Submitted to the European Solid State Circuits Conference, ESSCIRC '96

A Data-Driven IDCT Architecture for Low Power Video Applications

Thucydides Xanthopoulos   Anantha P. Chandrakasan   Charles G. Sodini   William J. Dally

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract. Analysis of transform coded (MPEG) video data streams reveals a large percentage of zero-valued Discrete Cosine Transform (DCT) coefficients. A Data-Driven 2D IDCT architecture (DDIDCT) is proposed which exploits this observation for energy efficiency. The DDIDCT architecture exploits variability in the computational workload caused by the presence of zero-valued DCT coefficients by adaptively changing the power supply and the clock frequency of the main computation units. Adaptive minimization of switching events and power supply voltage make the DDIDCT approach more energy efficient than a conventional fast row-column Distributed Arithmetic IDCT implementation by more than an order of magnitude for the same sample rate.

1 Introduction

Commercial video decompression chips exhibit power dissipation of more than 1 Watt. As a result, they are not suitable for battery operated portable systems. This work focuses on a low power architecture for the 2D Inverse Discrete Cosine Transform (IDCT), an integral part of the MPEG video decompression standard. This implementation, called the Data-Driven IDCT (DDIDCT), exploits data distribution at the algorithmic level to adaptively minimize the number of operations and hence switched capacitance per block.

Compressed video data contains a large percentage of zero-valued spatial frequency (DCT) coefficients. The occurrence of zero-valued coefficients is the primary reason for the use of transform coding in the MPEG compression standard. The histogram in Figure 1 shows the relative occurrence of non-zero coefficients for a typical MPEG-2 sequence. The average number of non-zero coefficients for this sequence is 6.37 per 8x8 block.

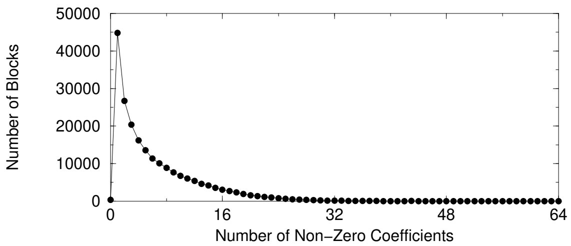

Figure 1: Histogram of Non-Zero DCT Coefficients in sample MPEG-2 Stream.

Conventional approaches to the computation of 2D IDCT on an 8x8 block use row column decomposition: first the 8-point 1D IDCT of each row is computed, then the result is transposed and the 8-point 1D IDCT of each column is computed. This process is equivalent to a 64-point 2D IDCT of the entire 8x8 block. Conventional approaches do not exploit data distribution to reduce energy dissipation. They perform a fixed number of operations per block independent of the data profile.

McMillan and Westover [1] have proposed an interesting direct realization of the 2D 8x8 IDCT. The forward mapped IDCT Algorithm (FMIDCT)

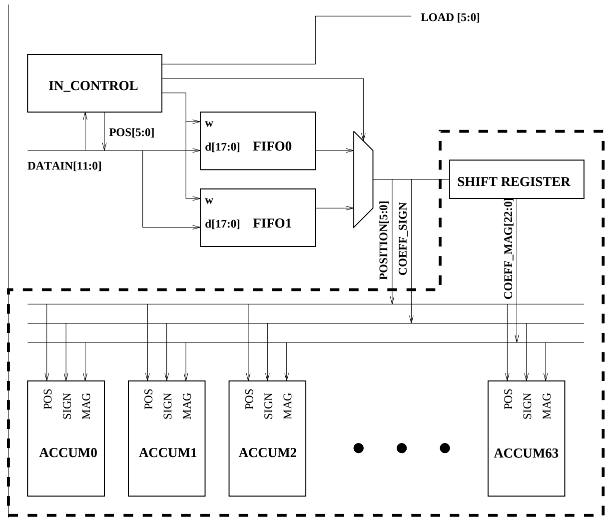

Figure 2: Data-Driven IDCT Block Diagram.

can be formulated as follows:

$$\vec{x} = y_0\vec{C_0} + y_1\vec{C_1} + y_2\vec{C_2} + \cdots + y_{63}\vec{C_{63}} \qquad (1)$$

where $\vec{x}$ is the 64-element reconstructed image block, $y_0, y_1, \ldots, y_{63}$ are the DCT coefficients (mostly zeroes) and $\vec{C_0}, \vec{C_1}, \ldots, \vec{C_{63}}$ are 64-element constant reconstruction vectors.

As seen from equation 1, each DCT coefficient is treated individually. Using this formulation combined with the observation that multiplication with a zero is a NOP, we can adaptively minimize the number of switching events by processing only the non-zero coefficients. The DDIDCT architecture exploits this fact to minimize average energy dissipation.

2 Data-Driven Architecture

Figure 2 shows a block diagram of the DDIDCT architecture. Data is pushed into one of two FIFOs while coefficients are read and processed from the FIFO not currently being written to (i.e a ping-pong buffer). Only non-zero values are pushed into the FIFOs and they are annotated with their position (0-63) within the 64-element block. This run length coding results in further energy savings [2] since zero values do not have to

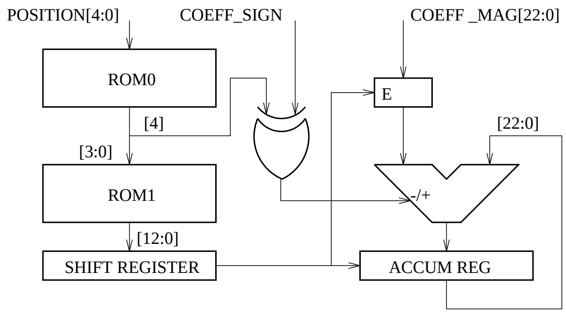

Figure 3: Accumulator Block Diagram.

be written or read from the FIFOs. Coefficients are represented in 12-bit sign-magnitude form. Each non-zero coefficient needs 13 cycles for processing and accumulation in the result registers. Coefficient magnitudes are read out of the FIFO and loaded in a 23-bit shift register. Within the next 13 cycles, the shift register will be continuously shifting the coefficient magnitude to the left and presenting the results to the 64 accumulators under local control. The accumulators will either accumulate the shifted coefficient magnitude or simply discard it based on their local control ROMs.

Figure 3 shows one accumulator in detail. The position of the coefficient is used as an index into a 2-level ROM. The first ROM is a 64x5 bit array which stores the sign and a 4-bit index of the magnitude of the constant value which must multiply the current coefficient for the block reconstruction. The second ROM stores the magnitude of the constant values represented in 13 bits. This bit width has been determined after running bit-level simulations of the architecture to determine compliance with IEEE Standard 1180-1990 concerning 2D IDCT precision. Figure 4 plots the Overall Mean Square Error (OMSE) as a function of constant magnitude bit width. The least precision required to meet the OMSE specified value was 13 bits. All other standard requirements are met.

The size of the magnitude ROM depends on

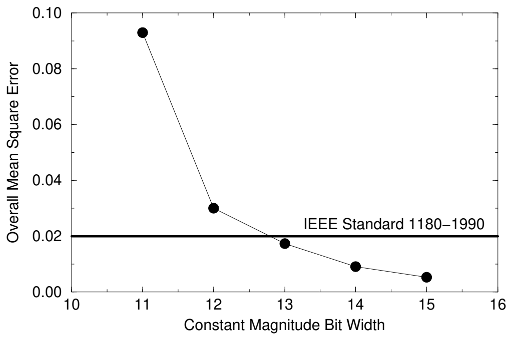

Figure 4: Overall Mean Square Error vs. Constant Bitwidth.

the position of the accumulator. The ROM can be as small as 1x13 bits and as large as 10x13 bits. The reason for breaking the table-lookup operation in 2 levels is to exploit redundancy in the reconstruction vectors of (1). This yields very small and power efficient ROMs. After the magnitude of the constant value has been read, it is loaded into a shift register. In the next 13 cycles, the LSB of this register will be used to enable the accumulation of the shifted coefficient magnitude. This LSB is also used to control a level-sensitive latch which prevents the adder inputs from switching when there is no operation to perform. This entire 13-cycle operation amounts to serial multiplication and subsequent accumulation as implied by equation (1).

3 Variable Power Supply

Since the number of operations per block varies with time, we propose to use an adaptive power supply and a variable clock rather than powering down the computational units when the processing load is less than the peak. This approach results in energy savings because of the quadratic dependence of energy on supply voltage. A variable clock tracks the critical path of the system under varying V_{DD}.

The block labelled IN_CONTROL in Figure 2 counts the total number of non-zero coefficients per block and produces the LOAD[5:0] signal. This quantity which can be between 0 and 64 is used to select both a supply voltage and a clock frequency for the circuitry included in the dashed box. The power supply can vary between 1.5V and 5V. The clock frequency can vary between $0.2 f_s$ (1 non-zero coefficient) and $13 f_s$ (64 non-zero coefficients) where f_s is the incoming DCT coefficient rate. Depending on the value of LOAD (0-64) an appropriate power supply and clock frequency is selected so that the throughput through the entire system is f_s.

One of the main issues in adaptive power supply systems is switching regulator latency. We will be using buffering and averaging techniques [4] to mask DC-DC converter delays. For our current simulations we have assumed that we have four distinct power supplies (1.5V, 2.7V, 3.9V, 5V) with zero switching latency. Moreover, the three thresholds of the LOAD signal are equally spaced apart between 0 and 64 (16, 32, 48).

4 Experimental Results

The architecture described in Sections 2 and 3 has been compared to the IDCT architecture in [3] in terms of first-order energy dissipation metrics such as number of additions and ROM/ RAM accesses per block. The reference architecture is an optimized Distributed Arithmetic (DA) approach of the Chen algorithm [5]. It is data independent and performs a fixed number of operations and memory accesses per block.

Figure 5(a,b,c) plots additions and ROM/ RAM accesses on a common time axis for the first 1000 blocks of the sample MPEG-2 stream of Figure 1. The fixed number of operations/ accesses for the DA approach and the average operations/ accesses for the entire stream for the DDIDCT approach are also plotted on the same three graphs. The DDIDCT clearly exhibits a smaller number of average operations and memory acesses per block.

In order to translate additions and memory accesses to energy dissipation figures, PowerPlay was used [6]. This tool is capable of estimating switched capacitance values for additions, mem-

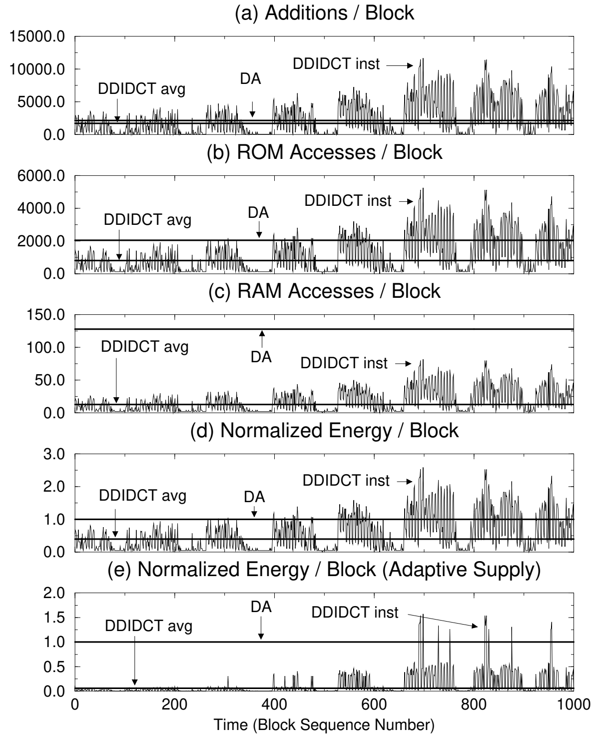

Figure 5: Comparison of the Proposed and Reference Architecture.

ory accesses and other high-level operations. Figure 5(d) plots normalized average energy dissipation per block for both approaches along with instantaneous energy dissipation for the first 1000 blocks of the sample MPEG-2 stream. Our first order energy dissipation metrics indicate that the DDIDCT dissipates less than half the energy of the DA and these savings are derived from purely algorithmic optimizations. Finally, Figure 5(e) shows the additional energy savings achieved by adaptively varying the supply voltage as described in section 3. In this case, energy savings up to a factor of 16 can be realized. The reference architecture runs at 5V.

5 Conclusion

This paper has presented an energy efficient VLSI architecture for the computation of the 2-D Inverse Discrete Cosine Transform. The DDIDCT architecture exhibits energy savings of a factor of 2 just by exploiting efficiencies at the algorithmic level over a conventional Distributed Arithmetic approach. If power supply variation is also employed energy savings of more than an order of magnitude can be achieved.

6 Acknowledgment

This research is supported by SHARP Corporation. The authors wish to thank Yoshifumi Yaoi for his help on this work, Rajeevan Amirtharajah, Jim Goodman, Abram Dancy and Vadim Gutnik for proofreading this paper.

References

[1] Leonard McMillan and Lee A. Westover, "A Forward-Mapping Realization of the Inverse Discrete Cosine Transform", *Proceedings of the Data Compression Conference (DCC '92)*, pages 219–228, IEEE Computer Society Press, March 1992.

[2] T. Meng et al., "Portable Video-on-Demand in Wireless Communication", *Proceedings of the IEEE*, pages 659–680, April 1995.

[3] S. Uramoto et al., "A 100 MHz 2-D Discrete Cosine Transform Core Processor", *IEEE Journal of Solid State Circuits*, pages 492–498, April 1992.

[4] Vadim Gutnik and Anantha Chandrakasan, "An Efficient Controller for Variable Supply Voltage Low Power Processing", *Symposium on VLSI Circuits*, June 1996.

[5] W.H. Chen et al., "A Fast Computational Algorithm for the Discrete Cosine Transform", *IEEE Transactions on Communications*, pages 1004–1009, September 1977.

[6] David Lidsky and Jan Rabaey, "Early Power Exploration - A World Wide Web Application", *33rd Design Automation Conference*, June 1996.