Computational Technique of 3D Reconstruction in Integral Imaging using FPGA Hardware

A.Hussein
Electronics and Communications Engineering
Mansoura University, Egypt

S.Kishk
Electronics and Communications Engineering
Mansoura University, Egypt

M. Abo-Elsoud
Electronics and Communications Engineering
Mansoura University, Egypt

ABSTRACT
This paper presents hardware implementation for a computational 3D integral imaging reconstruction technique. The reconstruction technique is based on a single pixel extraction from corresponding elemental images of a 3D scene, which is formed using an integral imaging system. The proposed hardware exploits both of the parallel processing and memory organization features, that is available in field-programmable gate array (FPGA) device to achieve several computational operations at real time. In this method the need for optical equipment such as micro-optics lens and special high-quality LCD to display 3D scene, which will make it suitable to field view TV application, is eliminated. Some experiments were carried out to show the feasibility of the proposed scheme, its result exposes that the hardware is able to process digital reconstruction at real time rate with low power consumption.

Keywords
Integral Image, Three-dimensional, CIIR, OIIR, Image reconstruction, FPGA, Power consumption

1. INTRODUCTION
Integral imaging (InIm) is a three dimensional (3D) imaging technique that is based on the principle of integral photograph, [IP] ([1]-[3]). IP uses a micro lens array (MLA) over a high resolution Charge Couple Detector (CCD) to pick up and record a group of 2D elemental images of 3D object. These recorded images have their own perspective and de-magnified images of a 3D object ([4], [5]). The operational principle of Integral imaging system is shown in Fig.1. Mainly, an integral imaging system consists of two processes: pickup and display. In the pickup process, intensity and directional information of rays coming from a 3D object through a micro lens array (MLA) is recorded by use of charge-coupled device (CCD) camera as shown in Fig.1 (a). Each of these small images recorded on the CCD is called an elemental image. On other hand, in the display process, the recorded EIs are displayed on a display panel inversely through each MLA to provide a 3D representation of the original object as shown in Fig.1 (b). Basically, there are two kinds of integral imaging reconstruction techniques, optical integral image reconstruction (OIIR) technique and computational integral imaging reconstruction (CIIR) technique ([6]-[9]). The OIIR technique has problems such as degraded image quality of the displayed 3D images caused by physical limitation of optical devices such as diffraction and interference between elemental images ([8], [10]). In order to overcome these drawbacks of OIIR technique, a CIIR technique based on a pinhole array model has been introduced, in which 3D object can be computationally reconstructed through digital simulation of geometrical optics [8]. Several attempts for 3D object reconstruction and visualization using InIms have been reported in the literature ([10]-[13]). Relying on hardware implementation of digital architecture, several attempts are carried out for 3D scene reconstruction using InIms and accelerating the reconstruction process ([5], [13]). In this paper, a simple computer-based 3D image reconstruction using a powerful hardware based on FPGA device is proposed. In this method the 3D scene is reconstructed by pixels extracting periodically from recorded elemental image array. By reconstruction of the 3D scene numerically using an FPGA device, the quality of the image can be improved, and a wide variety of digital Image processing techniques can be applied. The hardware implementation results reveals the speed of the reconstruction process improvement, resources utilization by applying specific memory data arrangements, the capability of changing the view angle of reconstructed 3D scene, and the low power consumption of the used FPGA device. The paper is organized as follows, Section 1 is an introduction. In section 2 the reconstruction process is shown. Section 3 is FPGA implementation. Section 4 discusses the practical experiment and results. Section 5 summarized the FPGA resources utilization, and finally section 6 is a conclusion.

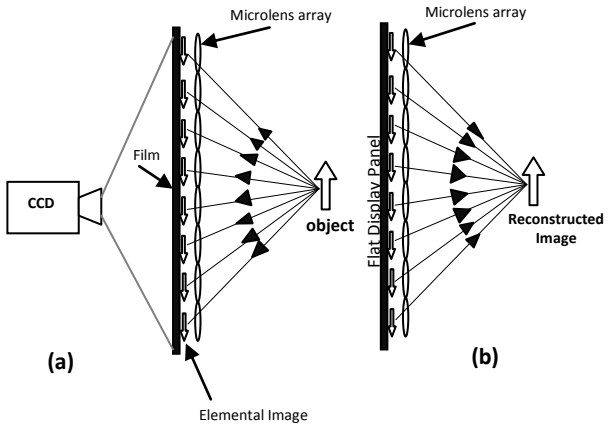
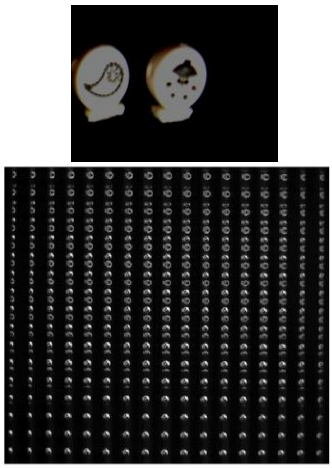


Fig.1:Conceptual diagram of pickup and display process in the integral imaging system: (a) pick up process and (b) display process.

2. 3D RECONSTRUCTION FROM INTEGRAL IMAGING METHODS
Recently, there has been a great interest in computational integral imaging reconstruction (CIIR) [6], [7], [14], [15]. These methods can be classified into plane-based CIIR technique and view-based CIIR technique [6], [14], [16]. Plane-based CIIR generates 3D image by changing the reconstructed image plane, in this case the reconstructed images are different according to the position of the plane. On the other hand, view-based CIIR generates 3D image by extracting a single pixel from each elemental image based on

a pinhole or micro-lens array, in this case the reconstructed images are different according to observation points. In this paper, the second method, view-based CIIR that depends on single pixel extraction is used to reconstruct 3D image from InIm [7]. This method is appropriate for applications that include optical measurement and remote sensing. Also the image processing may be used to compensate for some defects that exist in optical sensor that cause degradation in the reconstructed image. The 3D reconstruction algorithm can be summarized as follows; store the recorded EIs using the FPGA as it will be illustrated later, then pixel grid extraction is performed according to the required viewing angle, and finally the reconstructed image is displayed. The used input scene contains two circular objects having the same size. The object on the right is located 75mm from the microlens array and the second object is located 79mm from the microlens array plane as shown in Fig.2 (a). Each microlens has a diameter of 1mm and a focal length of 5.2mm. Elemental image arrays of the object are shown in Fig. 2 (b), it shows a part of the elemental images captured by the CCD camera. The CCD has a pixel size of 12µm x 12µm, each image element is recorded with ~23 square pixels. At this stage of the research and to reduce memory usage the observed elemental image array is stored in FPGA device as 2 bits/pixel.



(a) **(b)**

Fig.2. (a) Image of the 3D input scene. (b) Part of the recorded elemental image arrays.

In order to obtain the 3D scene from InIm simple extraction of points is performed periodically from the recorded elemental image array. As mentioned above, each elemental image contains 23×23 pixels. Therefore the pixels are extracted every 23 pixel in both the vertical and the horizontal lines of the elemental image array. The resolution of the reconstructed image is determined by the resolution of the CCD camera and the number of EIs. The number of pixels that compose a reconstructed image is the same as the number of the used EIs. Therefore the reconstructed image is 30 ×18 pixels. These extracted pixels reconstruct an image viewed from a particular angle. For reconstruction of images viewed from other angles, the grid positions of the pixels to be extracted are shifted gradually. During the experiment the grid can be shifted in both directions, horizontally and vertically to obtain different view angles for the reconstructed 3D scene.

3. FPGA IMPLEMENTATION

The view-based CIIR that depends on single pixel extraction (SPE) from recorded EIs to generate the 3D scene constitutes the core of the algorithm as detailed in previous section. Moreover to obtain real- time reconstruction and improve the performance, this algorithm favors hardware implementation. The immediate access to arbitrary EIs is achieved by using look-up table or by using RAM with optimized memory organization.

3.1 HARDWARE IMPLEMENTATION CONSIDERATION

In the proposed implementation, a SPE unit is designed as the core processing element in order to target a single FPGA device. The SPE unit can perform pixel extraction every clock cycle using a row and a column that is components for each EI. The design of FPGA memory takes into account the need for storage area required to store the recorded EIs array. One of the FPGA devices in the market is the Virtex-II Pro family, which used in this paper to carry out the reconstruction algorithm. Virtex-II Pro uses 0.13 µm CMOS nine-layer copper process at 1.5 V power supply [17]. In addition to advancements in its process technology, Virtex-II Pro is the first Xilinx FPGA with fully buffered interconnect, which may be considered as a turning point in its routing architecture. Fig.3 shows Virtex-II XC2VP30 architecture, as shown in this figure the Virtex-II pro includes a number of hard cores, including memory blocks, IO blocks, digital clock managers, encryption circuitry, and custom multipliers. However, most of the silicon area in the largest members of the family is consumed by what is referred to as programmable fabric.

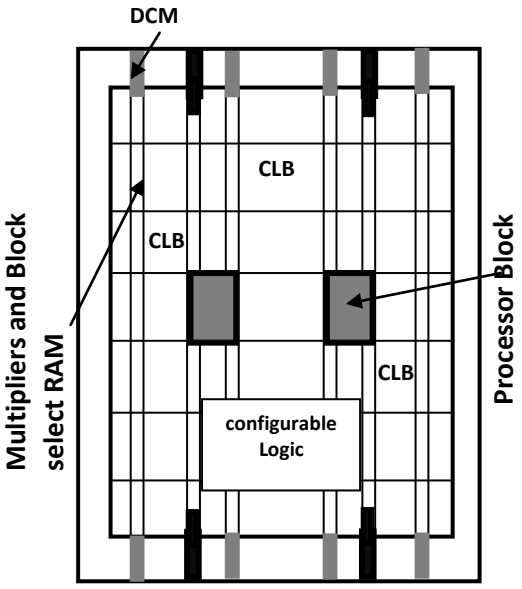


Fig.3. Virtex-II Pro platform FPGA.

3.2 IMPLEMENTATION DETAILS

Virtex-II Pro devices incorporate large amounts of 18Kb blocks RAM. These block RAMs is considered an important element in the proposed algorithm. Virtex-II Pro block RAM supports various configurations, including single- and dual-port RAM and various data/address aspect ratios, as shown in table1. Regardless of configuration type each port has the following types of inputs: Clock and Clock Enable, Write Enable, Set/Reset, and Address, as well as separate Data/parity data inputs (for write) and Data/parity data outputs (for read) [17]. Virtex-II Pro contains 136 block RAM, so the total internal memory will be about 2.2Mb. Virtex-II Pro

devices incorporate many embedded multiplier blocks. Virtex-II Pro multiplier block is an 18-bit by 18-bit 2's complement signed multiplier. These multipliers can be associated with an 18 Kb block RAM or can be used independently. Multiplier organization is identical to the 18 Kb block RAM organizations, because each multiplier is associated with a 18 Kb block RAM [17]. The Virtex-II Pro contains 136 hardware multipliers.

Table 1: Dual- and Single-Port Configurations

16K × 1 bit	2K × 9 bit
8K × 2 bit	1K × 18 bit
4K × 4 bit	512 × 36 bit

As detailed in the above section, each EI composed from 23x23 pixels, each pixel represented by 2 bits. In order to store the used part of EIs array (1350 × 810 pixels) in FPGA, the required memory will be 2.1Mb. The Block RAM is configured as single port ROM with 2 bits for data width and about 1M for depth. The memory arrangement for the stored EIs is depicted in Fig.4. SPE unit depends on combination of parameters such as Block RAM arrangement, hardware multiplier, and operating speed. In the current framework, the design of SPE unit exploits the parallel operation to immediate access the stored data in the block RAM, the outline of SPE unit is depicted in Fig.5. In the proposed implementation, the elemental image array are pre-determined and stored in Block RAM in single device FPGA based on Intellectual Property (IP) core generator [17]. In order to perform SPE operation, the index of each EI is determined; this is done by index generation as shown in Fig.5. The EI index depends on EI size; it considered an important parameter in reconstruction process. When it changes, the view angle of reconstructed image is changed. According to memory organization the EIs array represented as 1350×810 pixels, it can be converted to single vector contains 1093500 pixels, each 810 pixels represent a single row of the whole EIs array. The EI index can be determined according to following equation:

$$\left(\frac{(\text{single EI row size}+1)}{2}+N\right)\times(\text{EIs row size}+1)-(M\times \text{EIs row size}) \quad (1)$$

where N and M are integer values used to determine the viewing angle for the reconstructed image.

N =0, 1, 2, …,single EI row size and M =1, 2, …,single EI row size.So, if the EI row size equals 23 pixels, $\frac{(\text{single EI row size}+1)}{2}$ equal 12 pixels so, each 8922 pixels the SPE unit extracts and buffers a pixel in order to display it again. To continue reconstruction process, a pointer is used to access the memory.

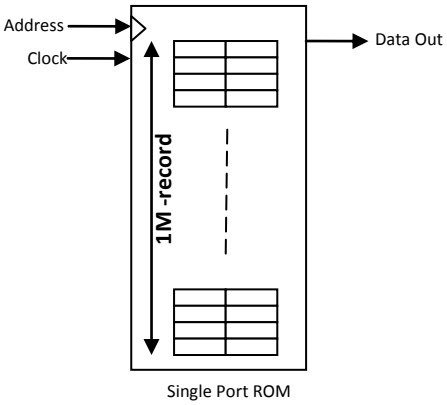


Fig.4. Memory arrangement.

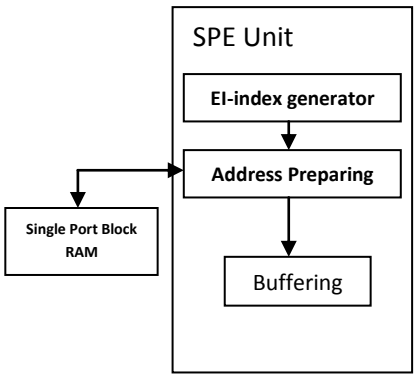


Fig.5.The implemented SPE unit outline.

4. PRACTICAL EXPERIMENT AND RESULT

The practical experiment considers the appropriate method to evaluate the architecture performance. The system was implemented using the Xilinx ISE tools on XUP Virtex-II Pro development system using Xilinx Virtex-II Pro XC2VP30 FPGA device. The algorithm is divided into three steps: pre-initialization Block RAM, pixel extraction and displaying step. In the step, the IP core generator software creates parameterizable versions of pre-defined "soft" IP optimized for Xilinx FPGAs. The IP core Generator includes memories and FIFOs as well as digital signal processing (DSP), math, standard bus interface, standard logic, and networking functions. In order to initialize Block RAM, it should pass an input coefficient file (COE file) to IP core generator. This is achieved by converting the stored elemental images array to binary form by using image processing tools such as Matlab tool. The COE file contains an array of data represent the EIs array. In this experiment the EIs is composed of 1350×810 pixels, so the COE file includes about 1093500 pixels, where each pixel represented by 2 bit as explained in subsection 3.(b). In the second step, the immediate memory access for recorded EIs depends on the resulted values from equation (1). When N or M are changed the viewing angle is changed. In the final step, the 3D scene is composed and refreshed on LCD after each SPE operation is performed. This manner of displaying is preferred than displaying a whole 3D image because it eliminates the need to use many buffers for the extracted pixels. In the displaying process the FPGA device must be connected with VGA interface. The VGA port has five active signals, including the horizontal and vertical synchronization signals, hsync and vsync, and three video signals corresponding to the red, green, and blue beams [20].

A video signal is an analog signal and the video controller uses a digital-to-analog converter to convert the digital output to the desired analog level. If a video signal is represented by a k-bit word, it can be converted to 2k analog levels. The pixel or refresh rate required for LCD is determined by three parameters: the number of pixels in a horizontal scan line and the number of lines in a screen and the number of screens per second. In this experiment 640-by-480 resolution is used therefore a pixel rate of 25-MHz is used. Since the development board uses the 100-MHz oscillator, the system clock rate is four times from the pixel rate. To generate 25-MHz from the system clock a mod-4 counter is used. The proposed architecture which is depicted in Fig.6 (a) implements a SPE unit and displaying 3D image. The SPE unit read data from Block RAM every 8922 word when N=0 and M=1 [using Eq.(1)] at clock rate equals 25-MHz. In this case each extracted pixel considers the center of each EI. This algorithm has the ability to reconstruct 3D image 30×18 pixels from recorded EIs 1350×810 pixels after 540 clocks (at frequency 25-MHz). The Image of the 3D input scene and the reconstructed image are shown in Fig.6(b) and 6(c).

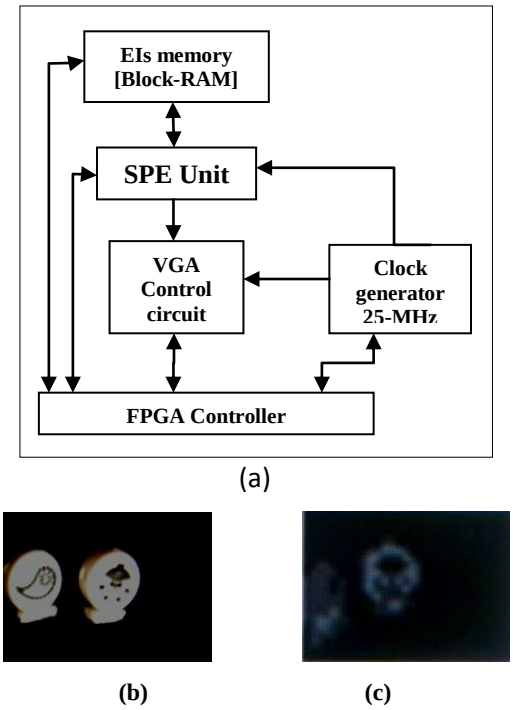


Fig.6. (a) The proposed FPGA architecture.(b) 2D Image of the 3D input scene. (c) reconstructed image

The hardware system implementation results on Virtex-II Pro XC2VP30 FPGA development board expose that the system can successfully process InIm data at frequency of 25-MHz according to the required resolution and the system clock. An efficient design for SPE unit gives the potential to do the reconstruction procedure over single recorded EIs or two recorded EIs, depending on the Block RAM configuration. The use of the embedded block RAM resources instead of external memory to store data gives more features, such as it can operate at the PFGA device frequency or less depending on the design and the short paths between FPGA core and Block RAM led to minimize power consumption. There are several techniques have been developed to estimate the power consumption for digital designs based on FPGA device, power estimation method can be based on statistics or probabilities propagation. General surveys about power estimation are presented in [21]-[23].There are two types of power consumption in FPGA devices; static and dynamic. The leakage current between power supply and ground is the only source of static power dissipation. There are two major sources of leakage current; the reverse biased PN-junction current and the sub threshold channel conduction. The static power of modern FPGAs such as the Virtex-II family (SRAM-based FPGA, 0.13µm technology), depends on the temperature, and the design. The dynamic power dissipation is caused by signal transitions at the device transistors. Frequencies of signal transitions are obviously related to the clock frequency. The most significant source of dynamic power consumption in CMOS circuits is the charging and discharging of capacitance. This can be modeled as:

$$\boldsymbol{P = \sum_{i} Ci.Vi^{2}.fi} \qquad (2)$$

Where C_i , V_i, and F_i , represent capacitance, the voltage swing, and clock frequency of the resource i, respectively [21], [23].The total dynamic power consumed by a device is the summation of the dynamic power of each resource. According to equation (2) the dynamic power consumption is supposed to increase linearly with changes of clock frequency and size of a design. Power efficiency of each particular design is investigated by XPower (one of the accessories of Xilinx Integrated Software Environment (ISE)) that provides estimations of power consumption [24]. The routed design, which is available in an internal format (called NCD), can be used for power estimation. Table 2 shows the power consumption results for the proposed hardware at different frequencies.

Table 2: dynamic power consumption for several frequencies at Vccint=1.5V.

Frequency MHz	Dynamic power mW	Total power mW
100	71	178
50	45	152

The total power is estimated at Vccint 1.5V (Internal supply voltage), it includes the static power which equal 107mW at 25o ambient temperature. When Vccint is changed to 1.2v the power calculations change as depicted in Table 3 and the static power become 92mW.

Table 3: dynamic power consumption for several frequencies at Vccint=1.2V.

Frequency MHz	Dynamic power mW	Total power mW
100	50	142
50	33	126

5. RESOURCES UTILIZATION

Resource utilization strongly depends on the system design. While the routed NCD file contains all the routing information including the utilized resources, it cannot be read directly. Therefore, the Xilinx Design Language (XDL) utility in Xilinx tools is used to convert the binary NCD file to text format. The FPGA resource utilities of this design are shown in table 4. The FPGA chip which is used in the design contains 30,816 logic cells, 13,696 slices, 136 block RAM, and 136 block multipliers (18 X 18 Bit).

Table 4: FPGA resource utilities.

FPGA Resources	Used	Utilization
Slices(13,696)	347	2%
Block RAM (136)	134	98%
MULT18x18 bits(136)	2	1%
Number of LUT (27,392)	646	2%
Number IOBs (556)	34	6%

6. CONCLUSION

FPGA device is used to reconstruct a 3 D image by numerical processing of an optically observed image array formed by microlens array. The single pixel extraction from each recorded elemental images is the core of the reconstruction operation. Both of the parallel processing and memory organization features give the hardware the ability to achieve several computational operations at real time. The proposed method stores EIs on block RAM which reduces the required time for accessing the memory and increase the system efficiency. The practical experiment result reveals that the implemented hardware system can successfully process 3D data in a rate that surpasses a contemporary InIm system's throughput capability. The power consumption for the used FPGA is estimated. It based on resource usage, I/O types, clock requirements, clock frequencies, and environmental conditions. Future work involves development in the reconstruction algorithm using FPGA in order to improve the viewing angle, resolution and determine the depth of reconstructed 3D image therefore the plane-based CIIR technique instead of plane-based CIIR technique could be used. In this paper, there are constraints on the storage area required to store EIs by using Block RAMs. So, it is better to use external memory to be suitable for large EIs and InIm video stream.

7. REFERENCES

[1] G. Lippmann," La Photographie integrale," C.R. Acad. Sci. 146 (1908) 446–455.

[2] H.-E. Ives, Optical properties of a Lippman lenticulated sheet, J. Opt. Soc. Am. 21(1931) 171-176.

[3] G. Saavedra, R. Martínez-Cuenca, M. Martínez-Corral, H. Navarro, M. Daneshpanah, B. Javidi, Digital slicing of 3D scenes by Fourier filtering of integral images, Opt. Express 16 (2008) 17154–17160.

[4] B.-G. Lee, Liliana, D.-H. Shin, Enhanced computational integral imaging system for partially occluded 3D objects using occlusion removal technique and recursive PCA reconstruction, Opt.Commun. 283(2010) 2084-2091.

[5] D. Chaikalis, G. Passalis, N. Sgouros, D. Maroulis, T. Theoharis, Near Real-Time 3D Reconstruction from InIm Video Stream, Springer-Verlag Berlin Heidelberg. 5112(2008) 336–347.

[6] S.-H. Hong, J.-S. Jang, B. Javidi, Three-dimensional volumetric object reconstruction using computational integral imaging, Opt. Express. 12 (2004) 483-491.

[7] H. Arimoto, Integral three-dimensional imaging with digital reconstruction, Opt. Lett. 26 (2001) 157-159.

[8] J.-S. Park, D.-C. Hwang, D.-H. Shin, E.-S. Kim, Resolution-enhanced 3D image correlator using computationally reconstructed integral images, Opt.Commun. 276 (2007) 72–79.

[9] D.-C. Hwang, K.-J. Lee, S.-C. Kim, E.-S. Kim, Extraction of location coordinates of 3-D objects from computationally reconstructed integral images basing on a blur metric, Opt. Express. 16 (2008) 3623- 3635.

[10] S. Kishk, B. Javidi, Improved Resolution 3D Object Sensing and Recognition using Time Multiplexed Computational Integral Imaging, Opt. Express. 11(2003) 3528–3541.

[11] Y. Frauel, B. Javidi, Digital three-dimensional image correlation by use of computer-reconstructed integral imaging, Appl. Opt. 41 (2002) 5488–5496.

[12] Y. Frauel, B. Javidi, Digital three-dimensional image correlation by use of computer-reconstructed integral imaging, Appl. Opt. 41 (2002) 5488–5496.

[13] D. Chaikalis , N.-P. Sgouros, D. Maroulis, A real-time FPGA architecture for 3D reconstruction from integral images, J. Vis. Commun. Image R. 21 (2010) 9–16.

[14] D.-H. Shin, M. Cho, K.-C. Park, E.-S. Kim, Computational technique of volumetric object reconstruction in integral imaging by use of real and virtual image fields, ETRI J. 27(2005) 208-712.

[15] T. Naemura, T. Yoshida, H. Harashima, 3-D computer graphics based on integral photography, Opt. Express. 8 (2001) 255-262.

[16] D.-H. Shin, B. Lee, E.-S. Kim, Improved Viewing Quality of 3-D Images in Computational Integral Imaging Reconstruction Based on Lenslet Array Model, ETRI J. 18 (2006) 521-524.

[17] Xilinx Virtex-II Pro Datasheet. Available from: <http://www.xilinx.com>.

[18] Pong P. Chu, FPGA Prototyping by VHDL Examples, John Wiley & Sons (2008).

[19] P. Pawel. Czapski, A. Sluzek, Power Optimization Techniques in FPGA Devices: A Combination of System- And Low Levels, World Academy of Science, Engineering and Technology. 28 (2007) 313-319.

[20] E. Todorovich, E. Boemo, F. Angarita, J. Vails, Statistical Power Estimation for FPGA'S, IEEE. 2 (2005) 515-518.

[21] L. Shang, A. S. Kaviani, K Bathala, Dynamic Power Consumption in Virtex™-II FPGA Family, ACM/SIGDA tenth international symposium on Field-programmable gate arrays. (2002) 157-164.

[22] Xilinx Inc., Chapter 11: XPower. In Development System Reference Guide, available from: <http://www.xilinx.com>.

[23] L. Shang, A. S. Kaviani, and K Bathala, "Dynamic Power Consumption in Virtex™-II FPGA Family," ACM/SIGDA tenth international symposium on Field-programmable gate arrays. (2002) 157-164.

[24] Xilinx Inc., "Chapter 11: XPower". In Development System Reference Guide, available at http://www.xilinx.com.