ARCHITECTURAL TEMPLATES FOR THE 4X4 TRANSFORMS OF THE H.264/AVC STANDARD TARGETING THE INTRA PREDICTION CODER

Felipe Sampaio, Robson Dornelles, Daniel Palomino, Guilherme Corrêa, Diego Noble, Luciano Agostini

Group of Architectures and Integrated Circuits (GACI)
Federal University of Pelotas (UFPel)

ABSTRACT

This paper presents an architectural investigation for the 4x4 transforms (4x4 Hadamard and 4x4 DCT) of the H.264/AVC standard, looking for high throughput and low latency architectures that fit in the Intra Prediction constraints. Five architectural templates were designed, and then applied to both 4x4 forward transforms for evaluation purpose. However, the templates can be also applied to the 4x4 inverse transforms. The architectures were described in VHDL and synthesized targeting Altera Stratix II FPGA. According to the synthesis results, the throughput range varies from 656 to 7,200 millions of samples processed per second.

1. INTRODUCTION

H.264/AVC [1] is the latest video compression standard and it was defined intending to double the compression rates reached by the previous standards.

The main modules of H.264/AVC coder are the Inter Frame, the Forward and Inverse Quantization (Q and Q^{-1}), Entropy Coding, Deblocking Filter, Intra Frame Prediction and Forward and Inverse Transforms (T and T^{-1}), which are the focus of this work.

Intra Prediction works with I type blocks [1]. Their codification does not depend on the previously processed frame. The reference values are in the previously coded blocks inside the same frame. So, it means that the predicted image blocks must be processed by the T, Q, Q^{-1} and T^{-1} modules before to be used as references to process the next block in the current frame. Thus, these modules belong to the critical path of intra prediction in the H.264/AVC encoder.

The goal of this work is to investigate hardware solutions that can present a good relation between the high throughput and the low latency in the transforms modules, since these are the main characteristics desired in these modules to enable Intra Prediction coding in real time (30 frames per second).

The Forward (T) and Inverse (T^{-1}) Transform modules defined in all H.264/AVC profiles are composed by two 4x4 transform operations: Forward and Inverse 4x4 DCT (FDCT and IDTC) and Forward and Inverse 4x4 Hadamard (FHAD and IHAD) [1].

The FDCT is applied to all luma (Y) or chroma (Cb and Cr) input data. When luma sample information was predicted using the Intra 16x16 mode [2], the 4x4 FHAD is applied over the DCs coefficients generated by the FDCT.

This paper proposes five different architectural templates for the 4x4 transforms defined in the H.264/AVC. They were designed intending to investigate the solution that best fits into the Intra Prediction constraints. To evaluate the characteristics of these templates, they were implemented in architectures that perform the 4x4 FHAD and the 4x4 FDCT transforms, in a total of ten different implementations.

The paper is structured as follows: Section 2 describes the architectural templates used to implement the both transforms operations. Section 3 shows the synthesis results and a discussion about them. Section 4 presents a comparison with previous works. Finally, Section 5 concludes this paper and presents future works.

2. ARCHITECTURAL TEMPLATES

In this work, were designed five architectural templates for the 4x4 transforms defined in the H.264/AVC standard.

Two different kinds of templates were implemented: fully parallel templates, which consume sixteen input samples per clock cycle; and the row based templates, which process four input samples per clock cycle. These templates were based on algorithms extracted from the transforms mathematical definitions [2].

Equation (1) presents the 4x4 2-D FDCT defined in the H.264/AVC, where **X** is the 4x4 input residual block and **Y** is the FDCT output.

$$Y = C_f X C_f^T \otimes E_f = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} \\ X \\ \\ \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 2 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \quad (1)$$

The 4x4 FDCT calculation transfers to the quantization module the scalar multiplication by the $\mathbf{E_f}$ matrix. The symbols *a* and *b* in the $\mathbf{E_f}$ matrix represent constants and the addition of this task to the Q module does not imply in an increase in the computational complexity on this module [2].

The 4x4 FHAD, 4x4 IHAD and the 4x4 IDCT are similar to the 4x4 FDCT previously presented and their mathematical definitions are not showed in this paper. All the 4x4 transforms are presented in the H.264/AVC Standard [1].

2.1. 4-Stage Pipeline with 16 Input Samples per Cycle (4P16S)

Fig. 1 shows the 4-stage pipeline with sixteen samples template.

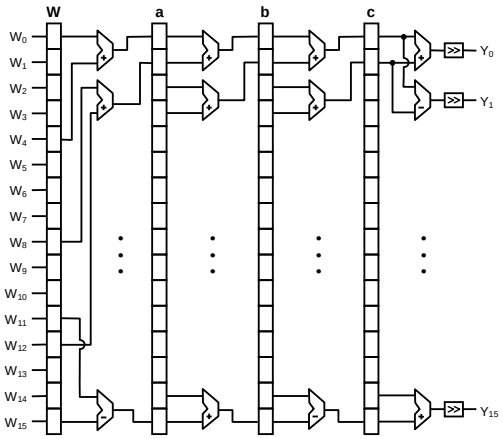

Figure 1. 4-stage pipeline architecture with 16 input samples per cycle.

This architectural template consumes an entire 4x4 block at each clock cycle. Since it uses four pipeline stages, the critical path is composed by only one operator. The combination of these two characteristics provides a high throughput for this template. In this case, the input samples take four clock cycles to be processed, so the latency of this solution is four cycles.

Architectures that use this template present high hardware consumption, since it uses four pipeline stages with sixteen operators each one. These operators are dedicated to perform just one operation, allowing a very simple control unity.

2.2. 2-Stage Pipeline with 16 Input Samples per Cycle (2P16S)

This solution is an intermediate architectural alternative for the 4P16S and the 1P16S templates. It searches the best relation between latency and throughput. The idea is to reduce the number of pipeline stages looking for a lower latency, trying to keep a high throughput.

This architectural template is similar to the previous one, the 4P16S. The difference is the number of used pipeline stages: only the **w** and the **b** register barriers presented in Fig 1 are used. It implies in a low latency of just two clock cycles in this case.

Due to the lower number of pipeline stages, the critical path (two operators serially connected) in this version is higher and the hardware consumption is lower when compared with 4P16S template.

2.3. 1-Stage Pipeline with 16 Input Samples per Cycle (1P16S)

This template is another modification of the 4P16S. This time, the **a, b** and **c** register barriers were removed (see Fig. 1). This version is also called as a "combinational version", since there are not intermediate register barriers, which leads to the most complex critical path (four operators serially connected) of all templates proposed in this work. This way, just one clock cycle is used to generate all the output values, i.e., the latency of this template is one clock cycle.

As only one register barrier is used, the hardware consumption is lower than in the previous templates.

2.4. 4-Stage Pipeline with 4 Input Samples per Cycle (4P4S)

The goal of the 4 input samples templates is to reduce the number of operators, reusing the same operator to perform more than one operation, allowing a lower parallelism level. However, it increases the control unity and the complexity of the operators.

This template presents a different hardware component: the ping-pong buffers. This component is used to keep the operands stable in each pipeline stage while new operands are being serially inserted. Fig. 2 shows the 4-Stage pipeline with four samples template.

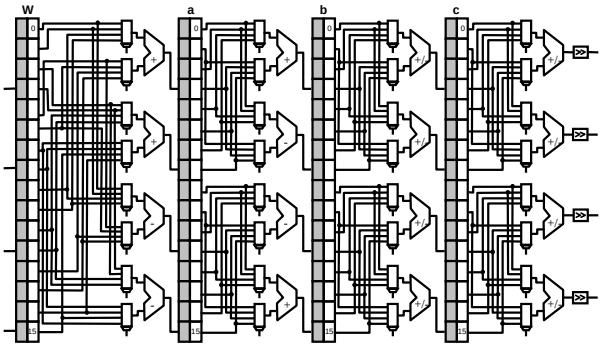

Figure 2. 4-stage pipeline architecture with four input samples per cycle.

This template presents a 16 clock cycles latency, since it implements four ping-pong buffers, and each pipeline stage takes four cycles to perform all their operations. This template presents a short critical path, of only one operator. So, this version should present a high operation frequency, similar to the 4P16S version. However, because of the operators versatility and the control unity complexity, the operation frequency is not as high as expected.

2.5. 2-Stage Pipeline with 4 Input Samples per Cycle (2P4S)

This template is an effort to achieve the lowest hardware cost of all versions. It is done by removing two ping-pong buffers of the 4P4S template (the **a** and **c**, in Fig. 2). This way, the latency is reduced to eight clock cycles. The critical path, however, is now longer (two operators serially connected), which reduces the operation frequency of the architecture.

3. SYNTHESIS RESULTS

All the architectural templates presented in the previous sections were applied to the forward 4x4 transforms defined in the H.264/AVC standard (4x4 FHAD and 4x4 FDCT). They were described in VHDL and synthesized for the EP2S15F672C3 device from

Altera Stratix II FPGA family [4] using the Altera Quartus II 7.2 software. The validation was done through the Mentor Graphics ModelSim Altera 6.1g tool, considering a behavioral and post place-and-route model of the architectures. Timing analysis was performed through the Altera Classic Timing Analyzer tool.

The architectures use 8-bits input samples. FHAD architectures use 11-bits output samples, while FDCT uses 14-bits outputs. This increase in the dynamic range occurs to avoid possible overflows in the operations.

Table 1 shows the maximum operation frequency achieved and the hardware consumption presented by the designed architectures. The first number in the Hardware Consumption fields represents the number of used ALUTs, while the second number represents the number of used DLR (Dedicated Logic Registers).

Table 1. Synthesis results to Stratix II FPGA.

Template	FHAD 4x4		FDCT 4x4	
	Freq. (MHz)	Hardware Consumption	Freq. (MHz)	Hardware Consumption
4P16S	448.03	672 and 756	438.8	676 and 789
2P16S	268.31	680 and 452	270.3	683 and 481
1P16S	162.02	681 and 340	163.2	684 and 329
4P4S	250.63	320 and 1052	241.1	494 and 1,092
2P4S	202.51	510 and 534	164.0	546 and 549

As seen, both transforms are able to reach a high frequency when using the presented architectural templates. The best operation frequency was reached by the 4P16S template, since it allows high parallelism level allied to a high number of pipeline stages. The lowest hardware cost was achieved by the 1P16S template, which does not use intermediate register barriers. Table 1 also shows that the 1P16S solutions present the lowest DLR consumption when compared to the others implementations.

Table 2 shows the processing rates achieved by the implementations using the templates that were designed in this work. The two first columns show the latency and the parallelism level of each architectural template, respectively. The performance results of the two implemented transforms (4x4 FDCT and 4x4 FHAD) are showed in three different metrics: the maximum throughput achieved in millions of samples processed per second, the maximum number of QHDTV frames (3840x2048 pixels) that can be processed per second and the minimum operation frequency necessary to allow real time when QHDTV frames are being processed.

The throughput data showed in Table 2 was obtained considering the operation frequency presented in Table 1 and the parallelism level of each used template. The number of QHDTV frames per second was calculated based in the number of samples per frame that each transform implementation needs to process.

Then, considering a QHDTV frame and a color relation of 4:2:0 (defined in the baseline, extended and main H.264/AVC profiles) [1], it was possible to determine the amount of samples that should pass through each transform operation in one frame. Since the 4x4 FDCT is applied to all information in the frame (chroma and luma), there are 11,796,480 samples to be processed by the 4x4 FDCT in one frame. In other hand, the 4x4 FHAD is used only for luma coefficients when the prediction mode was intra 16x16 [2]. In this case, only the DC coefficients generated by the DCT must be processed. The data presented in Tab. 2 for the 4x4 Hadamard throughputs considerers the worst case, where all luma input blocks were classified as intra 16x16. In this situation, there are 491,520 DC coefficients to be processed in one QHDTV frame. This is not a real situation, but is enough to evaluate the throughput of these transforms.

The best results were achieved by the fully parallel templates. They presented a higher throughput over the row based versions. The best throughput result was reached by the 4P16S template, which is able to process about 7 billions of samples per second in both transforms.

All versions are able to easily reach real time when high resolution videos are being processed (like QHDTV). Again, the fully parallel versions presented the best results, processing about 14 thousand of QHDTV frames per second in the FHAD architecture, and about 595 QHDTV frames per second in the FDCT architecture. The difference between these numbers of processed QHDTV frames per second can be explained by the number of samples per frame processed by each transform, as previously discussed.

This paper aims to find the best architectural template to be used in the implementation of the transforms, considering the Intra Prediction module constraints. Thus, the chosen template must present a good relation between low latency and high throughput. Then, the best results were reached by the 1P16S and the 2P16S in both transforms implementations.

Table 2. Architectures processing rates for Stratix II

Architectural Template	Latency (cicles)	// Level	Forward Hadamard 4x4			Forward DCT 4x4		
			Maximum Throughput (Msamples/s)	QHDTV Frames/s (@ Max. Freq.)	Min. Freq. QHDTV (MHz)	Maximum Throughput (Msamples/s)	QHDTV Frames/s (@ Max. Freq.)	Min. Freq. QHDTV (MHz)
4P16S	4	16	7,168	14,584	0.92	7,020.6	595.1	22.1
2P16S	2	16	4,292	8,734	0.92	4,324.3	366.6	22.1
1P16S	1	16	2,592	5,274	0.92	2,611.4	221.4	22.1
4P4S	16	4	1,002	2,039	3.69	964.3	81.7	88.5
2P4S	8	4	810	1,648	3.69	656.0	55.6	88.5

Disregarding the Intra Prediction constraints, it is important to note that all versions are able to reach real time even working in low frequencies, allowing lower power consumption if other type of application is considered. Tab. 2 shows that all implementations can reach real time with frequencies between 0.92 MHz and 88.5 Mhz.

4. RELATED WORKS

The comparison among works is presented in terms of latency and throughput. Since it is hard to find works using the same technology used in this paper, the comparison does not consider this fact. The parallelism level is another important characteristic to be analyzed, since it has a strictly relation with the throughput rates.

Table 3 presents the comparison between ours and related works. Then, the technology, parallelism level, latency and throughput rates of each work are presented. Since our obtained results are very similar for both transforms, comparing just one of them is sufficient. This way, we have chosen the FDCT results to be compared with related works. Thus, ours results, presented in Table 3, are relative to the FDCT results that best fits in the Intra Prediction (4P16S, 2P16S and 1P16S), as previously discussed.

The architecture presented in [5] implements just the FDCT algorithm. The solutions presented in [6, 7] are multitransforms, which means that it can handle more than one transform. The designs presented in [8] are dedicated forward transform architectures integrated in a complete T module.

Table 3. Comparison with related works

Solution	Technology	// Level	Latency (cycles)	Throughput (Msamples/s)
Our 4P16S	**Stratix II**	**16**	**4**	**7,021**
Porto [8] - FDCT	Virtex 2P	16	4	5,115
Porto [8] - FHAD	Virtex 2P	16	4	4,858
Our 2P16S	**Stratix II**	**16**	**2**	**4,324**
Our 1P16S	**Stratix II**	**16**	**1**	**2,611**
Agostini [6]	0.35 µ	16	6	3,499
Kordasiewicz [5]	-	16	1	1,720
Cheng [7]	0.35 µ	8	2	800

As shown in Table 3, the 4P16S solution presents the highest throughput values of all the compared architectures. So, the high throughput goal was reached.

The 1P16S solution presents the lowest latency, only comparable with the [5] solution. However, our solution presents a higher throughput than that solution. This way, another goal of this work was reached: low latency.

The relation between low latency and high throughput presented by our solutions is the best among the compared solutions. Thus, the main goal of this work was achieved.

5. CONCLUSIONS

This paper presented an architectural investigation for the 4x4 Forward DCT transform and the 4x4 Forward Hadamard transform of the H.264/AVC standard. The architectures were described in VHDL, and then synthesized targeting the Altera Stratix II FPGA. The validation was done through the ModelSim tool.

The main goal of this paper was to find an architectural template that fits in the Intra Prediction constraints, which means that the chosen architecture must present low latency and high throughput. This way, based in the shown results, the best options among the presented architectures are the 2P16S and 1P16S.

When compared to related works, these solutions presented the best relation between throughput and latency. Another important result is the good relation between parallelism level and throughput (the best among all compared works), which shows their efficient use of hardware. Thus, the main goal of this paper was achieved.

As future works we plan to design and to integrate the T, Q, Q^{-1} and T^{-1} modules, respecting the Intra Prediction restrictions.

6. REFERENCES

[1] INTERNATIONAL TELECOMMUNICATION UNION. ITU-T Recommendation H.264 (03/05): Advanced Video Coding for Generic Audiovisual Services, 2005.

[2] I. Richardson, *H.264 and MPEG-4 Video Compression*, Chichester: John Wiley and Sons, 2003.

[3] L. Agostini, et all, "Multiplierless and Fully Pipelined JPEG Compression Soft IP Directed to FPGAs", Journal Microprocessor and Microsystems. pp. 487-497, 2007.

[4] ALTERA CORPORATION. Altera: The Programmable Solutions Company. Available at:<http://www.altera.com.br>.

[5] R. Kordasiewicz and S. Shirani, "Hardware Implementation of the Optimized Transform and Quantization Blocks of H.264", Canadian Conf. on Electrical and Computer Engineering, pp. 943-946, 2004.

[6] L. Agostini, et all, "High Throughput Multitransform and Multiparallelism IP for H.264/AVC Video Compression Standard", IEEE International Symposium on Circuits and Systems, pp. 5417-5422, 2006.

[7] Z. Cheng, et all, "High Throughput 2-D Transform Architectures for H.264 Advanced Video Coders", IEEE Asia-Pacific Conf. on Circuits and Systems, pp. 1141-1144, 2004.

[8] R. Porto, et all, "High Throughput Architecture for Forward Transforms Module of H.264/AVC Video Coding Standard", IEEE International Conference on Electronics, Circuits and Systems, pp. 150-153, 2007.