Rapid-Prototyping Emulation System using a SystemC Control System Environment and Reconfigurable Multimedia Hardware Development Platform

DAVE CARROLL, RICHARD GALLERY
School of Informatics and Engineering,
Institute of Technology Blanchardstown, Dublin 15,
IRELAND

Abstract: - This paper describes research into the suitability of using SystemC for rapid prototyping of embedded systems. SystemC[1][2] communication interface protocols[3][4] are interfaced with a reconfigurable hardware system platform to provide a real-time emulation environment, allowing SystemC simulations to be directly translated into real-time solutions. The consequent Rapid Prototyping Emulation System Platform (PESP), suitable for the implementation of consumer level multimedia systems, is described, including the system architecture, SystemC Controller model, the FPGA configured MicroBlaze CPU system and additional logic devices implemented on the Multimedia development board, illustrated in the context of a typical application.

Keywords: - emulation, systemc, reconfigurable, prototyping, co-verification, real-time

1. Introduction

The paper describes research into the development for SystemC of an emulation environment in which SystemC based specifications can be translated directly into real-time hardware/software implementations. This would allow SystemC based designs to be implemented directly as real-time solutions, and would provide a rapid prototyping route (improved time-to-market) for complex hardware/software systems.

This paper describes a prototyping platform, incorporating hardware acceleration under software control, suitable for the real-time implementation of SystemC designs. The development of this system includes research into the architecture required to enable SystemC to be used directly for real-time systems, and the implementation of a prototyping system, with an application focus on consumer level multimedia systems, to investigate the viability of the approach.

The SystemC section of the prototyping system provides a tool for the specification, implementation and evaluation of applications, implemented across selected system architectures, and facilitates selection and evaluation of different communication interface approaches between the embedded SystemC functional models within the design.

The hardware acceleration platform, oriented towards consumer level multimedia applications, is based on an FPGA multimedia development platform. This provides a reconfigurable hardware emulation environment, for prototyping real-time application designs, in conjunction with the SystemC system model. The Hardware subsystem is designed around a Xilinx MicroBlaze and Multimedia Development Board[10] (MMDB), incorporating Xilinx Virtex II[8]. Hardware functional blocks developed as part of the system are written as VHDL models [16][17] .

2. Real-time SystemC based designs

Research questions include:

- Whether the underlying architecture of SystemC is suitable for direct real-time implementation of a SystemC based model

- If not, is there some subset of, or enhancement to, the SystemC approach, which is suitable for real-time implementation.
- Which modelling approaches, when deployed in SystemC based designs, are suitable for (real-time) emulation?
- The interaction of SystemC with real-time OS. This is an active development area within SystemC, and also in hardware-software co-verification systems (e.g. [5][6][7]).
- What hardware/software architectures for emulation platforms are required to allow real time implementation of SystemC designs?
- What hardware components are required within such real-time platforms to facilitate SystemC designs?
- What interfaces are required between the software and hardware aspects of such systems?
- What Industry standards (e.g. Open Core Protocol, OSCI-TLM standard, SCE-MI) are relevant, and in what manner should they be considered in conjunction with SystemC.

This paper discusses the PESP and in doing so presents a suitable SystemC based design, that when interfaced with a reconfigurable hardware system platform, using a suitable communication interface protocol, demonstrates that the underlying architecture of SystemC is suitable for direct real-time implementation of a SystemC based model. The authors present a hardware/software architecture for an emulation platform that facilitates real time implementation of the SystemC design and defines the hardware components used in that design.

The prototyping system is based around an embedded processor core controlling hardware components, and implements a subset of the SCE-MI specification to achieve this.

2.1. A Co-Verification Platform

The development of architectures and platforms suitable for the emulation of systems described with system level description languages is an active research area. This research is largely targeted towards co-verification platforms, where fast verification of a design is required, rather than real-time emulation [5]. Nevertheless there is a significant correspondence in the research required to develop both fast verification systems and real-time systems.

For example in [3] the functional requirements for a co-emulation modelling interface are discussed and an architecture and API is provided (SCE-MI). SCE-MI has been developed to meet growing industry demands for verification platforms for Systems-on-Chip (SoC) designs. It provides a mechanism by which the software aspects of an SoC design may be simulated on a workstation, the hardware aspects may be simulated on a hardware verification platform, with an efficient interface mechanism (based upon hardware-software transactors implementing Transaction Level Modelling [2]) between the two (see Fig. 1). SCE-MI facilitates the use of a system design language such as SystemC.

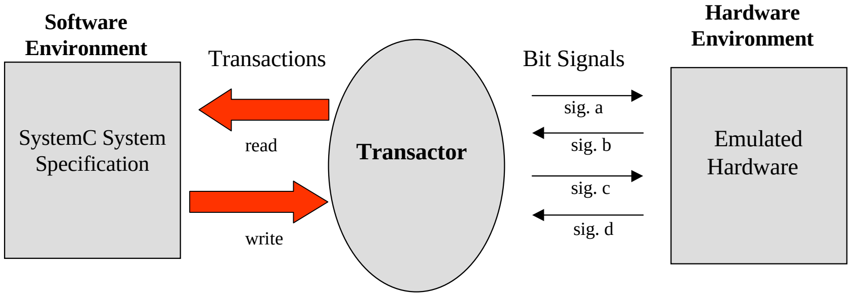

Fig. 1: co-emulation modelling interface

In a SystemC design it would be normal to initially produce a high-level abstract model of the entire system, with the process of refinement then being used to redesign the system for the target platform. Once this has taken place, the hardware components are simulated and verified. The Zebu system [18] provides a software/hardware co-verification platform using an implementation of the SCE-MI protocol. The software and hardware components of the design can be verified together, with the hardware emulation system allowing much more rapid verification time than would be the case using a software simulation of the hardware components.

The hardware-software transactor is a form of abstract gasket, which forms part of the SCE-MI infrastructure. The transactor communicates at the transactional level (e.g. Read and Write commands) with the software side of the system model, decomposing untimed messages into a series of cycle-accurate clocked signals. These clocked signals form the communication interface between the transactor and the hardware, on the hardware side of the system model. Similarly the hardware communicates with the transactor at the signal level, where the cycle accurate signals are recomposed into transactional level messages, for transfer to the software interface. Fig. 2 shows an example of a message being translated from the software side of a system model, through the transactor, to the hardware signal side of the transactor.

During the period from when the message is received by the transactor, controlled time will be suspended within the software environment via handshaking between the transactor and the SCE-MI infrastructure. The transactor will decompose the incoming message and generate required cycle accurate bit level signals to drive the hardware. Once the transactor actions have been completed, controlled time will be resumed via the transactor handshaking and the SCE-MI.

3. PESP Description

The PESP has been developed with a focus on investigating the application of SystemC based rapid prototyping in consumer level multimedia applications. Its is based around the concept of a SystemC based control system, supported by application specific hardware accelerators and other components, with communication between the software and hardware managed by transactors based upon the SCE-MI protocol.

Currently this system is based on a SystemC control system using set-up on a host PC (running a Microsoft Operating System) and a MMDB[10] hardware development platform, containing a Xilinx, MicroBlaze softcore processor[15] implemented on a Xilinx Virtex II FPGA fabric, along with other ASIC devices. The SystemC control system and the MMDB communicate through transactors, utilising a serial or other interface (such as an Ethernet link) at the physical layer. It should be apparent that the use of a PC in the system is for development purposes only, and that the System C control system would be implemented on the MicroBlaze softcore processor.

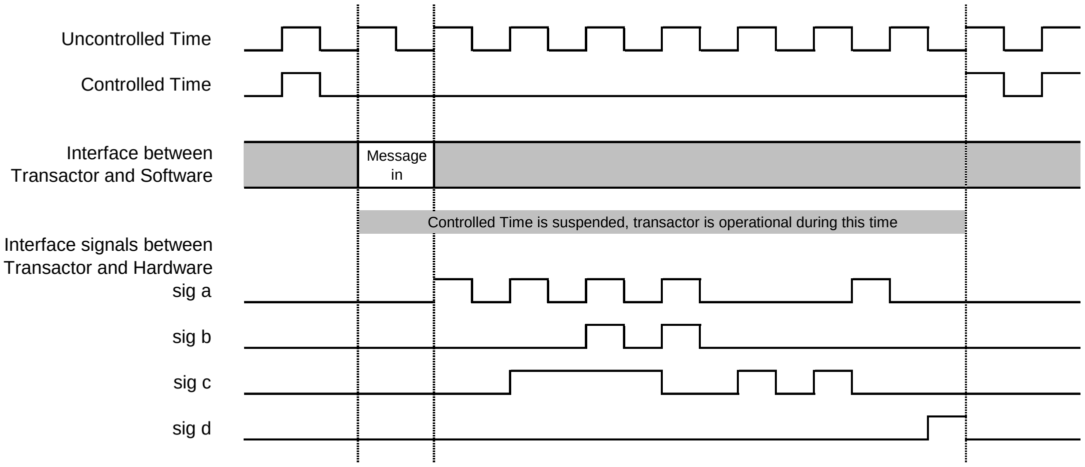
Uncontrolled Time
Controlled Time
Interface between Transactor and Software
Message in
Controlled Time is suspended, transactor is operational during this time
Interface signals between Transactor and Hardware
sig a
sig b
sig c
sig d

Fig. 2: message timing through transactor

3.1. Platform Architecture

The architecture of PESP is described using a multi-layered model, de-coupling the system architecture from the implementation details. The multi-layered architecture consists of an Application Layer (AL), Presentation Layer (PL) and Driver Layer (DL).

The DL controls each of the hardware components directly, passing required parameters and enabling the hardware components (which may be implemented in hardware, software, or a combination of both).

3.2. Communications Interface

The platform architecture described above is now mapped to an implementation incorporating SystemC and re-configurable hardware. The communications interface between the SystemC model and the re-configurable hardware is modelled on the Standard Co-Emulation API Modelling Interface (SCE-MI) [3].

SCE-MI describes a modelling interface based on a multi-channel abstract bridge, providing multiple communication channels that allow software models describing system behaviour, such as the PESP SystemC controller models, to connect to structural models describing implementation of a hardware emulation system.

Fig. 1 shows a high level view of the communications interface between the SystemC controller system and the MMDB based hardware emulation system. The SystemC system is on the left of the figure and consists of several SystemC models describing the controller functions, and the Xilinx MMDB platform is on the right. The Transactor performs the interface mechanism linking the high level SystemC controller description and the hardware implementation of functions within the FPGA MMDB. The Transactor implementation consists of several layers of software and hardware, from the SystemC model down to the PC hardware drivers, across the physical interface to the MMDB development platform and onto the MicroBlaze implemented on the FPGA fabric.

3.3. SystemC Controller System

The PESP SystemC is constructed using a modular approach to provide partitioning between the different functional elements of the overall controller. This facilitates the movement of functionality between different models, which proved useful during the Control model design. It also simplifies the addition and/or removal of models from the system. This was particularly useful when moving functionality from the SystemC Controller System to hardware, as the hardware platform was being constructed. Additional SystemC models were required in order to implement the SCE-MI protocol.

The ability to change or refine the communication interfacing between the different models proved very useful when deciding which interface channel type to use to connect the SystemC models that make up the overall SystemC Controller system. Different interface channel and port types affect the flow of data between SystemC models. Communication refinement is described in detail in [1][2] and [4]. Primitive channels (sc_fifo) were chosen for communications between SystemC models in this system. The sc_fifo primitive channel provides the ability to communicate between SystemC models using a blocking and non-blocking read/write instruction; which ensures that data does not get lost between data transfers.

Fig. 3 shows the SystemC models designed for this system and the interconnections used.

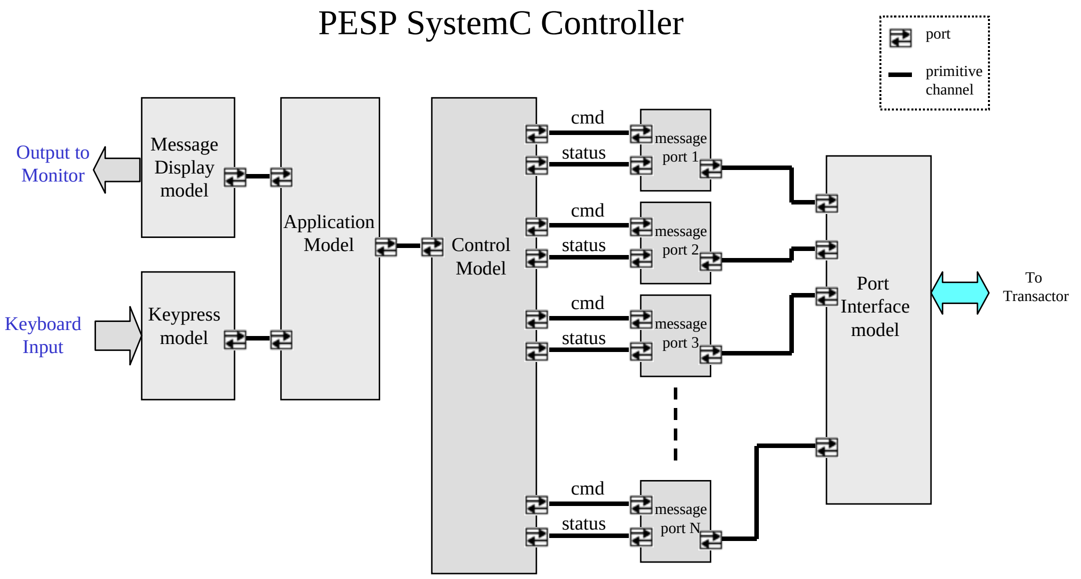

Fig. 3: Block diagram for PESP SystemC Controller

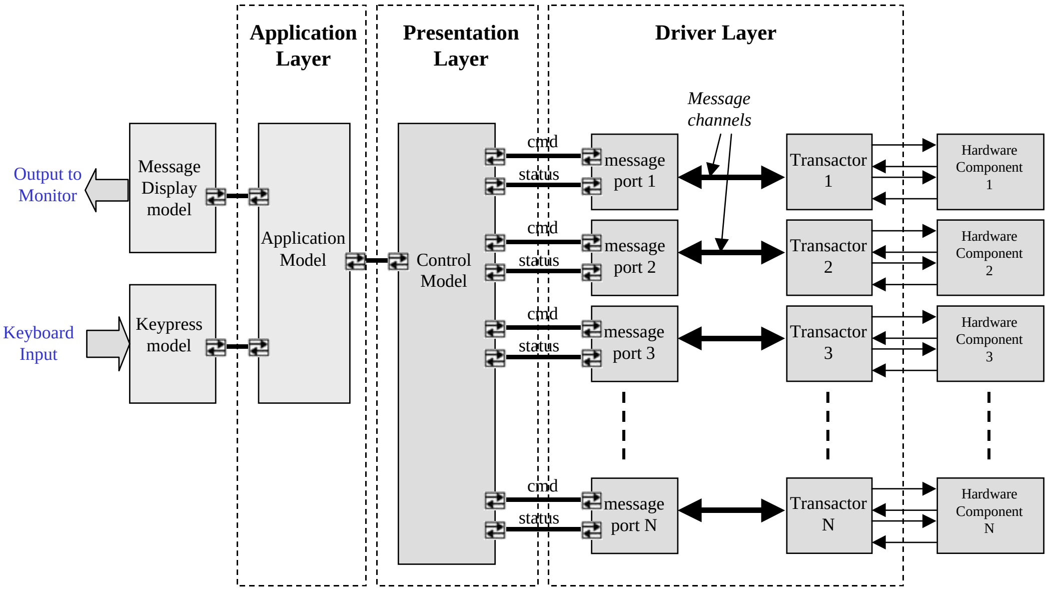

Fig. 4: multi-layered architecture of SystemC Controller model

The controller system architecture is represented by the multi-layered architecture model as shown in Fig. 4, where the AL of the system consists of the SystemC application model. The PL function of the controller system architecture is defined in the control model. The PL puts commands received from the AL in a defined order and communicates with the DL, sending requests and receiving status. In this system the DL is spread across the Port Interface model (Fig. 3) and on to the MMDB platform. The DL contains the transactors, and is responsible for communication with the hardware.

3.4. Reconfigurable Hardware

The MMDB was chosen as the reconfigurable hardware as it provides a range of multimedia oriented features, combined with an FPGA to allow additional functionality as required. It is built around the Xilinx MicroBlaze 32-bit RISC softcore processor, which, as currently configured, acts as part of the transactor, controlling hardware components on the MMDB, and managing dataflow between those hardware components themselves, and higher level SystemC modules.

The transactors, implemented through software and hardware, control the video hardware using clock cycle accurate bit signals through the MicroBlaze Peripheral Bus interface registers.

The higher level SystemC models are currently implemented on a PC. In a later stage of development, all software functionality would be transferred to the MicroBlaze itself.

Fig. 5 shows the hardware components that (developed as part of the research) are controlled by the transactor. The Display Engine Module, Image Grabber, Workarea Control Module and Resync Module are VHDL modules, designed, constructed and tested using the Xilinx ISE design environment [11] and interfaced to the MicroBlaze CPU via the OPB Interface Bus, a 32Bits data bus configured to run at 81MHz.

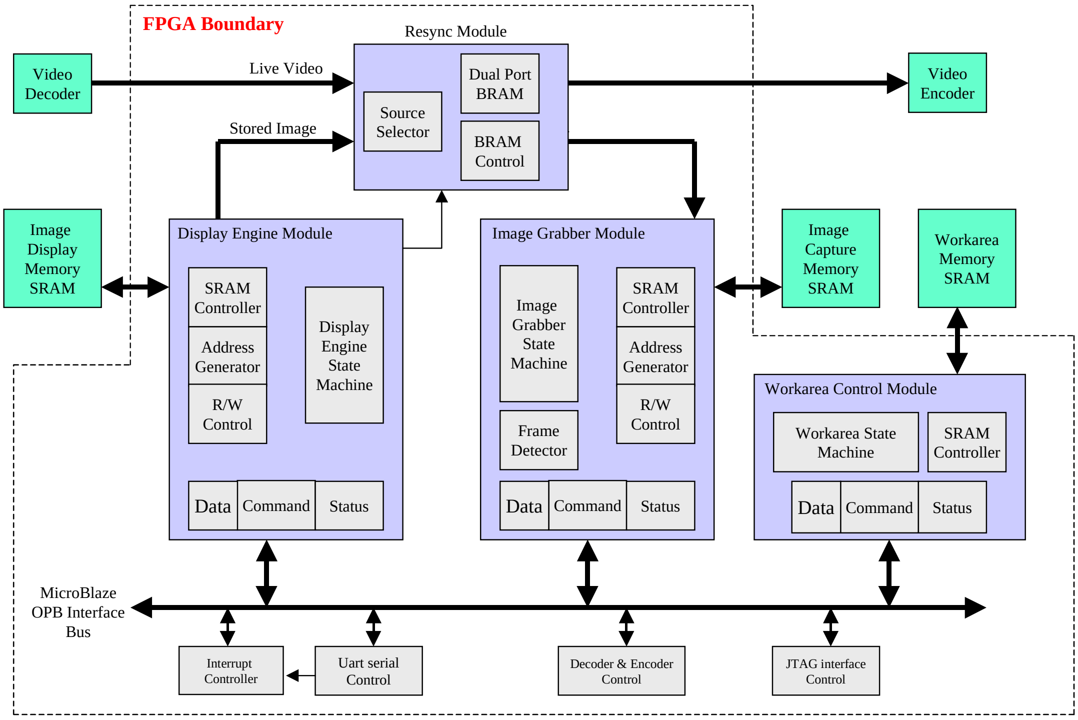

Fig. 5: FPGA based Video System Hardware Architecture

Video images are decoded external to the FPGA and enter the FPGA via the Video Resync module in 10-bit YCbCr PAL data format [12][13][14]. The data is synchronised with the output from a 27 MHz-clock generator model.

The Data exiting the Video Resync model is passed directly to the Video Encoder, a single chip, outside the FPGA boundary.

The Image Grabber VHDL model is used to capture and store images. Communications to and from, the software takes place through Control, Status and Data models connected to the OPB interface bus. Images are stored in the Image Memory device. The SRAM is controlled by the Image Grabber VHDL model.

The Display Engine provides a stream of 10-bit video data to the Video Resync module, synchronised with the system 27 MHz clock. It also controls the Image Display Memory.

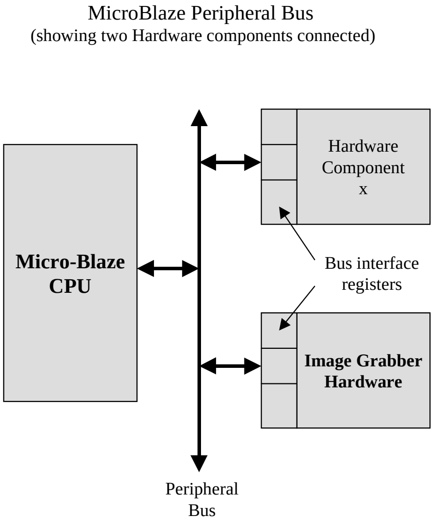

Fig. 6: simplified MicroBlaze Peripheral Bus connection

The Display Engine model is responsible for the display of captured images and provides a stream of 10-bit video data to the Video Resync module, synchronised with the system 27 MHz clock. It also controls the contents of the Image Display Memory device. The Display Engine is under the control of the MicroBlaze via Control, Status and Data models connected to the OPB interface bus.

The Workspace Memory consists of a single SRAM device on the MMDB and is controlled by the Workarea Controller connected to the OPB bus. This storage facility is used for storage of images as they are being manipulated.

4. Example Application

A digital camera, an example of a widely available digital system, which is sufficiently complex to provide scope for different architecture and technology configurations, was used as the focus application for functional testing of PESP. The digital camera function was successfully implemented on PESP.

5. Comparison of Related Methods

Other techniques used for building rapid prototyping platforms for verification of embedded system designs include, for example the Xilinx System Crafter SC[19] tool, whereby the partitioned hardware portion, of a hardware/software SystemC embedded system description, can be translated directly to RTL VHDL and implemented onto an FPGA hardware fabric. This approach differs from the PESP, in that PESP provides a hardware/software platform for the migration and implementation of selected functional blocks to the hardware platform, while maintaining the benefits of the SystemC simulation environment yet providing real-time system evaluation through the use of the SCE-MI interface. This provides the system designer a tool for evaluating different design scenarios in real-time.

Another SystemC based rapid prototyping technique is presented in [20], this paper describes a communications centric design methodology which differs from PESP in that the methodology defines the process to transfer the embedded system design, based in SystemC, entirely to an FPGA implementation. The defined process specifies the use of a SystemC High Level Interface Protocol (SHIP) for SystemC module communication interface, prior to transfer to the FPGA. In the PESP system, the high level SystemC architecture and module interface are not restricted. Thus the reconfigurable hardware components, used for real-time emulation, are connected to the SystemC based system via the SCE-MI interface, providing a real-time prototyping platform.

6. Conclusion & Future work

This paper discussed the PESP, which consists of an integration of both a SystemC controller and a reconfigurable hardware development platform.

This paper presented a suitable SystemC based design, that when interfaced with a reconfigurable hardware system platform, using a suitable communication interface protocol, demonstrates that the underlying architecture of SystemC is suitable for direct real-time implementation of a SystemC based model.

The paper discussed a modular modelling approach, which when deployed in the PESP SystemC based design, is suitable for emulation.

The paper also discussed an hardware/software architecture for an emulation platform that facilitates the real time implementation of the SystemC design and defines the hardware components used in that design.

The higher level SystemC models used in PESP are currently implemented on a PC; however models could be transferred from the PC platform to the MicroBlaze itself, thus implementing the entire system on the MMDB.

The interaction of SystemC with real-time OS was not considered as part of this paper and could be an area of focus for future work. This is an active development area within SystemC, and also in hardware-software co-verification systems (e.g. [5][6][7]).

References

[1] OSCI; "Functional Specification for SystemC 2.0", available at www.systemc.org

[2] *T. Grotker, S Liao, G. Martin, S Swan; "System Design with SystemC"; Kluer Academic Publishers, ISBN 1-4020-7072-1*

[3] Standard Co-Emulation Modelling Interface (SCE-MI) Reference Manual, Version 1.0, Published by Accellera, available from www.eda.org/itc

[4] J. Stickley, Duaine Pryor; "Functional Requirements Specification: Standard Co-Emulation Modeling Interface (SCE-MI)"; www.systemc.org

[5] Emulation and Verification Engineering; "Universal Design Verification Platform for IP, FPGA, ASIC Design and Embedded Software Developers"; www.eve-team.com

[6] Jon Connell, Bruce Johnson , Early Hardware/Software Integration Using SystemC 2.0, ESC San Francisco 2002.

[7] L. Benini et al., SystemC Cosimulation and Emulation of Multiprocessor SoC Designs, IEEE Computer, April 2003, pp. 53-59.

[8] Xilinx, Virtex II data sheet (Xilinx doc. no. DS031-1), www.xilinx.com

[9] Xilinx, Embedded Systems Tools Guide (EDK 3.2.2), www.xilinx.com

[10] Xilinx, Xilinx Virtex II MicroBlaze and Multimedia Development Board, www.xilinx.com

[11] Xilinx, ISE user guide, www.xilinx.com

[12] Charles Poynton, A Technical Introduction to Digital Video, John Wiley & Sons, ISBN 0-471-12253-X

[13] J. Whitaker and B. Benson, Video & Television Engineering, McGraw Hill, ISBN 0-07-069627-6

[14] Analog Devices, ADV7185 Video NTSC/PAL Decoder data sheet, www.analog.com

[15] Xilinx, MicroBlaze RISC 32-Bit Soft Processor data sheet, www.xilinx.com

[16] S. Sjoholm & L. Lindh, VHDL for Designers, Prentice Hall, ISBN 0-13-473414-9

[17] M.Zwolinski, Digital System Design with VHDL, Prentice Hall. ISBN 0-201-36063-2

[18] *C. Alquier, S. Guerinneau, L.Rizzatti, L. Burgun, Co-Simulation Between SystemC and a New Generation Emulator, DesignCon 2003*

[19] *Xilinx, System Crafter User Manual, www.xilinx.com*

[20] *Wolfgang Klingauf, Robert Günzel, Rapid Prototyping with SystemC and Transaction Level Modeling, Technical Report, Technical University of Braunschweig, Germany, 2005*