PIPELINE PROCESSORS FOR FAST TRIGONOMETRIC TRANSFORMS

D.Akopian[1], K.Egiazarian[1], S.Agaian[2], J.Astola[1]

[1]Signal Processing Laboratory,
Tampere University of Technology
P.O.Box 553, FIN-33101 Tampere, Finland

[2]Electrical Engineering Department,
Tufts University
MA 02155, Medford, USA

Abstract

Pipeline processors are proposed for computation of fast Discrete Trigonometric Transforms (DTT), particularly for Cosine (DCT) and Sine (DST) transforms. The transform time is $O(N)$, where $N = 2^n$ is the length of the input data vector. Because of the using the internal properties of the cosidered transforms, the numbers of adders, multipliers and registers are reduced when compared with the pipeline structures for fast Fourier transform and direct DCT.

1 Introduction

In the last years orthogonal Discrete Trigonometric Transforms (DTT) and particularly Discrete Cosine and Sine real value Transforms (DCT and DST) are widely used in such problems of digital signal processing as image coding, feature extraction, filtering, and, especially, image compression. DCT has the better energy compaction property and near optimal performance which is closest to that of the Karhunen-Loeve transform (KLT) among many discrete transforms for highly correlated signals, especially for the first-order Markov processes [1]. Jain have shown, that the performance of DST approaches is that of the KLT for a first-order Markov sequence with given boundary conditions, especially for signals with low correlated coefficients [3]. The JPEG (Joint Photographic Experts Group) coding algorithm, based on DCT (DCT-II by Wang classification [5]), has been established recently as an industry standard for still-frame, continuous-tone image compression [10].

In the same time with the search of effective algorithms, the problem of the hardware design needs to be solved also. Processors that are based on one arithmetic unit reflect the used algorithms and their performance is proportional to the number of arithmetic operations, involved in the transforms (see, for example, [8]). In spite of various fast algorithms, proposed to reduce the number of arithmetic operations, such processors transform the one-dimensional data set of length N in $O(N \log N)$ stages.

For high speed calculations, the processors with the performance of order $O(N)$ are required for processing data in real time. The systolic structures in [7] for DTT have such performance, using the $O(N)$ arithmetic units (cells).

In this paper we consider pipeline structures for DTT based on $O(\log N)$ arithmetic units and with the transform time $O(N)$ for the subset of the transform class, considered in [2], [5-6]. The delay-exchange procedures are simplified, compared with the pipelined structures for DTT in [9].

2 Discrete Trigonometric Transforms

A Discrete parametric Trigonometric Transform (DTT) of the vector $\vec{x} = [x_0, x_1, \ldots, x_{N-1}]^T$ is the vector $\vec{X} = V_N \cdot \vec{x}$, where $\vec{X} = [X_0, X_1, \ldots, X_{N-1}]^T$, and $V_N = \|v_{m,k}\|$ is the following matrix:

$$v_{m,k} = \mu_{m,k} \cdot \cos \psi_{m,k} + \nu_{m,k} \cdot \sin \psi_{m,k},$$

where

$$\psi_{m,k} = \frac{\pi}{M} \cdot \alpha_0 (m + \alpha_1)(k + \alpha_2);\ 0 \le m, k \le N - 1,$$

$(N, M, \alpha_0, \alpha_1, \alpha_2)$ are the parameters, and $(\mu_{m,k}, \nu_{m,k})$ are the coefficients.

The expression $V : (N; M; \alpha_0; \alpha_1; \alpha_2)$, $(\mu_{m,k}; \nu_{m,k})$ denotes the trigonometric transform with given parameters and coefficients. When the matrix V_N is an orthonormal trigonometric matrix, then DTT is called a discrete trigonometric transform. In this paper we consider the transforms:

Cosine Transforms

C_N^{II} : $(N; N; 1; 0; \frac{1}{2})$, $(\frac{2}{\sqrt{N}} b_1(m); 0)$

C_N^{III} : $(N; N; 1; \frac{1}{2}; 0)$, $(\frac{2}{\sqrt{N}} b_1(k) b_2(m); 0)$

C_N^{IV} : $(N; N; 1; \frac{1}{2}; \frac{1}{2})$, $(\frac{2}{\sqrt{N}}; 0)$

Sine Transforms

S_N^{II} : $(N; N; 1; 0; \frac{1}{2})$, $(0; \frac{2}{\sqrt{N}} b_2(m))$

S_N^{III} : $(N; N; 1; \frac{1}{2}; 0)$, $(0, \frac{2}{\sqrt{N}} b_2(k))$

S_N^{IV} : $(N; N; 1; \frac{1}{2}; \frac{1}{2}), (0, \frac{2}{\sqrt{N}})$

where

$$b_1(m) = \begin{cases} 1, & \text{if } m \neq 0; m \neq N \\ \frac{1}{\sqrt{2}}, & \text{if } m = 0; m = N \end{cases}$$

$$b_2(m) = \begin{cases} 1, & \text{if } m \neq N-1 \\ \frac{1}{\sqrt{2}}, & \text{otherwise} \end{cases}$$

The subscript represents the order of the matrix, and the superscript denotes the type of the DCT or DST. All the considered architectures transform vectors of data of length $N = 2^n$.

The peculiarity of the pipeline structures is that inputs (used in the operations) enter the cascade in different times, and the problem is to organize the data flow in a way, suitable for calculations when arithmetic units realize in sequential basic operations, parallelized with data reordering.

3 Pipeline processor for the fast DST-IV transform

Processor structure for DST sine transform is represented in the Fig.1a. It is analogous to the existing structure for the Cooley-Tukey FFT algorithm [4], but processes the real valued data using the other basic operation (see Table 1). The number of multipliers, adders and registers is reduced because of the utilization the internal properties of the DST transform. The processor contains $n = \log N$ units, arranged in a array. All units have the same structure and differ only by the sizes of shift register blocks. Unit with the number $(n-i), i = 0, \ldots, n-1$ contains shift register blocks with the size of 2^{n-i} registers, three adders S_1^i, S_2^i, S_3^i, multiplier M^i, memory element for coefficients CM^i, switch SW^i, and three shift register blocks $R_1^i - R_3^i$. The input of the $(n-1)$-th and the output of (0)-th units are the input and the output of the processor appropriately. The $(n-1)$-th unit differs from the others by the lack of the shift-register block R_1^{n-1}. The (0)-th unit has two additional multipliers M_o^1, M_o^2 and coefficient memories CM_o^1, CM_o^2. Input data enters the processor in reverse order [6]

$x_1 \ldots x_{N-1}\ x_N \rightarrow$

$0 \ldots 0 \quad 0 \rightarrow$ processor input

Blocks R_1^i, R_2^i and switches SW^i reorder the data, which enter the input of the cascade. Switches SW^i have two inputs I_1, I_2 and two outputs O_1, O_2. Every i tacts direct connection $I_1 = O_1, I_2 = O_2$ is changed on the cross connection $I_1 = O_2, I_2 = O_1$ and vice versa.

The $(n-i)$-th cascade processes data in the following way. We can represent data flow as it is shown in the Fig.4. Denote the variables, which enter through the upper and lower channels the $(n-i)$-th cascade as:

$p_{2^{n-i}-1} \ldots p_1\ p_0 \rightarrow$

$t_{2^{n-i}-1} \ldots t_1\ t_0 \rightarrow$

% 1. Reordering is performed by the shift registers blocks R_1^{n-i}, R_2^{n-i} and switch SW^{n-i} and in fact is the data exchanging between upper and lower chanels. See Fig.1d.

for $k' = 0, 1, 2, \ldots, 2^{i-2} - 1$ **do in sequential**

$k = 2k'$ $(= 0, 2, 4, \ldots, 2^{i-1} - 2)$

for $l = 0, \ldots, 2^{n-i} - 1$ **do in sequential**

do a., b., c. and d. in parallel

$a.\ p_{l+k\cdot 2^{n-i}} = p_{l+k\cdot 2^{n-i}}$

$b.\ t_{l+(k+1)2^{n-i}} = t_{l+(k+1)2^{n-i}}$

$c.\ p_{l+(k+1)2^{n-i}} = t_{l+k2^{n-i}}$

$d.\ t_{l+k2^{n-i}} = p_{l+(k+1)2^{n-i}}$

% 2. Order inverting is carried out by the SRB-s R_2^{n-i}, and R_3^{n-i}.

for $k = 1, 2, \ldots, 2^{i-1}$ **do in sequential**

for $l = 0, \ldots, 2^{n-i} - 1$ **do in parallel**

$\hat{p}_{l+k2^{n-i}} = p_{2^{n-i}-1-l+k2^{n-i}}$

% 3. Multiplication by coefficients in the multiplier M^{n-i}. Coefficients are stored in memory CM^{n-i} in advance.

for $k = 0, 1, \ldots, 2^{i-1} - 1$

for $l = 0, 1, \ldots, 2^{n-i} - 1$

$p_{l+k\cdot 2^{n-i}} = 2 \cdot d(2^i - 1 - k) \cdot p_{l+k\cdot 2^{n-i}}$

% 4. additions in the adder S_1^{n-i} of data from the register R_3^i with the data, passed through the lower chanel.

for $i = 0, \ldots, 2^{n-1} - 1$ **do in sequential**

$t_i = \hat{p}_i + t_i$

% 5. Butterfly operation is fulfilled by the paired adder and subtractor unit $S_2^{n-i} - S_3^{n-i}$.

$p_i = -p_i + t_i$

$t_i = p_i + t_i$

end

The output data then enter the two inputs of the $(n-i-1)$-th cascade. All the calculationes described are fulfiled without delays as far as data are received and so different operations are carried out in the same time. At the end in two output multipliers of the processor M_1^0 and M_2^0 the data from both the channels are multiplied with the output coefficients, stored in memories CM_1^0 and CM_2^0 in advance.

for $i = 0, 1, \ldots, 2^{n-1}$ **do in sequential**

a. and **b. do in parallel**

a. $p_i = p_i \cdot d(2 \cdot 2^n - 2i - 2)$

b. $t_i = t_i \cdot d(2 \cdot 2^n - 2i - 1)$

The characteristics of the structure are represented in Table 1, where

$$t = \tau_{\text{mul}} + 2\tau_{\text{add}} + o(t) \qquad (1)$$

and τ_{mul} is the multiplication time, τ_{add} is the addition time. Because of expressions [6]

$$[S_N^{IV}] = [\bar{I}_N][C_N^{IV}][D_N],\ [C_N^{IV}]^{-1} = [C_N^{IV}],$$

where

$$[D_N] = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 \end{pmatrix},$$

$$[\bar{I}_N] = \begin{pmatrix} 0 & \dots & 0 & 0 & 1 \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}, \qquad (2)$$

the DCT-IV and inverse DST-IV and DCT-IV transforms can be calculated in the same structure.

The coefficients, used in the algorithm can be calculated in advance and written in CM^i according to the following recursive expressions [6]:

$d(1) = \sqrt{0.5}$,

$d(2i) = [0.5(1 + d(i))]^{1/2}$,

$d(2i+1) = [0.5(1 - d(i))]^{1/2}$.

4 Pipeline processor for fast DCT-II transform

The processor structure for DCT-II is represented in Fig.1b. It contains $n = logN$ units, arranged in a array. Except the first (0)-th and the last $(n-1)$-th cascades, the others have the same structure and differ only by the sizes of shift register blocks (SRB). The unit with the number $(i), i = 1, 2, \dots, logN - 1$ contains the SRBs $R_1^i - R_4^i$ with the size of 2^i registers, a multiplier M^i, a coefficient memory CM^i, switches SW_1^i and SW_2^i, three adders $S_1^i - S_3^i$, an adress counter C^i for SRB R_4^i.

First (0)-th has the simple structure and does not contain the switch SW_2^0 and the SRB R_3^0. The last $(n-1)$-th cascade differs from others by lack of the switch SW_2^{n-1}, SRBs R_2^{n-1} and R_3^{n-1}.

Input data enter the processor in Hadamard order [14], the adresses of sequential data are formed recursively in the following way:

$h_1(0) = 0$

$h_{2k}(2i) = h_k(i)$

$h_{2k}(2i+1) = 2k - 1 - h_k(i)$.

At first, let us denote the variables, which enter the upper and lower channels of the i-th cascade in the following way.

$p_{2^{n-i}-1} \dots p_1\ p_0 \rightarrow$

$t_{2^{n-i}-1} \dots t_1\ t_0 \rightarrow$

The i-th cascade processes data by the following algorithm.

% 1. Butterfly procedure, which is performed by paired adder-subtractor block $S_2^i - S_3^i$.

for $j = 0, \dots, 2^{n-1} - 1$ **do in sequential**

a. and b. do in parallel

a. $p_j = p_j + t_j$

b. $t_j = p_j - t_j$

% 2. Procedure of multiplications by coefficients, which is performed after adders block in multiplier M^i.

for the (0)**-th cascade**

for $j = 0, \dots, 2^{n-1} - 1$ **do in sequential**

$t_j = d(2^{n-1} + j) \cdot t_j$

for the i**-th cascade**

for $k = 0, 1, \dots, 2^{n-i-1} - 1$ **do in sequential**

for $j = 0, 1, \dots, 2^i - 1$ **do in sequential**

if $j = 0$ **then** $p_{j+k\cdot 2^i} = d(2^{n-i-1} + k) \cdot p_{j+k\cdot 2^i}$

else $p_{j+k\cdot 2^i} = 2d(2^{n-i-1} + k) \cdot p_{j+k\cdot 2^i}$

% 3. Inverting procedure. This procedure is performed by SRB R_i^2, SRB R_i^3, SRB R_4^i and counter C^i. In the $(n-1)$-th cascade order inverting is performed only using the SRB R_4^{n-1} and the adress counter C^{n-1}, storing the first 2^{n-1} processing times and reading the next 2^{n-1} times in inverse order for calculations in the subtractor S_1^{n-1}.

for $k = 1, 2, \dots, 2^{n-1-i} - 1$ **do in sequential**

for $l = 0, \dots, 2^i - 1$ **do in parallel**

$\hat{p}_{l+k2^i} = p_{2^i-1-l+k2^i}$

% 4. Subtractions procedure is performed by subtractor S_1^i.

for $j = 0, \dots, 2^{n-1} - 1$ **do in sequential**

$t_j = t_j - \hat{p}_j$

% 5. Reordering procedure,which is performed by the SRB-s R_i^1, R_i^2 and by switch SW^i. Due to this procedure the blocks of i data are exchanged between upper and lower chanels as it is shown in Fig. 1d, where the expressions ' $n - i$ data' must be replaced by the 'i data'.

for $k' = 0, 1, 2, \dots, 2^{n-i-2}$ **do in sequential**

$k = 2k'$ $(= 0, 2, 4, \dots, 2^{n-i-1})$

for $l = 0, \dots, 2^i - 1$ **do in sequential**

% a.,b.,c. and d. do in parallel

a. $p_{l+k2^i} = p_{l+k2^i}$

b. $t_{l+(k+1)2^i} = t_{l+k2^i}$

c. $p_{l+(k+1)2^i} = t_{l+k2^i}$

d. $t_{l+k2^i} = p_{l+(k+1)2^i}$

end

SRBs R_1^i, R_2^i and switch SW_1^i reorder data in the same way as in the structure for DST-IV, exchanging between upper and lower chanels blocks of i data.

In time of this reordering some data are recalculated. Switch SW_2^i every i tacts changed the connection output SRB $R_1^3 \rightarrow$ second input S_1^i to the connection output SRB $R_4^i \rightarrow$ second input S_1^i, where R_4^i contains the data, which are red from the lower chanel after SW_1^i in inverse order. The output of the i-th cascade is two set of data, where upper set is formed as the output of the SBR R_2^i, and the lower set as the second output of the SW_1^i. All the calculations are fulfiled without accumulation in adders and multipliers as far as data are received, and reordering and arithmetic operations are carried out in the same time. The characteristics of the structure are represented in Table 1, where t is given by (4.1). Because of expressions from [16]:

$$[S_N^{II}] = [\bar{I}_N][C_N^{II}][D_N],\ [S_N^{III}] = [S_N^{II}]^{-1} = [S_N^{II}]^T,$$

where $\bar{I}_N$ and D_N are marices (4.2), DST-II and inverse DST-III can be calculated in the same structure.

5 Pipeline processor for the fast DST-III transform

This processor in the main is analogous to one for the DST-IV and is represented in the Fig.1c. Because of expressions from [16] this processor can be used also for computation of inverse DST-II and DCT-II transforms.

6 Conclusions

Pipeline processors for computation of fast trigonometric (Cosine and Sine) transforms are suggested. The transform time is $O(N)$, where $N = 2^n$ is the length of the input data vector. The proposed processors have less adders and multipliers comparing with the pipeline processors of the fast Fourier transform and existing pipeline structures for DCT. Although in this paper we present the architectures of processors for a few cosine and sine transforms, the presented approach makes it possible to construct analogous processors for a large class of DTT.

References

[1] Ahmed N., Natarajan T., Rao K.R. Discrete cosine transform. IEEE Trans., Comput., vol. C-23, pp. 90-93, Jan. 1974.

[2] Egiazarian K.O. Discrete Trigonometric Transforms. R. Trapple (ed.), Cybernetics and systems'88, 1215-1222, 1988 by Kluwer Academic Publishers.

[3] Jain A.K. Fundamentals of Digital Image Processing. Prentice Hall, Englewood Cliffs, NJ, 1989.

[4] Rabiner L.R., Gold B. Theory and Application of Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[5] Thompson C.D. Fourier transforms in VLSI. IEEE Trans.-1983. vol.C-32.p.1047-1057.

[6] Wang Z. Prunning the Fast Discrete Cosine Transform, IEEE Trans.,Commun., vol.39, no.5, May 1991.

[7] Wang Z. Fast Discrete Sine Transform Algorithms, Signal Processing 19(1990), 91-102, Elsevier.

[8] Agaian S.S., Akopian D.A. Sistolic Coders and Decoders for a Message Source Using Trigonometric Transforms. Pattern Recognition and Image Analysis, vol.2,no.3, 1992.

[9] Duhamel P., Guillemot C., Carlach J.C. A DCT Chip Based on a New Structured and Computationally Efficient DCT Algorithm. 1990 IEEE International Symposium on Circuits and Systems, May 1-3, 1990, pp.77-80.

[10] Wen K.-A., Cheng P.-W., Lin R.-Y. VLSI Design of the Shuffle-exchange Network for 2D Fast Transforms. 1993 IEEE International Symposium on Circuits and Systems, pp.754-757.

[11] JPEG Draft Technical Specification Revision 8. August 1990.

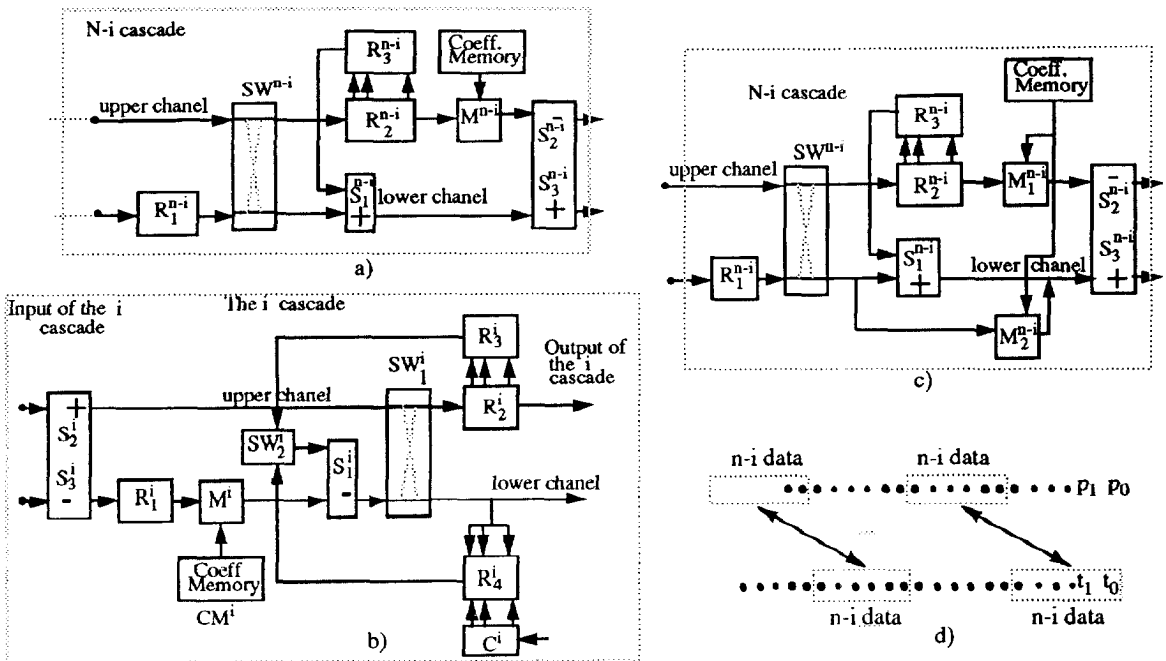

Fig.1. Processor structures for a) DST-IV b) DCT-II c) DST-III orthogonal transform when input vector size is $N=2^n$ and d) data exchange performed by the blocks of shift registers

Table 1: comparative characteristics of pipelined structures for trigonometric transforms, $N=2^n$

Processor type-	FST-IV	FST-III	FCT-II	FFT
number of multipliers for i-th cascade	1 for real-value data	2 for real value data	1 for real-value data	1 for complex-value data (4 real-value)
total number of multipliers	logN+2	2logN	logN-1	logN
number of adders for i-th cascade	3 real-value	3 real-value	3 real-value	2 complex-value (4 real-value)
total number of adders	3logN	3logN	3logN	4logN
number of registers for i-th cascade	$3*2^i$ (n-1)-th contain $2*2^{n-1}$	$3*2^i$-1 (n-1)-th contain $2*2^{n-1}$-1	$4*2^i$, (n-1)-th contain $2*2^{n-1}$, (0)-th - 2	$4*2^i$, (n-1)-th contain $2*2^{n-1}$
total number of the registers	$5*2^{n-1}$-3= 5N/2-3	$5*2^{n-1}$-3-n= 5N/2-3-logN	3N-6	3N-4
basic operation	a+d+k_1c, b+c+k_2d, a+d-k_1c, b+c-k_2d, a,b,c,d-real variables, k_1,k_2 -koefficients	the same as for FST-IV	a+b, a-b-k1(c+d), c+d, c-d-k2(a+b), a,b,c,d-real variables,k1,k2- koefficients	a+kb, a-kb, a,b-complex variables, k- complex koefficient
realized functions	DST-IV, DST^{-1}-IV, DCT-IV, DCT^{-1}-IV	DST-III, DST^{-1}-II, DCT^{-1}-II	DCT-II, DST-II, DST^{-1}-III	DFT, DHT
transform time	N*t	N*t	N*t	$N*t_2$, $t_2>t$
delay time	N*t	N*t	N*t	$N*t_2$