HIGH THROUGHPUT AND LOW COST ARCHITECTURE FOR THE FORWARD QUANTIZATION OF THE H.264/AVC VIDEO COMPRESSION STANDARD

Felipe Sampaio, Daniel Palomino, Robson Dornelles and Luciano Agostini
Federal University of Pelotas - UFPel
Group of Architectures and Integrated Circuits - GACI
Pelotas, RS - Brasil
{fsampaio.ifm, danielp.ifm, rdornelles.ifm, agostini}@ufpel.edu.br

Abstract

This work presents a dedicated hardware design for the Forward Quantization Module (Q module) of the H.264/AVC Video Coding Standard, using optimized multipliers. The goal of this design is to achieve high throughput rates combined with low hardware consumption. The architecture was described in VHDL and synthesized to the EP2S60F1020C3 Altera Stratix II FPGA and to the TSMC 0.18µm Standard Cell technology. The architecture is able to operate at 364.2 MHz as a maximum operation frequency. At this frequency, the architecture is able to process 117 QHDTV frames (3840x2048 pixels) per second. The designed architecture can be used in low power and low cost applications, since it can process high resolution in real time even with very low operation frequencies and with low hardware consumption. In the comparison with related works, the designed Q module achieves the best results of throughput and hardware consumption.

Keywords: H.264/AVC standard, quantization, video coding, VLSI design.

1 INTRODUCTION

The H.264/AVC is the newest video coding standard [1]. It was developed by the union of experts from ISO/IEC and ITU-T intending to double the compression rates when compared with previous standards.

Video coding is extremely important considering the amount of data in the high definition videos. The H.264/AVC standard has efficient tools to reduce the computational representation of these videos. However, the H.264/AVC encoders have high computational complexity as well. This way, hardware solutions are necessary, at least in the current technology, when real time processing (24 to 30 frames per second) and high resolutions videos are required.

This work is part of the Brazilian research effort to design hardware solutions for the SBTVD (Brazillian System of Digital Television), since the H.264/AVC is the chosen video coding standard to be used in the SBTVD [2].

The H.264/AVC standard explores the different kinds of video redundancy when coding a video: spatial redundancy, temporal redundancy and entropic redundancy [3]. The H.264/AVC block diagram is shown in Fig.1. The standard provides several coding profiles, and this work focuses in the Main profile which represents the color information in the YCbCr color space and considers a 4:2:0 subsampling relation [3].

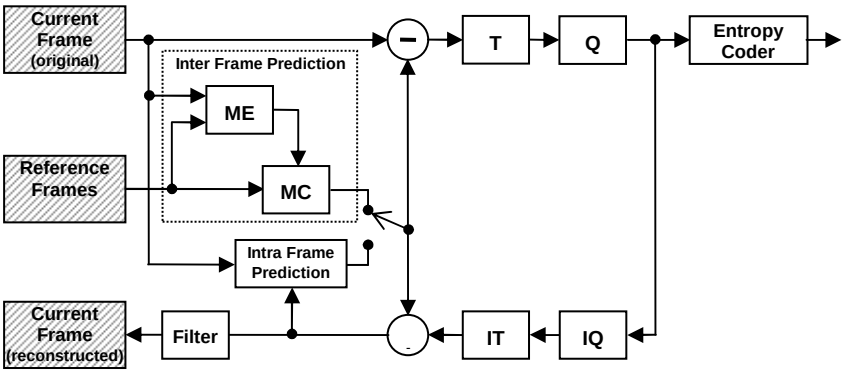

Figure 1: H.264/AVC block diagram.

The H.264/AVC Intra Frame Prediction (Fig. 1) reduces the spatial redundancy in a video. The Intra Frame Prediction is an innovation in the H.264/AVC coder, and it is responsible to explore the similarity between blocks of pixels in the same frame. This process is performed using the edge information which was already coded to represent neighbor blocks. This way, the predicted block is a copy of the edge pixels. The H.264/AVC standard defines several copy modes to the Intra Frame Prediction module. Considering luma(Y) samples, there are nine copy modes for 4x4 samples block size and four copy modes for 16x16 samples block size. For chroma (Cb and Cr) there are only four copy modes, since always 8x8 samples block size are used (these modes are equivalent to that used for 16x16 luma blocks) [3].

The temporal redundancy is treated for the Inter Frame Prediction which is composed by two modules: the Motion Estimation (ME in Fig. 1) and the Motion Compensation (MC in Fig. 1). In this case, the coding of the current block is made using the data from previously coded frames. The ME searches in the reference frames for the block that is closest to the current one and then it generates a motion information (motion vector), pointing to the block in the reference frame that will be used. After, the MC rebuilds the current block from the motion vectors generated by the ME [3].

After the prediction steps, the difference between the predicted block and the original block generates a residual information which must be processed by the transform and quantization modules in order to increase the codification rates. These steps contribute to reduce the spatial redundancy as the Intra Prediction step. However, there is data losses in the quantization process, which means that the block reconstructed in the decoder will be different from the original one [3]. As this block will be used as reference in the decoding process, it must to be used also as reference by the coder as well, guaranteeing that both, coder and decoder will use the same references. Thus, it is necessary to insert and reconstruction path in the coder side. This path is composed by the Inverse Quantization (IQ) and Inverse Transforms (IT), as is shown in the Fig. 1.

The entropic redundancy is reduced by the Entropy Coder presented in Fig. 1, which reduces the number of bits used to represent the video using lossless techniques [3].

Finally, the H.264/AVC encoder includes a deblocking filter (Filter in Fig. 1), which is used to improve the subjective quality of the coded video [3].

This work presents an architecture targeting high performance and low cost for the quantization module of the H.264/AVC coder. The main goal is to reach real time when processing high resolution videos using low operation frequencies. Then a high level of parallelism is explored to reach this goal. A secondary goal is to use the lowest possible amount of hardware resources maintaining a very high throughput, and then dedicated multipliers were designed instead of the high cost matrix multipliers. The low hardware use and the low frequencies allow reduced power consumption, since usually hardware use and frequency operation are directly proportional to power and energy consumptions. Then, considering all those features, the designed solution is able to be inserted in portable and mobile devices, which have highly non-functional restrictions, like hardware use, power and energy consumptions. The architecture consumes four samples per clock cycle (one line of a 4x4 block) and it was designed with the goal to reaches performance to guarantee the processing of high resolution videos in real time. The targeted technologies in this work were: Altera Stratix II FPGA and TSMC 0.18 µm Standard Cell. The FPGA version was done to be integrated with other encoder modules designed by other research teams in Brazil, since a FPGA prototype must be delivered by the Brazilian researchers in order to reach the goals of the SBTVD system deployment. The standard-cells version of this architecture was generated to allow the comparison with related works.

This work is organized as follow: Section 2 describes the Q module and show its definition; Section 3 presents the designed architecture; Section 4 describes the design and the verification methodologies; Section 5 shows the architecture synthesis results targeting the technologies already mentioned; Section 6 shows a comparison with related works and, Section 7 concludes this work and shows some future works.

2 THE H.264/AVC QUANTIZATION

The H.264/AVC standard defines that the residual information generated by the prediction modules must be transformed from spatial domain to the frequency domain. This operation is done by the transforms module (T in Fig. 1), which is composed by three different transforms [3]. In the frequency domain, the quantization discards or attenuates those frequencies that are less perceptible by the human eye. The quantization, typically, generates sparse matrixes which are easier to be compressed by entropy coder [3], increasing the compression rates.

The Forward Quantization Module inputs are the coefficients that come from the transforms module. After the transformation, the samples of a 4x4 block are called coefficients: the (0,0) element in a block is called DC coefficient and the others elements are called AC coefficients [3].

The H.264/AVC is the first compression video standard that uses three different transforms in the T and IT modules shown in Fig. 1. In the T module, all the input coefficients are processed by the forward DCT (4x4 FDCT). Besides, some DC coefficients are processed by the Hadamard transform, in order to explore a residual correlation among the DC elements of 4x4 blocks. All the chroma DC coefficients are processed by the 2x2 Hadamard transform (2x2 HAD), while the luma DC coefficients generated by the I16MB mode are processed by the 4x4 Hadamard transform (4x4 FHAD) [3].

The quantization process is made in different ways, according to the color component (Y, Cb or Cr) and the chosen prediction mode.

When the prediction mode is I4MB, both AC and DC coefficients quantization calculation are realized according to the Equation (1) [4]. This equation is applied to all AC coefficients in the I16MB and chroma modes. The quantization calculation for all residual data generated by the Inter Frame Prediction is also calculated through this equation.

$$\begin{aligned} |Z_{(i,j)}| &= (|W_{(i,j)}| \cdot MF + f) >> qbits \\ sign(Z_{(i,j)}) &= sign(W_{(i,j)}) \end{aligned} \quad (1)$$

For all luma DC coefficients where the prediction mode was I16MB, and for chroma DC coefficients, the quantization calculation is presented in the Equation (2) [4]. This way, all the output coefficients of the Hadamard transforms are applied to this equation.

$$\begin{aligned} |Z_{D(i,j)}| &= (|Y_{D(i,j)}| \cdot MF_{(0,0)} + 2f) >> (qbits + 1) \\ sign(Z_{D(i,j)}) &= sign(Y_{D(i,j)}) \end{aligned} \quad (2)$$

The constants used in equations (1) and (2) are defined in Tab. 1 and in the Equations (3) and (4). All these values are chosen considering the Quantization Parameter (QP). The QP value defines the "strength" of the quantization process, since a high QP value results in a higher compression rate, however the image quality decreases. The QP value is delivered by the encoder global control and it is an input for the Q module. The % operator in Tab.1 is the remainder operator.

Table 1: MF values according to the QP

QP%6	PF0 (0,0),(2,0),(2,2),(0,2) Positions	PF1 (1,1),(1,3),(3,1),(3,3) Positions	PF2 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

$$\begin{aligned} f &= 2^{qbits}/3, \text{ for Intra Frame Prediction} \\ f &= 2^{qbits}/6, \text{ for Inter Frame Prediction} \end{aligned} \quad (3)$$

$$qbits = 15 + \lfloor QP/6 \rfloor \quad (4)$$

Example: Considering a situation where the input value is Y=150, in the (1,1) 4x4 block position, the prediction mode was I4MB and QP=16. The operation QP%6 is 4, which means that MF value, according Tab. 1, is 3355. By the equations (3) and (4) the values generated are qbits=17 and f=43690. This way, applying these values in the equation (1) the output value will be Z=4.

More detailed definitions of the calculations used in the quantization module and how the constants were obtained, can be found in the H.264/AVC standard [1]. The derivation process to take the original definitions can be found in [3].

3 DESIGNED ARCHITECTURE

As presented in the previous section, the Forward Quantization process defined in the H.264/AVC is, basically, composed by one multiplication of the input value by an constant (MF), the sum with other constant (f or 2f) and the right shit operation controlled by other constant (qbits or qbits+1). This way, the data path of a generic architecture that performs the Forward Quantization must contain at least: one multiplier, one adder and one barrel shifter. Among these operations, the multiplication has the higher computational complexity. Thus, all possible optimizations in the multiplier architecture are extremely useful for the final performance of the complete architecture.

The MF constant value depends on the target coefficient position in the 4x4 block. Based in Tab. 1, it is possible to notice that there are three sets of positions where the MF is the same. This way, dedicated multipliers can be designed for each set of MF values, with the goal of reducing the multiplication impact in the architecture critical path. Then, three different data paths were designed, with dedicated multipliers according with the relative coefficient position.

Since the number of values that the constants can assume is small, these values were pre calculated and stored in memory. The control unit is responsible to address the right position in this memory using the QP value.

Several architectural solutions for the multipliers were analyzed, departing with (1) an conventional matrix multiplier, passing by (2) an multiplier that uses the shift-add decomposition and, at last, taking advantage of the other operations that compose the quantization process, the exploration arrives at (3) an dedicated multiplier that join the MF multiplication, the sum with the constant f and the 15-bit shift.

The shift-add decomposition is based on the binary arithmetic property which defines a relation between shifts-left and multiplications. The shift-left of one binary position in an input number is equivalent to a multiplication by two. Using different shifts to generate partial results and adding all these partial results is possible to multiply the input for any number. Fig. 2 shows an example where the multiplication was decomposed in two partial products and the result is obtained adding these two partial results.

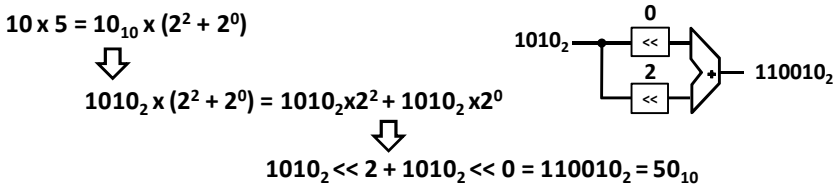

Figure 2: Example of a multiplication decomposed in shift-adds.

The designed architecture works over 15-bit input values, and its RTL diagram is presented in Fig. 3.

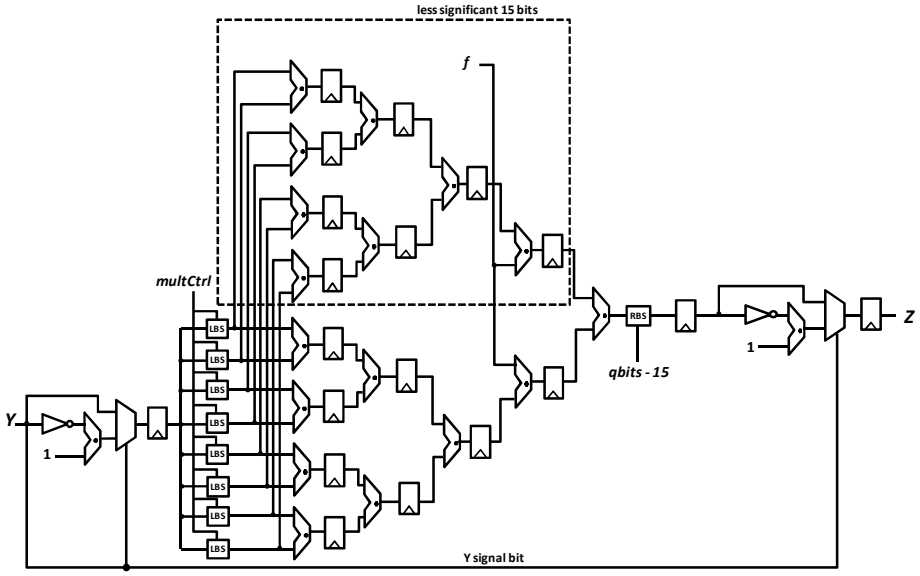

Figure 3: Designed Q module data path.

The quantization calculation defines a right shift operation of 15 bits, i.e., after the add with the f constant, the 15 less significant bits must be discarded. In the solution designed in this work, this shift operation was directly applied at the input values, even before the multiplication. This way, it was possible to reduce the adders bit width, decreasing the architecture critical path. In this case, the original adders, which have a bit width of 30 bits, could be replaced by adders with a half of the original critical path, with a bit width of only 15 bits. An auxiliary architecture performs the overflow verification with the 15 discarded bits to avoid arithmetic errors. If this verification detects the occurrence of overflow, then a correction factor is added to the final result. This optimization allowed an important reduction in the critical path of this multiplier, but it does not implies in the reduction of the used hardware, since one adder with 30 bits was broken in two adders with 15 bits.

The first and the last adders perform the 2-complement operation, since the quantization works over absolute input values, as defined in Equations (1) and (2).

The *multCtrl*, *dc_flag* and *qbits* are delivered to the datapath by the control unit, and their values are defined in function of: the prediction mode that was used in the current block coding, the target coefficient (AC or DC), the QP value and the coefficient position in the 4x4 block.

As the shift-left operations generate zero bits in the data less significant part, the adders that compose the operation tree could be simplified. It do not cause any gain in terms of critical path, since the last adder must operates over all bits. However, this simplification has a considerable impact in the hardware consumption. Considering a Ripple Carry implementation for the adders, the number of 1-bit full adders was substantially reduced.

For example, taking the data path that performs the (0,0) position quantization and 15-bit width for the input values, these simplifications appoints a reduction of about 22% in the 1-bit full adders.

Each adder column was isolated between temporal barriers (registers), in order to obtain the maximum possible performance. This way, the designed pipeline scheme contains seven stages and the critical path is defined by the larger bit width adder.

The entire architecture is composed by four datapaths working in a parallel way, as showed in Fig. 4. Then, the architecture consumes four coefficients generated by the T module at each clock cycle. The datapath names in the Fig. 4 (PF0, PF1 and PF2) are related to the supported multiplications and are associated to the names presented in Tab. 1.

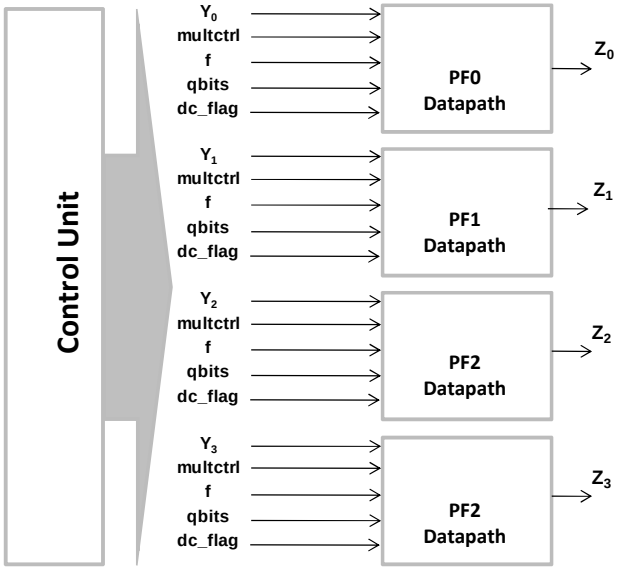

Figure 4: Q module architecture block diagram.

It is important to notice that the PF2 datapath is instantiated twice in the final architecture. It occurs because this datapath is responsible to process eight among the 16 coefficients of the 4x4 block, as presented in Tab. 1. The PF0 and PF1 datapaths work processing just four coefficients each one. Thus, since the PF2 datapath is duplicated, each datapath is responsible to operate over four specific coefficients.

The architecture latency is seven clock cycles, i.e., it takes seven cycles to deliver the first valid output. After the moment that the pipeline is filled, the architecture produces four quantized coefficients at each cycle.

The 2x2 block of DC chroma coefficients that is generated by the 2x2 HAD is completely consumed in four clock cycles in the designed architecture, since all coefficients must be processed by the same data path (PF0). In the same way, the 4x4 block of DC luma coefficients that comes from the 4x4 FHAD is processed by the architecture in 16 cycles, since it also must be processed by the PF0 datapath. All the datapaths are used for the 4x4 FDCT outputs, and the 4x4 block is completely processed in four clock cycles when processing the FDCT coefficients.

4 DESIGN AND VERIFICATION METHODOLOGY

The architecture presented in the previous section was described in VHDL and synthesized targeting the EP2S60F1020C3 Altera Stratix II FPGA [5] and TSMC 0.18 µm standard-cells technology [6]. The Altera Quartus II 9.0 tool was used for the FPGA synthesis. Leonard Spectrum was used to generate the synthesis results targeting the standard-cells implementation.

The architecture verification was performed using the Mentor Graphics ModelSim 6.4g tool. It was made following a sequence of steps. Firstly, code routines were inserted in the JM 16.0 Reference Software [7] of the H.264/AVC Standard (that is implemented in C language), in order to take the input and output data for the quantization module, which were used as golden model in this design.

The input and output data for the quantization module was generated running the modified reference software over real video sequences, this way the architecture could be verified with real data. The used video sequences were some traditional sequences used in this research area like: Foreman, City and Soccer. These videos had CIF resolutions (352x288 pixels) in a 4:2:0 color relation.

After that, one test bench was described in VHDL to feed the quantization architecture with the input data generated by the reference software. Then, it was possible to run both architectures in the ModelSim tool to take the output data. Another C code was designed to compare the output data from the architecture with the output data delivered by the reference software.

5 SYNTHESIS RESULTS

Tab. 2 shows the synthesis results in terms of maximum operation frequency and hardware consumption, considering the two targeted technologies: Altera Stratix II FPGA and TSMC 0.18 µm standard-cells.

Table 2: Synthesis results for the two targeted technologies

Complete Q Module		
Stratix II FPGA	**Freq. (MHz)**	291.7
	#ALUTs	1,308 (3%)
	#DLRs	956 (2%)
TSMC 0.18µm	**Freq. (MHz)**	346.2
	#Gate	14.5k

As expected, the standard-cell implementation achieves the best results. In this technology, the architecture is able to operate at 346.2 MHz, overcoming in 19% the maximum frequency achieved by the FPGA synthesis.

When mapped to FPGA, the architecture uses 1,308 ALUTs and 965 dedicated registers. This use represents only 3% of the target FPGA. This synthesis did not consider the use of internal multipliers of the target FPGA since the designed architecture was optimized exactly in the multiplier construction. The main gains of the designed solution (split the input bit-width and explore internal pipeline) would not be possible when using conventional multipliers as those available in the target FPGA.

Considering these results, it is possible to perform a processing rate evaluation. Tab. 3 shows this evaluation, considering the standard-cells synthesis.

Table 3: Processing rates for the standard cell implementation

	Throughput (billions of coeff./s)	**QHDTV frames/s**	**HDTV frames/s**	**Min. Freq. HDTV @ 30fps (MHz)**
Q Module	1.3848	117.4	445.2	23.3

The first column in Tab. 3 presents the throughput of the quantization architecture when running at the maximum allowed operation frequency. In this case, the architecture is able to process 1.4 billions of coefficients per second, which is a very high processing rate. This high throughput allows the architecture to process very high resolution videos in real time (30 frames per second). The architecture is able to process 117 QHDTV frames (3840x2048 pixels) per second or 445 HDTV frames (1920x1080 pixels) per second, when operating at its maximum operation frequency. These frame rates are very higher than the 30 frames per second which is necessary to the real time requirements. Then the proposed architecture is recommended for low energy consumption applications, since it can process high resolution videos even working at low frequencies. As presented in Tab. 3, an operation frequency of only 23.3 MHz is enough to the architecture reach real time when processing HDTV videos (1920x1080 pixels).

6 COMPARISON WITH RELATED WORKS

Comparison with related works of literature is not a trivial task. This is due to the particular goal of each work to achieve better results considering a set of specific restrictions. We did not find any work targeting FPGAs, and then we generated the standard-cells version of our architecture to allow the comparison with previously published works.

The comparison performed here presents four published works in literature. Kordasiewicz [8] presents two architectural solutions for the Q module: one with several optimizations with the goal to reduce the hardware consumption and other with optimizations targeting high operation frequencies. Besides, the solution designed by Peng [9] presents an entire solution for the transforms and quantization. The work proposed by Zhang [10] presents a solution that uses dedicated multipliers, like this work. However, it does not design the entire Q module. The architectural design proposed by Owaida [11] implements an optimization in the quantization algorithm targeting the QFHD resolution (3840x2160 pixels).

The work [10] presents six different multipliers for each set of 4x4 block positions, while this work proposes only one multiplier architecture that is able to perform the same operations. This reduces drastically the Q module hardware consumption when compared to a module that uses the multiplier proposed by [10]. More consistent comparisons with Zhang [10] results were not performed because that paper does not present synthesis results for the complete Q module.

The design presented in [11] considers the maximum possible parallelism (16 samples) and it does not use pipeline. The high parallelism level was used to meet the requirements of real time in QFHD videos. Our design decision was the use of a lower level of the parallelism level and the use of pipeline to accelerate the processing and to reach high throughput rates. This decision was based on the mainly innovation of our work: the reduced bit width operation tree allowed by the simplification in the quantization algorithm. As the technologies are different and the work [11] does not show results considering only the quantization module, a complete and fair comparison was not possible.

Tab. 4 shows the comparison among the solutions proposed in this work with the two architectural solutions proposed in [8] and with the solution designed in [9].

Table 4: Comparison with Related Works.

	This Work	**Kordasiewicz[8]** ***Area***	**Kordasiewicz[8]** ***Critical Delay***	**Peng [9]**
Technology	**0.18µm**	0.18µm	0.18µm	0.18µm
Parallelism Level	**4**	1	16	16
Throughput (billions of coeff./s)	**1.385**	0.02	1.085	-
Area (thousands of gates)	**14.5**	1.75	39.9	20.9
Throughput/ Area	**95.5**	11.4	27.2	-

With respect to the solution [8] with optimizations in the area, the results in hardware consumption of the architecture designed in this work are higher. It is justified because of the architecture [8] have been designed with

parallelism of one coefficient per cycle. Thus, only one datapath is used. This low level of parallelism implies in a low throughput; about 70 times lower than the performance achieved by the architecture designed in this work.

Considering the solution [8] with critical path optimizations, the throughput results are closed to those ones achieved in this work, since it uses the maximum parallelism allowed (16 coefficients per cycle). It causes an increasing in the hardware consumption in 23 times when compared with the area optimized solution. Even with this parallelism increasing, the achieved throughput is about 1.2 times lower than that achieved in this work.

Despite a gain in the area optimized solution proposed in [8], the architecture designed in this work achieves the best results in the relationship between throughput and area used from the two solutions proposed by [8].

The author of work [9] does not inform the results in terms of performance. This way, it was necessary an estimated throughput considering the description presented by the author. Considering the architectural design, the solution proposed in [9] parallelizes the quantization processes of the 4x4 coefficient block and it process one complete block per cycle. It also performs the whole quantization process in one clock cycle. Then the throughput of [9] is probably higher than that reached by our work, but it also used more hardware resources than our work our work.

7 CONCLUSIONS AND FUTURE WORKS

This paper presented an architecture with low cost and high throughput to the H.264/AVC forward quantization module. Initially, the H.264/AVC standard and its mainly coding tools were explained to show in what point this work focuses. Then, the quantization module was explained showing its complexity inside the entire encoder path. Once the quantization bottleneck was discovered (the multiplication), several architectures of classic multipliers were analyzed to find the best solution to be employed in the proposed quantization path. The final solution for the multiplier takes advantage of other operations in order to simplify the critical path. After that, the methodologies that were used in the development of this work were described.

The design considers the parallelism of four samples per cycle and explores the maximum use of pipeline stages between adders to speed up the coefficients processing derived from the transforms. The architecture was described in VHDL and synthesized targeting two different technologies: Altera Stratix II FPGA and TSMC 0.18µm standard-cells. The synthesis results appoint 346.2 MHz as maximum operation frequency. Thus, the architecture is able to process around 117.4 QHDTV frames per second. Considering applications with low power restrictions, the designed solution achieves real time processing for high resolution videos even in low frequencies, like 23.3MHz. When compared with related works, the designed architecture presents very good results, mainly when comparing throughput versus area, when our solution surpassed the related works with the same target technology.

As future works, it is intended the replacement of the operators by compressor adders to reduce the critical delay of the architecture using less pipeline stages. Furthermore, the power evaluation or the designed architecture is also a planned future work. This evaluation is important to guide some optimization to reduce the energy consumption, since this characteristic is extremely important in some applications, like mobile systems. Besides, with power results, the comparison with related works can aggregate this important non-functional characteristic.

Acknowledgements

The authors are thankful for the financial support of FAPERGS, CNPQ and CTIC/RNP Brazilian research support agencies.

References

[1] ITU-T: Recommendation H.264/AVC (11/2007). Advanced Video Coding for Generic Audiovisual Services. (2007)

[2] Brazilian Forum of Digital Television. ISDTV Standard. Draft. Dec. 2006 (in portuguese).

[3] Richardson, I.: H.264/AVC and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia. John Wiley&Sons, Chichester. (2003)

[4] Malvar, H.S.; Hallapuro, A.; Karczewicz, M.; Kerofsky, L.: Low-complexity transform and quantization in H.264/AVC. In: IEEE Transactions on Circuits and Systems for Video Technology. vol 13. (2003)

[5] Altera Corporation: Altera: The Programmable Solutions Company. Available at: www.altera.com.

[6] Artisan Components: TSMC 0.18 µm 1.8-Volt SAGE-XTM Standard Cell Library Databook. (2001)

[7] Joint-Video Model (JM 16.0). Available at: http://iphome.hhi.de/suehring/tml/

[8] Kordasiewicz, R.C., Shirani, S.: Asic and FPGA Implementations of H.264 DCT and Quantization Blocks. In: IEEE Int. Conf. on Image Processing, IEEE, Piscataway, (2005)

[9] Peng, C., Yu, D., Cao, X., Sheng, S. A New High Throughput VLSI Architecture for H.264/AVC Transform and Quantization. In: 7th International Conference on ASIC, 2007. (2007)

[10] Zhang, Y., Jiang, G., Yi, W., Yu, M., Li, F., Jiang, Z. and Liu, W.: An Improved Design of Quantization for H.264 Video Coding Standard. In: 8th Int. Conf. on Signal Processing, IEEE, Piscataway, vol. 2. (2006)

[11] Owaida, M., Koziri, M., Katsavounidis, I., Stamoulis, G. A Hoigh Performance and Low Power Hardware Architecture for the Transform and Quantization Stages in H.264. In: IEEE International Conference on Multimedia & Expo, IEEE, Piscataway, (2009)