A Prototype VLSI Chip Architecture for JPEG Image Compression

Mario Kovač[1], N Ranganathan[2] and Mario Žagar[1]

[1]Faculty of Electrical Engineering, University of Zagreb, Av. Vukovar 39, 41000 Zagreb, CROATIA
[2]Center for Microelectronics Research,Dept. Comp. Sc. and Eng.,Univ. of South Florida,Tampa,USA

Abstract

In this paper, we describe the design and implementation of a prototype single chip VLSI architecture for implementing the JPEG baseline image compression standard. The chip exploits the principles of pipelining and parallelism to the maximum extent in order to obtain high speed and throughput. The architecture for discrete cosine transform and the entropy encoder are based on efficient algorithms designed for high speed VLSI implementation. The chip was implemented using the Cadence tools and based on the prototype implementation the proposed chip architecture can yield a clock rate of about 100 MHz which would allow an input rate of 30 frames per second for 1024x1024 color images.

1. Introduction

Data compression is the reduction or elimination of redundancy in data representation in order to achieve savings in storage and communication costs. Data compression techniques can be broadly classified into two categories: lossless and lossy schemes. Lossless methods are used for text compression and image compression in certain environments such as medical imaging where no loss of information is tolerated and typically the compression ratio is around 3:1. Lossy compression methods are commonly applied in image and audio compression and depending upon the fidelity required compression ratios of even upto 100:1 can be obtained. Digital images require an enormous amount of space for storage. For example, a color image with a resolution of 1024 x 1024 picture elements (pixels) with 24 bits per pixel would require 3.15 M bytes in uncompressed form. At a video rate of 30 frames per second, this requires a data rate of 94 M bytes per second. With the recent advances in video applications such as video teleconferencing, HDTV, home entertainment systems, interactive visualization and multimedia, there is an increasing demand for even higher bandwidth computing and communication systems. Very high speed implementation of efficient image compression techniques will significantly help in meeting that challenge.

In recent years, a working group known as Joint Photographic Expert Group (JPEG) has defined an international standard for coding and compression of continuous-tone still images. This standard is commonly referred to as the JPEG standard. The primary aim of the JPEG standard is to propose an image compression algorithm that would be application independent and aid VLSI implementation of data compression [1].

In this paper, we propose an efficient prototype VLSI chip architecture for implementing the JPEG baseline compression standard algorithm. The architecture fully exploits the principles of pipelining and parallelism to achieve high speed and throughput. The JPEG baseline algorithm consists mainly of two parts: (i) Discrete Cosine Transform (DCT) computation and (ii) Entropy encoding. The hardware architecture for DCT is based on an algorithm proposed in [4]. The entropy encoding part consists of runlength encoding followed by Huffman encoding. The architecture for entropy encoding is based on a hardware algorithm designed to yield maximum throughput and clock speed.

2. Related Work

The JPEG baseline compression standard, described in detail in [1,2,3], uses DCT and the Huffman entropy coding method for achieving compression. Due to the wide spectrum of applications in which DCT is used, several researchers have worked on this topic resulting in a vast amount of literature. Similarly, there exist different software and hardware approaches towards the implementation of Huffman coding. Since it is difficult to cover the entire work in the literature, a brief outline and pointers to some of the important contributions are provided in [4].

The fact that there does not exist any paper in the literature that describes the complete architecture for implementing the JPEG standard was the initial motivation for our work. From the information available on the few commercial chips, it was clear that we can achieve much better speeds by designing a linear static pipeline architecture

M. Kovač's work was supported in part by Croatian Ministry of Science Grant and Fulbright Grant.
N. Ranganathan's work was partially sponsored by Enterprise Florida Inovation Partnership FTRI Fund.

with no global communication or global control logic. Such an architecture is advantageous in that higher clock speeds can be easily obtained by decreasing the granularity of processing in each stage. In other words, the clock period can be reduced by subdividing the critical delay path into smaller slices or stages.

In this paper, we propose a fully pipelined prototype VLSI architecture for implementing the JPEG baseline compression standard. The architecture does not require any global communication or global control logic. Thus, the entire architecture can be sliced into thin stages resulting in a small clock period. The architecture for DCT and for category selection and Huffman coding in the entropy encoder are based on the efficient algorithm that lead to high speed VLSI implementation. With the architecture proposed in this paper, it is possible to obtain data compression rates of 100 million bytes per second or more.

3. Prototype Chip Architecture

The prototype chip architecture of JAGUAR, the JPEG baseline compression chip, is shown in Fig. 1. The entire architecture is organized as a linear multistage pipeline in order to achieve high throughput. The hardware organization shown in Fig. 1 reflects the sequence of computation in the JPEG baseline process. The architecture consists of: (i) Encoder model and (ii) Entropy encoder. The encoder model consists of DCT module, quantization module and reordering logic. The entropy encoder consists of several modules such

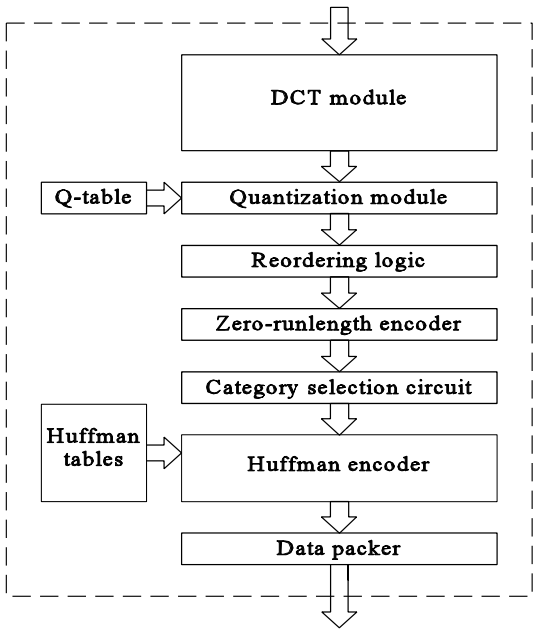

Fig. 1. JAGUAR Architecture

as zero-runlength encoder, category selection circuit, Huffman encoder and data packer. The image to be compressed is input to the architecture at the rate of one pixel per clock cycle. The input data is processed by the various modules in a linear fashion where each module itself is organized internally as a multistage linear pipe. The compressed data is output by the system at a variable rate depending on the amount of compression achieved. The design of each module is described in detail in the following sections.

4. Encoder Model

The encoder model consists of: (i) DCT module, (ii) quantization module and (iii) zig-zag reordering buffer.

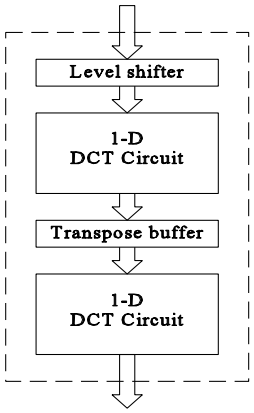

Fig. 2. DCT Module

4.1 DCT Module

The DCT module shown in Fig. 2 consists of a level shifter, two DCT circuits and a transpose buffer. The scaled two-dimensional DCT computation can be separated into two one-dimensional DCT operations and each one-dimensional DCT can be implemented by using modified DFT. The first DCT computation is performed row-wise and the second DCT computation is performed column-wise.

The circuit architecture for DCT computation is shown in Fig. 3. The circuit consists of six partitions as shown in the figure. Each partition contains a register set (RS) and an arithmetic unit with some associated control logic. Each register set consists of two columns of eight registers each except for the columns RS-d and RS-e which have nine registers per column. The circuit accepts one pixel per clock cycle and the entire processing is performed as a linear pipe. Our algorithm described in [4] is mapped directly onto the architecture such that each step in the algorithm corresponds

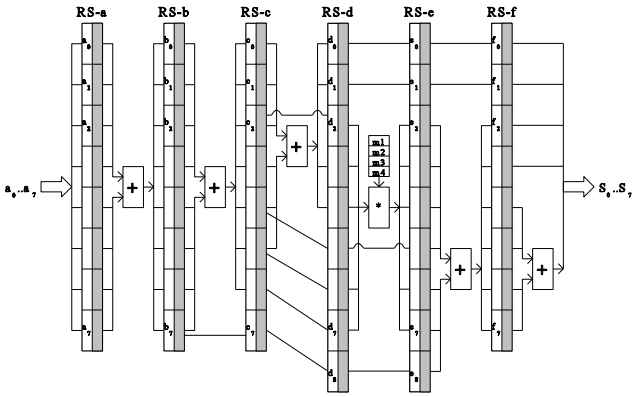

Fig. 3. One Dimensional DCT Circuit

to a partition in the architecture. During every clock cycle one data element is loaded in one of the left column registers. When the left column of register set RS-a is filled with eight data elements, the entire column is copied onto the corresponding registers in the right column. As the adder logic performs the computations of the algorithm, the left column keeps receiving new input data.It takes eight clock cycles to complete all the computations in step 1 which is the same number of cycles needed for filling up the left column. A similar process occurs in each of the partitions simultaneously. The adders are single stage units that have been reduced in size to the maximum extent without affecting the precision defined in the standard. The multiplier is a seven stage, reduced Wallace tree multiplier. During simulation we have found that the worst case crithical path for the entire chip is the 14-bit addition in the DCT computation. The DCT circuit has a latency of 50 clock cycles computed as 8 cycles per stage for all the stages except for the stage RS-a and RS-b that require 9 cycles. It should be noted that control logic is also minimized and is distributed in the register sets with no global control lines. Control logic consists of a simple 3-bit counter with a small decode logic. The same module is replicated for column-wise DCT computation.

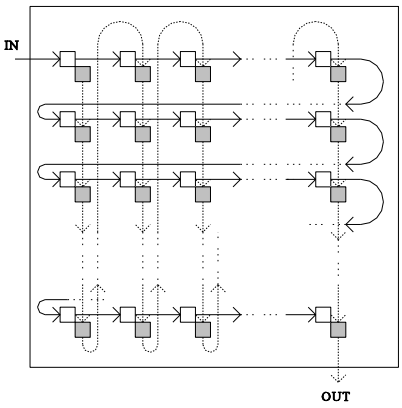

Fig. 4. Transpose Buffer

The transpose buffer is shown in Fig. 4. The buffer consists of an 8x8 array of register pairs organized as shown in figure. The data is input to the transpose buffer in row-wise fashion until all the 64 registers are loaded. The data in those registers are copied in parallel onto the corresponding adjacent registers which are connected in column-wise fashion. While the data is being read out from the column registers, the row registers will keep receiving further data from the DCT module.Thus, the output of row-wise DCT computation is transposed for column-wise DCT computation. The transpose buffer has a latency of 64 clock cycles.

4.2 Quantization Module

The quantization module consists of a RAM to store the quantization table and a 13x10-bit multiplier. This module is suitably combining scaling of the DFT coefficeints and the quantization operations required by JPEG. The latency of the quantization module is seven clock cycles which equals the number of stages in the multiplier. Control logic consists of a single 6 bit counter that serves as an address selector for selecting quantization factors from the RAM table.

4.3 Zig-zag Reordering Buffer

Zig-zag reordering is achieved by using an 8x8 array of register pairs organized similar to the transpose buffer with only difference that register pair outputs are connected in a zig-zag fashion.

5. Entropy Encoder

The function of the entropy encoder is to code the quantized coefficients from the encoder model using variable length encoding. The architecture of the entropy encoder is shown in Fig. 5. Each block of quantized pixel data consists of one DC coefficient followed by 63 AC coefficients. The various steps of the entropy encoder algorithm are briefly outlined as follows.

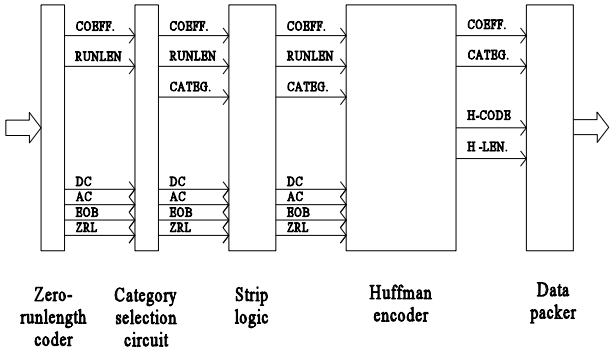

Fig. 5. Entropy encoding logic

5.1 Zero-Runlength Coder

The first step is to calculate $\triangle$ DC, which is the difference between the current and the previous DC coefficient, and to decrement all DC/AC coefficients by one if the sign of the coefficient is negative. The next step is to extract the zero-runlength count from the stream of the AC

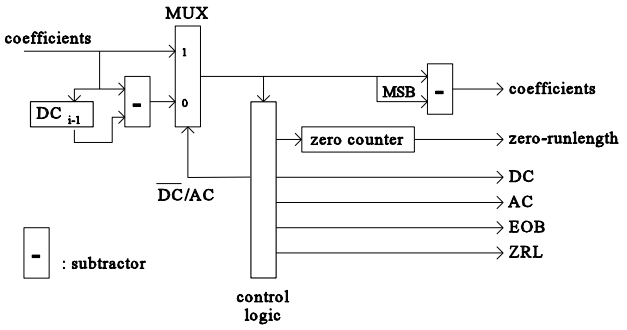

Fig. 6. Zero-runlength coder

coefficients within that block. The block data is thus converted into a stream of AC coefficients with an associated count value indicating the number of zeros preceding that coefficient. A 4-bit status field is generated corresponding to each coefficient which indicates if the data being output is a DC or AC coefficient, ZRL or EOB symbol. The above steps are performed within the zero-runlength coder shown in Fig. 6. The module consists of two stages and thus a latency of two cycles. The first stage consists of logic for computing $\triangle$DC while the second stage derives the runlength count and is used for decrementing negative coefficients. Control logic consists of a 6-bit increment circuit and a simple decoder. Increment circuit is driven by sytem clock and together with the decoder it generates all the necessary control signals.

5.2 Category Selection Circuit

Within the category selection circuit, each DC and AC coefficient is associated with a corresponding category depending on the magnitude of the coefficient. It should be noted that the data stream still contains all 64 coefficients including the streaks of zero coefficients which have been encoded as zero runlength counts.

The category selection is defined as a table in the JPEG compression standard document. Mathematical results and the algorithm which lead to a very efficient hardware structure for category selection are ommited for want of space.

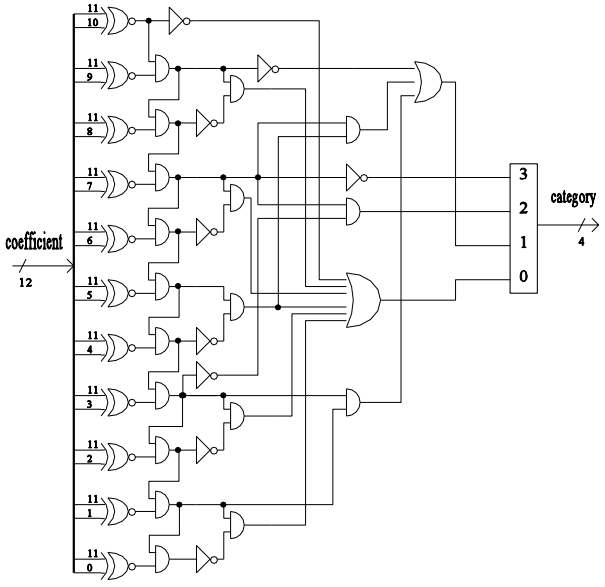

Fig. 7. Category selection circuit

The implementation of the category selection circuit is shown in Fig. 7. It should be noted that the chained delay of AND circuits shown in the Fig. 7. would exceed timing period and therefore the actual category selection is performed in two clock cycles.

5.3 Strip Logic

If an EOB symbol follows one or more ZRL symbols within the data stream the ZRL symbols are redundant and must be stripped off the data stream. The above function is performed within the strip logic.

The strip logic shown in Fig. 8. consists of five stages: load control stage and four strip stages. Each strip stage has three registers to hold the coefficient, runlength count and category fields corresponding to a data element output by the category selection circuit and a set of one-bit registers to hold the corresponding status. It should be noted that there could be a maximum of three ZRL symbols preceding an EOB symbol. The strip logic acts as a five stage buffer through which the compressed data elements after the removal of zero coefficients travel before being forwarded to the Huffman encoder. The valid bit signal is set to high whenever valid data is being output by the strip logic for Huffman encoding. ZRL modifier logic consists of eight gates and two one-bit latches.

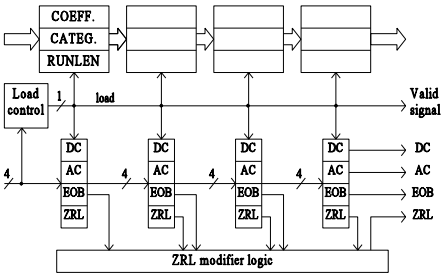

Fig. 8. Strip logic

5.4 Huffman Encoder Module

During the next step, each data element consisting of <AC/DC coefficient, runlength count, category, status > output by the strip logic is converted into a corresponding element: <AC/DC coefficient, category, Huffman code, Huffman code length>. The Huffman code is selected based on the runlength count, category and status fields. The set of Huffman codes are prestored in a table and can be changed depending on the application.

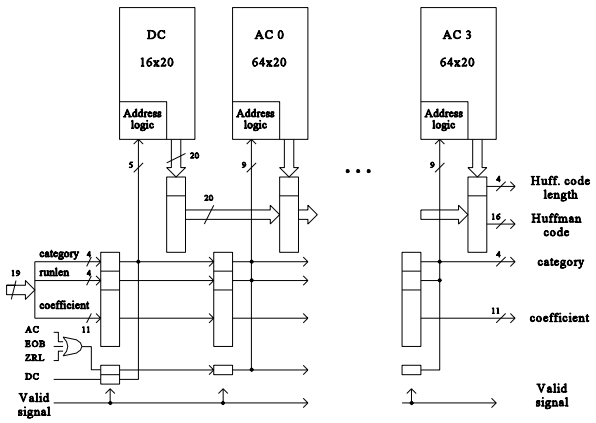

Fig. 9. Huffman encoder

The Huffman encoder module consists of Huffman code tables stored in random access memory modules and logic for replacing the category, runlength count pairs with the corresponding Huffman codes. Although the size of the DC coefficient code table is small, the code table storage for AC coefficients is relatively large. In order to keep the clock period small the memory for the code tables is organized as a set of five RAM modules arranged in a linear pipeline fashion. The idea is to reduce the access time by keeping the memory size small. The table is accessed by using the runlength, category pair for addressing. The input data passes through each of the five stages and depending on the address the corresponding Huffman code and the code length are output. The hardware organization is shown in Fig. 9 which is self explanatory.

5.5 Data Packer

The category and the Huffman code length fields are used in the data packer unit to pack the variable length compressed data (comprised of DC/AC coefficient and the Huffman code) into a stream of fixed length compressed data units to be output by the compression chip.

The data packer unit is used to convert variable length compressed data into fixed length compressed data stream. The logic consists of registers A and B, two left-shift units, two multiplexers and control logic which includes two registers A-length and B-length. The architecture and the description of the data packer are ommited for want of space.

6. VLSI implementation and simulation results

A prototype VLSI chip implementing the JAGUAR architecture was designed using CADENCE OPUS tools and CMOS 2-micron technology. Initially, the entire architecture was tested for functional correctness using C. During the actual design process the architecture was extensively simulated and tested using Verilog and other CADENCE tools. The chip was designed using two-phase non-overlapping clocking scheme. It was found that the architecture is fully functional at clock speed of 100 MHz under the worst case timing conditions. In order to test the operation of the chip three representative cells from the Encoder model (register set RS-a , register set RS-b and 13x10-bit multiplier) were fitted on a 6.8x6.9 mm^2 MOSIS standard frame and sent for fabrication. The layout of the chip is shown in Fig. 10. The circuitry consists of 21035 transistors and uses 80 pins. The multiplier required most of the area and has a total of 14767 transistors.

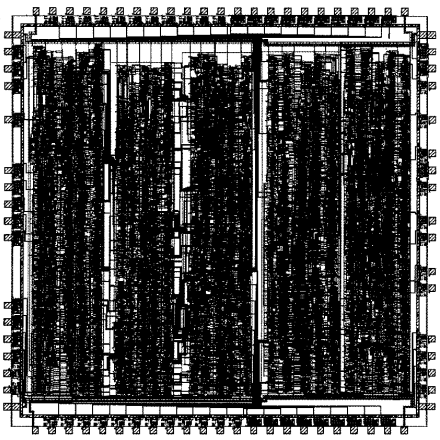

Fig. 10. JAGUAR chip layout

7. Conclusions

The proposed VLSI architecture has been simulated and verified using Verilog for functional correctness. The entire architecture is organized as a linear multistage pipeline. Thus the clock period can be reduced by increasing the level of fine grain parallelism. Since the entire architecture requires few major components such as three 13x10-bit multipliers, a few adders and small random access memory modules, the architecture can be realized as a single VLSI chip. For the architecture described in this paper, the critical path of the chip depends on a 14-bit adder circuit which can be easily achieved with a 10 nano seconds clock period. Thus, the proposed chip can function with an operating frequency of 100 MHz. This would allow an input rate of 100 million pixels per second leading to a rate of 30 frames per second for 1024x1024 color images. It should be noted that the 14-bit addition can be done in two clock cycles with the use of two 7-bit adders organized in two stages if a smaller clock is required. A prototype CMOS VLSI chip implementing the proposed JAGUAR architecture has been carried out using the Cadence design tools and sent for fabrication.

References

[1] ISO/IEC, International Standard DIS 10918, "Digital Compression and Coding of Continuous-Tone Still Images",

[2] W.B.Pennebaker, J.L.Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.

[3] G.K.Wallace, "The JPEG Still Picture Compression Standard", CACM, Vol.34, No.4, pp. 31-44, 1991.

[4] M. Kovač, N.Ranganathan, "JAGUAR: A High Speed VLSI Chip for JPEG Image Compression Standard", Proc. Intl. Conf. on VLSI Design, New Delhi, Jan 4-7, 1995.

[5] M. Kovač, N.Ranganathan, "VLSI Circuit Structure for Implementing JPEG Image Compression Standard", U.S. Patent pending