SECTION 7

Two Dimensional Fourier and Cosine Transforms

"To implement Eqn. 7-1, a two dimensional time sequence is decomposed according to its row or column."

Two dimensional Fourier transforms are widely used in image processing, image analysis, and video compression. Because the fast discrete cosine transform features high energy compaction and low implementing complexity, it is becoming more and more important in image and video compression.

7.1 Two Dimensional FFTs on the DSP96002

Two dimensional FFTs are simply an extension of one dimensional FFTs, and is shown by:

$$F(i,k) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) e^{(-j(2\pi mi))/N} e^{(-j(2\pi nk))/N}$$

Eqn. 7-1

where: i = 0,1,...N-1

k= 0,1,...N-1

To implement Eqn. 7-1, a two dimensional time sequence is decomposed according to its row or column. Eqn. 7-1 can be rewritten in Eqn. 7-2.

$$F(i, k) = \sum_{m=0}^{N-1}\left(\sum_{n=0}^{N-1} x(m, n)e^{(-j(2\pi mi))/N}\right)e^{(-j(2\pi nk))/N}$$

Eqn. 7-2

The one dimensional FFT code discussed in **SECTION 4** can be used in this extension. The code included on the Motorola DSP bulletin board (2DFFT.asm) implements the two-dimensional DIT FFT by calling subroutine CFFT96.asm N times, if an N by N 2D FFT is to be performed. Also, the code demonstrates the implementation of a double buffer by the DMA controllers on the DSP96002 as discussed in **SECTION 5**.

7.2 Discrete Cosine Transform on the DSP96002

7.2.1 One Dimensional Discrete Cosine Transform (DCT)

The one dimensional cosine transform of a discrete time sequence x(n), n = 0,1,...,N-1 is defined as:

$$F(k) = \frac{2c(k)}{N}\sum_{n=0}^{N-1} x(n)\cos\left[\frac{(2n+1k\pi)}{2N}\right];\ k = 0, 1, \ldots, N-1$$

Eqn. 7-3

where:

$$\begin{aligned} c(k) &= \frac{1}{\sqrt{2}}, \quad k = 0 \\ &= 1, \quad k = 1, 2, \ldots, N-1 \\ &= 0, \quad \text{elsewhere} \end{aligned}$$

and the inverse transform is:

$$x(n) = \sum_{n=0}^{N-1} c(k)F(k)\cos\left[\frac{(2n+1k\pi)}{2N}\right];\ n = 0, 1, \ldots, N-1$$

Eqn. 7-4

A fast discrete cosine transform (FDCT) proposed by Chen and Smith [see reference 1] is adapted in this application note, and it's flow diagram is plotted in Figure 7-1. Many optimized implementations on the FDCT have been published. The code given on the Motorola DSP bulletin board is not fully optimized; it simply demonstrates the simplicity of the DSP96002 assembly code.

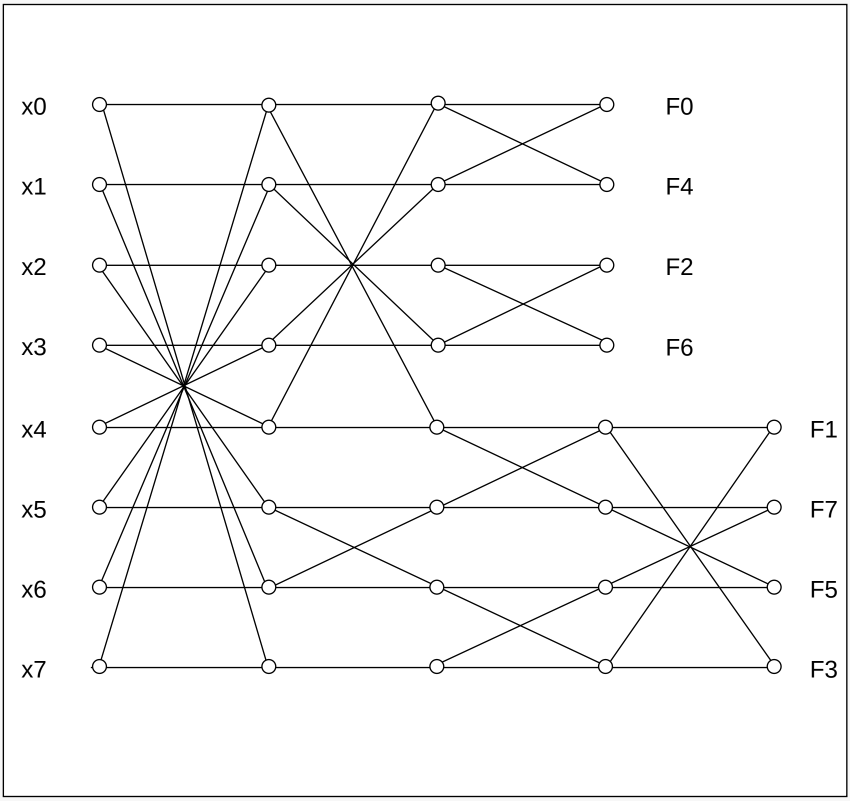

Figure 7-1 The flow diagram of an 8-point discrete cosine transform. Note that the output order of the transform is scrambled.

For $N=2^m$, $m > 2$, this algorithm requires:

$(3N/2)(\log_2 N-1)+2$ real additions and

$N\log_2 N - (3N/2)+4$ real multiplications.

7.2.2 Two Dimensional DCT

A one dimensional DCT can be easily extended to a two dimensional DCT as shown in .

$$F(j,k) = \frac{4c(j)c(k)}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)\cos\left[\frac{(2n+1)k\pi}{2N}\right] \times \cos\left[\frac{(2m+1)j\pi}{2N}\right]$$

Eqn. 7-5

Therefore, to calculate an N by N 2D DCT, repeat the N-point 1D DCT N times. An 8x8 2D DCT assembly code for the DSP96002 (DCT.asm) is presented on the Motorola DSP bulletin board . ■