A CORDIC Based Array Architecture for Complex Discrete Wavelet Transform

B. Das
Dept. of E & ECE
Indian Institute of Technology
Kharagpur, INDIA
Tel: +91 (0) 3222 81466
e-mail: bipul@ece.iitkgp.ernet.in

Swapna Banerjee
Dept. of E & ECE
Indian Institute of Technology
Kharagpur, INDIA
Tel: +91 (0) 3222 83500
e-mail: swapna@ece.iitkgp.ernet.in

ABSTRACT

In this paper an array architecture for computation of Complex Discrete Wavelet Transform has been proposed. The wavelet filter coefficients are realized using multiplier less pipelined CORDIC algorithm. The choice of pipelined CORDIC algorithm over the conventional one for realizing the filter coefficient of CDWT is hardware effective and also effects in high frequency operation. The controller unit clusters input samples into even and odd samples coming in proper sequence at each clock cycles. This clustering provides a good amount of parallelism for faster operation of the filter compared to direct filter realization. The 8-tap filter bank is implemented using array architecture, effecting in high throughput. The algorithm developed is implemented on FPGA using the Virtex XCV100 series.

Categories and Subject Descriptors

Architectures and Cache

Keywords

Complex Discrete Wavelet Transform, Image Padding, FIR filter, CORDIC, FPGA.

1. INTRODUCTION

Image analysis in transform domain has gained considerable importance in the recent years**.** *Wavelet Transform* [1],[2] have come a long way as an important tool for image analysis. The multiresolution property of this transform has proved to have immense application in reliable loss less image compression and reconstruction. Though the *Discrete Wavelet* has extensive application in motion vector estimation but the real wavelet transforms, e.g. Daubechies, Haar etc., suffer from the lack of rotation and shift invariance property. The Complex Discrete Wavelet Transform (CDWT) [3], [4], [5], [6] a phase-based method is a solution to these problems and is very effective in motion estimation and stereo image matching. For real-time application, hardware implementation of the transform is of immense importance.

An array architecture [7], for realizing CDWT is proposed, which uses pipelined *Co-Ordinate Rotation DIgital Computer* (CORDIC) [8] as the basic processing element (PE). As the CORDIC [9][10] element computes the trigonometric functions and does vectoring through *shift and adds*, avoiding any multiplication, the hardware overhead is drastically reduced and the speed is also enhanced. The algorithm developed through this paper separates the even and odd input samples to achieve a symmetric parallel architecture for the realization of the transform.

Section 2 of this paper develops the theory of CDWT. This section also gives the arrangement of the input signal, the padding pattern, which effectively produces the symmetry of the architecture. Section 3 proposes the array architecture for realizing the transform. The realization of the filter coefficients and the data sequencing is also detailed in this section. Section 4 shows the performance of the proposed architecture.

2. COMPLEX DISCRETE WAVELET TRANSFORM

Rational valued complex kernels realize the CDWT. These kernels can be modelled by **even length** FIR filter with approximate Gabor form, given by:

$$h(n) \approx a_0 e^{-\frac{1}{2}(\frac{n-n_0}{s_0})^2} e^{jw_0(n-n_0)} \qquad (1)$$

$$g(n) \approx a_1 e^{-\frac{1}{2}(\frac{n-n_0}{s_1})^2} e^{jw_1(n-n_0)}$$

With n_0 set to –0.5 to position the Gaussian window symmetrically in the interval *[-D, D-1]*, where *D* is the window half-length, a_0 and a_1 are the magnitude, ω_0 and ω_1 are the modulation frequencies and σ_0, σ_1 are the window standard deviation. The modulation frequencies ω_0 and ω_1 should be complementary, i.e.,

$$\omega_0 + \omega_1 = \pi \qquad (2)$$

to cover the frequency range *[0,p]*. With the Gaussian window functions, the Fourier transforms of *h* and *g* have conjugate symmetry about the modulation frequencies ω_0 and ω_1 respectively. Maximum coverage on frequency range [0, *p*] without significant gaps and with minimal overlap can be achieved by choosing filters with a spacing of *p/6*. The values of omega are *p/6* and 5*p/6* for low and high-pass filters respectively. Figure 1 illustrates the block level diagram for 2-d CDWT

For an input sequence the starting and terminating points of the sequence is of importance since these regions are prone to noise. When images are considered the termination at the start of the sequence introduces high noise level in the transform domain, thus reducing the scope of perfect reconstruction. Zero padding also introduces good amount of noise. Mirror padding is in that respect preferred in most application compared to the other forms of padding. For these reason mirror padding has been adopted here with slight modification for sake of hardware reduction and better symmetry.

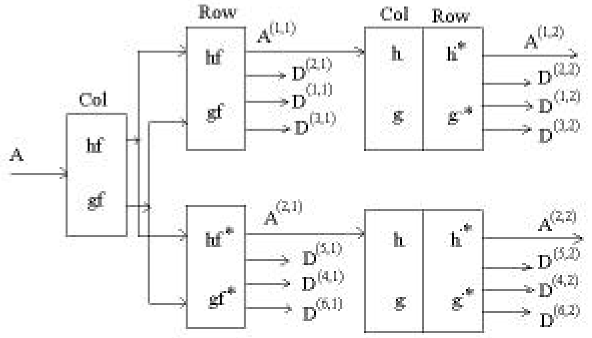

Fig 1. 2-level CDWT for images

The padding has been done in the following form

a_M a_{M-1} ..... a_2 a_1 | a_0 a_1 a_2 .... a_N | a_{N-1} a_{N-2} .... a_{N-M}

Where sequence to the left of a_0 and to the right of a_N is the padding signal. The remarkable feature of this padding is that it is similar to mirror padding except the fact that the edge values (a_0 and a_N) are not repeated in the padding sequence. The advantage of this can be seen in the later part.

The response can be verified for finite input sequence of length N. Let N is equal to 16. The convolution product gives the transfer properties of a digital FIR filter in the time domain

$$y(k) = \sum_{n=0}^{N-1} H(n)\, f(k-n) \qquad (6)$$

Where *N* is the filter length, *f (k)* is the input signal, *H (n)* is the *impulse response* of the filter and *y (n)* is the output signal. The transform relation of the filters is given by:

$$\begin{aligned}
C_0 &= H_0.a_0 + H_1.a_1 + H_2.a_2 + H_3.a_3 + ... +... + H_7.a_7 \\
C_1 &= H_0.a_2 + H_1.a_1 + H_2.a_0 + H_3.a_1 + ... +... + H_7.a_5 \\
C_2 &= H_0.a_4 + H_1.a_3 + H_2.a_2 + H_3.a_1 + H_4.a_0 +.... + H_7.a_3 \qquad (3) \\
&... \; ... \; ... \; ... \; .... \; ... \; ... \; ... \\
C_{11} &= H_0.a_8 + H_1.a_9 + H_2.a_{10} + H_3.a_{11} +...+…+ H_7.a_{15}
\end{aligned}$$

From these equations it can be seen that the even and odd indexed filter coefficients are multiplied with the even and odd samples of the input sequence respectively. This has been achieved using the modified form of padding. This form of padding also has the same performance as the mirror padding as far as the noise level is concerned **.**

3. ALGORITHM OF ARCHITECTURE

This section describes the architectural design of CDWT filter using CORDIC as the basic processing element. The symmetry shown in equation (3) has been exploited in design of the architecture. Based on the symmetry of even and odd samples and filter coefficients the filter structure is divided into two sections.

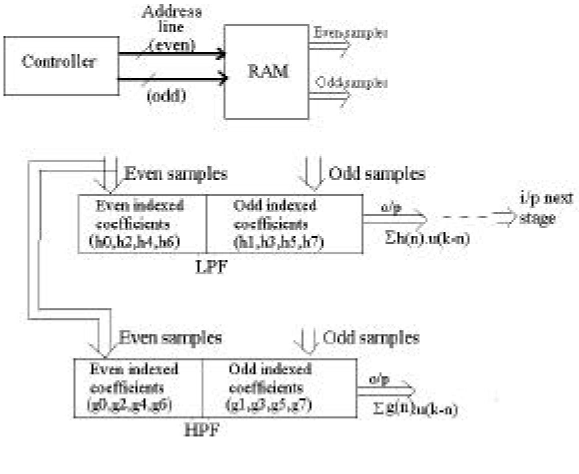
Address
line
(even)
Controller
(odd)
RAM
Even samples
Odd samples
Even samples
Odd samples
Even indexed coefficients (h0,h2,h4,h6)
Odd indexed coefficients (h1,h3,h5,h7)
o/p
Σh(n).u(k-n)
i/p next stage
LPF
Even samples
Odd samples
Even indexed coefficients (g0,g2,g4,g6)
Odd indexed coefficients (g1,g3,g5,g7)
o/p
Σg(n).u(k-n)
HPF

Fig 2. Scheme of architecture for 8-tap CDWT

The basic block level scheme for the realization of CDWT is given in Fig. 2. *h0, h1, h2, h3, h4, h5, h6, h7* are the filter coefficients of the 8-tap CDWT which are realized using CORDIC algorithm. The even filter coefficients take the even samples of the input, while the odd filter coefficients take odd-indexed samples as input.

The controller section generates address for the RAM, which stores the input data (ref. Fig. 2). Address generated by the controller section is of the form shown in table 1.

Filter	h_0, h_2, h_4, h_6	h_1, h_3, h_5, h_7
Clock	Even Samples	Odd Samples
1	a0, a2, a4, a6	a1, a3, a5, a7
2	a2, a0, a2, a4,	a1, a1, a3, a5
3	a4, a2, a0, a2,	a3, a1, a1, a3
4	a6, a4, a2, a0,	a5, a3, a1, a1
…		
11	a10, a12, a14, a14,	a11, a13, a15, a13
12	a8, a10, a12, a14,	a9, a11, a13, a15

Table 1: Order in which input samples are multiplied with filter coefficients

This separation of the input signal samples into even and odd sequence is done by the controller section. The output of the RAM is given to the filter stage, which performs the multiplication operation using CORDIC structure.

As the first sample is given to the CORDIC input, (ref. Fig. 3) the output is obtained after 24 clock cycles. Output of this stage is the product of input sample with the corresponding filter coefficient. This output is added to the product of the odd-indexed filter coefficient and odd input sample and registered. In the next clock the registered output of *Reg9* & *Reg10*, and *Reg 11 & Reg 12* are added and registered at *Reg13* and *Reg14* respectively. In next clock the output of *Reg13* and *Reg14* are added to give the output of the filter.

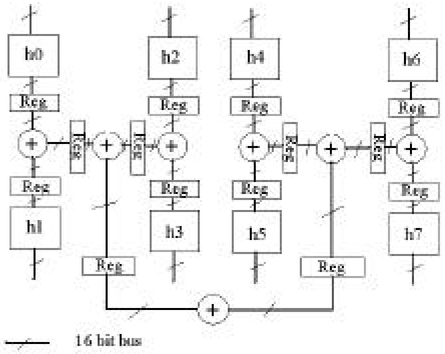

Fig 3. Array Architecture for CDWT filters (LPF section only)

3.1 Filter Design

The basic equations for CDWT (Equation.(1)) can be written in the following form:

$$H(n) = A\,[\cos((n-n_0)\omega_i)+j\sin((n-n_0)\omega_i)] \tag{4}$$

where *i=0* for low pass filter and *i=1* for high pass filter, and

$$A = a_i e^{-\frac{1}{2}\frac{(n-n_0)^2}{s_0^2}} \tag{5}$$

However, the structure of both the equations are same, so equation (4) can be taken as the generalized form for both high and low-pass filters. The amplitude values a_0 and a_1 are taken to be equal and of magnitude 0.5. This has been done without any loss of generality. Thus multiplication with a_0 and a_1 can be achieved using shift operations only. The structure of the filter is shown in Fig. 4.

The filters coefficients are given by:

$$\begin{aligned}
h_0 &= -A\begin{bmatrix}\cos 75^o & \sin 75^o\\ -\sin 75^o & \cos 75^o\end{bmatrix} & h_4 &= A\begin{bmatrix}\cos 15^o & -\sin 15^o\\ \sin 15^o & \cos 15^o\end{bmatrix}\\
h_1 &= A\begin{bmatrix}\cos 75^o & \sin 75^o\\ -\sin 75^o & \cos 75^o\end{bmatrix} & h_5 &= A\begin{bmatrix}\cos 45^o & -\sin 45^o\\ \sin 75^o & \cos 45^o\end{bmatrix}\\
h_2 &= A\begin{bmatrix}\cos 45^o & \sin 45^o\\ -\sin 45^o & \cos 45^o\end{bmatrix} & h_6 &= A\begin{bmatrix}\cos 75^o & -\sin 75^o\\ \sin 75^o & \cos 75^o\end{bmatrix}\\
h_3 &= A\begin{bmatrix}\cos 15^o & \sin 15^o\\ -\sin 15^o & \cos 15^o\end{bmatrix} & h_7 &= -A\begin{bmatrix}\cos 75^o & \sin 75^o\\ -\sin 75^o & \cos 75^o\end{bmatrix}
\end{aligned} \tag{6}$$

By realizing *15*^{o}, *45*^{o} and 75^o angles only, all the other orientations can also be covered. Here the tangent of the angles can be realized by using shift and add at five stages for both 15^o and 75^o, whereas for 45^o it is a one stage process without any shift. The tangent of the angles are given as

15^o (in rad) = $\tan^{-1}(2^{-2}) + \tan^{-1}(2^{-6}) + \tan^{-1}(2^{-10}) + \tan^{-1}(2^{-12})$
75^o (in rad) = $\tan^{-1}(2^{0}) + \tan^{-1}(2^{-1}) + \tan^{-1}(2^{-4}) - \tan^{-1}(2^{-9}) -$
$\tan^{-1}(2^{-11})$

The input sequence can be real or complex. In this paper a complex input sequence is chosen for generalization of the derivations.

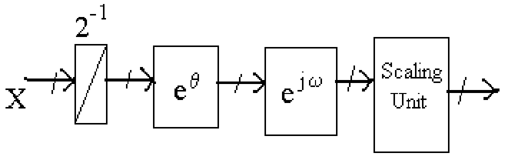

Fig 4. Realization of a filter coefficient

When a real signal is convolved with the complex filter response of the CDWT filters the output is a complex signal. Let the complex input sequence be described as:

$f = f_x + j f_y$

Multiplication of a signal coefficient with a filter coefficient of the form *A(cosq + j sinq)* gives the output as:

$$\begin{bmatrix} F_{re} \\ F_{im} \end{bmatrix} = A \begin{bmatrix} \cos q & \sin q \\ -\sin q & \cos q \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} \qquad (7)$$

where F_{re} is the real part of the output signal and F_{im} is the imaginary part of the signal. The above equation essentially represents a plane rotation operation, which can be efficiently computed by applying CORDIC algorithm.

3.2 Circular CORDIC

In CORDIC technique, the plane rotation through an angle a is achieved by decomposing the target angle into several elementary angles and carrying out rotations through each of these angles as follows [8],[10]:

$$a = \sum_{i=0}^{M-1} d_i q_i \qquad \text{where } q_I = tan^{-1}(2^{-i}) \qquad (8)$$

with M being the word length and $d_i = \pm 1$

Since $q_I /2 < q_{I+1} < q_i$, any arbitrary angle can be expressed in terms of elementary angles q_i with their signs properly chosen. An elementary plane rotation in two dimensions may be expressed as

$$x_{i+1} = cos\ q_i\ (x_i + d_i\ y_i\ tan(q_i))$$
$$y_{i+1} = cos\ (q_i)\ (-d_i\, x_i\ tan\ (q_i) + y_i) \qquad (9)$$

Where the value of δ_i decides the direction of rotation. This expression is identical to equation (7). For an 8-tap filter, the window half-length *D* is equal to 4. Thus the value of n ranges from -4 to 3. For low-pass filters the value of *q*, i.e. the orientation of the filter, is given by $\pm 15^o$, $\pm 45^o$, $\pm 75^o$, $\pm 105^o$.

3.3 Hyperbolic CORDIC

Equation (1) can be broken down into three parts, which are dealt individually. The first part of the equation involves the amplitude coefficients a_0 and a_1. Amplitude is multiplied with a Gaussian distribution function and the product is again multiplied with $e^{(j(n-n0)w)}$. The Gaussian function is expressed as:

$$e^{-\frac{1}{2}\frac{(n-n_0)^2}{s_0^{\,2}}}$$

where sigma is the standard deviation calculated from the frequency bandwidth and orientation bandwidth (ref. appendix A).

Solution of the Gaussian expression can also be done using Hyperbolic CORDIC scheme. In this case the recursion equation is given by:

$$x_{i+1} = x_i + d_i\, y_i\, 2^{-i}$$
$$y_{i+1} = y_i + d_i\, x_i\, 2^{-i} \qquad (10)$$
$$z_{i+1} = z_i - d_i\, e_i$$

for i ³ 1 where $e_i = tanh^{-1}(2^{-i})$

Equation (10) can be realized by using shifters and adders. Structures for circular and hyperbolic CORDIC (equation (9) and (10) are given in Fig 5.

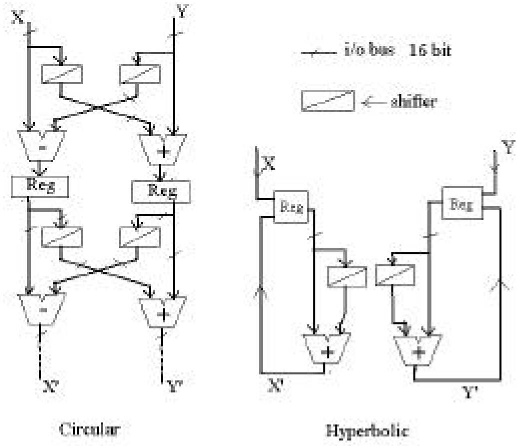

Fig 5. Pipelined CORDIC Architecture

4. PERFORMANCE ANALYSIS

Architecture of CDWT is probably the first of this type so, no comparisons can be provided. Number of clock cycles needed for computation of the CDWT using the proposed architecture for a N-tap filter ($N = 2^x$, x is an integer) is given by log_2N. Thus the time is $O(log_2N)$. The pipelined CORDIC algorithm ensures reduction of hardware overhead and also enhances the speed performance. The latency of the system is 24 (16 for hyperbolic CORDIC + 5 for circular CORDIC + 3), while the throughput is 1. Using pipeline CORDIC, the full parameter space with all possible rotation angles is reduced to five rotation stages for this particular application.

For verification of the algorithm the performance of the architecture is tested for a 4-tap CDWT on FPGA using Xilinx XCV100 series. Figure 6 shows FPGA implementation of 4-tap CDWT. Figure 6(a) shows the schematic for the architecture for 4 tap filter, figure 6(b) shows the circular CORDIC unit for 15° rotation, while scaling unit for the same is shown in figure 6(c). This architecture is found to operate at 27 MHz and gives accuracy up to *11th* bit in the worst case.

The 16-bit machine is used for 12-bit word length operation. A scheme of angle representation is done by choosing the weights as $-\pi$, $\pi/2$, $\pi/4$, ..., $\pi/2^{M-1}$ (M being the word length). The MSB and the last three LSB's are padded with zero for allowing the overflow and reducing the round-off error ($0\ k_0\ k_1\ ....\ k_{11}\ 0\ 0\ 0$. Because of non-availability of literature on architecture for CDWT, comparisons with other architectures cannot be provided.

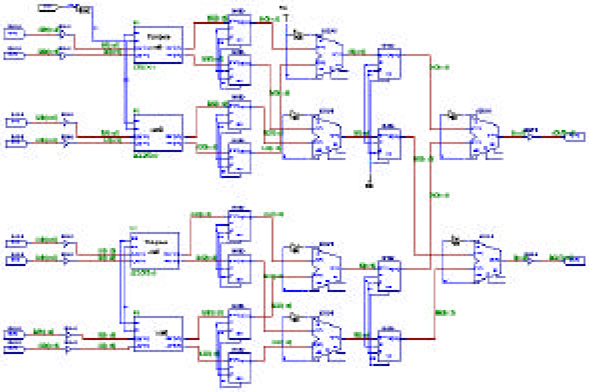

Fig. 6(a). Schematic of 4-tap CDWT filter implemented on Xilinx XCV100 series

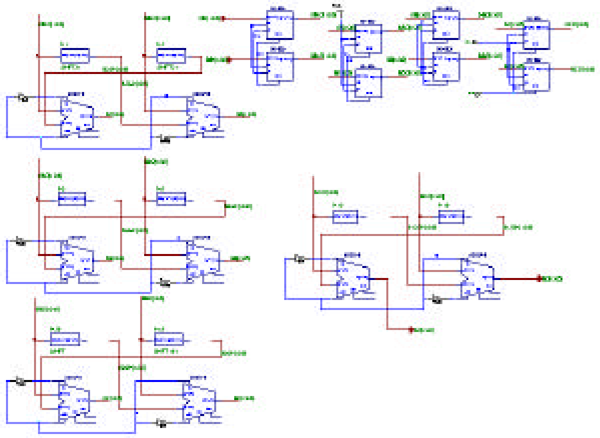

Fig. 6(b). Schematic of circular pipelined CORDIC for achieving 15^o

The algorithm described in this paper exploits the symmetry of the relation between the input samples. This results in a regular and parallel structure of the filter architecture. Parallelism results in low power and high-speed operation. But the tree structure retains a high throughput rate ($1/t_0$=frequency of the circuit) compared to the direct form of filter realization.

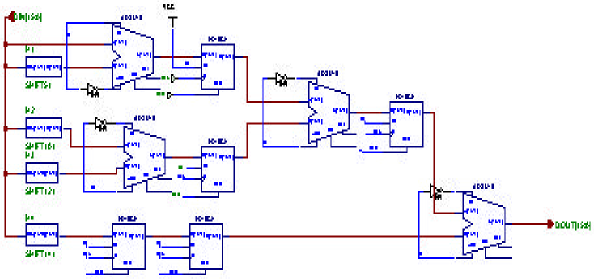

Fig 6 (c). Schematic of Scaling Unit for circular CORDIC of 15^o

4 APPENDIX

5.1 Standard Deviation

Determination of the standard deviation is done from the frequency bandwidth B_r and orientation bandwidth B_θ.

The frequency bandwidth is expressed as:

\begin{equation}

$$B_r = \log_2\left(\frac{n_0 + \sqrt{2\ln 2}s_u}{n_0 - \sqrt{2\ln 2}s_u}\right) \qquad (A.1)$$

Frequency bandwidth, in octaves, from frequency f_1 to frequency f_2 is given by $log_2(f_2/f_1)$. Image array with a width of N_c pixels where N_c is a power of 2, the following values of radial frequency n_0 are used:

1√2, 2√2, 4√2, ….

Taking the half-power consideration, for first octave B_r is equal to 1.

Thus

$$s_u = \frac{1}{3\sqrt{2\ln 2}} n_0 \qquad (A.2)$$

The orientation bandwidth B_θ is given by

$$\tan(\frac{B_q}{2}) = \frac{\sqrt{2\ln 2}s_v}{n_0}$$

It has already been seen that the frequency bandwidth $B_\theta = 30^o$.

6 REFERENCES

[1] Mallat, S.G., A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, IEEE Transactions on Pattern Recognition, vol. 89, pp. 674-693, 1989

[2] Daubechies, I, Ten Lectures on Wavelets, Capital City Press, Montpelier, Vermont, 1995

[3] Magarey, J. and Kingsbury, N.G. Motion estimation using a complex-valued wavelet transform, IEEE Trans. on Signal Processing, special issue on wavelets and filter banks, vol 46, no 4, pp 1069-84, April 1998

[4] Magarey, J. and Dick, A. Multiresolution Stereo Image Matching Using Complex Wavelets. in Proceedings of the 14th International Conference on Pattern Recognition, vol. 1, pp. 4-7, 1998

[5] Pan, H.P. General Stereo Image Matching Using Symmetric Complex Wavelets. in Proceedings of the SPIE Conference: Wavelet Application in Signal and Image Processing, vol. 2825, 1996

[6] Magarey, J. and Dick, N.G. An Improved Motion Estimation Algorithm Using Complex Wavelets, in

Proceedings IEEE International Conference on Image Processing, pp. 969-972, September 1996

[7] Kung, S.Y., VLSI Array Processors, Englewood Cliffs, NJ, Prentice Hall, 1986

[8] Deprettere, J.M., and Udo, R., The Pipelined CORDIC, International Rep. Network Theory Section, Delft Univ., Technol., 1983

[9] Haviland, G.L. and Tusznski, A.A., A CORDIC Arithmetic Processor Chip, IEEE Transactions on Computers. Vol C-29 (2), pp. 68-79, 1980

[10] Hsiao, S.F. and Delosome, J.M., Householder CORDIC Algorithm, IEEE Transactions on Computers, vol. 44 (8), pp. 990-1001, 1995