On the Design of Selective Coefficient DCT Module

Chung-Yen Lu and Kuei-Ann Wen

***Abstract*— In this transactions letter, an innovative selective coefficient discrete cosine transform (SCDCT) architecture is proposed which is designed for selective coefficient computation and straightforward row–column computation. Having these features, the selective coefficient DCT core will fit for various area/speed requirements. It can save the transposition delay to simplify the computation flow of two-dimensional (2-D) DCT and, in view of circuit implementation, SCDCT is multiply-free and thus area/speed efficient.**

***Index Terms*—DCT, selective coefficient, 2-D DCT.**

I. Introduction

THE discrete cosine transform (DCT) defined by Ahmed *et al.* [1] in 1974 has recently found a number of applications in the area of digital image processing [2]–[4]. Fast algorithms for the DCT are, therefore, of significant practical interest. For the fast computation of two-dimensional (2-D) DCT, there are two categories: row-column method form one-dimensional (1-D) DCT [5]–[9] and direct 2-D DCT [10]–[12]. However, for fully pipelined implementation of the row–column method, a complicated matrix transposition architecture as well as two 1-D DCT modules are required. On the other hand, large area and bandwidth are also required for the implementation of direct 2-D DCT.

We proposed a 1-D DCT module, called selective coefficient DCT (SCDCT), which is multiply-free, and it provides different bandwidth requirements with a buffer engine, in addition, it can avoid heterogeneous problems in video/image compression.

As illustrated in Fig. 1, the data flow for the transmitted video is inherently in the form of sequential data. No matter row–column method or direct 2-D DCT will generate 8 or 64 elements/cycle, which are far more than the bandwidth of quantization process.

To overcome this, we specified the design features to be: 1) sequential output; 2) straight forward row–column computation; and 3) selective coefficient computation. These three features avoid the transposed memory delay and straight match the throughput of the whole system. Being a building block of 2-D DCT, the SCDCT module provides the flexibility of the assembling and thus supply for the various area/speed tradeoffs.

Manuscript received March 21, 1995; revised January 18, 1996. This paper was recommended by Associate Editor K.-H. Tzou.

The authors are with the Institute and Department of Electronics Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan, R.O.C.

Publisher Item Identifier S 1051-8215(98)00111-6.

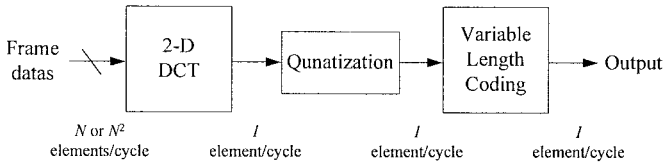

Fig. 1. Typical DCT coding system.

II. Selective Coefficient DCT Module

The 1-D DCT of a real data sequence $\{f(n) : n = 0, 1, ..., N-1\}$ is defined by

$$C(u) = \alpha(u) \sum_{n=0}^{N-1} f(n) \cos\left[\frac{\pi u(2n+1)}{2N}\right], \quad \text{for } u = 0, 1, \cdots, N-1 \tag{1}$$

where $\alpha(0) = \frac{1}{\sqrt{N}}$ and $\alpha(u) = \sqrt{\frac{2}{N}}$, $u \neq 0$. We take 8×8 as the standard size of a processing block. For the matrix form of (1), we defined constant matrix $\mathbf{F}$ and data matrix $\mathbf{D}(n)$ as

$$\mathbf{F} = \begin{bmatrix} A_1 & A_1 & A_1 & A_1 \\ B_1 & B_1 & B_2 & B_2 \\ C_1 & C_2 & C_3 & C_4 \end{bmatrix} \tag{2}$$

where $A_1 = \frac{1}{2}\cos[\frac{\pi}{4}]$, $B_i = \frac{1}{2}\cos[\frac{(2i-1)\pi}{8}]$, $C_j = \frac{1}{2}\cos[\frac{(2j-1)\pi}{16}]$, for $i = 1, 2$ and $j = 1, 2, 3, 4$.

$$\mathbf{D}(n) = \begin{bmatrix} D_0(n) \\ D_1(n) \\ D_2(n) \\ D_3(n) \end{bmatrix} \tag{3}$$

where $D_0(n) = f(0) + (-1)^n f(7)$

$$D_1(n) = (-1)^{p(n)} * [f(1) + (-1)^n f(6)]$$

where $p(n) = \lfloor \frac{n+2}{5} \rfloor$

$$D_2(n) = (-1)^{q(n)} * [f(2) + (-1)^n f(5)]$$

where $q(n) = \lfloor \frac{n+1}{3} \rfloor$

$$D_3(n) = (-1)^{r(n)} * [f(3) + (-1)^n f(4)]$$

where $r(n) = \lfloor \frac{n}{2} \rfloor$.

The matrix form of 1-D eight-point DCT will be expressed as

$$C(u) = [\mathbf{P}_u \mathbf{D}(u)]^T [\mathbf{S}_u \mathbf{F}] \tag{4}$$

for $u = 0, 1, \cdots, 7$ where $\mathbf{S}$ is selection matrix

$$\mathbf{S}_i = \begin{cases} [1 \quad 0 \quad 0], & i = 0, 4 \\ [0 \quad 1 \quad 0], & i = 2, 6 \\ [0 \quad 0 \quad 1], & i = 1, 3, 5, 7 \end{cases} \tag{5}$$

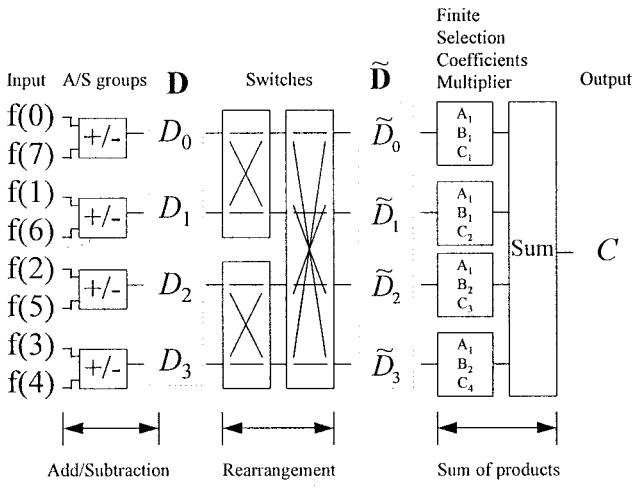

Fig. 2. Block diagram of selective coefficient DCT module.

and permutation matrices $\mathbf{P}_i, i = 0, 1, \cdots, 7$, as illustrated in the following:

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{P}_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{P}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{P}_5 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{P}_6 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{P}_7 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

From (4), the 1-D DCT computation could be decomposed into three steps.

1) Compute the data vector $\mathbf{D}$.
2) Rearrange $\mathbf{D}_0$, $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$ by mapping $\tilde{\mathbf{D}} = \mathbf{PD}$.
3) Sum up the products by $\mathbf{SF}\tilde{\mathbf{D}}$. The corresponding cosine factor set is chosen from $\mathbf{F}$ by selection matrix $\mathbf{S}$. Since the cosine factors are constant, the product could be implemented by shifting and adding operations. Therefore, we could obtain a multiplier-free DCT module.

The block diagram of SCDCT module is illustrated in Fig. 2. The input of the SCDCT module is a 1-D frame with eight elements, as denoted $f(n), n = 0, 1, \cdots, 7$. The output is one of the 1-D DCT coefficients, $C(u), u = 0, 1, \cdots, 7$.

III. Multiply-Free Implementation of SCDCT

The main functions of SCDCT are: 1) addition/subtraction operation; 2) arrangement process; and 3) sum of product of data vector with cosine factor set.

TABLE I
Booth's Representation of Cosine Factors

FSCM 1		FSCM 2		FSCM 3		FSCM 4	
A_1	$0.10\bar{1}0\bar{1}0101000$	A_1	$0.10\bar{1}0\bar{1}0101000$	A_1	$0.0101\bar{1}0101000$	A_1	$0.0101\bar{1}0101000$
B_1	$0.1000\bar{1}0\bar{1}00100$	B_1	$0.1000\bar{1}0\bar{1}00100$	B_2	$0.0101\bar{0}0010000$	B_2	$0.0101\bar{0}0010000$
C_1	$0.100000\bar{1}01000$	C_2	$0.10\bar{1}0101010\bar{0}1$	C_3	$0.0100\bar{1}00\bar{1}0010$	C_4	$0.000\bar{1}10010000$

TABLE II
Comparisons of Circuit Complexity in Real-Time Hardware Implementation of Eight-Point DCT

	Chen's [5]	Wagh's [6]	Lee's [7]	Malvar's [8]	Chan's [9]	SCDCT
Number of Multipliers	16	14	12	12	12	0
Number of adders	26	32	29	31	29	25
Input bandwidth requirement	8	8	8	8	8	1
Output bandwidht	8	8	8	8	8	1
Large transposition buffer for 2-D DCT	yes	yes	yes	yes	yes	no

The idea of multiply-free implementation of SCDCT is to take advantage of the fact that cosine factors are fixed so that multiplication of the factors are then shifted and added to produce output.

We define four sets of finite selection coefficients multipliers (FSCM's) for the right-most block as shown in Fig. 2

$$FSCM-1: A_1, B_1, C_1; \quad FSCM-2: A_1, B_1, C_2;$$
$$FSCM-3: A_1, B_2, C_3; \quad FSCM-4: A_1, B_2, C_4.$$

The booth's representation of the cosine factors are listed in Table I.

Therefore, the cosine factors control the addition or subtraction of the shifting data. Namely, if the ith bit of cosine factor is 1, then data$*2^{-i}$ are added, if it is $\bar{1}$, subtraction of the data$*2^{-i}$ will be processed, otherwise ignoring it. The block diagrams of the four FSCM's are shown in Fig. 3. The fully pipelined architectures of FSCM's consist of shifters, adders, and some simple logic gates.

The hardware implementation of FSCM is based on shifting and adding structure. Therefore, the circuit complexity is much less than many fast DCT algorithms. Table II compares the circuit complexity of SCDCT with fast 1-D DCT algorithms. The input bandwidth of SCDCT is determined with a buffer engine as illustrated in the Appendix.

IV. Conclusion

The SCDCT module is based on the property of arbitrary selective coefficient computation. It has three attributes: 1) sequential input/output could smooth the data flow and reduce the memory bandwidth requirement; 2) zero transpose memory

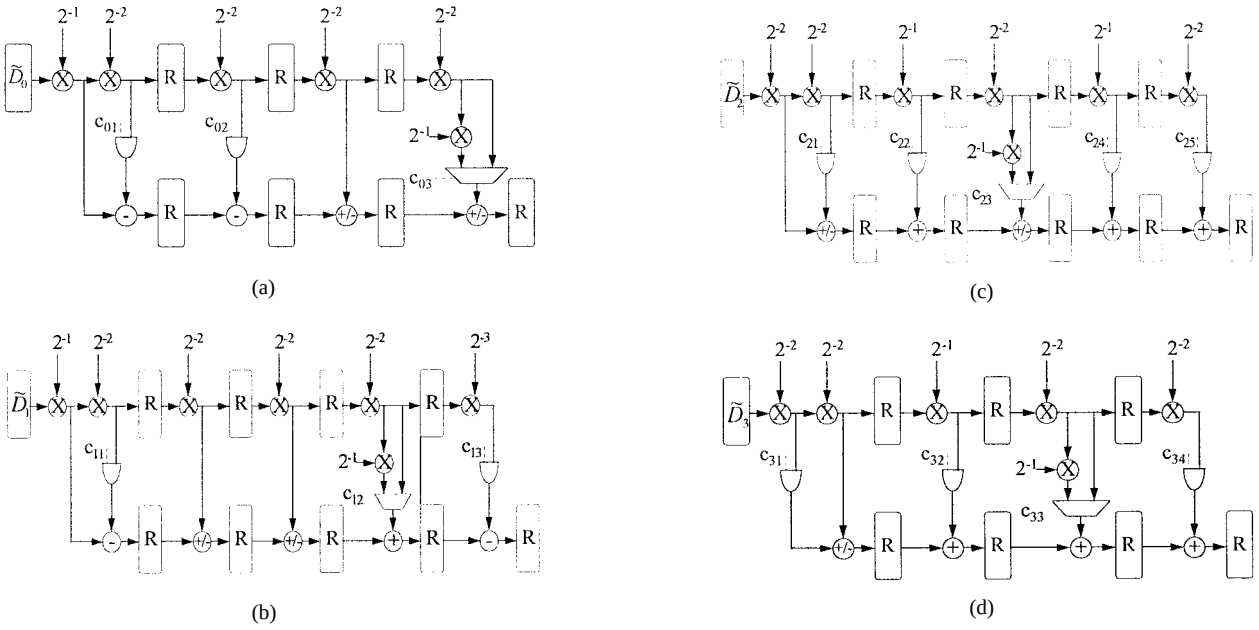

Fig. 3. The structures of FSCM's: (a) FSCM-1, (b) FSCM-2, (c) FSCM-3, and (d) FSCM-4.

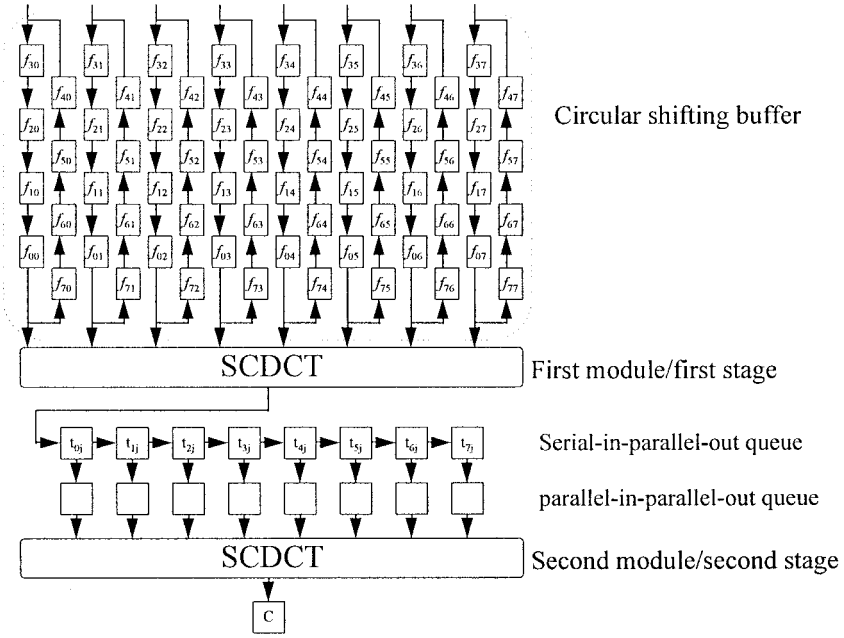

Fig. 4. Implementation of 2-D DCT by two SCDCT modules.

delay could straight forward the row-column computation flow; and 3) the flexibility of area/throughput tradeoff. In the architecture of the SCDCT module, with FSCM being proposed, the multiplier-free structure makes it area efficient.

APPENDIX
APPLICATIONS OF SCDCT

The SCDCT module is able to compute an arbitrary 1-D DCT coefficient. This property allows the SCDCT module to be adopted for various requirements, such as delay-free for 2-D DCT by row–column method, DCT truncation coding, adaptable area–time tradeoff and bandwidth requirement.

A. Implementation of Delay-Free Transposition 2-D DCT

The row data vectors $F_i = [f_{i0} \quad f_{i1} \quad \cdots \quad f_{i7}]$, $i = 0, 1, \cdots, 7$, are sequentially fed into the first SCDCT to get intermediate 1-D DCT coefficients. The column vectors of the intermediate DCT coefficients $T_j = [t_{0j} \quad t_{1j} \quad \cdots \quad t_{7j}]$, $j = 0, 1, \cdots, 7$, are sequentially fed into the second SCDCT to get 2-D DCT coefficients. The whole block diagram of

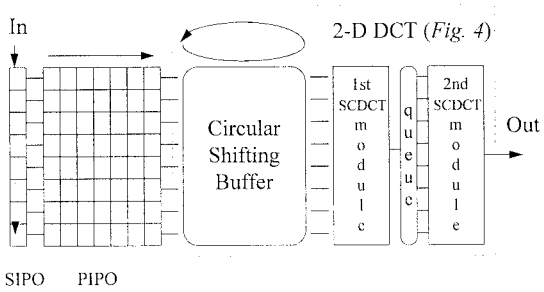

Fig. 5. The reduction of the bandwidth requirement.

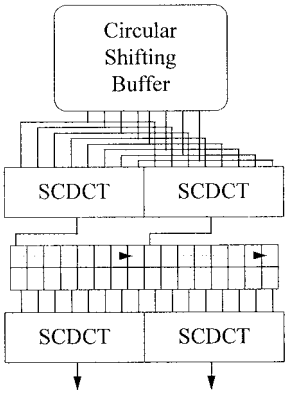

Fig. 6. The output bandwidth of 2-D DCT is two elements/cycle.

the 2-D DCT is illustrated in Fig. 4, the straight computation of column 1-D DCT coefficients save the large transposition buffer and transposition delay.

B. Reduction of Bandwidth Requirement with Buffer Engine

Many fast 2-D DCT's need eight elements per cycle for input/output. To match that, excessive I/O should be used. Using SCDCT combined with SIPO/PIPO as shown in Fig. 5, it is clear that the data flow is single-in single-out even for the 2-D DCT.

C. Modulization for Area–Time Tradeoff

Higher throughput of a 2-D DCT could be achieved with more SCDCT modules. Due to the property of free selective coefficient computation, different numbers of SCDCT modules can be used for various bandwidth requirements of 2-D DCT. Therefore, the different tradeoff between area and throughput can be easily fulfilled. As illustrated in Fig. 6, four SCDCT modules can double the output throughput.

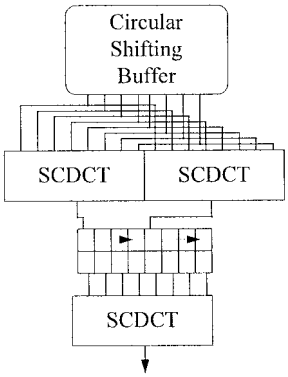

Fig. 7. Specific DCT application: Partial DCT.

D. DCT Truncation Coding

For DCT truncation coding, which forces the high frequency DCT coefficients to be zero, only a lower frequency DCT subblock is required. With SCDCT, a special DCT architecture could be designed to calculate any 4×4 subblock of the 2-D DCT as illustrated in Fig. 7.

REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, Jan. 1974.
[2] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, Apr. 1991.
[3] M. Liou, "Overview of the $p \times 64$ kbits/s video coding standard," *Commun. ACM*, vol. 34, no. 4, pp. 59–63, Apr. 1991.
[4] W. B. Pennebaker and J. L. Mitchell, *JPEG—Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
[5] W. H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004–1009, Nov. 1977.
[6] M. D. Wagh and H. Ganesh, "A new algorithm for the discrete cosine transform of arbitrary number of points," *IEEE Trans. Comput.*, vol. C-29, pp. 269–277, Apr. 1980.
[7] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1243–1245, Dec. 1984.
[8] H. Malvar, "Fast computation of discrete cosine transform through fast Hartley transform," *Electron. Lett.*, vol. 22, no. 7, pp. 352–353, Mar. 1986.
[9] Y. Chan and W. Siu, "A cyclic correlated structure for the realization of discrete cosine transform," *IEEE Trans. Circuits Syst.–II*, vol. 39, pp. 109–113, Feb. 1992.
[10] M. Vetterli, "Fast 2-D discrete cosine transform," in *Proc. ICASSP'85*, Mar. 1985, pp. 1538–1541.
[11] N. I. Cho and S. U. Lee, "DCT algorithms for VLSI parallel implementation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 121–127, Jan. 1990.
[12] ———, "A fast 4×4 DCT algorithm for the recursive 2-D DCT," *IEEE Trans. Signal Processing*, vol. 40, pp. 2166–2173, Sept. 1992.