HARDWARE PROTOTYPING FOR THE H.264 4×4 TRANSFORMATION

Ihab Amer[1], Wael Badawy[1], and Graham Jullien[2]

[1]Department of Electrical and Computer Engineering
University of Calgary
Calgary, Alberta, Canada, T2N 1N4
{amer, badawy}@enel.ucalgary.ca

[2]Advanced Technology Information Processing Systems (ATIPS)
Calgary, Alberta, Canada, T2N 1N4
jullien@atips.ca

ABSTRACT

This paper presents a hardware prototype of the H.264 transformation. The proposed architecture uses only add and shift operations to reduce the computational requirements for the 4×4 transform. The architecture is developed to be used in high-resolution applications such as High Definition Television (HDTV) and Digital Cinema. The developed architecture is prototyped and simulated using ModelSim 5.4®. It is synthesized using Leonardo Spectrum®. The results show that the architecture satisfies the real-time constraints required by different digital video applications.

1. INTRODUCTION

Digital video streaming is increasingly gaining higher reputation due to the noticeable progress in the efficiency of various digital video-coding techniques. This raises the need for an industry standard for compressed video representation with substantially increased coding efficiency and enhanced robustness to network environments [1].

In 2001, the Joint Video Team (JVT) was formed to represent the cooperation between the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) aiming for the development of a new Recommendation/International Standard.

The JVT are currently finalizing a new standard for the coding (compression) of natural video images [2]. The name H.264 (or MPEG-4 Part 10, "Advanced Video Coding (AVC)") is given to the new standard.

The H.264 standard has many new features when compared to conventional standards. It offers good video quality at high and low bit rates. It is also characterized by error resilience and network friendliness [3]-[7].

The standard does not use the traditional 8×8 Discrete Cosine Transform (DCT) as the basic transform. Instead, a new 4×4 transform is introduced that can be computed exactly in integer arithmetic, thus avoiding inverse transform mismatch problem [8].

The transform can be computed without multiplications, just additions and shifts, in 16-bit arithmetic. This minimizes the computational complexity significantly. Besides, the quantization operation uses multiplications avoiding unsynthesisable divisions.

To develop a hardware video codec for H.264, a VLSI architecture is required. Surveying the literature shows that there are few architectures that prototype the new 4×4 transformation.

In this present paper, a hardware prototype for the 4×4 transform adopted in the H.264 standard is introduced. The proposed architecture is developed to use only add and shift operations to reduce the computational requirements for the 4×4 transform.

The rest of this paper is organized as follows: Section 2 reviews the H.264 transform and quantization. In section 3, a description of the proposed hardware prototype of the H.264 standard is introduced. Section 4 presents the simulations and results achieved. Finally, section 5 concludes the paper.

2. H.264 4×4 TRANSFORM AND QUANTIZATION

The 4×4 transform adopted in the H.264 standard is an integer orthogonal approximation to the DCT. This allows for bit-exact implementation for all encoders and decoders. Another important feature in the new standard is the removal of the computationally expensive multiplications that appears in the conventional standards, which are based on the traditional DCT.

The encoder transform formula that is proposed by the JVT to be applied to an input 4×4 block is shown in Equation (1).

$$W = C_f X C_f^{\,T} \tag{1}$$

where the Matrix C_f is given by Equation (2).

$$C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \tag{2}$$

In Equation (2), the absolute values of all the coefficients of the C_f matrix are either 1 or 2. Thus, the transform operation represented by Equation (1) can be computed using signed additions and left-shifts only to avoid expensive multiplications.

The post-scaling and quantization formulas are shown in Equations (3) and (4).

$$qbits = 15 + (QP \; DIV \; 6) \tag{3}$$

$$Z_{ij} = round(W_{ij} . \frac{MF}{2^{qbits}}) \tag{4}$$

where QP is a quantization parameter that enables the encoder to accurately and flexibly control the trade-off between bit rate and quality. It can take any integer value from 0 up to 51. Z_{ij} is an element in the matrix that results from the quatization process. MF is a multiplication factor that depends on QP and the position (i, j) of the element in the matrix as shown in Table 1.

QP	(i, j) ∈ {(0, 0), (2, 0), (2, 2), (0, 2)}	(i, j) ∈ {(1, 1), (1, 3), (3, 1), (3, 3)}	Other Positions
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

Table 1. Multiplication Factor (*MF*)

The factor MF remains unchanged for $QP > 5$. It can be calculated using Equation (5).

$$MF_{QP>5} = MF_{QP=QP \bmod 6} \tag{5}$$

Equation (4) can be represented using integer arithmetic as follows:

$$|Z_{ij}| = SHR(|W_{ij}|.MF + f, qbits) \tag{6}$$

$$Sign(Z_{ij}) = Sign(W_{ij}) \tag{7}$$

where *SHR()* is a procedure that right-shifts the result of its first argument a number of bits equal to its second argument. f is defined in the reference model software as $2^{qbits}/3$ for Intra blocks and $2^{qbits}/6$ for Inter blocks [2].

3. HARDWARE PROTOTYPE

This section introduces the proposed hardware prototype of the 4×4 transform adopted by the H.264 standard. The proposed architecture uses 4×4 parallel input block. A block diagram of the architecture is shown in Figure 1.

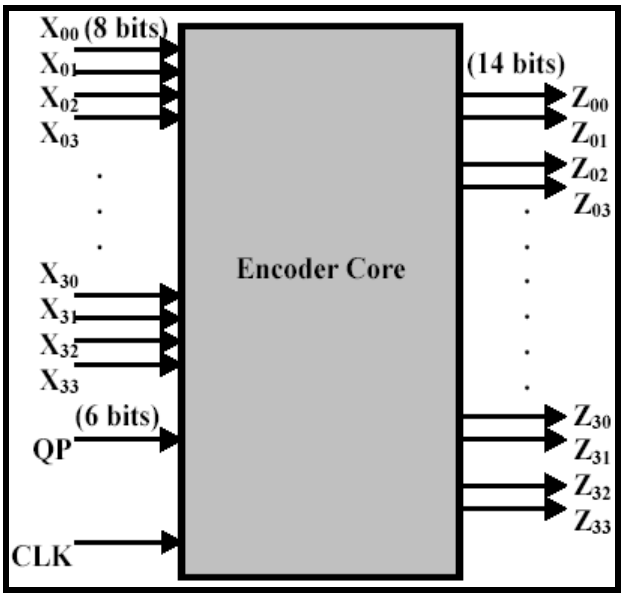

Figure 1. A block diagram of the proposed hardware architecture

This architecture requires a target chip with a number of I/O pins greater than or equal to 359. It is designed to perform pipelined operations. Therefore, with the exception of the first 4×4 input block; the illustrated architecture can output a whole encoded block with each clock pulse. The architecture does not contain memory

elements. Instead, a redundancy in computational elements shows up. This introduces an example of performance-area tradeoff.

A detailed description of the architecture is shown in Figure 2.

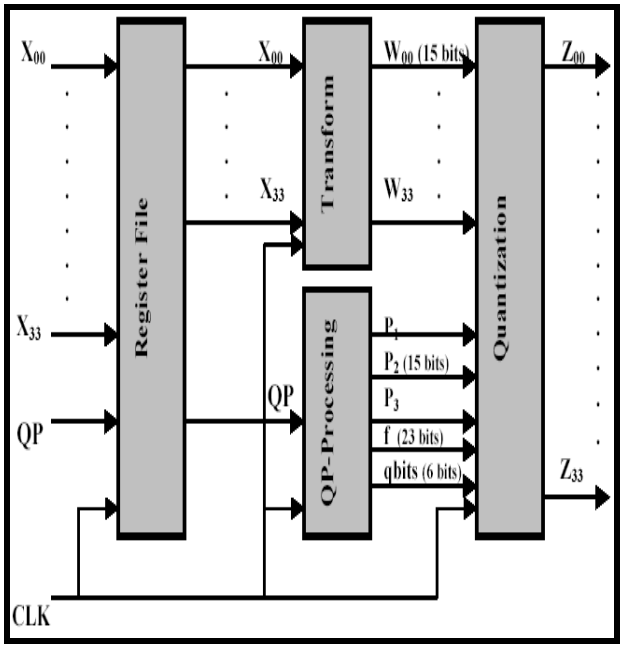

Figure 2. A detailed block diagram of the proposed hardware architecture

The architecture is composed of three main stages: The Register File stage, The Transform and the QP-Processing stage, and The Quantization stage.

Data is initially captured from the outside environment and stored in the Register File. Then the 4×4 input block is passed to the Transform block. This block consists of two cascaded sub-blocks. Each of them is responsible of multiplying two 4×4 matrices and is composed of four identical butterfly-adder blocks. Its operation is to perform a group of additions and shifts. Figure 3(a) shows the first butterfly-adder block in the first sub-block, while Figure 3(b) shows the first butterfly-adder block in the next sub-block. The Transform block is the hardware implementation that corresponds to Equation (1).

At the same time, the QP-Processing block is responsible for calculating *f*, *qbits*, and determining P_1, P_2, and P_3, which are the values of the multiplication factors at the three different groups of positions in the matrix as shown in Table 1. Finally, the Quantization process takes place in the last-stage block. The integer division by six that is required for implementing Equation (2) and Equation (5) is implemented by recursive subtraction.

Signed numbers are represented in the whole architecture using the standard signed two's complement representation.

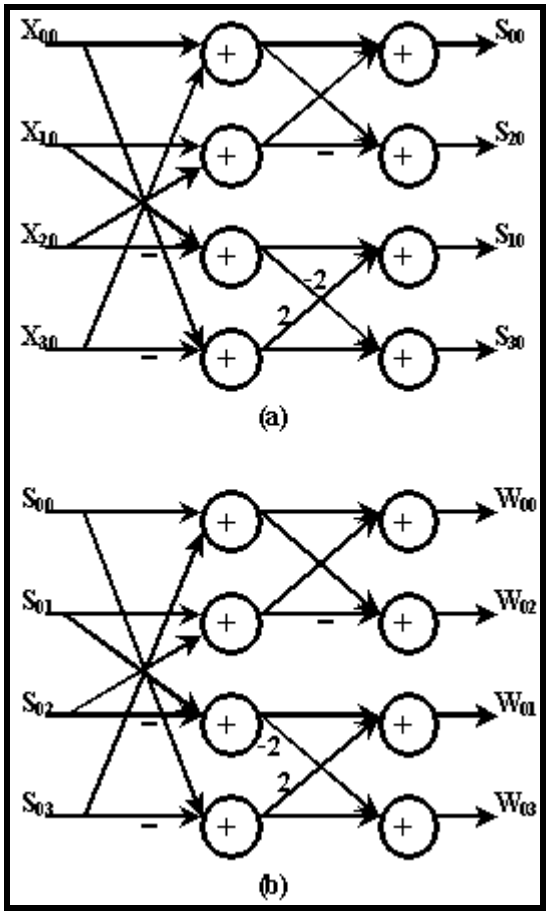

Figure 3. First butterfly-adder block in
(a) First sub-block
(b) Second sub-block

4. SIMULATIONS AND RESULTS

The architecture for the H.264 4×4 transformation is prototyped using VHDL language. It is simulated using the Mentor Graphics© ModelSim 5.4® simulation tool, and synthesized using Leonardo Spectrum®. The target technology is the FPGA device (2V3000fg676) from the Virtex-II family of Xilinx©.

The correctness of the implemented architecture is also checked. This is done by passing different input patterns to the architecture and comparing the output with the results obtained by passing the same inputs to the equations of Section 2.

Table 2 summarizes the performance of the prototyped architecture.

Critical Path (ns)	Clk Freq. (MHz)	# Of Gates	# Of Ports	# Of Nets
28.3	34.8	6212	359	718
# Dff's or Latches	**# Func. Generators**	**# CLB Slices**	**# B. Box Adders**	**# B. Box Subtr.**
4156	6407	3204	8	8

Table 2. Performance of the prototyped architecture

The critical path is estimated by the synthesis tool to be 28.3 *ns*. Since the chip outputs a whole 4×4 encoded block with each clock pulse (except for the first block), therefore the time required to encode a whole CIF frame (325×288 pixels) can be calculated as follows:

$$\begin{aligned} &\textit{Time required per CIF frame} = \\ &\qquad \textit{Time required per block} \times \textit{Number of blocks per frame} \\ &\qquad = 28.3\ ns \times \frac{\textit{Number of pixels per frame}}{\textit{Number of pixels per block}} \\ &\qquad = 28.3\ ns \times \frac{(352 \times 288)\ \textit{pixels per frame}}{(4 \times 4)\ \textit{pixels per block}} \\ &\qquad \simeq 0.18\ ms \end{aligned}$$

This value is 185 times less than the 33.3 *ms* standard time (assuming 29.97 *frames/sec*) required for frame encoding. Similarly, it can be shown that the time required to encode a whole High Definition Television (HDTV) frame of a 720×1280 pixels resolution, and a 60 *frames/sec* frame rate is 1.63 *ms*, which is about 10 times less than the 16.6 *ms* standard time. This leads to the suggestion of taking the input serially, integrating other operations on the same encoder chip, or targeting other applications that use more complicated-higher resolution video formats.

5. CONCLUSION

A novel architecture for prototyping the 4×4 transform adopted by the H.264 Standard is developed. The proposed architecture receives the parallel 4×4 input block and the Quantization Parameter (QP) as inputs.

The architecture satisfies the real-time constraints required by different digital video applications.

The system is simulated using ModelSim 5.4®, and synthesized using Leonardo Spectrum®, targeting the FPGA device (2V3000fg676) from the Virtex-II family of Xilinx©.

The system can be enhanced in different ways. Input may be taken serially. A hierarchical transform [8] structure can be implemented, and the pipeline stages can be balanced to increase performance.

6. ACKNOWLEDGEMENT

The authors would like to thank Advanced Technology Information Processing Systems (ATIPS), Natural Science and Engineering Research Council of Canada (NSERC), Canadian Microelectronics Corporation (CMC), Micronet R&D, Alberta Informatics Circle of Research Excellence (iCORE), Canada Foundation for Innovation (CFI), and the Department of Electrical and Computer Engineering at the University of Calgary for supporting this research.

7. REFERENCES

[1] "ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC," Draft Text of Final Draft International Standard for Advanced Video Coding, [Online]. Available: http://www.chiariglione.org/mpeg/working_documents.htm, March 2003.

[2] I. E. G. Richardson, "H.264/MPEG-4 Part 10: Transform & Quantization," A white paper. [Online]. Available: http://www.vcodex.com, March 2003.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems For Video Technology, Vol. 13, No. 7, pp. 560-576, July 2003.

[4] "Emerging H.264 Standard: Overview and TMS320DM642-Based Solutions for Real-Time Video Applications," A white paper. [Online]. Available: http://www.ubvideo.com, December 2002.

[5] K. Denolf, C. Blanch, G. Lafruit, and A. Bormans, "Initial memory complexity analysis of the AVC codec," IEEE Workshop on Signal Processing Systems, 2002 (SIPS'02), pp. 222-227, October 2002.

[6] T. Stockhammer, M. M. Hannuksela, T. Wiegand, "H.264/AVC in wireless environments," IEEE Transactions on Circuits and Systems For Video Technology, Vol. 13, No. 7, pp. 657-673, July 2003.

[7] M. Horowitz, A. Joch, F. Kossentini, A. Hallapuro, "H.264/AVC Baseline Profile Decoder Complexity Analysis," IEEE Transactions on Circuits and Systems For Video Technology, Vol. 13, No. 7, pp. 704-716, July 2003.

[8] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-Complexity Transform and Quantization in H.264/AVC," IEEE Transactions on Circuits and Systems For Video Technology, Vol. 13, No. 7, pp. 598-603, July 2003.