Efficient Implementation of the Two Dimensional Discrete Cosine Transform for Image Coding applications on the DSP96002 Processor

Srinath V. Ramaswamy and Gerald D. Miller
Department of Electrical Engineering
Northern Illinois University, DeKalb, Illinois 60115

Abstract--This paper describes the efficient implementation of the 2-D DCT for image coding on the DSP96002 processor. The DSP96002 is a general purpose, dual-bus IEEE floating point digital signal processor. Utilizing the DSP96002's inherent parallel processing capabilities, the execution of a 8 x 8 fast 2-D DCT takes 133 microseconds. The recently proposed 2-D and 1-D fast DCT algorithms are employed in this implementation. Transform coefficient zigzag ordering, used in the image coding process, executes in less than 28 microseconds. The fast DSP96002 routines, incorporated within this implementation, can be applied in a number of image coding applications such as video and still image coding as well as the newer JPEG still image and MPEG video standards.

INTRODUCTION

The Discrete Cosine Transform(DCT) is widely used in digital signal processing applications such as speech and image compression. The DCT's performance approaches that of the, difficult to implement, statistically optimal Karhunen-Loeve Transform [1]. The DCT application to the compression of a highly correlated image results in the image's energy being packed into fewer coefficients than other transform approaches. The DCT's adaptation, in the Joint Photographer Expert Group's(JPEG) and Motion Picture Experts Group's(MPEG) image compression standards, as a primary compression tool has added to its importance [2].

The DCT's popularity has stimulated the development of several fast 2-D DCT algorithms. Notable among these are the algorithms developed by Vetrelli, Duhamel, Hou [1] and recently by Cho and Lee[3]. These fast 2-D DCT algorithm are themselves generated by 1-D DCTs. The 2-d DCT's execution time is especially important in several real time applications such as video teleconferencing and video compression. The 2-D DCT's exhibit slow execution times on general purpose processors when compared to their execution on DSPs or application specific chips. However because the 2-D DCT is often used in conjunction with numerous other DSP algorithms, the usefulness of specialized DCT chips is greatly diminished[4].

The fast 2-D DCT algorithm proposed by Cho and Lee is one of the fastest algorithm proposed to date. This algorithm's highly regular and systematic computational structure lends itself to DSP implementation. This fast DCT algorithm is implemented on Motorala's DSP96002 processor. The DSP96002 processor is a general purpose, IEEE floating point DSP that provides several hardware features and software instructions which aid in fast algorithm implementations.

In image coding applications, the digital image is usually partitioned into 8 x 8 pixel blocks. These 8 x 8 pixel blocks are then sequentially compressed/decompressed. The coefficients generated by Cho and Lee's fast 2-D DCT algorithm for an 8 x 8 input pixel block are entropy coded. To facilitate entropy coding, the coefficients are rearranged into a zigzag order. This procedure arranges the coefficients in the order of increasing frequency. The 2-D DCT's memory requirements and execution time as well as zigzag ordering are addressed in this paper.

DSP96002 CHARACTERISTICS

The DSP96002 processor possesses several architectural features that allow efficient fast transform implementations[5,6]. The DSP96002 has a dual Harvard architecture and conforms to the IEEE 754 floating point standard. The X and Y data memories can be simultaneously accessed via separate address and data buses. By efficiently utilizing the two data memories, processor throughput is significantly increased. The three (3) DSP96002 CPU execution units, namely the Data Arithmetic Logic, Address Generation and Program Control Units, operate in parallel. This organization allows parallel data moves together with concurrent ALU operations. For example, results can be stored from and loaded in the data registers simultaneously with ALU operations. In addition, the on-chip dual channel DMA controller also permits block oriented I/O operations in parallel with CPU executions.

FAST 2-D DCT ALGORITHMAND and its IMPLEMENTATION

Cho and Lee's fast 2-D DCT algorithm was selected for implementation[3]. Because of this algorithm's highly regular and systematic structure, it is more suitable for VLSI implementation than many other algorithms. This fast algorithm generates the 2-D DCT of an 8 x 8 input sequence by means of eight 1-D DCTs, thereby reducing the number of multiplications by 50%. However, from a DSP perspective, the multiplication operation takes the same amount of time as the addition operation. Therefore, the reduction in the number of multiplications is not the primary criteria for this algorithm's selection. Its attractiveness is its

CH 3381-1/93/$01.00 ©1993 IEEE

highly regular and systematic structure which, computationally, involves numerous butterfly computations.

The butterfly computation is illustrated in Figure 1. The x and y input data, and the z and w results are stored respectively in the X and Y memory spectrums. By using the FADDSUB instruction, the sum and difference of x and y can be simultaneously computed. By means of parallel data move operations, the results can be stored, and the next set of input data can be concurrently retrieved as well. since the butterfly's computation complexity is equivalent to only three instructions, it forms the basis for fast implementations.

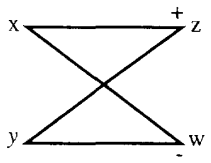

Figure 1. Butterfly Structure

The Signal Flow Graphs(SFGs) utilized by Cho and Lee's fast 2-D DCT algorithm are illustrated in Figures 2, 3 and 4. The fast 2-D DCT algorithm's execution time is further reduced by utilizing a fast 1-D DCT algorithm to implement the DCT modules as illustrated in Figure 2. B. G. Lee's proposed fast 1-D DCT algorithm [7] is utilized in this paper. The DCT module's input data for the f_{pl} computations is stored in the X memory space, while the input data for the g_{pl} computations is stored in the Y memory space. These two memory spaces are simultaneously accessed by the DSP96002. The DSP96002 can also perform a multiplication, addition and subtraction in a single instruction. This capability is used to concurrently execute two DCT modules: one computes f_{pl} while the other computes g_{pl}.

The SFGs illustrate that a complicated data access scheme is required. Some of these complications are simplified by utilizing the addressing modes and modifiers available on the DSP96002 processor. For example, the input SFG values, in Figure 1 have indices consistent with the following set of equations:

$$j(p;a) = (pi + (p-1)/2)\ modulus 8$$

or (1)

$$j(p;b)) = 7 - ((pi + (p-1)/2)\ modulus 8)$$

for $p = 1, 3, 5, \cdots, 7$ $i = 0, 1, 2, \cdots, 7$

The DSP96002's modulo arithmetic addressing facility is used to efficiently implement equation 1. The large number of internal registers makes it feasible to in-place reorder some of the input data. B. G. Lee's fast 1-D DCT outputs bit-reversed ordered data, which is efficiently unscrambled using bit-reversed address modifiers.

The DCT module's reordered output coefficients are distributed between the two separate X and Y data memories. This scenario efficiently implements the computational structure illustrated in Figures 3 and 4. Figures 3 and 4 SFG's output coefficients are rearranged utilizing DSP96002's addressing capabilities. If the scale factors are ignored, the inverse 2-D DCT's signal flow graph is the same as the forward transform. Consequently, the inverse transform is similarly implemented except for some necessary scaling factor modifications. Nested hardware DO loops are used instead of straight line coding since they involve little overhead.

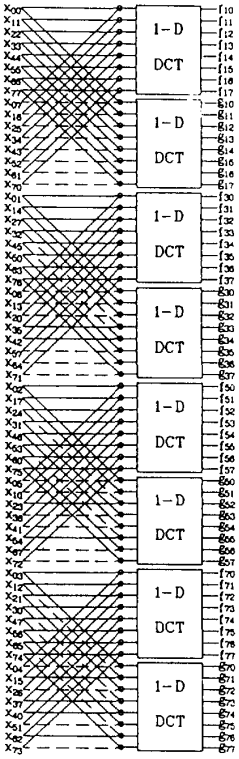

Figure 2. Signal Flow Graph from x_{ij} to f_{pl} and g_{pl}

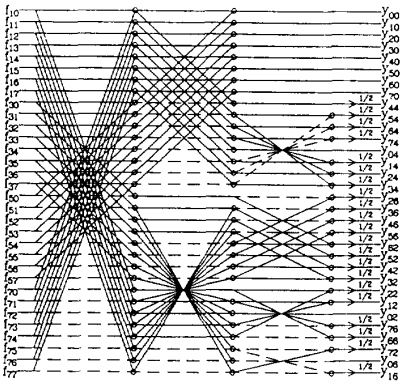

Figure 3. Signal Flow Graph from f_{pl} to Y_{mn}, Where n is even

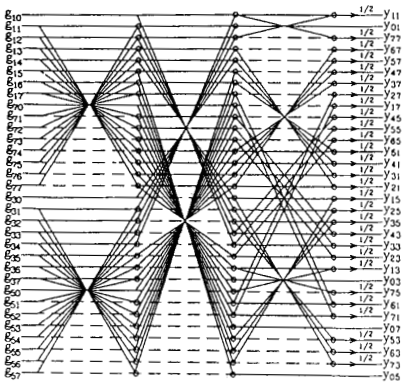

Figure 4. Signal Flow Graph from g_{pl} to Y_{mn}, where n is odd

EFFICIENT ZIGZAG CODING IMPLEMENTATION

The coefficients generated by the 2-D DCT are rearranged into a zigzag order as shown in Figure 5. Prior to zigzag reordering, the coefficients are quantized using a quantization table.

The two-dimensional array elements are stored horizontally in rows. The elements are reordered in a zigzag fashion as dictated by the lines connecting the elements. Each 8 x 8 array element is denoted by $x_{i,j}$, where i indicates the row and j indicates the column. The adjacent line elements are denoted by $x_{i,j}$ and $x_{i+1,j-1}$. These adjacent line elements are separated by a constant distance:

$$d = [(i-1) - (i)] \times N + [(j-1) - (j)] \qquad (2)$$

which reduces to:

$$d = N - 1$$

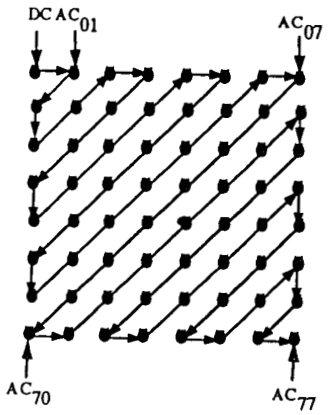

Figure 5. Zig Zag Sequence

An inspection of alternate diagonal lines in Figure 5, reveals that they contain an even/odd number of elements. Lines which contain an even number of points are directed downward to the left while those containing an odd number of points are directed upward to the right. Consequently, a single loop is constructed for reordering elements in a line with an even number of elements, and a second loop in constructed of lines with an odd number of elements. By using the DSP96002's register indirect addressing capabilities, where the offset register is loaded with constant distance d derived above, the zigzag ordering of the two-dimensional array of elements was implemented efficiently and quickly. A line with an even number of elements is referenced using a post increment with an offset. A line with an odd number of elements if referenced using a post decrement with an offset. The zigzag ordering facilitates the entropy coding process by rearranging the coefficients in the order of increasing frequency.

EXTERNAL DATA TRANSFERS

The externally stored 8 x 8 pixel blocks are sequentially transferred to the DSP96002's internal memory utilizing the on-chip dual channel DMA controller. Similarly, the coefficients generated by the 2-D DCT routine are transferred to the external memory via the DMA. Since the DMA controller and CPU operate concurrently, these two transfers can take place in parallel. Therefore, the routines which execute the image sample coefficient transfers between the internal and external memories involve very low overhead.

RESULTS

The forward and reverse 8 x 8 DCT execution times and memory requirements are summarized in Table 1. In contrast to the forward transform, some of the inverse transform instructions do not utilize the DSP96002's inherent parallelism. These implementation considerations account for the difference between the program word requirements of the forward and inverse transform computations.

Routine	Memory Requirements			Execution Time (micro-seconds)
	Program Words	X Space	Y Space	
Forward 8 x 8 DCT	629	192	72	133.3
Inverse 8 x 8 DCT	682	200	256	131.05
Forward zigzag	58	64	64	27.85
Inverse zigazg	60	64	64	27.85

Table 1. Memory Requirements and Execution Times of the DCT and Zigazg Routines

The zigzag routine's execution time and memory requirements, associated with the 2-D DCT routine, are also provided in Table 1. The forward and inverse DCT transform with the zigzag feature results in total computation times of 161 and 159 microseconds respectively (with a 40 MHz clock). Since the DSP96002 contains 1024 words of full speed on-chip program memory, it can accommodate and quickly execute both the forward and inverse DCT routines which incorporate the zigzag feature.

SUMMARY

The DSP96002 is a general purpose, IEEE floating point digital signal processor containing hardware and software features which lend themselves to the efficient implementation of fast DCTs. An efficient implementation of the 8 x 8 pixel 2-D DCT and zigzag ordering have been described. These implementations use the most recently proposed fast DCT algorithms. The fast DCT and zigzag routines execute in 133 and 28 microseconds respectively. These fast routines can be utilized in image coding applications; these routines are especially useful in those applications which adhere to the JPEG /MPEG image coding standards.

REFERENCES

[1] Ran, K. R. and Yap, P. "Discrete Cosine Transform. Algorithm, Advantages, Applications." *Academic Press,* San Diego. 1990

[2] Watchcases G. K. "The JPEG still Picture Compression Standard." *IEEE Transactions on Consumer Electronics.* Vol.38 No.1, February 1992, pp.xvii-sixties.

[3] Cho, N. I. and Lee, S. U. "Fast Algorithm and IMplementation of 2-D discrete Cosine Transform." *IEEE Transactions on Circuits and Systems.* Vol.38. No.3, March 1991, pp297-305.

[4] David Shear, "EDN's DSP Benchmarks" *EDN.* September 29,1988.

[5] *DSP96002, IEEE Floating-Point Dual-Port Processor User's Manual.* MOtorala, inc., 1989.

[6] Guy R. L. solei and Kevin L. Kookier, "A Digital Signal Processor with IEEE Floating-Point Arithmetic". *IEEE Micro. Vol.*8, No.6, December 1988, pop. 49-67.

[7] Lee, B. G. "A New Algorithm to Compute the Discrete Cosine Transform." *IEEE Trans. Acoustics, Speech, and Signal Processing.* Val. ASSP-32. No.6, December 1984, pop 1243-1245.