A pipelined 8×8 2-D forward DCT hardware architecture for H.264/AVC high profile encoder

Conference Paper · December 2007

DOI: 10.1007/978-3-540-77129-6_5 · Source: DBLP

6 authors, including:

Thaisa Silva
Faculdade Meridional
38 PUBLICATIONS 169 CITATIONS
SEE PROFILE

Cláudio Machado Diniz
Universidade Católica de Pelotas (UCPel)
41 PUBLICATIONS 190 CITATIONS
SEE PROFILE

João Alberto Vortmann
Universidade Federal de Pelotas
5 PUBLICATIONS 19 CITATIONS
SEE PROFILE

Luciano Volcan Agostini
Universidade Federal de Pelotas
312 PUBLICATIONS 1,618 CITATIONS
SEE PROFILE

A Pipelined 8x8 2-D Forward DCT Hardware Architecture for H.264/AVC High Profile Encoder

Thaísa Leal da Silva[1], Cláudio Machado Diniz[1], João Alberto Vortmann[2],
Luciano Volcan Agostini[2], Altamiro Amadeu Susin[1], and Sergio Bampi[1]

[1] UFRGS – Federal University of Rio Grande do Sul - Microelectronics Group
Porto Alegre - RS, Brazil
{tlsilva, cmdiniz, bampi}@inf.ufrgs.br, altamiro.susin@ufrgs.br
[2] UFPel – Federal University of Pelotas – Group of Architectures and Integrated Circuits
Pelotas - RS, Brazil
{agostini, jvortmann}@ufpel.edu.br

Abstract. This paper presents the hardware design of an 8x8 bi-dimensional Forward Discrete Cosine Transform used in the high profiles of the H.264/AVC video coding standard. The designed DCT is computed in a separate way as two 1-D transforms. It uses only add and shift operations, avoiding multiplications. The architecture contains one datapath for each 1-D DCT with a transpose buffer between them. The complete architecture was synthesized to Xilinx Virtex II - Pro and Altera Stratix II FPGAs and to TSMC 0.35µm standard-cells technology. The synthesis results show that the 2-D DCT transform architecture reached the necessary throughput to encode high definition videos in real-time when considering all target technologies.

Keywords: Video compression, 8x8 2-D DCT, H.264/AVC standard, Architectural Design.

1 Introduction

H.264/AVC (MPEG 4 part 10) [1] is the latest video coding standard developed by the Joint Video Team (JVT) which is formed by the cooperation between ITU Video Coding Experts Group (VCEG) and ISO/IEC Moving Pictures Experts Group (MPEG). This standard achieves significant improvements over the previous standards in terms of compression rates [1].

H.264/AVC standard was firstly organized in three profiles: Baseline, Extended and Main. A profile defines a set of coding tools or algorithms which can be used to generate a video bitstream [2]. Each profile is targeted to specific classes of video applications. The first version of H.264/AVC standard was focused on "entertainment-quality" video. In July 2004, a extension was added to this standard, called the Fidelity Range Extensions (FRExt). This extension focused on professional applications and high definition videos [3]. Then, a new set of profiles was defined and this set was generically called High profile, which is the focus of this work. There are four different profiles in the High profile set, both targeting high quality videos: High profile (HP) includes support to video with 8 bits per sample and with an YCbCr

D. Mery and L. Rueda (Eds.): PSIVT 2007, LNCS 4872, pp. 5 – 15, 2007.

color relation of 4:2:0. High 10 profile (Hi10P) supports videos with 10 bits per sample and also with a 4:2:0 color relation. High 4:2:2 profile (H422P) supports a 4:2:2 color relation and videos with 10 bits per sample. Finally, High 4:4:4 profile (H444P) supports a 4:4:4 color relation (without color subsampling) and videos with 12 bits per sample.

One improvement present in High profiles is the inclusion of an 8x8 integer transform in the forward transform module. This transform is an integer approximation of the 8x8 2-D Discrete Cosine Transform (DCT) and it is commonly referred as 8x8 2-D DCT in this standard [3].

This new transform is used to code luminance residues in some specific situations. Other profiles support only 4x4 DCT transform. However, significant compression performance gains were reported for Standard Definition (SD) and High Definition (HD) solutions when larger than 4x4 transforms are used [4]. So, in High profiles, the encoder can choose adaptively between the 4×4 and 8×8 transforms, when the input data was not intra or inter predicted using sub-partitions smaller than 8x8 samples [5][6].

Fig. 1 presents a block diagram of the H.264 encoder. The main blocks of the encoder [7], as shown in Fig. 1, are: motion estimation (ME), motion compensation (MC), intra prediction, forward and inverse (T and T^{-1}) transforms, forward and inverse quantization (Q and Q^{-1}), entropy coding and de-blocking filter.

This work focuses on the design of an 8x8 2-D forward DCT hardware architecture, which composes the T Block of H.264/AVC coders when the high profile is considered. T module is highlighted in Fig. 1. This architecture was designed without multiplications, just using shift-adds operations, aiming to reduce the hardware complexity. Besides, the main goal of the designed architecture was to reach the throughput to process HDTV 1080 frames (1080x1920 pixels) in real time, allowing its use in H.264/AVC encoders targeting HDTV. This architecture was synthesized to Altera and Xilinx FPGAs and to TSMC 0.35µm standard-cells and the synthesis results indicated that the 2-D DCT designed in this work reaches a very high throughput, making possible its use in a complete video coder for high resolutions videos. We did not find any solution in the literature which presents a H.264/AVC 8x8 2-D DCT completely designed in hardware.

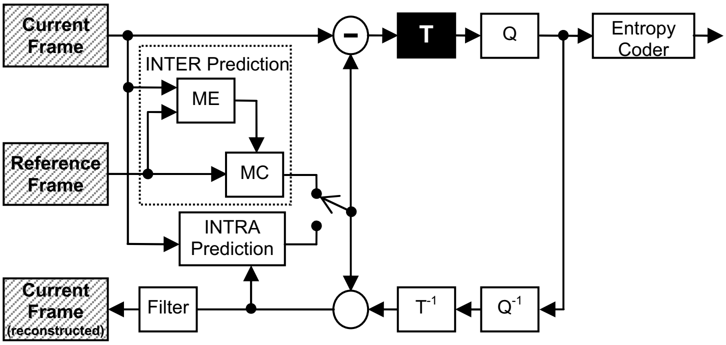

Fig. 1. Block diagram of a H.264/AVC encoder

This paper is organized as follows: section two presents a review of the 8x8 2-D forward DCT transform algorithm. The third section presents the designed architecture. Section four presents the validation strategy. The results of this work and the discussions about these results are presented in section five. Section six presents comparisons of this work with related works. Finally, section seven presents the conclusions of this work.

2 8x8 2-D Forward DCT Algorithm

The 8x8 2-D forward DCT is computed in a separable way as two 1-D transforms: a 1-D horizontal transform (row-wised) and a 1-D vertical transform (column-wised). The 2-D DCT calculation is achieved through the multiplication of three matrices as shown in Equation (1), where X is the input matrix, Y is the transformed matrix, C_f is the transformation matrix and C_f^T is the transposed of the transformation matrix. The transformation matrix C_f is showed in Equation (2) [5][8].

$$Y = C_f X C_f^T \tag{1}$$

$$C_f = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix} \cdot \tfrac{1}{8} \tag{2}$$

This transform can be calculated through fast butterfly operations accordingly to the algorithm presented in Table 1 [5], where *in* denotes the vector of input values, *out* denotes the transformed output vector and *a* and *b* are internal variables.

Table 1. 2-D Forward 8x8 DCT Algorithm

Step 1	**Step 2**	**Step 3**
a[0] = in[0] + in[7];	b[0] = a[0] + a[3];	out[0] = b[0] + b[1];
a[1] = in[1] + in[6];	b[1] = a[1] + a[2];	out[1] = b[4] + (b[7]>>2);
a[2] = in[2] + in[5];	b[2] = a[0] - a[3];	out[2] = b[2] + (b[3]>>1);
a[3] = in[3] + in[4];	b[3] = a[1] - a[2];	out[3] = b[5] + (b[6]>>2);
a[4] = in[0] - in[7];	b[4] = a[5] + a[6]+ ((a[4]>>1) + a[4]);	out[4] = b[0] - b[1];
a[5] = in[1] - in[6];	b[5] = a[4] - a[7] - ((a[6]>>1) + a[6]);	out[5] = b[6] - (b[5]>>2);
a[6] = in[2] - in[5];	b[6] = a[4] + a[7] - ((a[5]>>1) + a[5]);	out[6] = (b[2]>>1) - b[3];
a[7] = in[3] - in[4];	b[7] = a[5] - a[6] + ((a[7]>>1) + a[7]);	out[7] = - b[7] + (b[4]>>2);

This algorithm was derived from Equation (1) and it needs of three steps to compute the 1-D DCT transform. However, in this work the algorithm presented in [5] was modified in order to reduce the critical path of the designed architecture and to allow a better balanced pipeline when the architecture was designed. This modified algorithm was divided in five steps, allowing the architectural design in a five stages pipeline.

The modified algorithm is presented in Table 2 and it computes the 1-D DCT transform in five steps. This algorithm uses only one addition or subtraction to generate each result, allowing the desired best balancing between the calculation stages.

Table 2. 2-D Forward 8x8 DCT Modified Algorithm

Step 1	**Step 2**	**Step 3**
a[0] = in[0] + in[7];	b[0] = a[0] + a[3];	c[0] = b[0];
a[1] = in[1] + in[6];	b[1] = a[1] + a[2];	c[1] = b[1];
a[2] = in[2] + in[5];	b[2] = a[0] - a[3];	c[2] = b[2];
a[3] = in[3] + in[4];	b[3] = a[1] - a[2];	c[3] = b[3];
a[4] = in[0] - in[7];	b[4] = a[5] + a[6];	c[4] = b[4] + (b[8]>>1);
a[5] = in[1] - in[6];	b[5] = a[4] - a[7];	c[5] = b[5] - (b[10]>>1);
a[6] = in[2] - in[5];	b[6] = a[4] + a[7];	c[6] = b[6] - (b[9]>>1);
a[7] = in[3] - in[4];	b[7] = a[5] - a[6];	c[7] = b[7] + (b[11]>>1);
	b[8] = a[4];	c[8] = b[8];
	b[9] = a[5];	c[9] = b[9];
	b[10] = a[6];	c[10] = b[10];
	b[11] = a[7];	c[11] = b[11];

Step 4	**Step 5**
d[0] = c[0];	out[0] = d[0] + d[1];
d[1] = c[1];	out[1] = d[4] + (d[7]>>2);
d[2] = c[2];	out[2] = d[2] + (d[3]>>1);
d[3] = c[3];	out[3] = d[5] + (d[6]>>2);
d[4] = c[4] + c[8];	out[4] = d[0] - d[1];
d[5] = c[5] - c[10];	out[5] = d[6] - (d[5]>>2);
d[6] = c[6] - c[9];	out[6] = (d[2]>>1) - d[3];
d[7] = c[7] + c[11];	out[7] = (d[4]>>2) - d[7];

3 Designed Architecture

Based on the modified algorithm presented in Section 2, a hardware architecture for the 8x8 2-D Forward DCT transform was designed. The architecture uses the 2-D DCT separability property [9], where the 2-D DCT transform is computed as two 1-D DCT transforms, one row-wised and other column-wised. The transposition is made by a transpose buffer. The 2-D DCT block diagram is shown in Fig. 2.

The designed architecture was designed to consume and produce one sample per clock cycle. This decision was made to allow an easy integration with the other

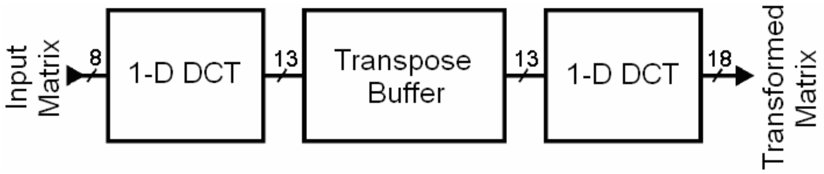

Fig. 2. 2-D DCT Block Diagram

transforms designed in our research group for the T module of the H.264/AVC main profile [10], which were designed with this production and consumption rates.

The two 1-D DCT modules are similar; the difference is the number of bits used in each pipeline stage of these architectures and consequently, in the number of bits used to represent each sample. This occurs because at each addition operation could generate a carry out and the number of bits to represent the data increases in one bit. Both 1-D DCT modules were designed at the same way and the input and output bit-widths are changed for the second 1-D DCT module.

The control of this architecture was hierarchically designed and each sub-module has its own control. A simple global control is used to start the sub-modules operation in a synchronous way.

The designed architecture for the 8x8 1-D DCT transform is shown in Fig. 3. The hardware architecture implements the modified algorithm presented in Section 2. It has a five stage pipeline and it uses ping-pong buffers, adders/subtractors and multiplexers. This architecture uses only one operator in each pipeline stage as shown in Fig. 3.

Ping-pong buffers are two register lines (*ping* and *pong*), each register with *n* bits. The data inputs serially in the *ping* buffer, one sample at each clock cycle. When *n* samples are ready at the *ping* buffer, they are sent to the *pong* buffer in parallel [11]. There are five ping-pong buffers in the architecture and these registers are necessary to allow the pipeline synchronization.

The 1-D DCT was the first designed module. A Finite State Machine (FSM) was designed to control the architecture datapath.

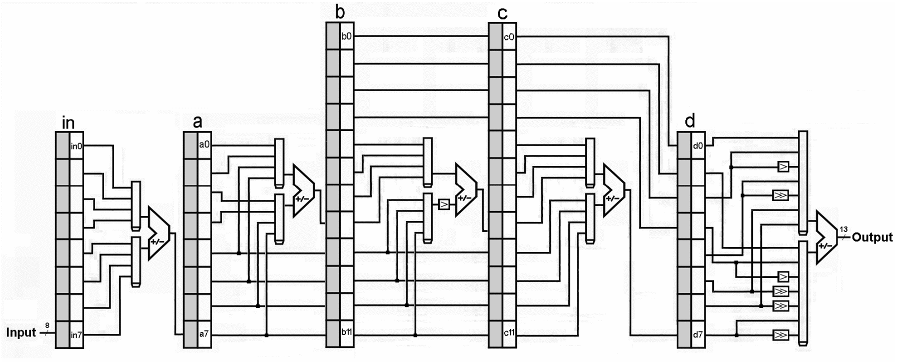

Fig. 3. 1-D 8x8 DCT Architecture

A transpose buffer [11] was designed to transpose the resulting matrix from the first 1-D DCT, generating the input matrix to the second 1-D DCT transform.

The transpose buffer is composed of two 64-word RAMs and three multiplexers besides various control signals, as presented in Fig. 4. The RAM memories operate in an intercalated way: while one of them is used for writing, the other one is used for reading. Thus, the first 1-D DCT architecture writes the results line by line in one memory (RAM1 or RAM2) and the second 1-D DCT architecture reads the input values column by column from the other memory (RAM2 or RAM1).

The signals *Wad* and *Rad* define the address of memories and the signals *Control1* and *Control2* defines the read/write signal of memory. The main signals of this architecture are also controlled by a local FSM.

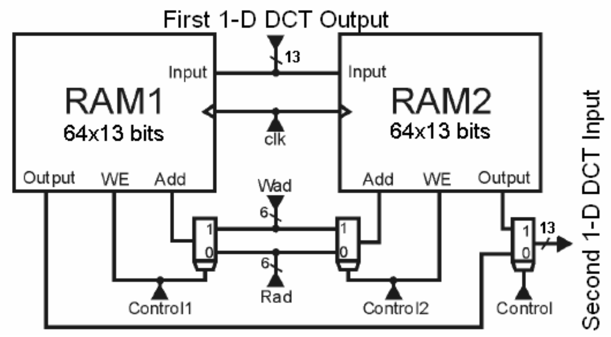

Fig. 4. Transpose Buffer Architecture

Each 1-D DCT architecture has its own FSM to control its pipeline. These local FSMs control the data synchronization among these modules.

The first 1-D DCT architecture has an 8-bit input and a 13-bit output. The transpose buffer has a 13-bit input and output. In the second 1-D DCT architecture a 13-bit input and an 18-bit output is used. Finally, the 2-D DCT architecture has an 8-bit input and an 18-bit output.

The two 1-D DCT architectures have a latency is of 40 clock cycles. The transpose buffer latency is of 64 clock cycles. Then, the global 8x8 2-D DCT latency is of 144 clock cycles.

4 Architecture Validation

The reference data for validation of the designed architecture was extracted directly from the H.264/AVC encoder reference software and ModelSim tool was used to run the simulations.

A testbench was designed in VHDL to generate the input stimulus and to store the output results in text files. The used input stimuli were the input data extracted from the reference software. The first simulation considers just a behavioral model of the designed architecture. The second simulation considers a post place-and-route model of the designed architecture. In this step the ISE tool was used together with

ModelSim to generate the post place-and-route information. The target device selected was a Xilinx VP30 Virtex-II Pro FPGA. After some corrections in the VHDL descriptions, the comparison between the simulations results and the reference software results indicates no differences between them.

The designed architecture was also synthesized for standard-cells, using Leonardo Spectrum tool, the target technology was TSMC 0.35um. After, the Modelsim tool was used again to run new simulations considering the files generated by Leonardo and to validate the standard-cells version of this architecture.

5 Synthesis Results

The architectures of the two 1-D DCTs and the Transpose Buffer were described in VHDL and synthesized to Altera Stratix II EP2S15F484C3 FPGA, Xilinx VP30 Virtex II Pro FPGA and TSMC 0.35µm standard-cell technologies. These architectures were grouped to form the 2-D DCT architecture which was also synthesized for these target technologies.

The 2-D DCT architecture was designed to reach real time (24fps) when processing HDTV 1080 frames and considering the HP, Hi10P and H422P profiles. Then, color relations of 4:2:0 and 4:2:2 are allowed and 8 or 10 bits per sample are supported. In this case, the target throughput is of 100 million of samples per second.

This section presents the synthesis results obtained considering a 2-D DCT input bit width of 8 bits. The synthesis results of the two 1-D DCT modules, transpose buffer module and the complete 2-D DCT targeting Altera and Xilinx FPGAs are presented in Tables 3 and 4, respectively.

From Table 3 and Table 4 it is possible to notice the differences between the use of hardware resources and the maximum operation frequency reached by the two 1-D DCT modules, since the second 1-D DCT module uses a higher bit width than the first 1-D DCT module. It is also possible to notice in both tables that the transpose buffer uses few logic elements and reaches a high operation frequency, since it is basically two Block RAMs and a little control.

From Table 3 it is very important to notice that the 8x8 2-D DCT uses 2,718 LUTs of the Altera Stratix II FPGA and it reaches a maximum operation frequency of161.66MHz. With these results this 2-D DCT is able to process 161.66 million of

Table 3. Synthesis results to Altera Stratix II FPGA

Blocks	**Total Logic Elements**			**Period**	**Throughput**
	LUTs	**Flip Flops**	**Mem. Bits**	**(ns)**	**(Msamples/s)**
First 1-D DCT Transform	1,072	877	-	5.03	198.77
Transpose Buffer	40	16	1,664	2.00	500
Second 1-D DCT Transform	1,065	1,332	-	5.18	193.09
2-D DCT Integer Transform	2,718	2,225	1,664	6.18	161.66

Selected Device: Stratix II EP2S15F484C3 samples per second. This rate is enough to process HDTV 1080 frames in real time (24fps) when the 4:2:0 or 4:2:2 color relations are considered.

Table 4 presents the results for Xilinx Virtex II Pro FPGA and this synthesis reported an use of 1,430 LUTs and a maximum operation frequency of 122.87MHz, allowing a processing rate of 122.87 million of samples per second as presented in Table 4. This processing rate is also enough to reach real time when processing HDTV 1080 frames.

Table 4. Synthesis results to Xilinx Virtex II - Pro FPGA

Blocks	Total Logic Elements			Period (ns)	Throughput (Msamples/s)
	LUTs	Flip Flops	Mem. Bits		
First 1-D DCT Transform	562	884	-	6.49	153.86
Transpose Buffer	44	17	2	2.31	432.11
Second 1-D DCT Transform	776	1,344	-	7.09	141.02
2-D DCT Integer Transform	1,430	2,250	2	8.13	122.87

Selected Device: Virtex II - Pro 2vp30ff896-7

Table 5 shows the synthesis results targeting TSMC 0.35µm standard-cells technology for all designed blocks. Besides, this table emphasizes the synthesis results of the 2-D DCT architecture including and not including the Block RAMs synthesis. From these results it is possible to notice that the number of used gates in the architecture with Block RAMs synthesis is almost the double of the architecture without Block RAMs. This difference is caused because the memories were mapped directly to register banks.

But nevertheless this architecture is able to process 124.1 million of samples per second, also reaching the throughput to process HDTV 1080 frames in real time.

The presented synthesis results indicate that the 2-D DCT architecture designed in this work reaches a processing rate of 24 HDTV 1080 frames per second considering all

Table 5. Synthesis results to TSMC 0.35µm standard-cells technology

Blocks	Total Logic Elements (Gates)	Period (ns)	Throughput (Msamples/s)
First 1-D DCT Transform	7,510	6.33	158.1
Transpose Buffer	15,196	4.65	215.2
Second 1-D DCT Transform	11,230	7.58	131.9
2-D DCT Transform (without RAM)	19,084	7.58	131.9
2-D DCT Transform (with RAM)	33,936	8.05	124.1

technology targets. This processing rate allows the use of this architecture in H.264/AVC encoders for HP, Hi10P and H422P profiles which target high resolution videos.

6 Related Works

There are a lot of papers that present dedicated hardware designs for 8x8 2-D DCT in the literature, but papers targeting the complete 8x8 2-D DCT defined in the H.264/AVC High profile were not found in the literature. There are some papers about the 4x4 2-D DCT of the H.264/AVC standard, but not about the 8x8 2-D DCT.

Only three papers were found about the High profile transforms, but not reporting the complete hardware design 8x8 2-D DCT defined in the standard. The first work [12] proposes a new encoding scheme to compute the classical 8x8 DCT coefficients using error-free algebraic integer quantization (AIQ). The algorithm was described in Verilog and synthesized for a Xilinx VirtexE FPGA. This work presented an operation frequency of 101.5 MHz and a consumption of 1,042 LUTs, and not presented throughput data.

The second work [13] proposes a hardware implementation of the H.264/AVC simplified 8x8 2-D DCT and quantization. However, this work implements just the 1-D DCT architecture and not the 8x8 2-D DCT architecture.

The comparison with the first paper [12] shows that the architecture designed in this paper presented a higher operation frequency and a little increase in the hardware resources consumption. A comparison in terms of throughput was not viable, once this data not presented in [12]. The comparison with the second paper is not possible, once it reports only an 8x8 1-D DCT and quantization design and this work presents an 8x8 2-D DCT.

Finally, the third work [14] proposes a fast algorithm for the 8x8 2-D forward and inverse DCT and it also proposes an architecture for this transforms. But this architecture was not implemented in hardware, therefore, it is not possible to realize comparisons with this work.

Other 8x8 2-D solutions presented in the literature were also compared with the architecture presented in this paper. These other solutions are not compliant with the H.264/AVC standard. Solutions [11], [15], [16], [17] and [18] presents hardware implementations of the 8x8 2-D DCT using some type of approximation to use only integer arithmetic instead of floating point arithmetic originally present in the 2-D DCT. A comparison of our design with others, in terms of the throughput and the used technology, is presented in Table 6. The differences between those implementations will not be explained, as they used completely different technologies, physical architectures and techniques to reduce area and power.

Throughputs in Table 6 show that our 8x8 2-D DCT implemented in Stratix II surpasses all other implementations. Our standard-cells based 8x8 2-D DCT is able to process 124 millions of samples per second and it presents the highest throughput among the presented standard-cells designs.

Our FPGA based results could be better had we used macro function adders, that are able to use the special fast carry chains that are present in the FPGAs.

In function of these comparisons, it is possible to conclude that the 8x8 2-D Forward DCT architecture designed in this paper has interesting profits in relation to other published works.

Table 6. Comparative results for 8x8 2-D DCT

Design	Technology	Throughput (Msamples/s)
Our Standard-cell version	**0.35µm**	**124**
Fu [15]	0.18µm	75
Agostini [11]	0.35µm	44
Katayama [16]	0.35µm	27
Hunter [17]	0.35µm	25
Chang [18]	0.6µm	23.6
Our Stratix II version	**Stratix II**	**162**
Agostini [11]	Stratix II	161
Our Virtex II version	**Virtex II**	**123**

7 Conclusions and Future Works

This work presented the design and validation of a high performance H.264/AVC 8x8 2-D DCT architecture. The implementations details, the synthesis results targeted to FPGA and standard-cells were also presented. This architecture was designed to reach high throughputs and to be easily integrated with the other H.264/AVC modules.

The modules which compose the 2-D DCT architecture were synchronized and a constant processing rate of one sample per clock cycle is achieved. The constant processing rate is independent of the data type and it is important to make easy the integration of this architecture with other modules.

The synthesis results showed a minimum period of 8.13ns considering FPGAs and a minimum period of 8.05ns considering standard-cells. These results indicate that the global architecture is able to process 122.87 million of samples per second when mapped to FPGAs and 124.1 million of samples per second when mapped to standard-cells, allowing their use in H.264/AVC encoders targeting HDTV 1080 @ 24 frames per second.

As future works it is planned an exploration in others design strategies for the 8x8 DCT of the H.264/AVC standard and a comparison among the obtained results. The first design strategy to be explored is to implement other 8x8 2-D DCT transform in a parallel fashion with a processing rate of 8 samples per clock cycle. Other future work is the integration of this module in the Forward Transform module of the H.264/AVC encoder.

References

1. Joint Video Team of ITU-T, and ISO/IEC JTC 1: Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 or ISO/IEC 14496-10 AVC). JVT Document, JVT-G050r1 (2003)
2. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Transactions on Circuits and Systems For Video Technology 13, 560–576 (2003)

3. Sullivan, G.J., Topiwala, P.N., Luthra, A.: The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions. In: SPIE Conference on Application of Digital Image Processing, Denver, CO, vol. XXVII (5558), pp. 454–474 (2004)
4. Gordon, S., Marpe, D., Wiegand, T.: Simplified Use of 8x8 Transforms. JVT Document, JVT-I022 (2004)
5. Gordon, S., Marpe, D., Wiegand, T.: Simplified Use of 8x8 Transforms - Updated Proposal & Results. JVT Document, JVT-K028 (2004)
6. Marpe, D., Wiegand, T., Gordon, S.: H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas. In: International Conference on Image Processing, ICIP 2005, Genova, Italy, vol. 1, pp. 593–596 (2005)
7. Richardson, I.E.G.: H.264 and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia. John Wiley & Sons, Chichester, UK (2003)
8. Malvar, H.S., Hallapuro, A., Karczewicz, M., Kerofsky, L.: Low-Complexity Transform and Quantization in H.264/AVC. IEEE Transactions on Circuits and Systems for Video Technology 13, 598–603 (2003)
9. Bhaskaran, V., Konstantinides, K.: Image and Video Compression Standards: Algorithms and Architectures, 2nd edn. Kluwer Academic Publishers, Norwell, MA (1997)
10. Agostini, L.V., Porto, R.E.C., Bampi, S., Rosa, L.Z.P., Güntzel, J.L., Silva, I.S.: High Throughput Architecture for H.264/AVC Forward Transforms Block. In: Great Lake Symposium on VLSI, GLSVLSI 2006, New York, NY, pp. 320–323 (2006)
11. Agostini, L.V., Silva, T.L., Silva, S.V., Silva, I.S., Bampi, S.: Soft and Hard IP Design of a Multiplierless and Fully Pipelined 2-D DCT. In: International Conference on Very Large Scale Integration, VLSI-SOC 2005, Perth, Western Australia, pp. 300–305 (2005)
12. Wahid, K., Dimitrov, V., Jullien, G.: New Encoding of 8x8 DCT to make H.264 Lossless. In: Wahid, K., Dimitrov, V., Jullien, G. (eds.) Asia Pacific Conference on Circuits and Systems, APCCAS 2006, Singapore, pp. 780–783 (2006)
13. Amer, I., Badawy, W., Jullien, G.: A High-Performance Hardware Implementation of the H.264 Simplified 8X8 Transformation and Quantization. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005, Philadelphia, PA, vol. 2, pp. 1137–1140 (2005)
14. Fan, C.-P.: Fast 2-D Dimensional 8x8 Integer Transform Algorithm Design for H.264/AVC Fidelity Range Extensions. IEICE Transactions on Informatics and Systems E89-D, 2006–3011 (2006)
15. Fu, M., Jullien, G.A., Dimitrov, V.S., Ahmadi, M.: A Low-Power DCT IP Core Based on 2D Algebraic Integer Encoding. In: International Symposium on Circuits and Systems, ISCAS 2004, Vancouver, CA, vol. 2, pp. 765–768 (2004)
16. Katayama, Y., Kitsuki, T., Ooi, Y.: A Block Processing Unit in a Single-Chip MPEG-2 Video Encoder LSI. In: Workshop on Signal Processing Systems, Shanghai, China, pp. 459–468 (1997)
17. Hunter, J., McCanny, J.: Discrete Cosine Transform Generator for VLSI Synthesis. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1998, Seattle, WA, vol. 5, pp. 2997–3000 (1998)
18. Chang, T.-S., Kung, C.-S., Jen, C.-W.: A Simple Processor Core Design for DCT/IDCT. IEEE Transactions on Circuits and Systems for Video Technology 10, 439–447 (2000)