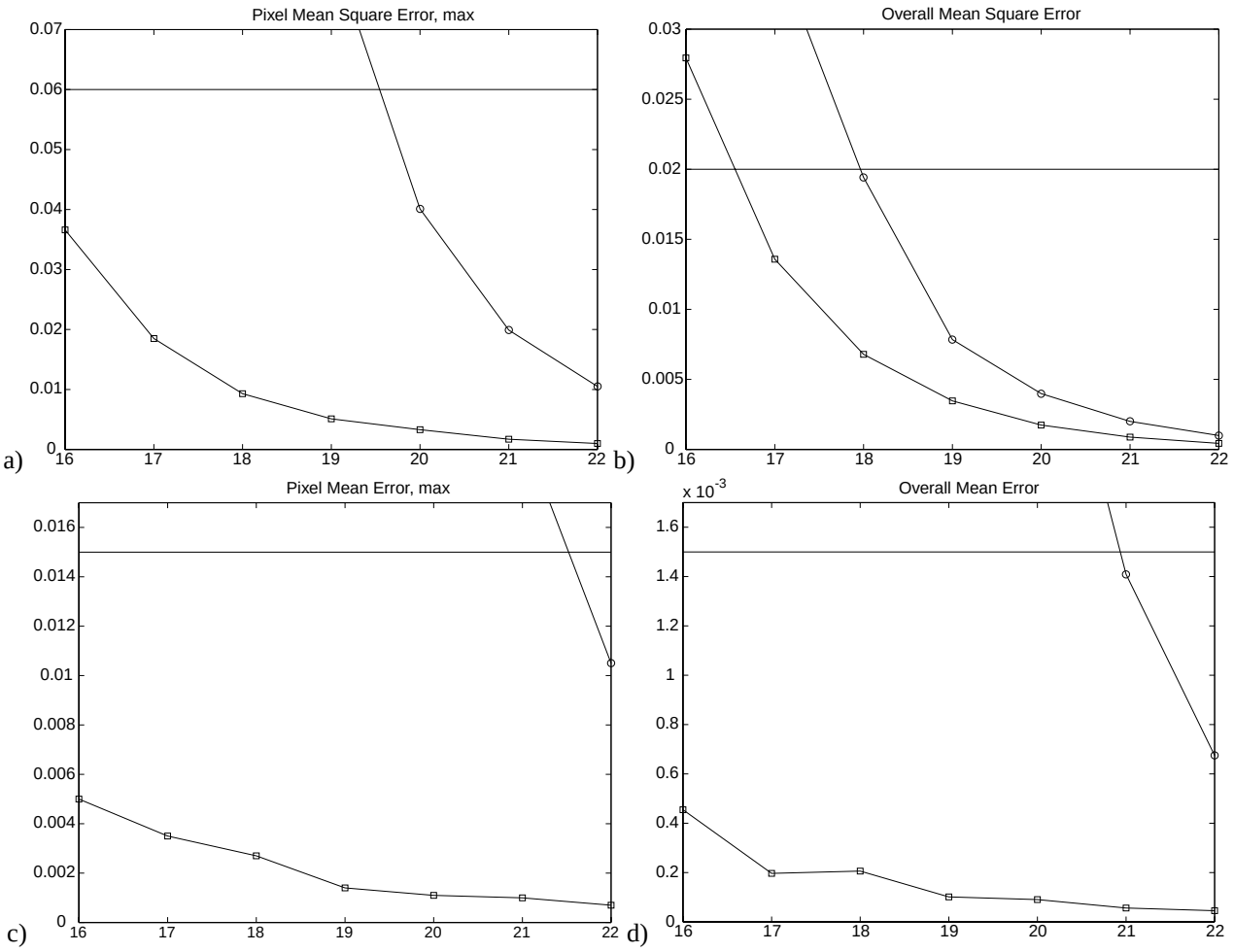

Fig. 6. Error behaviour of the proposed architecture as a function of internal word width: a) pixel mean square error, b) overall mean square error, c) pixel mean error, and d) overall mean error. Line with squares: rounding, line with circles: two's complement, and the solid line: requirement of the IEEE Standard.

simulations. The architecture requires internal word width of 17 bits with rounding and 22 bits with truncation of two's complement to satisfy IEEE Standard 1180-1990. The two-dimensional IDCT architecture yields arithmetic complexity of 6 multipliers, 6 adder/subtractors, and 4 adders. The overall latency with pipeline stages of single arithmetic unit is 83 system cycles.

REFERENCES

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

[2] H. L. Groginsky and G. A. Works, "A pipeline fast Fourier transform," *IEEE Trans. Comput.*, vol. 19, no. 11, pp. 1015–1019, Nov. 1970.

[3] J. Nikara, J. Takala, D. Akopian, J. Astola, and J. Saarinen, "Sequential architecture for discrete cosine transform," in *Proc. 18th NORCHIP Conference*, Turku, Finland, Nov. 6–7 2000, pp. 279–282.

[4] P. Pirsch, *Architectures for Digital Signal Processing*, John Wiley & Sons, Ltd., Chichester, United Kingdom, 1998.

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

[6] J. C. Carlach, P. Penard, and J. L. Sicre, "TCAD: a 27 MHz 8x8 discrete cosine transform chip," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, Glasgow, UK, May 23–26 1989, pp. 2429–2432.

[7] J. Takala, J. Nikara, D. Akopian, J. Astola, and J. Saarinen, "Pipeline architecture for 8×8 discrete cosine transform," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 5–9 2000, pp. 3303–3306.