

System-on-Chip Design: Impact on Education and Research

HUGO DE MAN

Katholieke Universiteit
Leuven/IMEC, Belgium

In his invited, well-received keynote speech at the VLSI Test Symposium held last April in Dana Point, California, Hugo De Man offered his view of how we can prepare future designers to tackle the challenges inherent in SOC design. De Man is an electrical engineering professor at the Katholieke Universiteit in Leuven, Belgium. He cofounded IMEC, Europe's large, independent semiconductor research institute, where he serves as a senior research fellow. He is an IEEE fellow and a member of the Royal Academy of Sciences in Belgium.

DEEP-SUBMICRON TECHNOLOGY is rapidly leading to exceedingly complex, billion-transistor chips. Within a decade, these chips will deliver enormous computing power as well as RF and analog interfaces for information and communication technology (ICT).^{1,2} At the same time, portability and the need for inexpensive packaging will limit power consumption to a few watts or less.

These systems-on-chip (SOCs), designed at the processor-memory level, will fuel the future information society. Such designs depend, however, on SOC architects who can bridge the gap between software-centric system specifications and their implementation in novel, energy-efficient silicon architectures. Designing such hardware-software platforms will require a global-system approach from concept to implementation, which may well require a rethinking of present engineering schools.

Chips will no longer be stand-alone components but complete silicon boards encapsulating complex system knowledge. These boards will be specified far above the hardware-description-language level and implemented in new, heterogeneous architectures designed at the processor-memory level.

Today we know very little about the nature of these future architectures, let alone the design methodology. We will instead need what I call the design research center concept. The research center's goal will be to perform cross-disciplinary system design research to create new methodologies, tools, libraries, and courses—distributed via the Internet—to produce enough SOC architects worldwide.

SOC and paradigm shifts

Major paradigm shifts in the electronics industry will occur due to rapid progress toward 0.1-micron technology. These paradigm shifts—which will lead to a methodology crisis if we are unprepared—have a major impact on the skills needed from future SOC architects.

Impact of the silicon board. Even if we started a new education program for SOC architects today, its first PhDs would not graduate until 2005. In 2005, deep-submicron technology will have reached the 0.1-micron level, leading to 6-cm² dies with more than 200 million transistors.³ What will such a

The megatransistor designs of the next millennium will require a new breed of system architect. What skills will designers need, and how will designers acquire them? A design research center, melding the best of industry and academia, may be the answer.

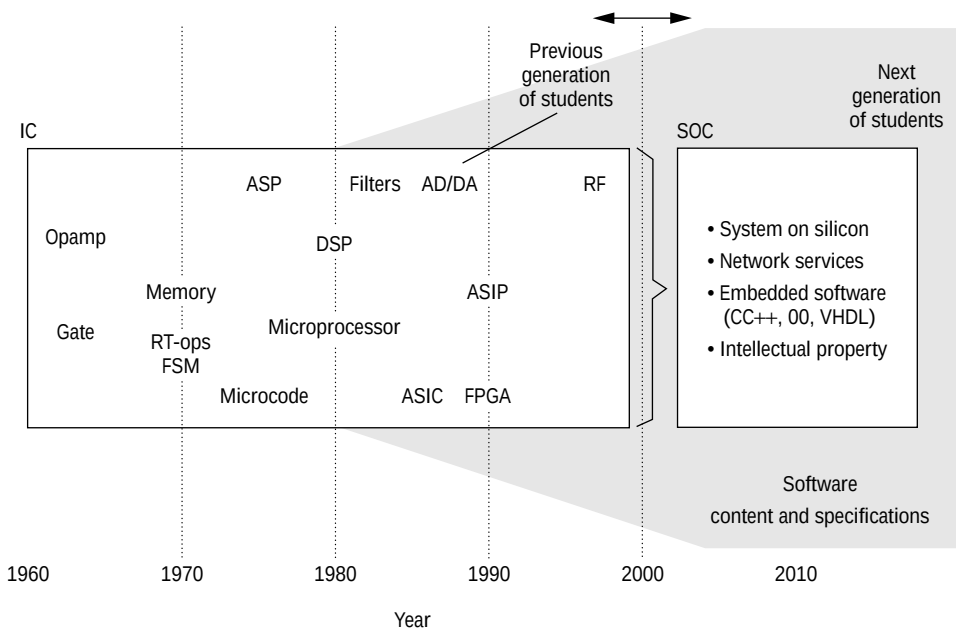


Figure 1. Evolution of the chip from a stand-alone component to a silicon board merging all design paradigms on the same carrier. The future silicon board will be embedded in a global network providing new services to the information and communication technology (ICT) consumer.

chip architecture look like? What products will they represent? Will the design be affordable? What design skills will be required? Figure 1 shows some of these aspects, including the anticipated dividing line between student generations.

Until now, integrated circuit (IC) designers were taught to design stand-alone hardware components such as ASICs, instruction-set processors, memory, FPGA, analog, and even RF CMOS chips. Engineers would then design these components into systems, systems and components being clearly separate. This situation is rapidly changing because, as Figure 1 shows, today's 0.25-micron technology lets us merge components as hard or soft intellectual property (IP) blocks on the same silicon substrate. The substrate then becomes a huge silicon board. Now a chip is no longer assembled at the gate level but at the IP block level and IP interfaces.

Hardware-centric design, as Figure 1 shows, is evolving to software-centric design. Soft IP blocks are described in hardware description languages (HDLs) while programmable IP blocks have their specifications implemented in real-time embedded software. Additionally, SOC specifications are now coming from ICT systems engineers at a level far above the HDL RT-level descriptions used by today's chip designers.

An SOC thus encapsulates system knowledge in a single medium—silicon—which blurs the separation between component design and system design. We must educate system

architects to think in terms of IP-based silicon design.

Architectural gap. One of the main bottlenecks to SOC design is the *architectural gap*—the widening gap between the software-centric system description and the system implementation in new silicon architectures.

Until recently, semiconductor technology drivers were expensive PC processors. Designers continuously increased clock speeds and designed DRAMs with ever-higher capacity to keep pace with memory-hungry software. But the Internet is rapidly changing this situation. Business is shifting from the expensive PC to inexpensive, broadband communication Web gadgets.

Higher performance servers, routers, and high-speed interfaces will always be needed, but these do not provide the production volume needed to fill the fab lines. The demand for the huge production volume of SOC will instead come from inexpensive Internet appliances in homes, small offices, and third-generation mobile communication devices.

For the home, three main application domains for high-volume production of SOC are emerging.

- **Set-top box.** An intelligent terminal that connects to the appropriate broadband Web channel, the set-top box will function as a base station for the home local-area network. The box, reconfigurable by Java agents, will adapt to the in- and outgoing information type and broadband source. The box will provide a wired LAN connecting to digital PCTV and inexpensive Internet appliances such as the Internet phones that will gradually replace today's expensive PCs.
- **Digital PCTV.** This will provide interactive HDTV services, via the new, object-oriented MPEG 4-7 compression standard⁴ permitting concurrent use of regular TV, interactive 3D graphics, and Internet services. The total computational power will range between 100 and 700 gigaoperations per second.⁵ Memory bandwidth will exceed 8 gigatransfers/sec while inexpensive packaging costs will restrict power consumption to below 10 watts.
- **Set-top box LAN service.** The set-top box will also serve a

multiuser wireless LAN (WLAN) with more than 100 Mbits/s of shareable bandwidth to provide data, speech, and images to home Internet appliances. To efficiently use the shared frequency spectrum, sophisticated modulation techniques might require computational power of roughly 10 gigaoperations per second (GOPS), while portability imposes power far below 1 watt.

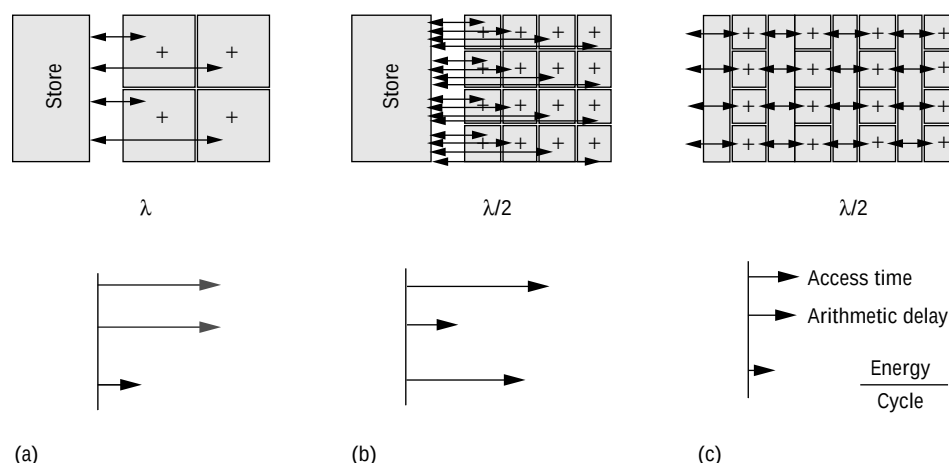


Figure 2. Energy-efficient computing requires on-chip distributed computing by localizing computation and storage.

Such consumer SOC's will require computing platforms significantly different from today's general-purpose PCs. First, SOC power efficiency must exceed 10 GOPS/watt—about two orders of magnitude better than that of current programmable microprocessors. Second, such systems are mixed-signal systems containing RF front-ends, high-speed A/D conversion, high-performance DSP stream processing, and complex control for protocol setup and mode control. Third, increased reuse of IP blocks and a much greater embedded software content are mandatory to reduce nonrecurring engineering (NRE) costs, redesign risk, and excessive time to market.

Embedded software itself will require multiprocessor, configurable computing platforms, which, for power consumption and cost reasons, will necessarily differ from today's platforms. This situation results from today's "infinite resource" software, which leads to the memory bandwidth and power crisis of classical microprocessor computing architectures.⁶ Infinite research software is written without concern for the underlying hardware limitations.

Future design solutions will be more domain-specific, and an inexpensive, combined memory-logic CMOS technology may emerge to implement low-power consumer products. That same CMOS technology⁷ or some form of chip-package codesign⁸ will also implement RF parts for local wireless access on the same silicon carrier.

Beyond 2007, silicon technology will likely hit some economical barriers at the 0.05-micron level. Because of its overwhelming integration potential, the problem will no longer be hardware design but software productivity. Lewis has said, for instance, that software productivity increases at 4.6% per year versus 21% per year for hardware.⁹ IP reuse may increase hardware productivity but leave us with the worsening problem of filling RAMs with reliable embedded software.

Bridging the architectural gap requires the availability of well-educated designers, which presupposes that faculty is available with experience in a system design methodology yet to be invented. The latter, in turn, supposes we can organize system-level design at the university level. I address this problem later.

Physical gap. As specifications move to higher abstraction levels, the gap seemingly widens from the bottom up. Deep-submicron physics indicates that wires, not transistors, dominate power and performance. This leads to the so-called timing-closure problem for digital systems requiring layout-driven logic and RT-level synthesis.

Lower operating voltages and higher current spikes lead to challenging, signal integrity problems and impose hard limits on the dynamic range of analog and RF components—this is the *physical gap*. However, the architectural gap, because it's a creativity gap requiring research in new methodologies lacking support tools, will likely be more severe.

To overcome the more-generic physics gap, the major CAE vendors, IP providers, and the VSIA are embarking on a massive effort. Evidence indicates that, by 2002, tools and IP libraries will emerge that reestablish the link between RT level and deep-submicron layout structures. The strongest impact of deep-submicron physics will be on the SOC architecture itself if we want to exploit silicon's intrinsic energy efficiency.¹⁰

Figure 2a shows a processor with a central memory and four computational units. Scaling the technology by a factor of 2, as in Figure 2b, yields potentially four times more computational units that potentially operate twice as fast. However, that quadruples data traffic to the central memory, per clock cycle. Since energy consumption in instruc-

tion-set computing primarily goes into memory transfers,¹¹ per-cycle energy increases at least four times.

Moreover, large central-memory access delay does not scale with gate delay, and power consumption goes up with size.⁶ To exploit performance and reduce energy consumption, designers must abandon the uniprocessor and central-memory concepts. As Figure 2c shows, energy efficiency requires locality of computation and storage, which leads to on-chip distributed computation on multiprocessor architectures.

Although such architectures are harder to program, especially for general-purpose programming, they are a better match for stream-based data processing of speech, images, and multimedia-system packets. The improved match is due to the multidimensional data and to the nested-loop structure of the code to be compiled. Next-generation SOC designers must bridge the architectural gap between system specifications and energy-efficient, IP-based architectures, while CAE vendors and IP providers will bridge the physical gap.

Design challenges

The megatransistor SOC of 2005 will likely be a self-contained system communicating directly with broadband wireline or wireless information services. It will probably also interact with users and appliances through multimedia transducers such as microphones, gloves, goggles, and other displays. It may contain a 5-GHz RF CMOS front end and A/D interface. Other predictions are these:

- Half of the chip can be filled with 200 low-power, RISC-like processors interconnected by field-programmable buses, embedded in 20 Mbytes of distributed DRAM and flash memory. Such memory-dominated chips will require memory or logic to be embedded, in view of multimedia stream processing and embedded software's dominance.
- A 20-Mbyte memory is small for running today's infinite-resource PC software, which underscores the need for efficient, real-time embedded software production techniques for heterogeneous systems. This discipline, however, is not widely taught today. Computational power will probably not result from multi-GHz clocking but from parallelism, with a clock frequency below 200 MHz. This will greatly simplify the design for correct timing, testability, and signal integrity, as well as design automation related to the physical gap.
- The SOC may still have room for one-million-gate reconfigurable computing architectures and one-million-gate hardwired logic. Configurability will significantly reduce design risks and increase the production volume of the chip for which NRE costs will become dominant.

All these predictions will lead to a computational power of 50 GIPS for programmable components or 500 GOPS for dedicated hardware architectures. Such chips will operate at the 1-volt level; power consumption should be restricted to the watt range.

Bridging the architectural gap. Bridging this gap will require skills learned from practice, as methodologies do not yet exist.

Product reliability. To functionally describe a 200-million-transistor chip at the VHDL RT level would require an estimated 2.5 million lines of code containing 250 bugs and requiring 20,000 staff-years to develop. Therefore, to maintain product reliability, designers must design at a level far above the RT level, with reuse factors in excess of 100.

Because SOC design reliability will be a major roadblock, SOC architects must be educated in production techniques for high-quality software and in the use of formal verification techniques. An SOC chip must be flexible enough to survive several product generations and evolving industry standards.

Trade-off. The biggest SOC design challenge may be the conflict between flexibility and energy efficiency. Computational efficiency can be expressed in MOPS/watt. Several researchers have identified the enormous energy overhead of instruction-set computing; Claasen, for example,¹⁰ stated that, in 0.1-micron technology, a programmable microprocessor will reach about 100 MOPS/watt while the hardwired computational power of silicon reaches 100 GOPS/watt. Reconfigurable computing lies somewhere in the middle; programmability is in the interconnect between a large number of concurrently operating computing nodes with a very restricted instruction set.¹² In the Pleiades project at the University of California, Berkeley, such architectures have a computational efficiency between instruction-set programming and hard-wired logic.¹³ To combine flexibility and energy efficiency, we must map different system behaviors into a flexible, energy-efficient architecture that combines multiprocessors, reconfigurable computing, hardwired logic, and analog interfaces.

Software for low power. Memory-intensive multimedia applications are characterized by concurrent tasks containing code with a highly nested loop structure. In such applications, power dissipation is dominated by memory accesses. Appropriate transformations of both tasks¹⁴ and loop structures¹⁵ tremendously affect the energy efficiency of hardware and of software implementations. Task transformations can reduce the size of memory buffers by an order of magnitude while loop transformations can lead to much higher

temporal and spatial locality of data. In fact, some researchers have reported power reductions of up to two orders of magnitude.¹⁵

Behavior-platform-compiler codesign. Trends show that system specifications, made at a much higher level than HDLs, will soon be described with an object-oriented, multithreaded language, based on C++ or Java.¹⁶

Energy-efficient platforms, on the other hand, will be heterogeneous mixtures of multiprocessors, reconfigurable computing, hardwired logic, and distributed memory. These platforms must be codesigned with the high-level-language system specifications, describing their hardware and software implementation. Besides the transformations for low energy, we need to develop techniques for efficient codesign of compilation techniques to map specifications efficiently on these novel architectures, which is another challenge to electrical engineering and computer science.

Global system engineering. An SOC encapsulates highly specialized system knowledge into a customized computing platform. Experience shows that such systems must be globally optimized within the environment—network or service level—in which they operate. For example, designing a single-chip wireless transceiver requires an understanding of interdependencies between RF-analog-digital-software architectures and the properties of the network into which it is to be embedded. Correct partitioning determines the ultimate power, feasible dynamic range, final area, packaging cost, time-to-market, and hence economic product viability.

Global system engineering is exemplified in chip-package codesign. Especially in future wireless system designs, single-chip SOC solutions will probably not be as cost- and energy-efficient as multichip module solutions for signal integrity, yield, and technology reasons. On these MCMs, one can integrate high-quality passive RF components, and mount separate RF front-end chips, MEMS, and DSP chips on them using flip-chip technology.

Nontechnical skills. This, too, will be a design challenge. Complaints are common in industry about engineering students' lack of project management, teamwork, and communication skills. The complexity and cross-disciplinary nature of SOC architectures can only be mastered by a team effort and the application of concurrent engineering principles. Therefore, faculty must educate SOC architects as team players rather than as individual contributors, however brilliant they may be.

Summary. Multidisciplinary system thinking is required for future designs. Academically, this can be achieved by submerging students and faculty into actual system-design projects to formalize and encapsulate experience into

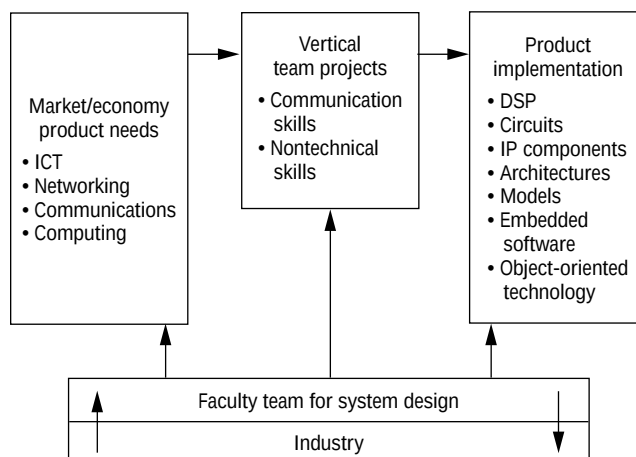


Figure 3. The training of the SOC architect creates a bridge between product needs in ICT and their implementation in heterogeneous software-hardware architectures within socioeconomic constraints. Design and nontechnical skills result from major cross-disciplinary projects managed by a professor team, which derives its vision from interaction with industry.

reusable methods, libraries, and tools shared by the entire educational community. We must take an interdisciplinary approach.

SOC architect option

Today's electrical engineering and computer science departments deliver either the classical VLSI designer or computer specialist. To create a new breed of engineers—the SOC architects—I propose an option that lies between the traditional VLSI design and computer science options.

Figure 3 illustrates how product needs are bridged with product implementation through vertically integrated team-based design projects.

The combined electrical and computer engineering education envisioned here provides the basics for the eventual product implementation skills. The computer engineering component must be significantly increased, in anticipation of increased software content of future system designs. The electrical engineering kernel should focus on analog and digital signal processing for signal and data processing and their link to power-area-time trade-offs.

The computer engineering component should have three main ingredients: real-time embedded software linked to multiprocessor architectures; computational models¹⁷ for specification and formal verification; and object-oriented programming and modeling techniques for tool development and specification. Team teaching, along with modern CAD tools and IP libraries, should be the bootstrapping mechanisms to focus on the architectural level.

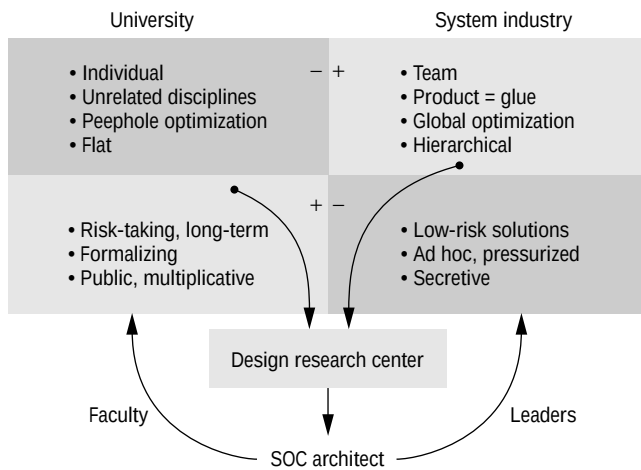


Figure 4. Comparison of university research to industrial system design. Combining the best of both worlds leads to the concept of a design research center.

Starting in the third year, the education emphasis should shift toward ICT by introducing the basics of digital communications and multimedia processing, and networking as well as computing architectures. Gradually, students will learn to reason from circuit to architecture to ICT subsystem and finally to global ICT system level.

Training must be oriented to problem solving, best learned through the vertically integrated project component. Students would operate in multidisciplinary concurrent engineering teams at the system level rather than the superindividualistic layered education typical today. Projects must evolve from analysis at the undergraduate level to open-end product design at the graduate level.

Such training must include communication skills and reporting. Projects must therefore have a management structure in which students and faculty play “leader and team” roles. Support from industrial psychology and the economics department might be helpful. Entrepreneurial principles must also be taught: Students should be encouraged to create start-ups in innovative products resulting from research.

Our future SOC architects must be able to embed, formalize, and document design expertise, methodology, libraries, and algorithms in well-conceived, reusable software objects. These objects should be made available over the Internet such that evolving design knowledge is available for bootstrapping the next generation of students, which would reflect the trend of making IP available over the Web.

The Web availability of IP also points to the important role of the semiconductor and computer-aided engineering (CAE) industries in making IP and software inexpensively available to these design-centric schools. In return, the added value of academic IP is available to industry, and it

assures that graduating engineers will have the desired skills.

Designing such an educational curriculum presupposes that faculty have a global view of system design and are prepared to team up to organize and manage vertically integrated teaching projects. The teamwork such a view implies would be a radical departure from the present academic reward system that favors individualism.

Design research center

Is system design research even possible in a university setting? The fundamental obstacle to such research in today’s engineering school is that the school’s professors and students act as individuals. Research domains are horizontally structured, with little attention to how they interact vertically, concerning design and product creation.

Furthermore, the complexity of global systems exceeds the research domain of a single professor. A university’s flat organizational structure ignores the possible impact of a professor’s research on strategically important problems at the system level, and the structure discourages cross-disciplinary research. Few engineering professors have experience in system design or have participated in industrial system design projects. In contrast, a complex product in industry can only result from teamwork within a clear, cross-disciplinary management structure.

The answer is to blend the best characteristics of the university with the best characteristics of industry. In its favor, university research has a risk-taking, long-term perspective. Industry, however, is under market pressure and therefore inclined to keep the risk under control and hence take only incremental steps. Universities can formalize methods whereas industrial research, pressured by time, tends to seek ad hoc solutions. These solutions are not made available in the public domain and hence do not create common knowledge, which is needed for faster progress. Universities, however, make methods and techniques available in the public domain and multiply knowledge through education of experts who can apply it.

So, can schools conduct system design? Yes, if they are willing to combine the best of both worlds, as Figure 4 shows. The result is a new system research organization called the design research center.

In such a center, professors form a team focused on a long-term strategic system design problem. The team accepts a management structure with the goal of performing research on a long-term, cross-disciplinary system basis, from specification to prototype implementation. An example long-term problem might be to investigate future wireless system architectures in below 0.1-micron technology. The problem would take a few generations of graduate students to solve. The center would then widely disseminate the formalized results.

As Figure 5 shows, a design research center is conceptually positioned between affiliated universities and industries. Ideally, it would be physically located on or near a major campus. A research team would have sufficient critical mass to tackle all tasks in the vertical design flow and ultimately solve the system design problem.

The team of faculty, students, and industry representatives would come from the affiliated universities and industries.

Recently, a number of such centers have emerged such as ALBA,¹⁸ Yamacraw,¹⁹ Berkeley Wireless Research Center,²⁰ Gigascale Silicon Research Center,²¹ Eindhoven Embedded Systems Institute,²² Broadband Access Project IMEC,²³ Human Computer Interface Institute CMU,²⁴ and so on.

These centers are characterized by long-term, vertically structured, system design projects. The projects result from the co-ordinated efforts of cross-disciplinary research teams with strong systems industry interaction. Rather than by traditional theorem-proving methods, concepts are proven by working demonstration models and publicly available software.

The design research center's mission is to produce the technical leaders of tomorrow by immersing faculty and students alike in daring system-level projects defined by academic and industrial visionary leadership. The goal is that "PhD students must become true leaders of tomorrow rather than super master degree holders of yesterday."²⁵

Design centers such as these should have the following characteristics:

- Research leads to the demonstration of the feasibility of a global system concept at the leading edge of technology. Research is theme-driven and 5 to 10 years ahead of current practice. Besides producing SOC architects, the research goal is formalizing design methodologies and then encapsulating them in reusable software. Direct technology transfer is not the goal. Actual transfer takes place through people trained for technical leadership, ready to create new products either in start-ups or in the affiliated major system houses or CAE companies.
- Faculty teams are committed to share and build knowledge to reach a common goal such that "concepts and ideas rub and spark against each other to create new approaches and new solutions."²⁴ This may not be consistent with today's science-dominated publication policy but it does help to produce the right people. It also requires a deeply respected visionary academic—a project champion—to lead such a center.
- Design centers also must have a strong industrial affiliate program to keep research goals realistic and generate long-term system requirements by visionary dialogues. Moreover, such a program entitles the industry to prototype background and foreground research results and

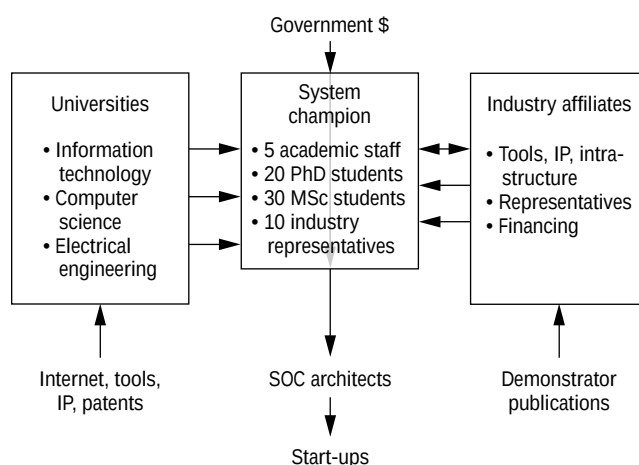


Figure 5. The concept of a system design research center and key features.

a direct contact with the center's most important product: well-trained SOC engineering students. Industry representatives ideally would participate in project research; in fact, they could play a team leader role for the PhD researchers in the design centers.

Such a design center reflects the modus operandi of the industry in which the engineering students will likely work after graduation. It provides the framework for global thinking by induction from their team professors. This does not, however, mean that smaller universities are excluded from such a system. Instead, I anticipate that vertically structured system projects would generate many fundamental open problems requiring in-depth, focused research that suits a traditional university structure. The design centers could announce such open problems on the Web, and smaller universities could then obtain funding to provide solutions to these problems.

The Internet eliminates geographical barriers. Just like virtual companies, virtual engineering schools are possible. Networked sharing of educational IP of the best schools will contribute to a better education, a better use of education funding, and faster technological progress. System design research centers will

- create faculty who understand the global system issues
- enable faculty to put horizontal disciplines in a vertical context
- attract more students as projects near reality
- better prepare students for industry

SOC education program

Figure 6 shows a tentative MSc and PhD program within the design center context for ICT-SOC training. A five-year

MSc program is a prerequisite. The PhD program takes place in the design research center.

In the SOC curriculum, regular courses are restricted to basic concepts and are based on team teaching. Team teaching focuses on engineering aspects and context. As the arrows in Figure 6 indicate, at least 30% of the time is devoted to project work, most of which is interdisciplinary.

The first two years are analytical and should provide the scientific basis to engineering. Basic science should be taught by engineers for engineers. In addition, the basics of both electrical and computer engineering are taught. Basic science is linked to engineering by having 25% of the time devoted to product analysis seminars. These seminars are hands-on, in which basic science is related to practice by analyzing aspects of popular ICT products (for example, CD players, modems, and digital videodisks). Engineers in industry are invited to explain the global design aspects, economics, and marketing issues of such products.

Because software takes on such an important role in future information systems, the curriculum should provide a balance between electrical engineering and computer science. Computer science should have an electrical engineering orientation; for example, it should focus on

- embedded software: compilation and real-time aspects, interface synthesis, parallel processing
- software impact on power dissipation, memory architecture, and so on
- computational models (data flow, control flow, and concurrency)
- object-oriented programming as a means for system modeling, knowledge reuse, and tool development

- the use of networking as an engineering tool
- the use of commercial and university CAE to focus on architecture

During the last three years of the MSc program, the electrical and computer engineering subjects gradually give way to the ICT subjects. In the last year, special attention is paid on implementing ICT systems in software-hardware architectures, as that is the final goal of the design center education.

The emphasis of the master program is on the ability of problem solving, both individually and in teams. At least 25% of the time, therefore, students work on interdisciplinary design projects. Students get involved in increasingly complex design problems from specification to implementation. A management structure involving both faculty and students from the system design research center should run these projects. Concurrent-engineering principles, teamwork, and communication skills are thus introduced in practice. The master thesis is devoted to a more in-depth design task (including tool development, if appropriate). A team of two or three students supervised by a PhD student working in the design center can prepare a master thesis.

The PhD program is devoted to research within the theme project of the design center. PhD students would focus on aspects of large system design problems, whereby emphasis is placed on combining engineering disciplines. The PhD program should be oriented to create enhanced design productivity by demonstrating new design methodologies. Such methodologies should be documented by reusable software available over the Internet. This form of publication may turn out to be more useful than the overproduction of academic papers of which the impact cannot be easily tested. Methods,

tools, and libraries shared by researchers would naturally be selected. Their acceptance rate can be a new and, from an engineering viewpoint, more interesting "citation index."

DEEP-SUBMICRON TECHNOLOGY, combined with the tremendous growth of global networking, requires a rethinking of computing and communication architectures and of new multidisciplinary design methodologies. In turn, new methodologies create the need for SOC architects able to bridge the gap between

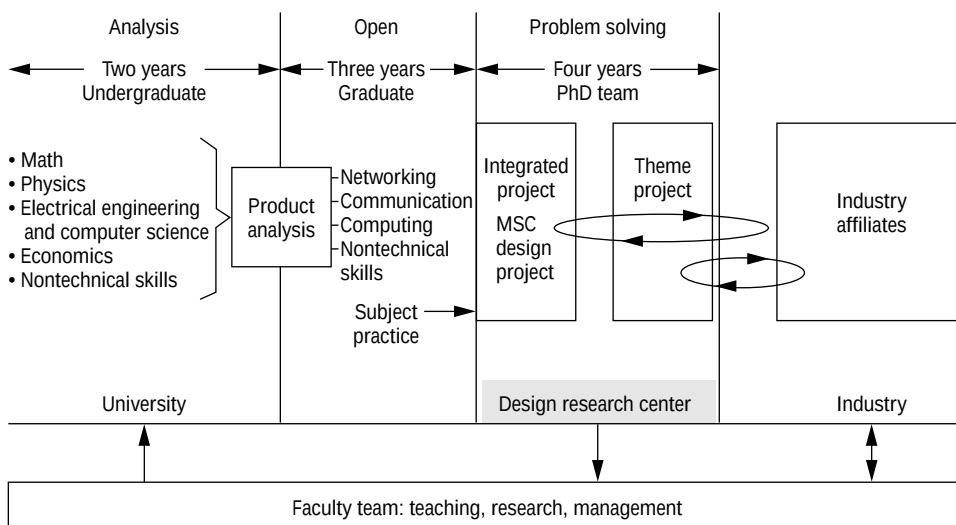


Figure 6. Tentative five-year curriculum of a design-centric school for the ICT-oriented SOC architect.

system-level concepts and their implementation. A design research center is the ideal environment for educating the SOC architect. In such schools, emphasis is on broad education in the basics of computing, communications, and components. SOC architects learn design practice (including nontechnical skills) by team-based integrated projects in product design. 

References

1. P. Chatterjee, "Gigabits: Deliver Affordable Digital Multimedia for Work and Play via Broadband Network and Set-Top Box," *Digest of Tech. Papers ISSCC 95*, IEEE Press, Piscataway, N.J., 1995, pp. 26–30.
2. J. Borel, "Technologies for Multimedia Systems on a Chip," *Digest of Tech. Papers ISSCC 97*, IEEE Press, 1997, pp. 18–21.
3. *The National Technology Roadmap for Semiconductors*, Sematech Inc., Austin, Tex, 1994.
4. <http://drogo.cselt.stet.it/mpeg/standards/mpeg-4/mpeg-4.htm>
5. H. Nakatsuka, "The New Frontier Created by High-Bandwidth Digital Video Systems and Services," *Digest of Tech. Papers ISSCC 99*, IEEE Press, 1999, pp. 16–19.
6. D. Patterson, "Intelligent Ram: Chips that Remember and Compute," *Digest of Tech. Papers ISSCC 97*, IEEE Press, 1997, pp. 224–252; <http://iram.cs.berkeley.edu/publications.html>.
7. T.H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press, Cambridge, UK, 1998.
8. J. Lin, "Chip-Package Codesign for High-Frequency Circuits and Systems," *IEEE Micro*, July/Aug. 1998, pp. 24–32.
9. T. Lewis, "The Next 10,000 Years," *Computer*, Part I: April 1996, pp. 64–70; Part II: May 1996, pp. 78–86.
10. T. Claasen, "High Speed: Not the Only Way to Exploit the Intrinsic Computational Power of Silicon," *Digest of Tech. Papers ISSCC 99*, IEEE Press, 1999, pp. 22–25.
11. V. Tiwari et al., "Instruction-Level Power Analysis and Optimization of Software," *J. VLSI Signal Processing*, No. 13, Kluwer Academic, Boston, 1996, pp. 223–238.
12. A. DeHon, "Trends Toward Spatial Computing Architectures," *Digest of Tech. Papers ISSCC 99*, IEEE Press, 1999, pp. 362–363.
13. Pleiades project; http://bwrc.eecs.berkeley.edu/Research/Configurable_Architectures/
14. F. Catthoor et al., "Proposal for Unified System Design Metaflow in Task-Level and Instruction-Level Design Technology Research for Multimedia Applications," *Proc. 11th ACM/IEEE Int'l Symp. System-Level Synthesis*, ACM Press, New York, 1998, pp. 89–95.
15. K. Danckaert et al., "Platform Independent Data Transfer and Storage Exploration Illustrated on a Parallel Cavity Detection Algorithm," *ACM Conf. Parallel and Distributed Processing Techniques and Applications*, ACM Press, June 1999.
16. P. Schaumont et al., "A Programming Environment for the Design of Complex High Speed ASICs," *Proc. 35th Design Automation Conf.*, ACM Press, 1998, pp. 315–320.
17. S. Edwards et al., "Design of Embedded Systems: Formal Models, Validation, and Synthesis," *Proc. IEEE*, Vol. 85, No. 3, IEEE Press, 1997, pp. 366–390.
18. <http://www.london.press.net/issues/8/58/pagefour.htm>, <http://www.sli-scotland.org.uk/>
19. <http://www.gcatt.gatech.edu/yamacraw/about.html>
20. <http://infopad.eecs.berkeley.edu/BWRC/>
21. <http://www-cad.eecs.berkeley.edu/GSRC/>
22. <http://www.eesi.tue.nl/english/eesihome.html>
23. <http://www.imec.be/vsdm/domains/dista/>
24. <http://www.cs.cmu.edu/afs/cs.cmu.edu/usr/hcii/www/index.html>
25. B.J. Sheu, Y.D. Lee, and W.C. Young, "Higher Education for the Information Age," *IEEE Circuits and Devices*, Vol. 14, No. 3, May 1998, pp. 23–28.

Direct questions about this article to Hugo De Man, K.U. Leuven, Kapeldreef 75, 3001 Leuven, Belgium; deman@imec.be.

The Road to Systems on a Chip Design and Test of SOCs: Creative Artwork or Engineering Effort?

The complexity and integration density of custom integrated circuits has increased dramatically, and the old paradigm of ASICs being a rather small part of a whole system is no longer valid. With system-on-chip (SOC) integration, new paradigms must be followed that may significantly change the way engineers design and test their systems. Will designing an ASIC remain an engineering effort, or will it be merely an artwork pulling together predefined and pretested intellectual property blocks?

- Will the usage of tools to design and test SOCs make high-qualified engineers obsolete?
- How will SOC design changes affect the job market and education?
- Will we see a migration of the work force to third-world countries?
- Where are the engineering challenges that could keep the qualification needs at a high level?

For continuing updates on the panel, technical program, and workshops, watch the conference Web site at <http://asic.union.edu>. For more information, contact the ASIC/SOC 99 Conference Office, Suite 270, 101 Lakeforest Blvd., Gaithersburg, MD 20877; phone: (301) 527-0900

Panel Discussion: September 15, 1999, 5:30-7:00 p.m.
Double Tree Hotel National Airport, Washington, D.C.
Sponsored by the
IEEE Solid State Circuits Society and IEEE Rochester Section

Coorganized by the 12th Annual IEEE International
ASIC/SOC Conference
and
IEEE Design & Test of Computers