A project driven digital design course using FPGAs

António J. Araújo, José C. Alves
Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, 4200 – 465 Porto, Portugal
Email: aja@fe.up.pt; jca@fe.up.pt

***Abstract*— This paper presents a project based teaching experience in an advanced digital systems design course with emphasis on design methodologies and laboratory assignments. Projects are the core of the practised teaching methodology and are structured in a pedagogical format according to the course programme. The use of the FPGA technology as the most suitable implementation technology for digital design teaching purposes is discussed. The course structure, oriented to the development of real working digital systems, challenges the students and increases their motivation. This way, the learning process is improved and the classes are more productive. A laboratory development infrastructure based on a FPGA device, used to implement a real-time video processing system, is presented. Examples of laboratory projects implemented with this infrastructure in a recent course edition are also presented.**

I. Introduction

The course discussed on this paper is a course on advanced digital systems design taught at Faculty of Engineering of the University of Porto (FEUP), Portugal. It is taught in the 4th year of the integrated Master's Degree (Bologna-compliant graduation) in Electrical and Computer Engineering (ECE) and is the first in a stream of two complementary courses on digital design. It calls Digital Systems Design (DSD), EEC0055 [1], and addresses the front-end design phase of complex digital systems for implementation on microelectronic technologies. The second one, VLSI Circuit Design (EEC0056 [1]), concentrates on the physical design of digital systems in current silicon-based IC (Integrated Circuit) fabrication technologies. The subject of this paper is the DSD course, focusing on the teaching methodology and on the laboratory assignments.

Assessment is most commonly associated with testing whether students have achieved the learning outcomes of the course. However, used imaginatively, assessment techniques can be used as a part of the learning experience to help foster deeper engagement with the material under study. Student participation is widely recognized as being a key point in effective learning. It is also known that the traditional lectures can be augmented with learning experiences that involve and challenge the students [2], [3]. This is particularly true in a technological area, where a key aspect for a successful learning is the contact of the students with laboratory equipments. By this way they can really experiment the solution of their assignments and apply their skills in an effective manner.

The technology and Electronic Design Automation (EDA) tools employed by industry in the design of digital hardware evolve rapidly. Well-prepared engineers, who are able to produce actual designs and adapt to this dynamic world, are in demand. Thus, the training of a digital hardware engineer should be more focused on acquiring designing skills than on learning a specific Hardware Description Language (HDL) or software tool. However, proposals such as [4], [5], [6] are too focused on the details of the HDL and do not answer to the necessities of the industry [7]. Nowadays, many proposals are being made to try to reduce the gap between the educational community and industry [8], [9], [10], [11]. In a digital design course, students should work similarly to digital hardware engineers in a company, in particular when it is a course near the end of their graduation.

The main objective of the DSD course is to face the students with industrial methodologies and EDA tools, and apply them to complete real working digital systems. Motivation of the students increase if they can see, touch and play with the final results of their projects, going far beyond computer simulations and model abstractions. This aspect has a key role that contribute for an effective strategy of active learning being used in the DSD course.

After these considerations the contents of the remaining sections are described. Section II discusses key aspects about alternative implementation technologies and justifies the most appropriate choice to teach an advanced digital systems course. Section III describes the goals of the DSD, the current course programme and the practised teaching methodology. Section IV describes the laboratory assignments and explains the adopted approach. A laboratory infrastructure, used by the students to experiment their projects, is presented and examples of some proposed assignments are also included. Section V ends the paper by presenting the main conclusions from the central issues discussed along it.

II. Choice of the technology

To practice the complete *design-verification-physical implementation* design cycle for a complex digital system within a 14-week course (one semester) it is necessary to adopt a convenient implementation technology that allow the students to see and experiment the results of their projects in the laboratory. Although FEUP, as a member of the European EUROPRACTICE programme [12], has easy access to the fabrication of integrated circuit prototypes, the typical 2-3 month turnaround time invalidates the utilisation of this solution as the target technology. Besides that, the relative high costs of IC fabrication represents a huge limitation for

practical design iterations, which is inevitable to happen during the learning process.

Field-Programmable Gate Array (FPGA) is now a mature technology for the implementation of digital systems [13], [14]. These are completely fabricated commercial ICs that can be configured an unlimited number of times by the end user in a matter of seconds using low cost equipment. It allows the rapid implementation of multi-million gate-equivalent digital systems with no real costs of silicon fabrication, providing high-performance digital systems competing, in same cases, with custom made ICs. Although the emphasis of the DSD course is not FPGA-targeted design, this technology was adopted in the course as the implementation target for the laboratory projects due to various reasons: fast design cycle with no costs of design iterations, support for high-performance and complex digital systems, and availability of a large variety of commercial development platforms that include various peripherals and interfaces (e.g. RAM, analogue-digital signal converters, network interfaces, etc.). According to published experiences [15], [16], [17], [18], several other engineering schools around the world have adopted FPGA circuits to teach their digital design courses.

By the given reasons and considering the context of the DSD course in the ECE graduation programme, the FPGA technology was chosen to implement the laboratory projects.

III. Digital Systems Design Course

The aim of DSD course is to supply the students with technological skills and methodological aspects about the design of complex digital systems, considering their implementation in microelectronic technologies or in reconfigurable digital systems. For that, it is intended to give them the necessary competences to design digital systems using a structured approach, based on digital HDLs at different abstraction levels and using commercial design tools.

A. The design process

The digital systems design is centred in modelling, validation and synthesis tasks. They can occur at different abstraction levels and representation domains, depending on the complexity of the project. Figure 1 represents them.

At the system level, the starting point is generally the study and evaluation of an algorithm to use in the system to be designed. For that, behavioural models are used to choose and evaluate a resolution method for the problem. The modelling and validation processes are based on modelling and simulation tools, e.g. Matlab/Simulink and programming languages. Synthesize a model at the system level consists on translate it to a representation in a lower abstraction level, i.e. the Register Transfer Level (RTL), based on RTL blocks that implement functional elements.

Perform a design at the RTL level consists on to build and to validate a RTL model, in a behavioural or structural mode, and translate it to a structural representation at the logic level that is functionally equivalent to the initial model. This constitutes an important task when designing complex digital systems and

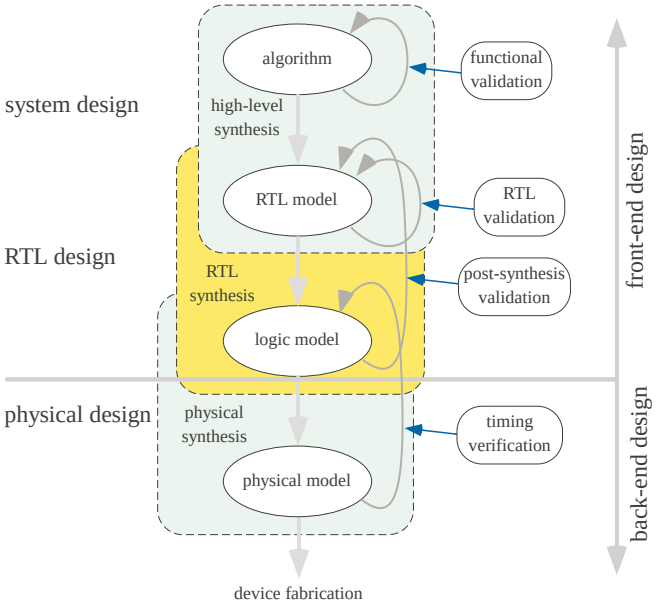

Fig. 1. Digital design tasks, abstraction levels and representation domains.

is supported by simulation and automatic synthesis tools that reveal essential in this activity. The result of this process is a model that represents a netlist of available logic elements of the target technology, each one corresponding to a physical cell on that technology. This phase and the previous one complete the front-end design phase where an algorithm is converted in a digital system.

The translation of that structural model into a physical representation is the physical design task, also known as back-end. This process is specific of the target technology. It involves the spatial organization of blocks over the physical area of the target device, the placement and routing of cells, and the extraction of physical characteristics after that placement and routing.

B. Course programme

The course is composed by the following main topics:

- Structured design of digital systems: design flow, top-down and bottom-up design methodologies, modelling at different abstraction stages, modularity and hierarchy concepts;
- Hardware description languages: modelling, verification and synthesis perspectives, and Verilog HDL;
- Verification of the project: functional validation, logical verification and temporal analysis, and techniques for writing testbenches;
- Synthesis of digital systems: RTL and behavioural synthesis, and RTL synthesis under temporal constraints;
- Synchronous digital systems: problems related with the clock generation, management and distribution, and systems with several clock domains;
- Reconfigurable implementation technologies: FPGAs and combined FPGA/microprocessor systems;

- Control and datapath synthesis: synthesis of control units and custom processing units, architectures of both fixed-point and floating-point arithmetic operators;
- Low power: design principles and techniques oriented for the reduction of power consumption.

As far as possible, these topics are maintained independent of the implementation technology.

C. Teaching methodology

The topics in the DSD programme are taught on the theoretical classes. During the lectures, slides are employed as a teaching aid. The slides and other materials of the course are available for the students before the start of the course, allowing the students to make better use of in-class time. Whenever opportune, exercises are solved to exemplify or clarify some particular aspects. The evaluation of the knowledge acquired by the students on the theoretical classes is done by two mini-exams, each one graded with 4 in 20. This examination is mainly used to clarify for possible inequities of effort among team members when working on laboratory projects.

The practical classes are used mainly to develop laboratory assignments that exercise the design methodologies, the development tools and concepts presented in the theoretical classes. In terms of assessment, the grade of the assignments is 12 in 20, since the core of the adopted teaching methodology is the set of laboratory projects.

By the reasons described in Section II, the laboratory assignments are addressed to use reconfigurable circuits to implement the developed digital systems. Concretely, Xilinx FPGAs are used for that purpose with the ISE development tools [19] and the used HDL is Verilog [20]. The design approach consists on using the top-down methodology, automatic synthesis and implementation on a Xilinx Spartan-3 FPGA contained in a prototyping board [21].

The first stage of the design cycle, the functional verification, is essential for a successful design. Wrong models that not satisfy the project specifications can not result in correct hardware solutions. As part of the teaching methodology, the students are alerted to this fact and to the importance of the functional verification of a digital system as a way to guarantee its correctness. To put this in practice, the students are encouraged to exchange the testbenches among them. This reveals a very useful practice to avoid incomplete or wrong design verification because their testbenches can be addicted. If the model under checking is wrong, due to a mistake on the interpretation of the project specification, then the testbench also tends to be wrong.

The organization and the contents of the laboratory assignments where this methodology is applied are described in the next section.

IV. Project based laboratory assignments

The DSD course has a strong hands-on practical component based on three main laboratory assignments. The practical classes are planned to allow the execution of these projects in articulation with the theoretical classes. By this way the background topics are guaranteed. These assignments allows the students to progressively contact with the design flow and methodologies as well as the corresponding tools. Next section details the sequence of the three projects focusing on their objectives and explaining their corresponding pedagogical contributions.

A. Organization of the assignments

1) Lab 1: The first of these projects intends to face the students with modelling and functional verification of digital systems, familiarizing with the Verilog HDL and the ModelSim simulator [22]. For this purpose a control-dominated synchronous digital system is designed. The main issues addressed on this assignment are the correct understanding of RTL modelling of sequential and combinational logic circuits, building basic testbenches with automatic verification procedures, practice with the front-end design stage through the design cycle and verify the project against specifications.

This first assignment is initiated after a tutorial about ModelSim utilization. The specification is given to the students in a class, where they should answer to a short questionnaire. This intends to motivate them to clarify what they should do and how. At the end of the session the students have their own architecture of the circuit to be modelled.

2) Lab 2: The second project is intended to obtain a circuit, implementing it in a FPGA and see it working. For the first time the students visit all the design stages: modelling, functional validation, synthesis, implementation and final experimentation. A basic ISE project is supplied with the basic necessary resources to implement it in the Spartan-3 prototyping board. The students add the module that they should develop to this ISE project.

It is intended with this approach that the students concentrate on the circuit to be developed, once this second project still usually reveals some flaws of the students at the modelling level. The reason for that is the improper use of the HDL constructions that not synthesize properly. This is one of the main difficulties when students start work with a HDL, because they tend to confuse a HDL with a programming language. Several authors relate this problem [23]. This approach still allows the students to understand the overall organization of a complete ISE project and learn how to define design constraints, like assigning signals to the input/output pins of the FPGA circuit, and setting the maximum delay for some critical signals. Additionally, the students familiarize with the remaining design tools.

As occurred in the previous project, this one starts after a tutorial about the ISE package that shows the use of the main tools.

3) Lab 3: The third project is a more complex assignment. Students have to conceive and develop core parts of a given design, exploring and evaluating alternatives to meet realistic design constraints and optimize some quality criteria. In the past few years several different projects were proposed, spanning diverse application areas such as audio coding and

processing, software radio, processor design and real-time video processing. Although the proposed projects are quite simple to be compatible with the student's skills, they result in interesting demonstrators and motivate the students into an active learning process.

Due to the design of a concrete application and to the used methodologies and tools, the students are involved in an activity that is close to the realities that they will find in companies within the area of electronic digital systems design at the job market.

The largest duration and complexity of this last project justify its greater weight comparing to the other course assignments. The grade of the third project is normally 6 to 8 in 12. The evaluation is made considering aspects like the organization and quality of the Verilog code of both developed modules and testbenches, the produced datasheet about the developed system including a brief characterization and the results of the implementation, and obviously considering the fundamental aspect that is the correct functionality of the system. The optimization of the system implementation, measured by the gate count, is also valued serving to distinguish the several teams.

B. Laboratory infrastructure

This infrastructure was developed and implemented two years ago and since that it is being used successfully. It consists on a processing chain that receives data from a OmniVision 7120 [24] monochromatic camera, perform some image processing operation and supplies the resultant signals to the board that interfaces with the VGA monitor. This basic real-time video processing system, 30 frames per second and 8 bits per pixel, is built on a XILINX Spartan-3 FPGA board. The FPGA implements the video processing system and other modules. It includes circuitry to interface with a PC and to define some parameters that control the operation of the digital camera. This infrastructure is depicted in Figure 2, showing their composition and the physical resources.

An ISE project with all the blocks that support the implementation of image processing operations is normally given to the students with a module that exemplifies how to integrate in the system the modules that they must develop. Normally, a module that calculates the negative image is included, serving also to show how they can use the buses that transmit the image data. Figure 3 shows a structural view of the given top-level design, with emphasis on the video processing module inserted on the chain.

One of the blocks included in the given ISE design, synchronize the signals coming from the digital camera with the 100 MHz system clock. The pixel values are generated at 12.5 MHz, and for each one the correspondent coordinates are also generated, considering that $(0, 0)$ corresponds to the upper left corner of the image. Besides that, horizontal and vertical synchronism signals are also produced. The system still includes an interface with the digital camera I2C bus, allowing the configuration of several parameters of the camera by setting proper configuration registers, e. g., it is possible to

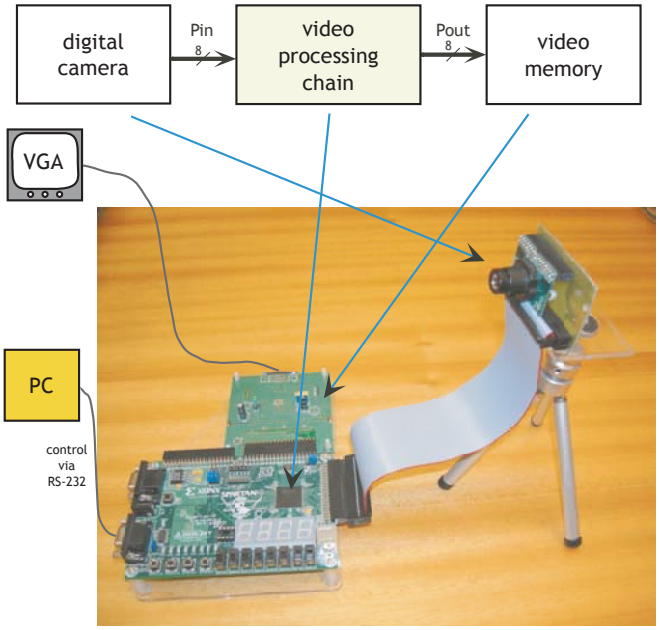

Fig. 2. Laboratory infrastructure.

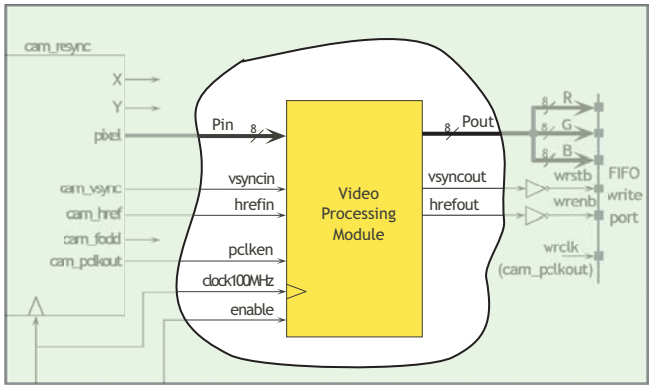

Fig. 3. Insertion of a video processing module on the given design.

define the image scan mode, choosing the interlaced form or the sequential one, etc.

Various image processing operations, normally known by the 4th year graduation students, can be implemented. For all of them, the implementation methodology consists on defining an architecture that satisfies the required specification, its modelling and functional verification, its integration on the system by joining it to the whole design, and finally the synthesis and implementation of the entire design. To implement the entire system in a synchronous manner, the input signals should be read with the 100 MHz global clock using the signal that validates each new arrived pixel as a clock enable. In spite of that, the local processing operations can use all of the global clock cycles. By this way, a processing module can use up to 8 clock cycles to do some operation over each pixel of the image. However, this is only possible since that the line and frame synchronism signals suffer a delay with exactly the same number of cycles, guaranteeing the correct timing

of the signals outputted by that module. This is the reason why the synchronism signals pass throughout the module to be added.

After being processed, the image is sent to an external board that is equipped with a set of FIFO memories where it is saved. These memories are then read, synchronously with the produced VGA synchronism signals, and applied to a digital-analogue converter that produces the analogue signal to the VGA monitor.

Complementing this hardware infrastructure, it is also supplied a PC application to interface the FPGA circuit, allowing a bi-directional communication via RS-232. This tool reveals very useful, once that the students can control their modules and, particularly, they can debug them by observing problematic signals to understand what can be wrong with them.

C. Examples of proposed assignments

Some of the laboratory projects proposed on the third assignment in recent course editions are presented to show what is involved in them. In all of them, the students have to integrate some proposed processing functions in the video processing system, i.e. they add some modules to a given base ISE project to then implement and try it.

1) Histogram making: This project consisted on the determination and visualization of the real-time histogram corresponding to the captured image. To define the visualization place, the coordinates are defined using the PC communication interface (section IV-B). The counting of the pixel's frequencies on a frame is saved in the FPGA internal memory to allow the visualization of the current histogram within the next frame. Actually, this was the principal challenge to the students as well as the normalization of the histogram to allow a convenient representation scale.

2) Motion detection: A basic technique for motion detection is based on the difference of two consecutive video frames. This was applied in a project that consisted in the implementation of a motion detector based on the video image system. It allows the end user setting threshold levels that establish when it is considered motion. To help the end user having movement perception, the level of movement is represented as a bar over the original image. The size of that bar is proportional to the detected movement. It is also possible to specify in run-time the coordinates of the window where the motion should be detected. That region is highlighted to differentiate it in the whole image on the screen.

3) Bi-dimensional convolution: A 2D convolution was also implemented in a recent laboratory assignment. A 3×3 matrix was used in the convolution operator. Choosing appropriate values to that 9 coefficients result various filtering effects, e.g. edge detection, blur, emboss, sharpen, etc. The principal challenges were the arithmetic involved, i.e. how to allocate and schedule the operators, saving the line of the current pixel and the previous two, and also the effect of the image borders. These situations were problematic due to the FPGA available space and system timing constraints to allow the real-time system operation.

V. Conclusion

FPGAs are currently the main implementation technology of digital systems when a fast design cycle is required. The fast design cycle of this technology and the possibility to reconfigure the circuits allow the students to perform unlimited design iterations without additional costs. By this way they can explore the solution space of a digital system implementation considering the consumed resources and its performance. These features improve the productivity of the classes and the learning process results more appellative and attractive.

This paper presented a project driven digital systems design course considering the implementation of such projects in FPGA devices. The teaching methodology and the laboratory projects of the course were presented. With this approach, the students acquire valuable knowledge and skills on an actual technology that assumes a role with growing importance in the development of digital hardware, once FPGA based prototyping impels a design approach based on HDLs that favors the design portability to other target technologies.

References

[1] `http://www.fe.up.pt`.

[2] Juan A. Marin-Garcia and Jaime Lloret. Improving teamwork with university engineering students. The effect of an assessment method to prevent shirking. *WSEAS Transactions on Advances in Engineering Education*, 5(1), January 2008.

[3] Saumil Merchant, Gregory D. Peterson, and Don Bouldin. Improving embedded systems education: Laboratory enhancements using programmable systems on chip. In *Proceedings of the 2005 International Conference on Microelectronic Systems Education*, Anaheim, CA, USA, June 2005.

[4] Zainalabedin Navabi. The impact of hardware description languages on the education of hardware design. In *Proceedings of the 8th IEEE University/Government/Industry Symposium*, pages 235–239, Westborough, MA, USA, June 1989.

[5] Assim Sagahyroon and Mehran Massoumi. On the use of hardware description languages in teaching VLSI design courses. In *Proceedings of the 26th. ASEE/IEEE Frontiers in Education Conference*, pages 713–716, Salt Lake City, UT, USA, November 1996.

[6] Assim Sagahyroon. From AHPL to VHDL: A course in hardware description languages. *IEEE Transactions on Education*, 43(4):449–454, November 2000.

[7] Don M. Gruenbacher. An active learning/teamwork approach to implementing an integrated circuit design cycle in an advanced hardware description language course. In *Proceedings of the 29th. ASEE/IEEE Frontiers in Education Conference*, San Juan, Puerto Rico, November 1999.

[8] José N. Amaral, Paul Berube, and Paras Mehta. Teaching digital design to computing science students in a single academic term. *IEEE Transactions on Education*, 48(1):127–132, February 2005.

[9] Valery Sklyarov and Iouliia Skliarova. Teaching reconfigurable systems: methods, tools, tutorials and projects. *IEEE Transactions on Education*, 48(2):290–300, May 2005.

[10] Hugo Hedberg, Joachim N. Rodrigues, Fredrik Kristensen, Henrik Svensson, Matthias Kamuf, and Viktor Owall. Teaching digital ASIC design to students with heterogeneous previous knowledge. In *Proceedings of the 2005 International Conference on Microelectronic Systems Education*, pages 15–16, Anaheim, CA, USA, June 2005.

[11] Igone Vélez and Juan F. Sevillano. A course to train digital hardware designers for industry. *IEEE Transactions on Education*, 50(3):236–243, August 2007.

[12] `http://www.europractice.com`.

[13] Stephen D. Brown, Robert J. Francis, Jonathan Rose, and Zvonko G. Vranesic. *Field-Programmable Gate Arrays*. Kluwer Academic Publishers, 1992.

[14] R. C. Cofer and Benjamin Harding. *Rapid System Prototyping with FPGAs: Accelerating the Design Process*. Elsevier, 2005.

[15] Don Bouldin. Impacting education using FPGAs. In *Proceedings of the 18th International Symposium on Parallel and Distributed Processing*, Santa Fe, USA, April 2004.

[16] Stephen A. Edwards. Experiences teaching an FPGA-based embedded systems class. In *Proceedings of the Workshop on Embedded Systems Education (WESE)*, New Jersey, USA, 2005.

[17] Yong-K. Jung. Work in progress – a rapid design methodology for FPGA-based processor platform design education. In *Proceedings of the 35th. ASEE/IEEE Frontiers in Education Conference*, Indianapolis, USA, October 2005.

[18] Dong-Soo Kang, Soo Yun Hwang, Kyoung-Son Jhang, and Kang Yi. A low cost and interactive rapid prototyping platform for digital system design education. In *Proceedings of the 2007 International Conference on Microelectronic Systems Education*, San Diego, CA, USA, June 2007.

[19] `http://www.xilinx.com`.

[20] The Institute of Electrical and Electronics Engineers, New York. *IEEE Standard Verilog hardware description language*, September 2001. IEEE Standard 1364-2001.

[21] Xilinx. *Spartan–3 Starter kit board User Guide*, 2005. `http://digilentinc.com/Data/Products/S3BOARD/S3BOARD-rm.pdf`.

[22] `http://www.mentor.com`.

[23] R. James Duckworth. Embedded system design with FPGAs using HDLs (lessons learned and pitfalls to be avoided). In *Proceedings of the 2005 International Conference on Microelectronic Systems Education*, Anaheim, CA, USA, June 2005.

[24] OmniVision Technologies, Inc. OV7120 single-chip CMOS VGA b&w digital camera, 2001. `http://www.ovt.com`.