

Power Saving of DHT in Digital Image Photography

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University,
Bhubaneswar, India

Prasant Kumar Patra
School of Electronics Engineering,
KIIT University,
Bhubaneswar, India

Sitakanta Maharatha
School of Electronics Engineering,
KIIT University,
Bhubaneswar, India

ABSTRACT

Digital camera is the main medium for digital photography. Digital image takes more memory space and more energy to store, as a digital camera operates on portable battery. Hence, it limits its performance and longevity. An efficient energy management is required to limit this drawback. This paper proposes a new technique by using Discrete Hartley Transform which will give the same result as that of the existing algorithms with low power consumption.

General Terms

Image Analysis and Image Processing.

Keywords

Image Compression, DCT, JPEG, DHT.

1. INTRODUCTION

Image compression is an important issue in digital image processing and finds extensive applications in many fields. This is the basic operation performed frequently by any digital photography technique to capture an image. For longer use of the portable photography device it should consume less power so that battery life will be more. To improve the Conventional techniques of image compressions using the DCT have already been reported. The JPEG is a lossy compression scheme, which employs the DCT as a tool and used mainly in digital cameras for compression of images. In the recent past the demand for low power image compression is growing. As a result various research workers are actively engaged to evolve efficient methods of image compression using latest digital signal processing techniques. The objective is to achieve a reasonable compression ratio as well as better quality of reproduction of image with low power consumption. Keeping these objectives in mind the research work in the present paper has been undertaken.

For solving for low power image compression we have modified the existing JPEG architecture which is the basic technique used for image compression and proposed some new low complexity method of compression technique. The performances of the proposed compression techniques have been evaluated and have been compared with that of the standard technique. Finally the proposed algorithms implemented in hardware and the power consumption is estimated. It is in general, observed that the proposed techniques are efficient than the conventional one.

2. JPEG COMPRESSION

All the JPEG (Joint Photographic Experts Group) standards has been around for some time and is the only standard for lossy still image compression. There are quite a lot of interesting techniques used in the JPEG standard and it is important to give an overview of how JPEG works. There are several variations of JPEG, but only the 'baseline' method is discussed here.

As shown in the figure 1, the image is first partitioned into non-overlapping 8×8 blocks. A Forward Discrete Cosine Transform (FDCT) is applied to each block to convert the spatial domain gray levels of pixels into coefficients in frequency domain. To improve the precision of the DCT, the image is 'zero shifted', before the DCT is applied. This converts a 0 - 255 image intensity range to a -128 - 127 range, which works more efficiently with the DCT. One of these transformed values is referred to as the DC coefficient and the other 63 as the AC coefficients.

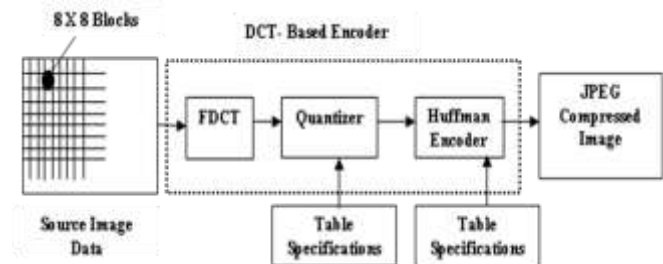


Figure 1: JPEG Encoder

After the computation of DCT coefficients, they are normalized with different scales according to a quantization table provided by the JPEG standard conducted by psycho visual evidence. The quantized coefficients are rearranged in a zigzag scan order for further compressed by an efficient lossless coding algorithm such as run length coding, arithmetic coding, Huffman coding. The decoding process is simply the inverse process of encoding.

3. DISCRETE COSINE TRANSFORM (DCT)

The DCT is a widely used transformation for data compression. It is an orthogonal transform, which has a fixed set of (image independent) basis functions, an efficient algorithm for computation, and good energy compaction and correlation reduction properties. Ahmed et al found that the Karhunen L  ve Transform (KLT) basis function of a first order Markov image closely resemble those of the DCT. They become identical as the correlation between the adjacent pixel approaches to one. The DCT belongs to the family of discrete trigonometric transform,

which has 16 members. The 1D DCT of a $1 \times N$ vector $x(n)$ is defined as

$$Y[k] = C[k] \sum_{n=0}^{N-1} x[n] \cos \left[\frac{(2n+1)k\pi}{2N} \right] \quad (1)$$

Where $k = 0, 1, 2 \dots N-1$ and

$$C[k] = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{1}{N}} & \text{for } k = 1, 2, \dots, N-1 \end{cases}$$

The original signal vector $x(n)$ can be reconstructed back from the DCT coefficients $Y[k]$ using the Inverse DCT (IDCT) operation and can be defined as

$$x[n] = \sum_{k=0}^{N-1} C[k] Y[k] \cos \left[\frac{(2n+1)k\pi}{2N} \right] \quad (2)$$

Where $n = 0, 1, 2 \dots N-1$

The DCT can be extended to the transformation of 2D signals or images. This can be achieved in two steps: by computing the 1D DCT of each of the individual rows of the two-dimensional image and then computing the 1D DCT of each column of the image. If represents a 2D image of size $x(n_1, n_2) N \times N$, then the 2D DCT of an image is given by:

$$Y[j, k] = C[j] C[k] \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x[m, n] \cos \left(\frac{(2m+1)j\pi}{2N} \right) \cos \left(\frac{(2n+1)k\pi}{2N} \right) \quad (3)$$

Where $j, k, m, n = 0, 1, 2 \dots N-1$

$$C[j] \text{ and } C[k] = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } j, k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } j, k = 1, 2, \dots, N-1 \end{cases}$$

Similarly 2-D IDCT can be defined as:

$$x[m, n] = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} C[j] C[k] Y[j, k] \cos \left(\frac{(2m+1)j\pi}{2N} \right) \cos \left(\frac{(2n+1)k\pi}{2N} \right) \quad (4)$$

The DCT is a real valued transform and is closely related to the DFT. In particular, a $N \times N$ DCT of $x(n_1, n_2)$ can be expressed in terms of DFT of its even-symmetric extension, which leads to a fast computational algorithm. Because of the even-symmetric extension process, no artificial discontinuities are introduced at the block boundaries. Additionally the computation of the DCT requires only real arithmetic. Because of the above properties the DCT is popular and widely used for data compression operation. $2N \times 2N$ The DCT presented in equations (1) and (4) is orthonormal and perfectly reconstructing provided the coefficients are represented to an infinite precision. The coefficients of the DCT are always quantized for high compression, but DCT is very resistant to quantization errors due to the statistics of the coefficients it produces. The coefficients of a DCT are usually linearly quantized by dividing by a predetermined quantization step. The DCT is applied to image

blocks $N \times N$ pixels in size (where N is usually multiple of 2) over the entire image. The size of the blocks used is an important factor since they determine the effectiveness of the transform over the whole image. If the blocks are too small then the images is not effectively de-correlated but if the blocks are too big then local features are no longer exploited. The tiling of any transform across the image leads to artifacts at the block boundaries. The DCT is associated with blocking artifact since the JPEG standard suffers heavily from this at higher compressions. However the DCT is protected against blocking artifact as effectively as possible, without interconnecting blocks, since the DCT basis functions all have a zero gradient at the edges of their blocks. This means that only the DC level significantly affects the blocking artifact and this can then be targeted.

Ringling is a major problem in DCT operation. When edges occur in an image DCT relies on the high frequency components to make the image shaper. However these high frequency components persist across the whole block and although they are effective at improving the edge quality they tend to 'ring' in the flat areas of the block. This ringing effect increases, when larger blocks are used, but larger blocks are better in compression terms, so a trade off is usually established.

4. DISCRETE HARTLEY TRANSFORM (DHT)

The In 1942, R. V. L. Hartley proposed a real integral transform for the analysis of transmission problem. Based on that integral transform, Bracewell proposed a real valued discrete transform called the DHT. The DHT is a real valued alternative to the DFT as the even and odd parts of the DHT of a real valued sequence are same as the real and negative imaginary parts of the corresponding DFT components. The DHT of an N -point real valued sequence $x(n)$ is

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \text{cas} \left[\frac{2\pi kn}{N} \right] \quad (5)$$

Where $j, k, m, n = 0, 1, 2 \dots N-1$

and the inverse transform is

$$X(n) = \sum_{k=0}^{N-1} X(k) \text{cas} \left[\frac{2\pi kn}{N} \right] \quad (6)$$

Where $\text{cas} \frac{2\pi kn}{N} = \cos \frac{2\pi kn}{N} + \sin \frac{2\pi kn}{N}$

Where $k, n = 0, 1, 2 \dots N-1$

Equations (5) and (6) may respectively be expressed in the matrix product form as

$$X = (1/N) C_N x \text{ and } x = C_N X \quad (7)$$

Where x is the $N \times 1$ input vector and X is the $N \times 1$ DHT vector is the $N \times N$ Hartley matrix whose elements are given by

$$C_N(k, n) = \cos \frac{2\pi kn}{N} + \sin \frac{2\pi kn}{N} \quad (8)$$

for $k, n = 0, 1, 2, \dots, N-1$ from (8) it can be seen that

$$C_N(k, n) = C_N(n, k) \quad (9)$$

and

$$\sum_{n=0}^{N-1} C_N(n, k) C_N(n, k') = N \delta_{kk'} \quad (10)$$

where Kronecker delta $\delta_{kk'}$ is given by

$$\delta_{kk'} = \begin{cases} 1 & \text{for } k = k' \\ 0 & \text{for } k \neq k' \end{cases}$$

From (9) and (10) it can be found that the Hartley matrix is a non-singular Hermitian matrix of Eigen value $\pm \sqrt{N}$. An important feature of this transform that makes it more advantageous compared to the DFT and DCT is that the inverse transform defined by (6) is identical to the forward transform given by (5), except a scale factor of N . Therefore, only one routine may be coded and stored for the forward as well as inverse transform.

Just like 1D, the 2D DHT computation can be developed and has a potential application in the field of image processing. Development of efficient schemes for its fast computation is, therefore, a subject of interest. Many researches have been carried out to solve this problem. Bracewell et al have proposed an efficient algorithm to compute the multidimensional DHT by adding certain number of intermediate arrays where each of the arrays is computed using 1-dimensional fast DHT algorithm. Another scheme of computation of the DHT has been proposed, where the computation is based on the prime-factor decomposition. According to this scheme, multidimensional DHT can be computed using a 1D fast DFT algorithm and a 1D fast DHT algorithm. This scheme has been reported to be less compulsive as well as structurally less complex over the earlier scheme. In the JPEG scheme we have incorporated the DHT that has been computed using the prime factor.

The 2D DHT of an array $[x(m, n)]$ of size $M \times N$ may be defined as

$$X(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(m, n) \left[\cos 2\pi \left(\frac{km}{M} + \frac{ln}{N} \right) \right] + \left[\sin 2\pi \left(\frac{km}{M} + \frac{ln}{N} \right) \right] \quad (11)$$

By splitting the arguments of sine and cosine function of (11), we get

$$X(k, l) = \sum_{n=0}^{N-1} u(k, n) \left[\cos \frac{2\pi ln}{N} + \sin \frac{2\pi ln}{N} \right] + \sum_{n=0}^{N-1} v(k, n) \left[\cos \frac{2\pi ln}{N} - \sin \frac{2\pi ln}{N} \right] \quad (12)$$

where

$$u(k, n) = \sum_{m=0}^{M-1} x(m, n) \cos \frac{2\pi km}{M}$$

(13)

$$v(k, n) = \sum_{m=0}^{M-1} x(m, n) \sin \frac{2\pi km}{M}$$

(14)

It may be noted that $u(k, n)$ and $v(k, n)$, for $k = 0, 1, \dots, M-1$, represent the real parts and the negative imaginary parts of M -point DFT of n th column of, respectively.

Substituting $n = N - n$ on the second sum of (12), we can get

$$X(k, l) = \sum_{n=0}^{N-1} w(k, n) \left[\cos \frac{2\pi ln}{N} + \sin \frac{2\pi ln}{N} \right] \quad (15)$$

$$\text{Where } w(k, n) = u(k, n) + v(k, (N - n)_N) \quad (16)$$

Equations (13) - (16) indicate that a 2D DHT of an array of size $M \times N$ may be computed in the following sequences:

1st, the M -point DFT of each column of $[x(m, n)]$ be computed.
2nd, the real parts of the DFT of the n th column be added with the -ve of corresponding imaginary parts of the DFT of the $(N-n)$ th column, for $n=1, 2, \dots, N-1$. The real part of the DFT of zeroth column, however, is added with the -ve imaginary part of the DFT of the same column. The results thus obtained, be stored in the corresponding position of $[x(m, n)]$ under its new variable name $[w(m, n)]$. 3rd, finally, the N -point DHT of each row of $[w(m, n)]$ be computed to obtain the desired 2D DHT.

5. VLSI OVERVIEW AND RECONFIGURABLE COMPUTING

The semiconductor industry has evolved the first integrated circuits (ICs) that matured rapidly. The era of large-scale integration (LSI) packed even larger logic functions, such as the first microprocessors, into a single chip. Then the evolution of very large scale integration (VLSI) has developed when millions of transistors can be integrated into a single chip. By using VLSI, the design of 64-bit microprocessors, complete with cache memory and floating point arithmetic units has become possible. Based on all the new technologies which have been grown from several years, the digital integrated circuit (IC) is one of the most phenomenal growths in terms of circuit complexity, switching speed and the power dissipation. Among the possible design methodologies of full-custom, mask programmable and field programmable logic devices, field programmable gate array (FPGAs) with different architecture and programming capacity are recently used overwhelmingly in different Application Specific Integrated Circuit (ASIC) development. Various types of sophisticated Computer Aided Design (CAD) tools are now available which really made the whole process feasible economically and timely.

5.1 Advantages of Using ASIC

The major advantages of using an ASIC are: a) Miniaturization: The usage of custom ICs will reduce the size of the end product. b) Lesser inventory: The reduced number of components per system reduces the inventory. This in turn reduces the overall cost. c) Reduced maintenance cost: Lesser components lead to fewer failures and lesser system down time. Maintenance will be easy. d) Lower power consumption: Lesser number of components in a system reduces the power consumption. e) Performance: More number of functions can be integrated to the ASIC, without increasing the size, cost or power consumption of the product.

5.2 Major Risks of Using ASIC

The major risks of using an ASIC are as follows: a) Higher Cost: ASIC will be always expensive than the standard components. b) Time to market: The product lead time will be more for an ASIC based system. The market research team should define the requirement of the end product well in advance. c) First time success: The ASIC design should be properly simulated and thoroughly tested to insure the first time success. Any failure will affect the time to market and resulting huge loss in revenue.

6. VLSI DESIGN METHODOLOGIES

The basic steps involved in designing an ASIC can well be understood as an iterative process of development and testing. ASIC can broadly be classified into following categories.

6.1 Full Custom ASIC

Possibly all logic cells and mask layers are customized in a full custom ASIC. So, full custom ICs are more costly. Manufacturing time (not including design time) requires 8 to 10 weeks. Designers go for full custom if there is no suitable existing cell library which is fast enough or the logic cells are small enough or consume too much power. This ASIC technology is chosen, based on many reasons like, a. Power consumption, b. Functional requirement, c. Speed of the ASIC, d. Non availability of suitable library, e. Product life, etc. Microprocessor and Microcontroller are examples of full custom IC.

6.2 Semi-Custom ASIC

In the semi custom ASIC, the designer will be using the pre-characterized and sometimes prefabricated logic cells. This approach reduces the design time and increases the turn around time of the ASIC. Again the semi-custom ASIC can be subcategorized into 1. Standard Cell based and 2. Gate Array base.

6.3 Programmable ASIC

The programmable ASIC is the latest invention of the IC family. A programmable ASIC can be reprogrammed according to a change in the specification. It reduces the development time and cost. The programmable ASIC can be sub grouped into two types according to their architecture and function: 1. Programmable Logic Device (PLD) and 2. Field Programmable Gate Array (FPGA).

7. FPGA DEVICES

The FPGA is a high capacity programmable logic device. It consists of an array of programmable basic logic cells surrounded by programmable interconnect. It can be configured by end user (field programmable) to have specific circuitry with in it. Any combinational or sequential circuit can be designed using FPGA. The programmable logic array was introduced in the late 1970s and was followed by the introduction of the first FPGA, XC2000 series, by Xilinx in 1985. The advantage of FPGAs is that they combine the performance that can be achieved by ASICs with the flexibility of programmable microprocessors. With these merits, FPGAs destroyed the balance of gate array market and have taken a significant proportion of the standard cell market. Conceptually a programmable FPGA has three key elements: i. *Programmable logic cells*: It provides the functional elements for constructing the user's logic. ii. *Programmable Input/output (I/O) blocks*: It provide the interface between the package pins and the logic cell. iii. *Programmable interconnects*: It provides routing paths to connect the inputs and outputs of the logic cell and I/O blocks.

For any programmable device the control information is stored in the memory which can be programmed. According to the select input from the programmable memory the accurate data are selected by the MUX and thus the switches are connected accordingly.

7.1 Selection of FPGA device

The performance and cost of the final FPGA based product, depends on the target FPGA. Therefore before making a prototype, it is very much essential to choose the target FPGA device. Large variations of FPGA devices are available, by different vendors. In this present research work the target FPGA *Virtex*, *XCV1000* is selected and is a product from the vendor Xilinx. The resources offered by this family of FPGA are summarized in Table 1 and Table 2. The Virtex FPGA family delivers high-performance, high-capacity programmable logic solutions. Dramatic increases in silicon efficiency result from optimizing the new architecture for place and- route efficiency and exploiting an aggressive 5-layer-metal 0.22 μm CMOS process. These advances make Virtex FPGAs powerful and flexible alternatives to mask-programmed gate arrays.

Table 1 –: FPGA Resources

Device	System Gates	CLB Array	Logic Cells	Available I/O	Block RAM
XCV1000	1,124,022	64 X 96	27,648	512	131,072

Table 2 –: FPGA Resources

Device	Slices	Flip Flops	4 Input LUTs	Available I/O	Block RAM
XCV1000	12,288	24,576	24,576	512	131,072

Virtex devices feature a flexible, regular architecture shown in figure 2, comprises an array of configurable logic blocks (CLBs)

surrounded by programmable input/output blocks (IOBs), all interconnected by a rich hierarchy of fast, versatile routing resources. The abundance of routing resources permits the Virtex family to accommodate even the largest and most complex designs. Virtex FPGAs are SRAM-based, and are customized by loading configuration data into internal memory cells. In some modes, the FPGA reads its own configuration data from an external PROM. Otherwise, the configuration data is written into the FPGA (Select- MAP™, slave serial, and JTAG modes). The standard Xilinx Foundation™ and Alliance Series™ Development systems deliver complete design support for Virtex, covering every aspect from behavioral and schematic entry, through simulation, automatic design translation and implementation, to the creation, downloading, and read-back of a configuration bit stream. Virtex devices provide better performance than previous generations of FGA. Designs can achieve synchronous system clock rates up to 200 MHz including I/O. Virtex inputs and outputs comply fully with PCI specifications, and interfaces can be implemented that operate at 33 MHz or 66 MHz.

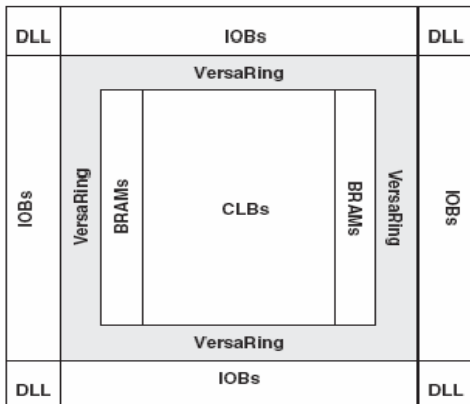


Figure 2: Virtex Architecture

7.2 Simulation Results

All the compression techniques are synthesized by using the AccelChip DSP Synthesis tools. FPGA *Virtex*, *XCV1000* device is used for synthesize all the algorithms to get an approximate hardware requirements for their calculation. The comparative study is performed in terms of number of slices, number of Flip Flops, number of LUTs used, Table 3 presents lists for calculation of DCT, IDCT, JPEG quantization, energy quantization (EQ), modified energy quantization (MEQ), rule based energy quantization (RBEQ), DHT and IDHT, Arithmetic compression technique respectively. Table 4 presents a comparative study for different image compression techniques in terms of size and power consumption in mW calculated using the Xilinx websites power calculation tool. For calculation of the power consumption we have assumed that all the devices are operated at 10MHz frequency.

Table 3: Synthesize reports of different techniques

	Number of Slices Used	Number of Flip Flops used	Number of 4 Input LUTs used
DCT Calculation	4836	6561	8538
IDCT Calculation	6295	8346	11188
JPEG Quantisation	15338	22827	27529
JPEG Dequantisation	14146	11250	25811

Table 4: Comparison between different image compression technique in terms of size and power consumption

Compression Techniques	Number of Slices Used	Number of Flip Flops used	Number of 4 Input LUTs used	Power consumed in mW
DCT + IDCT + JPEG Quantisation	40615	48984	73066	2011
DCT + IDCT + EQ	29058	40723	52825	1637

8. SUMMARY AND CONCLUSION

DHT based JPEG with rule based energy quantization is an ideal solution for lossy image compression technique as the hardware requirement is less and power consumption is very low compare to other techniques. If we want to go for good quality, low compression and high speed then the arithmetic compression with snake scanning is the best choice, which also consumes very less power compare to the rest compression techniques.

This paper primarily focuses on image compression with less computation and low power requirement. The investigation made in this paper pertains to the development of some alternative compression- reconstruction schemes, which offer superior performance compared to the conventional works. The proposed quantization techniques for image compression has aimed to enhance the CR required for some multimedia applications while reducing the computational complexity for coding and decoding. The different image compression techniques developed in the paper can suitably be applied for 3D video signals.

9. REFERENCES

- [1] David Salomon. 1998. *Data Compression, The Complete Reference*, 2nd Edition Springer-Verlag.
- [2] E. Fossum. Dec. 1995. "CMOS Image Sensors: Electronic Camera on a Chip", IEEE Int. Electron. Devices Meeting Tech. Digest, pp. 1-9.
- [3] Eyadat, M. and Muhi, I. April 2005. "Compression standards roles in image processing: case study", International Conference on Information Technology: Coding and Computing. Volume 2, Page(s):135 – 140.
- [4] Frank Vahid and Tony Givargis. 2002. "Embedded System Design: A Unified Hardware/Software Introduction", John Wiley & Sons.
- [5] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen. April 1992. "Low-power CMOS digital design", IEEE Journal of Solid-State Circuits, vol. 27, no. 4, pp. 473– 483.

- [6] G. K. Wallace. Feb 1992. “*The JPEG Still Picture Compression Standard*”, IEEE Trans. On Consumer Electronics, vol.38, No.1, pp. xviii – xxxiv.
- [7] N. Ahmed, T. Natrajan, and K. R. Rao. July 1989. “*Discrete Cosine Transform*”, IEEE Transactions on Computers, vol. 23, pp. 90-93.
- [8] S. Martucea. May 1994. “*Symmetric convolution and the discrete sine and cosine transform*”, IEEE Transaction on Signal Processing, vol. 42, pp. 1038-1051.
- [9] O, Buneman, et al. Feb 1987. “*Multidimensional Hartley transform*”, Proc. Of IEEE, vol.75, no. 2, pp. 267.
- [10] P. K. Meher, J. K. Satapathy, and G. Panda. June 1992. “*Fast computation of multidimensional discrete Hartley Transform*”, Electronics letter, vol. 28, no.12, pp. 1077-1078.
- [11] R. N. Bracewell, et al. Dec 1983. “*Discrete Hartley Transform*”, J. Opt. Soc. Amer., vol. 73, no. 12, pp. 1832-1835.
- [12] R. N. Bracewell, O. Buneman, et al. Sept 1986. “*Fast two-dimensional Hartley transform*”, Proc. IEEE, vol. 74, no. 9, pp. 1282-1283.
- [13] R. V. L. Hartley, et al. Mar 1942. “*A more symmetrical Fourier analysis applied to transmission problem*”, Proc. IRE, vol. 30, pp. 144-150.
- [14] W. B. Pennebaker and J. L. Mitchell. 1993. *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York.
- [15] M.J.S. Smith. 2003. *Application Specific Integrated Circuits*, Pearson Education.
- [16] The Xilinx Home page: www.xilinx.com.