EPLD-BASED ARCHITECTURE OF REAL TIME 2D-DISCRETE COSINE TRANSFORM AND QUANTIZATION FOR IMAGE COMPRESSION

S. Ramachandran, S. Srinivasan* and R. Chen***
*Department of Electrical Engineering, Indian Institute of Technology, Chennai - 600 036, India
**Electrical Engineering Department, San Jose State University, San Jose, CA 95192, USA

ABSTRACT

This paper proposes a novel implementation of Two Dimensional Discrete Cosine Transform (2D-DCT) and Quantization (Q) using an Embedded Programmable Logic Device (EPLD). The salient features of this scheme are that its architecture is regular, linear, highly pipelined and it fits into just one piece of a commercially available EPLD. It is capable of processing images of size 1024 X 768 pixels at the rate of 25 frames per second. The chip does not require any extra hardware for interfacing and can be used directly in conjunction with other processors. The architecture is realized as a modular implementation using VHDL and Schematic Packages. The hardware complexity, speed and accuracy of the proposed DCT and Quantization processor compare favorably with those of other known implementations.

1. INTRODUCTION

Video standards such as HDTV coding, JPEG and MPEG use 2D-DCT and quantization for effective image compression. The DCT is computationally very intensive and needs an efficient and a very large scale integration implementation to achieve real time speeds. Many attempts have been made at efficient mapping of DCT algorithms into VLSI chips.

In the architecture proposed by Sun-Chen et.al. [1], all the multipliers are replaced by ROMs, and the number of ROMs required is large. For example, an 8 x 8 DCT realization requires more than forty ROMs of size, 1k x10 bits. Furthermore, a very stringent utilization of VLSI technology is required for the design to meet the specified speed criteria. Direct mapping of Fast Cosine Transform (FCT) algorithm using SIMD architecture is presented in [2]. This implementation uses a large number of switching networks and processing elements to achieve real time speeds. Cho and Lee [3] have implemented 2D-DCT using only six bit precision for cos coefficients. With increase in precision, the processing speed decreases drastically.

A linear, highly pipelined, parallel architecture has been proposed in this present work for the implementation of 2D-DCT and Quantization on an EPLD. This architecture suffers from none of the limitations cited in the earlier references. The present work is an improvement over that of Murthy et. al. [4] as regard to speed. Furthermore, the present work incorporates the additional feature of quantization whereas Murthy et. al. have implemented only the DCT.

This paper is organized as follows. The next section presents a parallel algorithm for evaluating DCT/Q efficiently. Section 3 deals with the architecture of the scheme and Section 4 with the implementation of the design on an EPLD. Results are discussed in Section 5 and conclusions are presented in Section 6.

2. ALGORITHM FOR PARALLEL MATRIX MULTIPLICATION OF DCT/Q

DCT is an orthogonal transform consisting of a set of vectors that are sampled cosine functions [5]. 2D-DCT of a block of size 8 x 8 pixels is defined as

$$Z = C\,X\,C^T, \qquad (2.1)$$

where X is the input image matrix, C, the cosine coefficient matrix and C^T, its transpose. In order to achieve a regular and efficient method of implementation, a parallel matrix multiplication algorithm has been proposed in [4] and the same is re-arranged to suit the implementation of the design in the present work.

The two stage matrix mulplication of Eq. 2.1 can be implemented by a parallel architecture wherein eight partial products, which are the row vectors of CX generated in the first stage, are fed to the second stage. Subsequently, eight DCT coefficients, corresponding to a row of $C\,X\,C^T$, are generated by multiplying row vectors of CX by the C^T matrix. While computing the '(i+1) th' partial products of CX, the 'i th' row DCT coefficients can also be computed simultaneously since the 'i th' partial products of CX are already available. Quantized outputs can be obtained by dividing each of the 64 DCT coefficients by the corresponding quantization table values. These stages can be pipelined in such a way that the entire matrix multiplication is completed in 79 clock cycles.

3. ARCHITECTURE

The basic architecture of the proposed DCT and quantization (DCT/Q) processor is shown in Fig. 3.1. It

0-7803-5471-0/99/$10.00©1999 IEEE

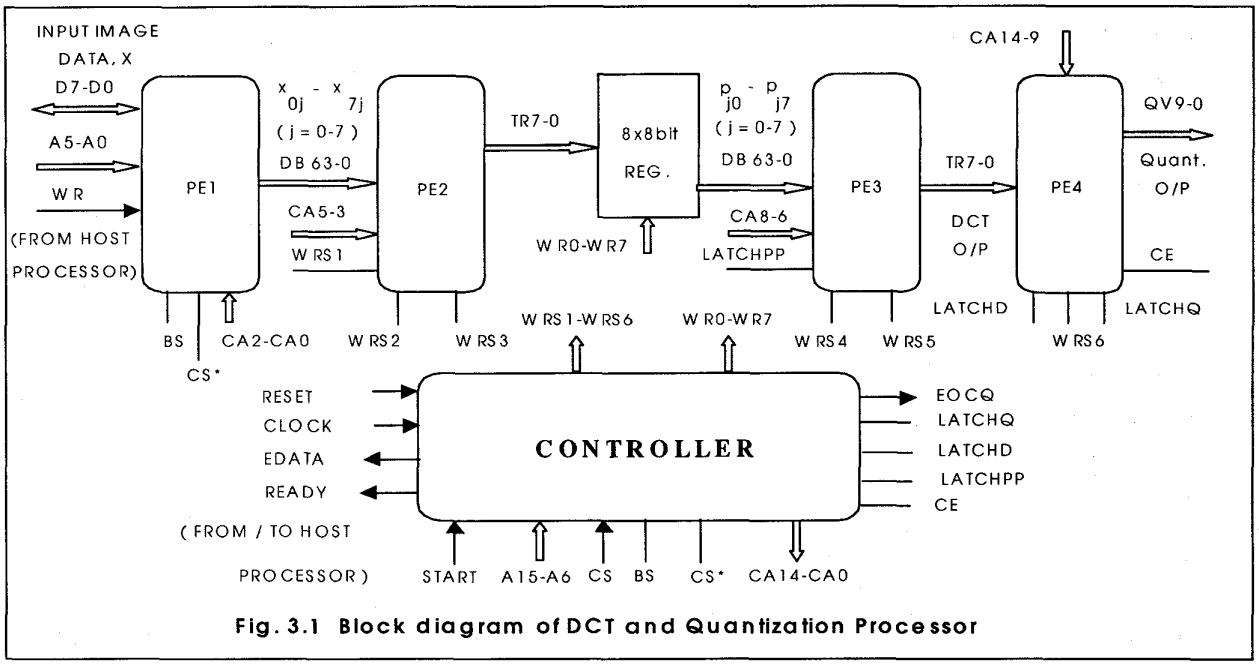

Fig. 3.1 Block diagram of DCT and Quantization Processor

consists of four processing elements, PE1 to PE4. PE1 and PE2 constitute the first stage of DCT, PE3 its second stage, and PE4 the final quantization stage. The input image (64 bytes) is written into PE1 through the data bus D0-D7, byte by byte. A0-A15 serve as the byte address, WR is the write pulse and CS is the chip select for writing. BS signal selects one of the two register banks to write the input image into, and the other register bank for processing DCT/Q simultaneously. Address lines CA0-CA2 select j, the column of input matrix X to read the same into PE2. Signals CA3-CA5 address a ROM to access the corresponding cos table. The DCT/Q processor is cleared by the external signal RESET and processing is initiated by asserting the START signal. After the first eleven CLOCK cycles, the partial products p_{00} to p_{07}, which are the row elements of CX, are latched into 8x8 bit registers sequentially, using WR0-WR7 write pulses. Corresponding cos coefficients (C^T) are multiplied (using CA6-CA8 as the address) with these partial products during the next five clock cycles to generate the DCT/Q output. While this DCT/Q output computation is under progress, computation of the next set of first stage partial products goes on concurrently. Thus, the two computations are pipelined. Once the pipeline is full (after a latency of fifteen clock cycles), the 64 values of the quantized outputs are available at QV0-QV9, one every clock cycle, at the rising edge of EOCQ (PE4 stage). Lines CA9-CA14 are used to addresss the quantization ROM. Thus, the computation of DCT/Q of an 8x8 image block requires a total of only 79 clock cycles.

The Controller generates the various control signals as shown in Fig. 3.1, which are used to latch the incoming data, store the data in pipeline registers (WRS1-WRS6, WR0-WR7, LATCHPP/D/Q), clear the data (RESET), select register bank (BS) etc. It basically consists of an eighty-state program counter and control cicuitry. The Program Counter is reset after transmitting all the 64 quantized coefficients and the DCT/Q processor will be ready to process the next 8x8 block of image data.

Architecture of PE1

As shown in Fig. 3.2, PE1 consists of two banks of registers, each of size 64 x 8 bits, tristated output buffers, input buffers, and decoders for selecting output buffers and writing into the selected registers. Input image data X is fed into the selected registers through the data bus in the order $x_{00},x_{01},...x_{07}$; $x_{10},x_{11},...x_{17}$; etc. whereas, while processing, its order is: $x_{00},x_{10},...x_{70}$; $x_{01},x_{11},...x_{71}$; etc. While the host processor is busy filling one bank of registers, the DCT/Q processing takes place in parallel from the other bank, thus eliminating the input loading time of the host processor.

Architecture of PE2/PE3

Each of these processing elements consists of four sets of 8X7 multipliers (MUL87), four two's complement converters (TRANSF) , 8-byte ROMs to store cos / cos

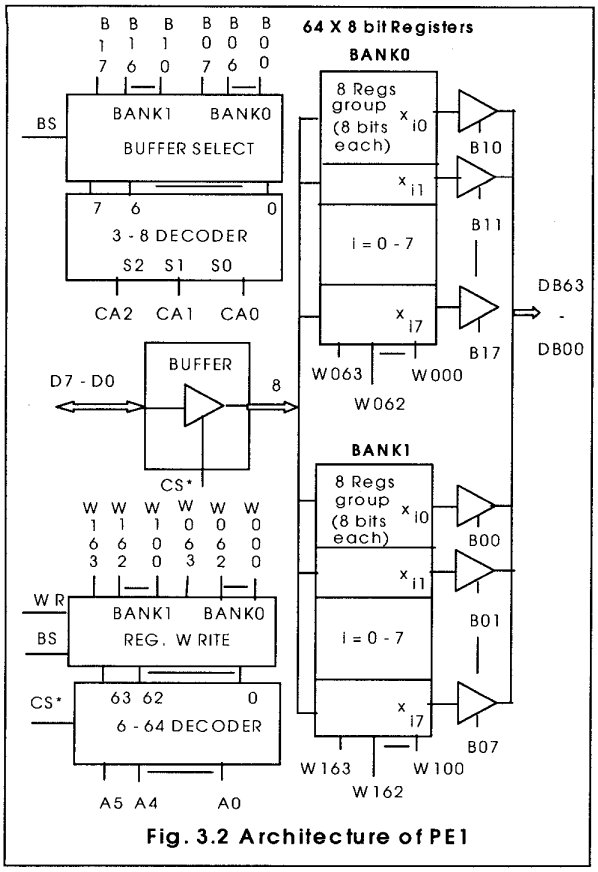

Fig. 3.2 Architecture of PE1

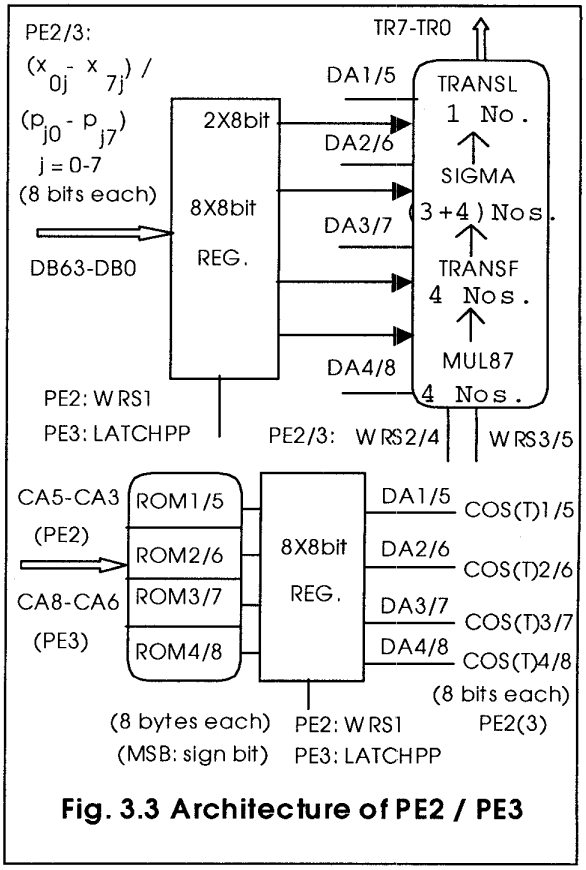

Fig. 3.3 Architecture of PE2 / PE3

transpose (COS/COST) tables, signed full adders (SIGMA), sign - magnitude converter (TRANSL) and pipeline registers. The data stored in input registers are multiplied with the corresponding cos coefficients and accumulated as shown in Fig. 3.3. Transform components convert the products into 2's complement representation to take care of the sign during addition. The partial products (p_{00} - p_{07} etc.) are generated in PE2, whereas the DCT coefficients are generated in PE3. The TRANSL converts two's complement number back to sign magnitude number.

Architecture of PE4

As shown in Fig. 3.4, the Processing Element PE4 consists of an 8X8 multiplier, a 64-byte ROM to store the quantization table (8 bit unsigned numbers), three pipeline registers and an output buffer. The DCT coefficients (64 in number) generated at PE3 are fed one after another into TR0-TR7 input and the corresponding 10 bit quantised outputs (QV0-QV9) are available one at a time, and can be latched at the rising edge of EOCQ output signal.

3.1 Sequence of Operations

The step by step operation sequence of the host and DCT/Q processor follows.

1. An 8x8 block of image data (64 bytes) is written by the host processor into one block of the 64-byte register, byte by byte, via the data bus after ascertaining that EDATA is high. The writing time is 40 ns per byte.
2. If READY is high, then the host processor issues START to begin DCT/Q processing, else it waits for READY to become high.
3. If EDATA is set high now, the next 8x8 block of image data is written into another block of the 64-byte register, otherwise the host processor waits.
4. Steps 2 and 3 are repeated for processing subsequent blocks.

4. IMPLEMENTATION

The proposed DCT and Quantization Processor has been implemented using a single Altera's FLEX 10k EPLD device [6] with each module being synthesized using either the Schematic or the VHDL package. The utilization of logic gates for the present design is about

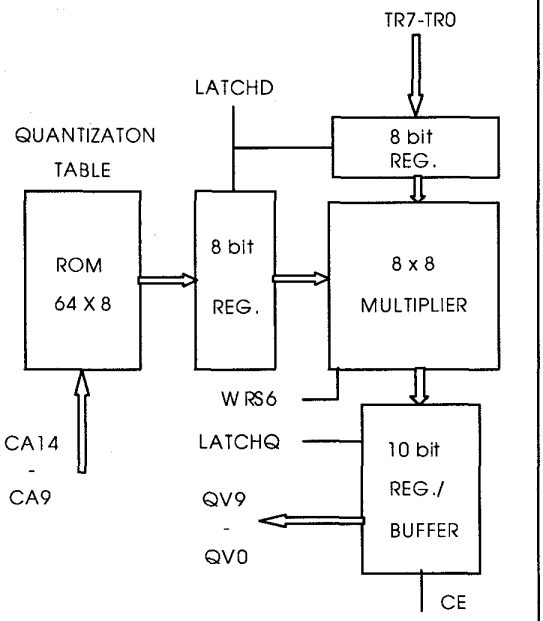

Fig. 3.4 Architecture of PE4

1,60,000 which is 80% of the total chip capacity. Thus it provides enough room for incorporation of additional features such as Huffman coding. As mentioned in Section 3, DCT/Q computation for an 8X8 pixel block requires 79 clock cycles or 3160ns. For a 1024X768 pixel image frame, the processing time will be approximately 40ms. Hence the DCT/Q is capable of processing images of size 1024X768 pixels at the rate of 25 frames per second.

The present implementation has better features when compared to the scheme suggested in [4], in which, only one set of input registers are used and it requires loading the same input data, row-by row, a number of times. Further, it has only nine pipelining stages. Its speed of execution, therefore, is slow compared to the present work, which incorporates dual-redundant input image registers, sixteen stages of pipelining and optimized controller design. These changes result in three times higher speed and reduction of hardware compared to the earlier scheme. Further, the use of dual input registers eliminates the input loading time of the host processor.

5. RESULTS AND DISCUSSIONS

Synthesis, functional testing and timing analysis were carried out for all the individual modules. The proposed architecture was tested with 40ns clock and it was found to work satisfactorily.

5.1 Comparison of DCT/Q processor speed with other implementations

The execution speed of the present DCT/Q implementation is compared with three other parallel architectures as shown in Table 5.1. It is found that the speed of the proposed DCT/Q processor is higher than that of other architectures eventhough they have implemented only DCT and not quantization.

Table 5.1 Execution times for 8 x 8 block DCT and Quantization

Proposed DCT/Q processor (25 MHz)	3.2 μs
ROM Multiplier based DCT Chip [1]	5.2 μs
SIMD processor [2]	8.2 μs
DCT processor [4]	9.7 μs

6. CONCLUSIONS

From the results obtained and comparisons made with the existing architectures, it can be concluded that the implementation of the DCT and quantization using EPLD is an easy and cost effective solution meeting the real time requirements and offers better speed than other implementations. The speed of this highly pipelined architecture can be improved further by deploying fast algorithms without sacrificing the concept of pipelined architecture and also by using faster EPLDs as and when they become available.

References

[1] M.T. Sun, T.C. Chen, and A.M. Gottlieb, "VLSI implementation of a 16x16 Discrete Cosine Transform", IEEE Trans. on Circuits and Systems, vol. 36, pp 610-617, April 1989.

[2] C.M. Wu and A. Chiou, "A SIMD Systolic Architecture and VLSI chip for the Two - Dimensional DCT and IDCT", IEEE Trans. on Consumer Electronics, Vol. 39, No. 4, pp 859-869, Nov. 1993.

[3] N.I. Cho, S.U. Lee, "Fast Algorithm and Implementation of 2D-Discrete Cosine Transform", IEEE Trans. on Circuits and Systems, Vol. CAS 38, pp 297-305, Mar. 95.

[4] D.V.R. Murthy, S. Ramachandran, and S. Srinivasan, "Parallel implementation of 2D-Discrete Cosine Transform using EPLDs", The twelfth international conference on VLSI design, Goa, Jan. 1999.

[5] K.R. Rao and P. Yip, "Discrete Cosine Transforms: Algorithms, Advantages, and Applications", New York Academic, 1990.

[6] ALTERA MAX + PLUS II manuals and ALTERA FLEX 10K Data Sheets, Data Book, 1998.