Sohm, O. P., Canagarajah, C. N., & Bull, D. R. (1999). Fast 2D-DCT implementations for VLIW processors. In IEEE Multimedia Signal Processing Workshop. (pp. 655 - 660). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/MMSP.1999.793940

FAST 2D-DCT IMPLEMENTATIONS FOR VLIW PROCESSORS

O.P. Sohm, D.R. Bull, C.N. Canagarajah
Centre for Communications Research
University of Bristol
Bristol, UK

Abstract - This paper analyzes various fast 2D-DCT algorithms regarding their suitability for VLIW processors. Operations for truncation or rounding which are usually neglected in proposals for fast algorithms have also been taken into consideration. Loeffler's algorithm with parallel multiplications [1] was found to be most suitable due to its parallel structure.

INTRODUCTION

The Discrete Cosine Transform (DCT) forms the basic building block of many image and video coding standards including the recently finalized MPEG-4 standard. Together with motion estimation the 2D-DCT is one of the most computationally intensive algorithms in a video coder. Therefore, for high bit rate applications, the DCT has usually been realized in dedicated hardware. The high performance of recently introduced VLIW-type digital signal processors [2] permits the DCT to be implemented in software.

This paper discusses properties of DCT algorithms that influence their performance on VLIW processors, such as arithmetic complexity, computational structure and accuracy. Various approaches to fast DCT algorithms are summarized and analyzed regarding their suitability for VLIW processors. Finally, performance results are discussed for implementations of selected algorithms on the Texas Instruments DSP TMS320C6201.

VLIW PROCESSORS

VLIW (Very Long Instruction Word) processors achieve their high performance by exploiting instruction-level parallelism (ILP) [3] through multiple parallel functional units. The main task remains with the compiler or programmer that has to optimize the data and control flow of a program with the aim to schedule as many operations as possible in parallel.

DCT algorithms, as most other numeric algorithms, have a highly parallel data flow and no control flow except the loop structure. at all. Therefore they contain a high amount of instruction-level parallelism. Conventional digital signal processors, however, are not able to exploit much of this parallelism.

PROPERTIES OF DCT ALGORITHMS

A reduction in the number of arithmetic operations is often essential for both hardware and software implementations, but is not the only requirement for a fast implementation. Another very important property of an algorithm is its computational structure. For hardware implementations the minimization of the number of multiplications and a regular structure are essential. However, for VLIW processors, a structure that is more suitable to their architecture can easily outweigh the advantage of saving one or two multiplications. Because there are typically more functional units available for additions than for multiplications or shift operations, multiplications and shifts are computationally more expensive. Shift operations which are required in practical implementations for truncation/rounding of the results of fixed-point multiplications and normalization are usually neglected in many proposals of fast algorithms. Other resource requirements, such as the number of registers, cannot be reflected in a measure of arithmetic complexity. For this purpose the structure of the algorithm has to be analyzed.

The accuracy of the computation is determined by the number of bits of the multiplier. Cosine coefficients have to be converted into a fixed-point representation for which the number of bits can be chosen according to the desired accuracy. For improved accuracy, results of multiplications can be rounded instead of just truncated. To further improve accuracy it is beneficial to keep the extended precision representation after a multiplication by rounding after summing up the products.

FAST ALGORITHMS

This section summarizes different approaches to fast DCT algorithms and analyses their suitability for VLIW processors.

Direct 2D approaches, such as [4], have a lower arithmetic complexity [5, 6] than row-column approaches but the processing of large data vectors causes register values to be stored temporarily in memory (register spilling) if the CPU does not have a sufficient number of registers. This can degrade execution speed drastically. Also, the complex computational structure prevents an implementation in a simple loop structure which results in increased code size.

Other approaches alter the cosine coefficient matrix in such a way that the number of multiplications and additions are reduced [7]. A further saving in the number of multiplications can be achieved by evaluating 2×2 sub-matrices as *rotations* [1].

Another method for computing a 2D-DCT is to first perform real-DFT's on the columns and then on the rows. A complex 2D-DFT is then established and rotations applied to obtain the 2D-DCT [8]. This is a kind of hybrid approach involving methods from the row-column and the direct 2D approach. Therefore, the same problems as with other direct 2D approaches exist.

Polynomial approaches, such as [9], map the DCT into complex polynomials. This representation of the DCT is similar to that of the DFT. A fast algorithm is then obtained through a polynomial transform similar to the FFT. This method can be used for 2D and row-column approaches.

Recursive algorithms [10] compute the DCT through lower order DCT's, e.g. the 8-point DCT is computed using two 4-point DCT's each of which are in turn computed using two 2-point DCT's. However, due to a required post-computation stage the overall number of stages is greater than that of other algorithms.

In multiplication-free structures [11], multiplications have been replaced by shift operations. This might be beneficial for hardware implementations, but in the case of VLIW processors such an implementation would not fully utilize resources since the multiplier units would be left unused. Additionally, the computational complexity would increase because one multiplication is replaced with up to 6 shift operations.

IMPLEMENTATION RESULTS

Among the various algorithms analyzed [4, 9, 6, 7, 1, 10, 11], the algorithm by Loeffler et. al [1] with parallel multiplications (LLM PM) was found most suitable for VLIW processors. The standard algorithm (LLM) is shown in Figure 1. To obtain the version with parallel multiplications, the odd part (after the first stage) has to be replaced with the flow graph shown in Figure 2. The algorithm requires 12 multiplications and 32 additions. Although there are algorithms with fewer multiplications and additions, such as the standard LLM algorithm, the suitability of the LLM PM algorithm becomes apparent if practical implementation issues are considered.

Generally, in a row-column DCT algorithm without rounding shift operations are required for two purposes: (1) truncation after fixed-point multiplications and (2), at the end of the column DCT, truncation to normalize the output values. To perform rounding instead of truncation, addition operations for adding rounding constants have to be introduced.

The difference of the two LLM algorithms lies only in the odd part which shall now be compared to highlight the advantages of the PM structure. The LLM PM algorithm requires 4 shift operations and, if rounding is desired, one rounding add. For the row DCT the shifts perform the truncation required after the multiplications, and in the column DCT the shifts perform both functions, truncation and normalization. The cascaded multiplications structure of the standard LLM algorithm (Figure 1) requires two more shift operations and two more addition operations for rounding. The shifts are required because the multiplier results from the rotation C1 are fed again into multipliers and need to be truncated before. Both, the LLM standard and PM algorithm were implemented on the TMS320C6201 in assembly language with and without rounding. Table 1 shows a summary of the characteristics and the performance of the implementations.

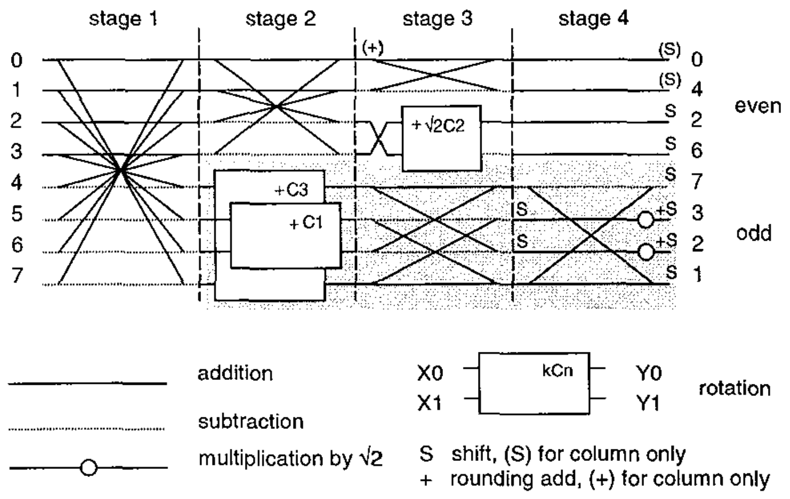

Figure 1: Signal flow graph of standard LLM algorithm [12]

As can be seen in the table, the PM algorithm requires fewer shift operations than the standard algorithm because no multiplications are in cascade. Without rounding operations, the standard algorithm appears to have a slight advantage with its fewer additions and multiplications and therefore executes faster. However, considering the implementations with rounding, one can see that the advantage of fewer additions is no longer relevant if the additions for rounding are taken into account. Both implementations require 34 and 35 additions for the row and column DCT, respectively. The disadvantage of the PM algorithm having one more multiplication is outweighed by the reduced number of shift operations. This makes the PM algorithm with 266 cycles clearly faster than the standard algorithm (297 cycles). As expected, the PM algorithm is superior in terms of accuracy because cascaded multipliers are avoided. The achieved accuracy of 0.0364 (mean square error) compares well with the accuracy of the floating-point implementation which is 0.0292.

CONCLUSIONS

For practical software implementations of fast DCT algorithms, other issues than multiplicative complexity have to be taken into account in order to achieve good performance. The structure of the algorithm determines the number of rounding (additions) and shift operations required. In particular shift operations, which can be implemented with almost no cost at all for VLSI, can be as computationally expensive as multiplications for software

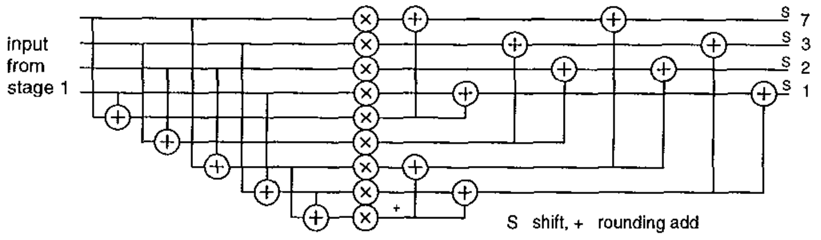

Figure 2: Odd stage of Loeffler's algorithm with parallel multiplications (LLM PM)

Alg.	Rounding	Adds	Mults	Shifts	Total	Accuracy	Cycles
Std	no	29	11	8/10	98	0.4845	255
PM	no	32	12	6/8	102	0.2968	272
Std	yes	34/35	11	8/10	109	0.0552	297
PM	yes	34/35	12	6/8	108	0.0364	266
Float	n/a	26	16	n/a	84	0.0292	n/a

Table 1: Number of operations and performance results of implementations of the LLM standard and PM algorithm with and without rounding (the number of operations is the same for both row and column DCT unless stated as row/column)

implementations. The structure of an algorithm also determines how efficient the resources of an VLIW processor can be exploited. As shown, resource requirements, the most important being the number and type of functional units and the number of registers, can be estimated through analysis of the algorithm structure, but only the actual implementation will reveal the true performance.

References

[1] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1D DCT algorithms with 11 multiplications," in *IEEE ICASSP*, vol. 2, pp. 988–991, February 1989.

[2] P. Faraboschi, G. Desoli, and J. A. Fisher, "The latest word in digital and media processing," *IEEE Signal Processing Magazine*, vol. 15, no. 2, pp. 59–85, 1998.

[3] B. R. Rau and J. A. Fisher, "Instruction-level parallel processing: History, overview, and perspective," *The Journal of Supercomputing*, vol. 7, no. 1/2, pp. 9–50, 1993.

[4] N. I. Cho and S. U. Lee, "Fast algorithms and implementation of 2-D discrete cosine transform," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 297–305, March 1991.

[5] P. Duhamel and H. H'Mida, "New 2^n DCT algorithms suitable for VLSI implementation," in *Proc. ICASSP-87*, pp. 1805–1808, 1987.

[6] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2174–2193, September 1992.

[7] F. A. McGovern, R. F. Woods, and M. Yan, "Novel VLSI implementation of (8x8) point 2-D DCT," *Electronic Letters*, vol. 30, pp. 624–626, April 1994.

[8] M. Vetterli, P. Duhamel, and C. Guillemot, "Trade-off's in the computation of mono- and multi-dimensional DCT's," in *ICASSP*, pp. 999–1003, 1990.

[9] P. Duhamel and C. Guillemot, "Polynomial transform computation of 2-D DCT," in *IEEE ICASSP*, pp. 1515–1518, 1990.

[10] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1455–1461, October 1987.

[11] C.-Y. Lu and K.-A. Wen, "On the design of selective coefficient DCT module," *IEEE Trans. Circuits, Systems for Video Technology*, vol. 8, pp. 143–146, April 1998.

[12] H. R. Wu and D. Tan, "Implementation of Cho and Lee's 2D DCT algorithm using LLM 1D DCT algorithm," in *1997 IEEE Int. Workshop on Intelligent Signal Processing and Communications Systems*, pp. 2.1–2.5, November 1997. Sect. 24.