AREA-EFFICIENT NEDA ARCHITECTURE FOR THE 1-D DCT/IDCT

Archana Chidanandan

Rose-Hulman Institute of Technology,
Computer Science and Software Engineering,
Terre Haute, IN, U.S.A.

Magdy Bayoumi

University of Louisiana at Lafayette
Center for Advanced Computer Studies
Lafayette, LA, U.S.A.

ABSTRACT

New Distributed Arithmetic has been been applied to the 1-D DCT to produce a low power, high throughput architecture. In this paper, we apply NEDA to the even-odd decomposition matrices of the 8 x 8 forward and inverse DCT. We show that, with the proposed approach, the number of adders required for the adder array for the forward DCT and the inverse DCT is fewer than required if NEDA is applied directly to the 8 x 8 DCT and IDCT matrices. This reduction will result in power savings, without decreasing the throughput. Also, for the inverse DCT, the number of adder stages is reduced, resulting in faster decoding.

1. INTRODUCTION

In [1, 2, 3], the idea of New Distributed Arithmetic or NEDA is introduced and defined. The DCT is used as an example of how NEDA can be applied. It is shown that by distributing the bits of the constant DCT coefficients, it is possible to perform the DCT operation with just addition operations. The ROMS that are required in DA techniques and the multiply operations required in MAC architectures are replaced by additions. This results in a low power, high throughput architecture for the DCT. Further, by applying a compression algorithm which removes the redundant addition operations, the number of adders was further reduced. In [1, 2, 3], NEDA is applied to the 8 x 8 DCT/IDCT matrices directly.

In this paper, we propose the application of NEDA to the even-odd decomposition matrices of the 8 x 8 DCT/IDCT matrices. The even-odd decomposition results in 4 x 4 matrices. Applying NEDA to these matrices, we will show, reduces the number of adders required, especially in the case of the inverse DCT. We first describe the even-odd decomposition of the DCT/IDCT. In the section following that, we describe how NEDA can be applied to the smaller matrices. We examine the number of adders required and compare with the results obtained when applying NEDA to the 8 x 8 DCT/IDCT matrix directly and other architectures then summarize our conclusions.

2. EVEN-ODD DECOMPOSITION OF THE 8 X 8 DCT AND IDCT

The even-odd decomposition of the 8 x 8 DCT matrix has been described previously in [4, 5] and others. Assume that

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{bmatrix} = \sqrt{\frac{2}{N}} \begin{bmatrix} cos\frac{\pi}{4} \\ cos\frac{\pi}{16} \\ cos\frac{\pi}{8} \\ cos\frac{3\pi}{16} \\ cos\frac{5\pi}{16} \\ cos\frac{3\pi}{8} \\ cos\frac{7\pi}{16} \end{bmatrix} \quad (1)$$

Then the even-odd decomposition of the 8 x 8 DCT matrix can be written as

$$\begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ c & f & -f & a \\ a & -a & -a & a \\ f & -c & c & -f \end{bmatrix} \begin{bmatrix} X(0)+X(7) \\ X(1)+X(6) \\ X(2)+X(5) \\ X(3)+X(4) \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} = \begin{bmatrix} b & d & e & f \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} X(0)-X(7) \\ X(1)-X(6) \\ X(2)-X(5) \\ X(3)-X(4) \end{bmatrix} \quad (3)$$

X(0), X(1), ... X(7) are the inputs and Y(0), Y(1), ... Y(7) are the transformed values.

The 8 x 8 DCT matrix has been replaced by two 4 x 4 matrices, which can be computed in parallel.

The IDCT is computed as follows:

$$\begin{bmatrix} Z(0) \\ Z(1) \\ Z(2) \\ Z(3) \end{bmatrix} = \begin{bmatrix} a & c & a & f \\ a & f & -a & -c \\ a & -f & -a & c \\ a & -c & a & -f \end{bmatrix} \begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} \quad (4)$$

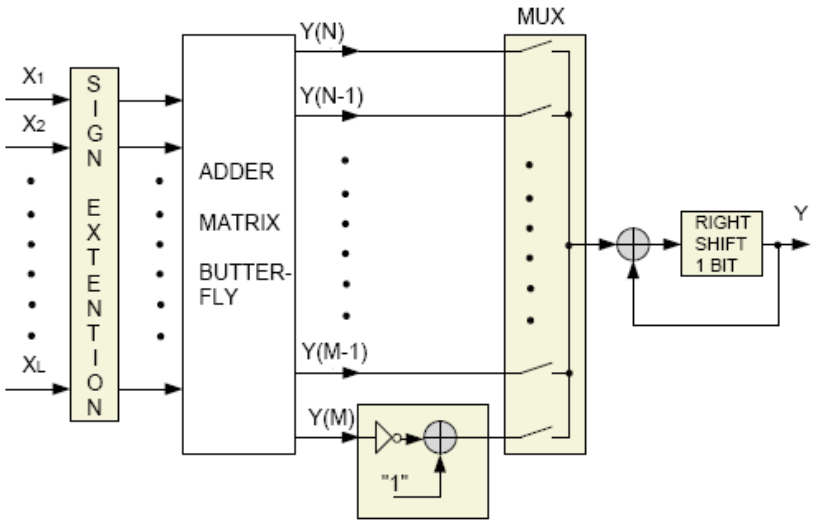

Fig. 1. NEDA Architecture with Only Addition Operations[3]

$$\begin{bmatrix} Z(1) \\ Z(3) \\ Z(5) \\ Z(7) \end{bmatrix} = \begin{bmatrix} b & d & e & f \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} \quad (5)$$

Y(0), Y(1), . . . Y(7) are the DCT transformed values again and Z(0), Z(1), . . . Z(7) are the values obtained after the inverse transform.

3. APPLYING NEDA TO THE 4 X 4 MATRICES

In NEDA, the bits of the constant DCT coefficients are distributed, unlike Distributed Arithmetic architectures where the bits of the input values are fed in a bit-serial manner. Distributing the DCT coefficients allows us to feed the input values in parallel and this results in a speedup in the processing. Also, all multiplication operations are replaced by add and shift operations. Figure 1 shows the NEDA architecture to calculate the inner product.

If we choose to represent the signed coefficient values in 13 bits(NEDA internal precision, P), then for each row of the DCT matrix, we have an equivalent P x M adder array matrix created where P is the NEDA precision and M is the number of columns in the DCT array. Since, in the proposed work, NEDA is being applied to the matrices decomposed into even-odd rows, and the NEDA precision chosen is 13, there are four 13 x 4 matrices generated.

By examining the four adder matrices, and applying the compression algorithm of [3], all the possible combinations of additions can be determined. All redundant addition operations can be removed to obtain the final number of addition operations.

3.1. Forward DCT

The adder matrix for row 1 of the DCT matrix in equation 2 is shown below:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

where row 1 represents the LSB and row 13 represents the sign-bit.

By examining the four adder matrices obtained from the matrix in equation 2, we determined that there are 11 unique addition operations required, as listed in table 1. Similarly, the 4 x 4 matrix in equation 3, resulted in 15 unique addition operations, as shown in table 2.

Figure 2 shows the adder matrix butterfly structure for row 1 of the matrix in equation 2.

Stage	Addition combinations
1	h, i, j, k
2	h + i, h + j, i + k, j + k, i + j, k + h
3	h + i + j + k

Table 1. Additions required for the matrix in equation 2. Here, h = X(0) + X(7), i = X(1) + X(6), j = X(2) + X(5), k = X(3) + X(4)

Stage	Addition combinations
1	p, q, r, s
2	p + q, r + s, q + s, p + s, p + s, p + r, q + r
3	p + q + s, p + q + r, p + q + r + s, p + r + s, q + r + s

Table 2. Additions required for the matrix in equation 3. Here, p = X(0) - X(7), q = X(1) - X(6), r = X(2) - X(5) and s = X(3) - X(4)

3.2. Inverse DCT

The even-odd decomposition of the inverse DCT also results in two 4 x 4 matrices as shown in equations 4 and 5. For the same NEDA internal precision, we determine the number of additions required. Tables 3 and 4 show the number of add operations required.

Note that the matrices in equations 3 and 5 are identical. Note also that for the IDCT only two levels are additions are required.

4. COMPARISON WITH NEDA APPLIED TO THE ORIGINAL 8 X 8 DCT AND IDCT MATRICES

In table 5, we show the total number of adders required with the proposed method and with NEDA applied to the original 8 x 8 DCT/IDCT matrix. It can be seen that in both cases the number of adders required is fewer and the overall savings is 40%.

Also, for the IDCT we have noted that the stages of additions required is reduced by one. This implies that the decoder will be faster than with the direct application, which is an essential requirement for the decoder.

Stage	Addition combinations
1	l, m, n, o, t, u, v
2	n + Y(6), m + Y(6), v + Y(6), n + Y(4), m + o

Table 3. Additions required for the matrix in equation 2. Here, l = Y(4) + Y(6), m = Y(2) + Y(4), n = Y(0) + Y(2), o = Y(0) + Y(6), t = Y(2) + Y(4), u = Y(2) + Y(6), v = Y(0) + Y(4)

Stage	Addition combinations
1	w + x, y + z, x + z, w + z, w + z, w + y, x + y
2	w + x + z, w + x + r, w + x + y + z, w + y + z, x + y + z

Table 4. Additions required for the matrix in equation 5. Here, w = Y(1), x = Y(3), y = Y(5), z = Y(7)

5. COMPARISON WITH OTHER ARCHITECTURES

Once the partial products have been generated using NEDA, they have to be shifted and added together to generate the final values. To perform these additions, we will need 16 adders (8 for the DCT and 8 for the IDCT). This will result in a total of 57 + 16 = 73 adders. In table 6, we compare our results with that of other reported 8 x 8 DCT and IDCT architectures.

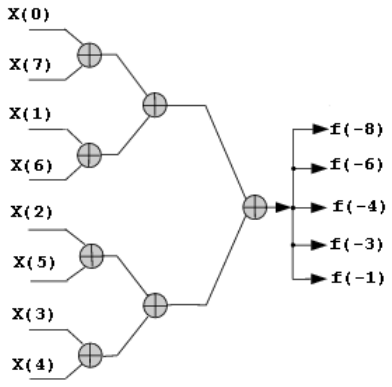

Fig. 2. Adder matrix butterfly structure for row 1 of matrix of equation 2. Here, f(-1), f(-3), etc are the partial products.

	Direct NEDA implementation	Proposed implementation
DCT	35	26
IDCT	60	31
Overall	95	57
Savings	-	40%

Table 5. Number of adders for the DCT and IDCT

	Madisseti et al[5]	Fanucci et al[6]	Proposed implementation
Number of multipliers	14	0	0
Number of adders	32	16	73
Number of ROM words	0	128	0

Table 6. Comparison with other 8 x 8 DCT/IDCT architectures

6. CONCLUSIONS

Using the even-odd decomposition of the 8 x 8 DCT/IDCT matrix, we obtain four 4 x 4 matrices. We applied NEDA to these four matrices, using a NEDA internal precision of 13. This resulted in eight 13 x 8 matrices. As is the case with NEDA, these matrices represent the addition operations that have to be performed to generate the DCT or the IDCT of input values. No multiplication is required, also no ROMs are required. By applying NEDA to the 4 x 4 matrices, instead of the 8 x 8 matrix directly, it is determined that there is a 40% reduction in the number of adders. Also, in the decoder with the proposed method, the number of adder stages is fewer. Therefore, applying NEDA to the 4 x 4 matrices results in smaller area, lower power and higher throughput.

7. REFERENCES

[1] A. Shams and M. Bayoumi, "A 1.8Gbps, 1.5GHz 1D-DCT architecture," in *Applications-specific Systems, Architectures, and Processors (ASAP 2000)*, 2000.

[2] A. Shams, W. Pan, A. Chidanandan, and M. Bayoumi, "A low power high performance Distributed DCT architecture," in *2002 IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2002)*. IEEE, 2002, pp. 26–34.

[3] A. Shams, A. Chidanandan, W. Pan, and M. Bayoumi, "NEDA: A low power high throughput DCT architecture," *to be published in the IEEE Transactions on Signal Processing*.

[4] S. Yu and E.E. Swartzlander Jr., "DCT implementation with Distributed Arithmetic," *IEEE Transactions on Computers*, vol. 50, pp. 985–991, Sept. 2001.

[5] A. Madisetti and A. N. Willson, "A 100 MHz 2-D 8 x 8 DCT/IDCT processor for HDTV application," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 158–164, Apr. 2002.

[6] L. Fanucci and S. Saponara, "Data driven VLSI computation for low power DCT-based video coding," in *9th International Conf. Electronics, Circuits and Systems*. IEEE, 2002, vol. 1, pp. 541–544.