THE PLATFORM AS AN INTERFACE IN A SOC DESIGN CURRICULUM

INGO SANDER, AXEL JANTSCH, HANNU TENHUNEN
Royal Institute of Technology, Stockholm, Sweden

1. TRENDS IN SYSTEM DESIGN

The steady increase of complexity of single chip systems drives the search for restricting the design space in meaningful ways. The trick is to restrict the design space such that the design process becomes fast and efficient while the resulting product is still close to optimal. One widely used way to do this is to provide architectural templates that allow to quickly assemble a large number of components in a very systematic way. The recent trend to platform based design [1,2,3] is emerging because it is increasingly difficult to organize a large number of pre-designed intellectual property (IP) blocks on chip. Nexperia [4,5] is a successful example of a platform. As figure 1 shows, a typical instance of the platform may consist of a RISC and a VLIW media processor, four buses, a central memory controller and a fairly large number of dedicated functional blocks. It is much easier to start with a platform like this and optimise and fine tune it for a

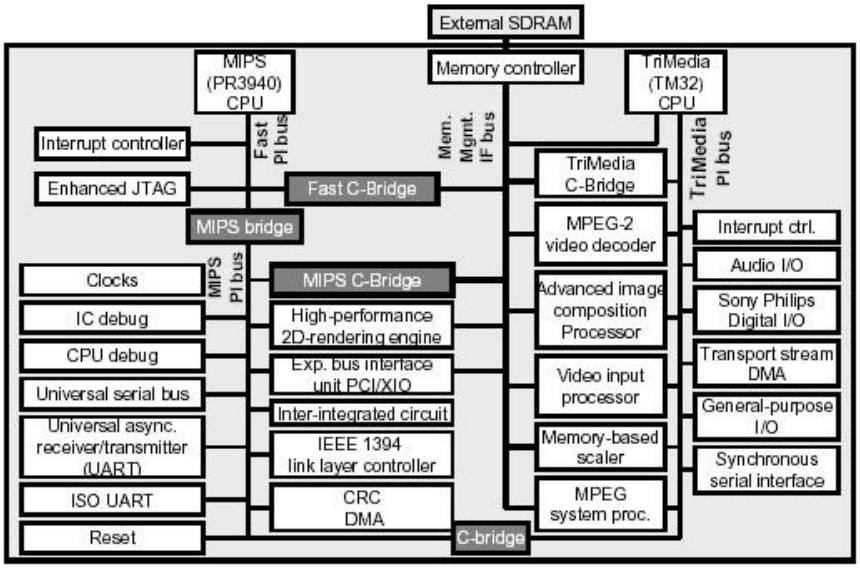

Figure 1. PNX8500 Viper processor based on the Philips Nexperia platform

particular product, than to start from scratch. Also, analysis and design tools can be developed for the platform and reused for every product development, which is a significant advantage because the development of specific tools is typically far beyond the possibilities of individual design projects.

2. CONSEQUENCES FOR EDUCATION

What are the skills needed when platform based design becomes the dominant style for designing SoC ASICs? Apparently we have to distinguish between *platform designers* and *application designers*.
Platform designers will resemble traditional hardware and ASIC designers and will need a strong competence in digital and analog hardware design, a solid knowledge of electrical and physical properties of transistors and wires, and a fair understanding of applications and the relevant system properties such as performance, cost and power. They will need an appreciation of how the platform is used by application designers and they will typically cooperate with application designers because platforms are usually developed with a particular application design project as driver. However, the key difference to the earlier VLSI generation is that the emphaise is on system level electrical issues instead of transistor or circuit block level issues as used to be in VLSI.
Application designers have to deal a lot with embedded software of various kinds. They need a strong competence in embedded operating systems, real-time systems and modelling and analysis of complex, distributed systems. They need to be able to design and verify the functionality of complex mixed hardware/software systems and the will have to conduct sophisticated analysis to verify the system performance and power consumption constraints. Here the emphasis is more on embedded and firmware software. The key difference to traditional embedded systems is that the hardware platforms are more complex (multiprocessor) and even reconfigurable with different techniques.
Obviously there are some joint issues for both groups, e.g. such as fault tolerance, architectures, security, low power, and low cost.
Apparently, these two groups require sufficiently distinct skills and knowledge to justify two separate educational tracks. However, as it stands today, the concept of platform is not sufficiently mature and stable with many overlaps in day-to-day engineering practice to propose entirely unconnected curricula. If a few standard platforms, stable of several generations, emerge, the education of platform designers and application designers can be separated further much like we have specialized curricula for hardware design and for software engineering with the processor as dividing platform. However, since this is not yet a reality we propose a connected curriculum with two specialization tracks.

3. THE CURRICULUM

The program is a 1½ years Master of Science program and attracts students with computer science, computer engineering or electrical engineering background. Figure 2 gives an overview of the program structure and lists the individual courses. The numbers denote EECS credit units. The program is a further development of the SoC Master program [6], which is currently operated at KTH. The new program will become active in the fall 2004.

3.1 BASE BLOCK

First the students pass a common block of courses which takes approximately half a year and starts with an introductory course to Embedded System, which also provides a survey of many important topics many of which are elaborated much more in other courses later on.
SoC Architecture is one of the central courses in the program. It discusses all important SoC processor and communication architectures and presents several

Master Thesis (30)		
Application Design		**Platform Design**
Elective Courses (15) SOC-Applications (7.5) Anatomy of CAD-tools (7.5) Fault-Tolerant Systems (6) SOC Modeling (7.5) System Verification (7.5) ***Compulsory Course*** Embedded Software (7.5)		***Elective Courses (15)*** Embedded Software (7.5) Anatomy of CAD-tools (7.5) Fault-Tolerant Systems (6) Electronic System Packaging (7.5) Low Power & Mixed Signal IC (7.5) Radio Electronics (7.5) ***Compulsory Course*** Digital Circuit Design (7.5)
Embedded Systems (7.5) SOC-Architecture (6)	**Common Courses** ASIC Design (7.5)	Digital System Engineering (7.5) Digital Hardware Design (9)

Fig. 2.The curriculum consists of a common basic block (37.5 ECTS-credits); two alternative specializations with one compulsory course (7.5 ECTS-credits) and several elective courses (15 ECTS-credits); and finally a master thesis (30 ECTS-credits).

important platforms. The Digital Systems Engineering and Digital Hardware Design courses convey the basics in hardware design and the important transistor and wire abstractions. The latter is a necessary foundation to allow for sensible analysis of performance and power at the system level later on. The ASIC course introduces high level hardware design, synthesis, verification, simulation and testing techniques and tools.

3.2 PLATFORM DESIGNER

In the platform designer track the Digital Circuit Design course is compulsory. It is a continuation of the Digital Systems Engineering course and elaborates further on the physical properties of transistors and wires. In two –3 elective courses the student can study topics such as radio electronics, low power, fault tolerance, CAD tools of embedded software.

3.3 APPLICATION DESIGNER

The only compulsory course in the application designer track is embedded software. It introduces the topics layered embedded software architectures, hardware drivers, communication services, resource management and real-time operating systems. This course is an innovation and does not yet exist at KTH and perhaps nowhere else in this particular form. In 2-3 elective courses the student

can further deepen his/her knowledge in advanced verification and modelling techniques, fault tolerance, CAD tools and SoC applications.

4. Summary

At KTH we have developed the successfully operating SoC Master Program further to meet the expected requirements on SoC engineers in the next few years. Based on the assumption that platforms will play a central role in SoC design in the near future, we have designed a curriculum consisting of two tracks, one for platform designers and one for application designers.

5. REFERENCES

[1] Alberto Sangiovanni Vincentelli, "Defining Platform-based Design", EEDesign of EETimes, February 2002.

[2] Henry Chang, Larry Cooke, Merrill Hunt, Grant Martin, Andrew McNelly, and Lee Todd, Surviving the SOC Revolution - A Guide to Platform-Based Design, Kluwer Academic Publishers, 1999.

[3] edited by Grant Martin and Henry Chang, Winning the SoC Revolution, Kluwer Academic Publisher, 2003.

[4] J. Augusto De Oliveira and Hans van Antwerpen, "The Philips Nexperia Digital Video Platform", Winning the SoC Revolution, pp. 67-96, Kluwer Academic Publisher, edited by Grant Martin and Henry Chang, 2003.

[5] Philips Semiconductors. *Nexperia*. http://www.semiconductors.philips.com/products/nexperia/

[6] L. Hellberg, A. Hemani, J. Isoaho, A. Jantsch, M. Mokhtari, and H. Tenhunen, "System Oriented VLSI Curriculum at KTH", Proceedings of the International Conference on Microelectronic Systems Educations, MSE97, 1997.