The Heritage of Mead & Conway: What Has Remained the Same, What Has Changed, What Was Missed, and What Lies Ahead

By MARCO CASALE-ROSSI
Synopsys, Inc.

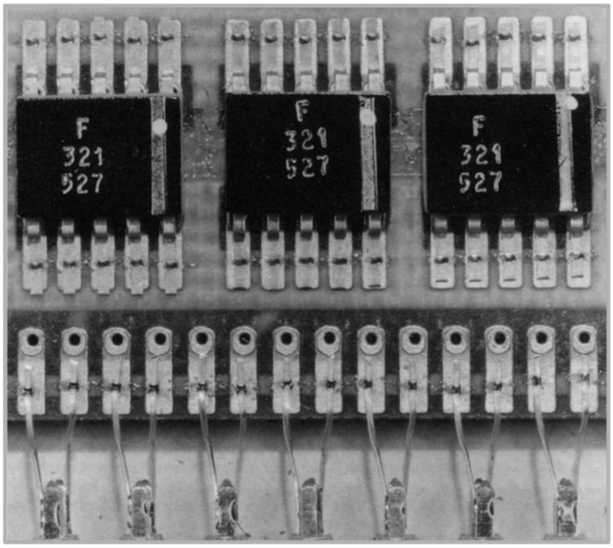

Fig. 1. *AGC logic module, detail (source: NASA).*

I. INTRODUCTION

I was an undergraduate student when I read *Introduction to VLSI Systems* [1] for the first time, in the early 1980s; most likely a copy brought to Italy by Prof. Alberto Sangiovanni-Vincentelli on the occasion of one of his summer visits to the Politecnico di Milano (Milan, Italy); "the book" introduced me to a world full of new ideas, and contributed to changing the voyage of my life, from mathematics and computer programming to electronic design automation (EDA). At that time, the students at the Electrical Engineering Department, Politecnico di Milano, were using KIC, a lambda-rules-based polygon layout editor devised by Giles Billingsley and Kenneth Keller, two Ph.D. students of Prof. Richard Newton at the University of California Berkeley (Berkeley, CA, USA) [2]. Running under UNIX BSD 4.1 on a Digital VAX 11/780 with a Ramtek 6211 Colorgraphics terminal, KIC was one of the first EDA tools inspired by Mead & Conway's methods, and was used to contribute designs to multiproject chips (MPCs) to be manufactured by STMicroelectronics, then SGS Microelettronica (Geneva, Switzerland), acting as a foundry service for various Italian universities and research centers—another of Mead & Conway's ideas that had quickly crossed the ocean. I joined STMicroelectronics Central R&D in 1985 to work on one of the first commercial incarnations of physical implementation.

I reread "the book" during summer 2012 while working on a customer presentation, and I thought that Mead & Conway's methods still looked incredibly modern. I wondered what had really changed in the last 30 years, if anything; whether there were fundamental misses in Mead & Conway's methods; and if and how they could help address the grand challenges and the "red bricks" that lay ahead.

These thoughts inspired me to organize a special session at the DATE 2013 conference in Grenoble, France.

Digital Object Identifier: 10.1109/JPROC.2013.2295871

0018-9219 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

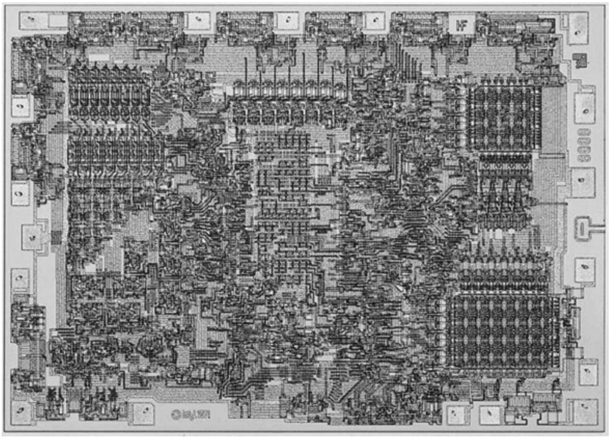

Fig. 2. *Intel 4004 (source: Intel Corp.).*

The title of the special session was "The Heritage of Mead & Conway: What Has Remained the Same, What Has Changed, What Was Missed, and What Lies Ahead." It was introduced and moderated by Alberto Sangiovanni-Vincentelli, with the participation of—in alphabetical order—Luca Carloni, Bernard Courtois, Hugo de Man, Antun Domic, and Jan Rabaey. My only regret is that Lynn Conway could not attend, as the special session ended up being one of the most successful of the conference; Mead & Conway's methods were still a hot topic, actually much hotter than I had imagined.

II. A GIANT LEAP...

Electronic engineering sprang from the electrical engineering tree in the late 19th, early 20th century, but it was not until after World War II that it really took off; in 1947 came the transistor, and in 1959 the integrated circuit (IC), indeed "a giant leap forward for mankind..."

In 1961, the NASA Apollo Guidance Computer (AGC) design (Fig. 1) was awarded to the MIT Instrumentation Laboratory (Cambridge, MA, USA). Manufactured at Raytheon (Waltham, MA, USA), assembling approximately 2900 just-invented Fairchild Semiconductor monolithic integrated circuits, each implementing two 3-input NOR logic gates, the AGC ran at 512 KHz, and consumed 70 W at 28 V; it also had a power-saving mode that reduced power consumption by 5×. Thus, the first digital computer was born. It was the first computer using ICs and the smallest computer ever—it weighed only 70 lb. Yet it was easily understandable to computer scientists: all the 24 logic modules consisted of 120 identical ICs assembled in a flat pack and connected via a wire wrap [4].

Ten years later, in 1971, Intel Corporation (Santa Clara, CA, USA) announced the 4004 (Fig. 2), the first single-chip microprocessor. Manufactured in 10-μm technology on 2-in wafers, it integrated approximately 2300 transistors and ran at 108 KHz. Quoting from Intel's first, 12-page annual report as a public company: "innovative design work by our engineers in 1970 and 1971 culminated with the introduction of MCS-4 and MCS-8, which are microcomputers built around one-chip Central Processor Units (CPUs) selling for less than \$100 each. The low cost of the CPUs will allow them to move into hundreds of product areas which never before could justify the use of a general purpose computer" [6].

While architecturally simple, the 4004 design implementation and fabrication details were obscure to most computer scientists. According to Lynn Conway, the question back then was "whether the design of VLSI systems would be possible outside Intel moving forward" [3].

The answer began taking shape in 1975 with a serendipitous meeting of visionary people from California Institute of Technology (Caltech, Pasadena, CA, USA) and Xerox PARC (Palo Alto, CA, USA): the Sutherland brothers, Lynn Conway, Douglas Fairbairn, Carver Mead, James Rowson, and David Johannsen. Their vision attracted many brilliant people who devised new designs and new design tools exploiting Mead & Conway's methods.

Lynn Conway said, "as in the case of the impressionists, who held their own separate exhibition in Paris in 1874 to suddenly bring their work forward and make it visible," [3] Mead & Conway's methods were suddenly brought forward in 1978–1980 and made visible through a set of courses across the country and abroad.

Concepts such as simplified design methods, new, electronic representations of digital design data, scalable design rules, "clean" formalized digital interfaces between design and manufacturing, and widely accessible silicon foundries suddenly enabled thousands of chip designers to create tens of thousands of chip designs. A completely new way of creating VLSI systems on silicon was born.

In 1981, when Intel announced the 286 (Fig. 3)—manufactured in 1.5-μm technology, integrating approximately 134 000 transistors, and running at 6 MHz—its layout showed a much greater regularity than that of its predecessors, evidence of the impact of Mead & Conway's methods. In its first ten years, the microprocessor had been "designed into more than 100 000 products," [7] and was showcased on the cover page and six out of 26 pages of the Intel's annual report.

In 2011, Intel announced the third-generation Core (Fig. 4). Manufactured in 22-nm technology, it integrates approximately 1.4-billion trigate transistors, runs up to 4 GHz, and consumes 77–130 W. The regularity of its layout is amazing, both at the floorplan and transistor levels.

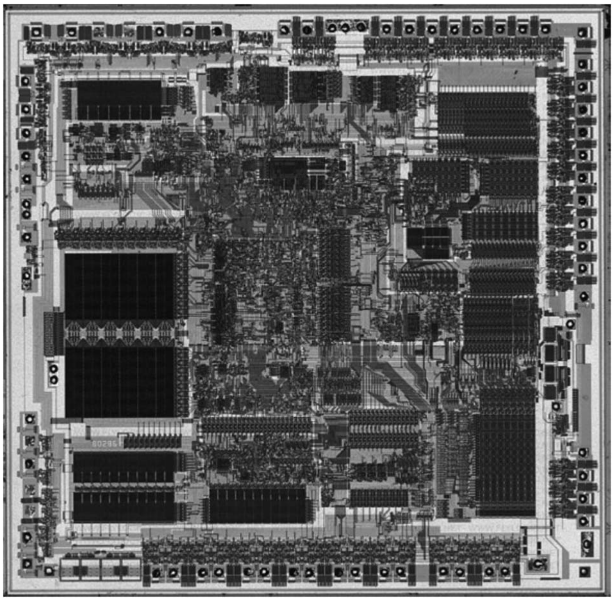

Fig. 3. *Intel 286 (source: Intel Corp.).*

III. THE HERITAGE OF MEAD & CONWAY

In the mid-1960s, the ICs used for the AGC project absorbed approximately two-thirds of the worldwide semiconductor industry capacity at that time. Today, as Moore's law—a term coined by Carver Mead—has brought us from 100 μm to 10 nm, and worldwide semiconductor revenue has passed the \$300 billion threshold, what is the heritage of Mead & Conway?

I believe that Carver Mead and Lynn Conway's vision and ideas have been a cornerstone of the IC revolution in three respects: what Alberto Sangiovanni-Vincentelli calls the "orthogonalization of concerns," i.e., the separation of design from manufacturing; the use of design methodology and design tools to cope with complexity; and finally, education at large. Using an often-abused term, Carver Mead and Lynn Conway gave birth to an ecosystem: "workers from many backgrounds, computer scientists, electrical engineers, and device physicists are collaborating on a common problem area" [1].

IV. SEPARATION OF DESIGN FROM MANUFACTURING

The concepts of silicon foundries and fabless design houses, and the separation of design from manufacturing were publicly introduced by Carver Mead in January 1979 at the first Caltech Conference on VLSI: "if innovation by a myriad of small groups and individuals is to carry us into the VLSI revolution, we must not expect these groups and individuals to provide their own fabrication facilities. The level of innovation required can be achieved only if fabrication is provided as a service by a few well-capitalized firms. [...] What is needed is a clean, standard interface between a multitude of small, diverse VLSI design groups and a few state-of-the-art fabrication suppliers" [8].

For this "VLSI revolution" to happen, however, the definition of the rules that designers had to comply with was required, so that their layout could be manufactured. These "design rules" needed to be independent of rapidly changing manufacturing processes as much as possible. For this reason, Lynn Conway proposed to express the "design rules in dimensionless form, as constraints on the allowable ratios of certain distances to a basic length unit [...] equal to the fundamental resolution of the process itself [...] all dimensions are given in terms of this elementary distance unit, which we call λ [...] a least common denominator likely to be representative of the design rules for a reasonable period of time," the lambda rules [1].

Lambda rules have been used to design the layout of ICs for more than a decade, from 3-μm *n*-channel metal–oxide–semiconductors (NMOSs) of the late 1970s to 0.7-μm complementary metal–oxide–semiconductors (CMOSs) of the early 1990s. Thanks to the lambda rules, layout engineers could move smoothly from one technology node to the next, and layout generators fueled productivity from a handful of transistors a day to several hundred transistors a day. Lambda rules sparked the development of the first generation of modern, commercial polygon layout and design rules checking tools.

In fall 1979, Lynn Conway and her colleagues at Xerox PARC organized a "live demo" of the full flow: they rapidly implemented and demonstrated it in a very risky, large-scale,

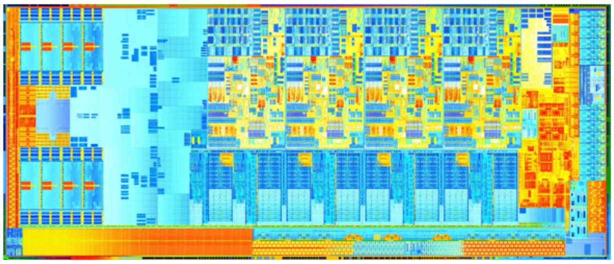

Fig. 4. *Intel's third-generation Core (source: Intel Corp.).*

entrepreneurial effort. One hundred twenty four designers from universities around the world contributed 82 chip designs in the CIF2.0 format. Hewlett-Packard (Palo Alto, CA, USA) agreed to donate wafer fabrication, and the packaged chips were shipped back to the designers in January 1980. MPC79 and then MOSIS were based on the Internet—then ARPANET—and contributions were sent electronically. As of today, MOSIS has fabricated more than 50 000 chip designs for hundreds of universities and research centers; silicon foundries have borrowed and expanded the concept of the multichip project; and today "taxis," "shuttles," and "trains" continue to make silicon accessible to startups, and small-/medium-size businesses.

The first modern silicon foundry, TSMC, was incorporated in 1987. Over the course of 25 years, it has become a $17 billion revenue company with an installed manufacturing capacity of 1.3-million 200-mm equivalent wafers per month. Silicon foundries have become a $40 billion industry at the heart of an $80 billion fabless semiconductor business. Collectively, silicon foundries and fabless semiconductor companies represent about 40% of the entire semiconductor industry, and the forecast is that over the next five years, they will double their revenue.

Today, we can design ICs that integrate hundreds of thousands more transistors than in 1981, but we went from Lynn Conway's "just two pages of very basic, easy-to-understand design rules" [3] to several thousands of complex design rules to be complied with, and it takes tens of thousands of lines of code to describe these design rules. Is mastering both design and manufacturing, after all, a strategic advantage again? Intel CEO Paul Otellini in his last earnings call commented that "in the first quarter [of 2013], we shipped our 100-millionth 22-nanometer processor, using our revolutionary tri-gate transistor technology, while the rest of the industry works to ship its first unit" [10]. Is the question, again, whether the design of VLSI systems will be possible outside a (shrinking) handful of IDMs moving forward?

Moreover, contrary to the common belief, this separation is far from being complete. While it is an (endangered) reality for "simple" CMOS processes, when it comes to analog and mixed-signal processes, silicon foundries and fabless companies still have a minority share. The complexity and fragmentation of process variants multiply the efforts and the costs required to set up the design infrastructure and the foundation IP that are the key building blocks of a foundry/fabless relationship. Finally, the separation of design from manufacturing never started for memory processes.

Complexity threatens the separation of design from manufacturing, an extremely valuable heritage of Mead & Conway that we should strive to preserve today and in the future. I believe that fighting complexity should be our industry's number one objective; quoting Alberto Sangiovanni-Vincentelli, "only by carrying along the least amount of unnecessary baggage at each step," [4] i.e., only by hiding all the inessential details, will we be able to preserve what we have achieved in the last 30 years. And, given that we cannot fight against the device physics that keep adding details, it is in the design methodology and in the design automation technology that we must find the solution.

V. DESIGN METHODOLOGY TO COPE WITH COMPLEXITY

"The first microprocessors produced by the semiconductor industry were fairly direct mappings of early generation central processor architectures into LSI. Such LSI systems, directly mapping data path and control functions appropriate in earlier component technologies, of necessity contained a great deal of random logic. However, the extensive use of random logic results in chip designs of very great geometrical and topological complexity, relative to their logic processing power. To deal with such complexity, system design groups have often stratified the design problem into architecture, logic design, circuit design, and finally circuit layout, with specialists performing each of these levels. Such stratification often precludes important simplifications in the realization of system functions" [1].

The introduction to the third chapter of "the book" contains both the diagnosis (the explanation of why the first microprocessors looked so simple and understandable at the block-diagram level and so complicated and obscure at the actual layout level) and the therapy: "the process of designing an LSI system is sufficiently complex that only by adopting some type of regular, structured design methodology can one have hope that the resulting system will function correctly" [1].

Carver Mead and Lynn Conway brought ideas from software programming into hardware design, such as abstraction, hierarchy, structure, use, and reuse of "common blocks" that have enabled modern VLSI systems design.

It has been argued that Mead & Conway's methods had missed the imminence of logic synthesis, place-and-route, and hardware design languages, which were just around the corner: "switching theory provides formal methods for minimizing the number of gates required to implement logic functions. Unfortunately, such methods are of little value in VLSI systems since the area occupied on the silicon surface by circuitry is far more a function of the topological properties of the circuit interconnections than it is of the number of logic gates implemented" [1]. Aart de Geus' intuition that logic mapping and minimization, albeit very elegant, were useless without timing led to Design Compiler; the move to three routing layers—which made river and channel routing suddenly obsolete, and "circuit interconnections" much more efficient—brought utilization, i.e., "the area occupied on the silicon surface by circuitry," from 40% to 80%; and finally, Verilog wiped schematic entry.

However, it would not be fair to call them misses: Carver Mead and

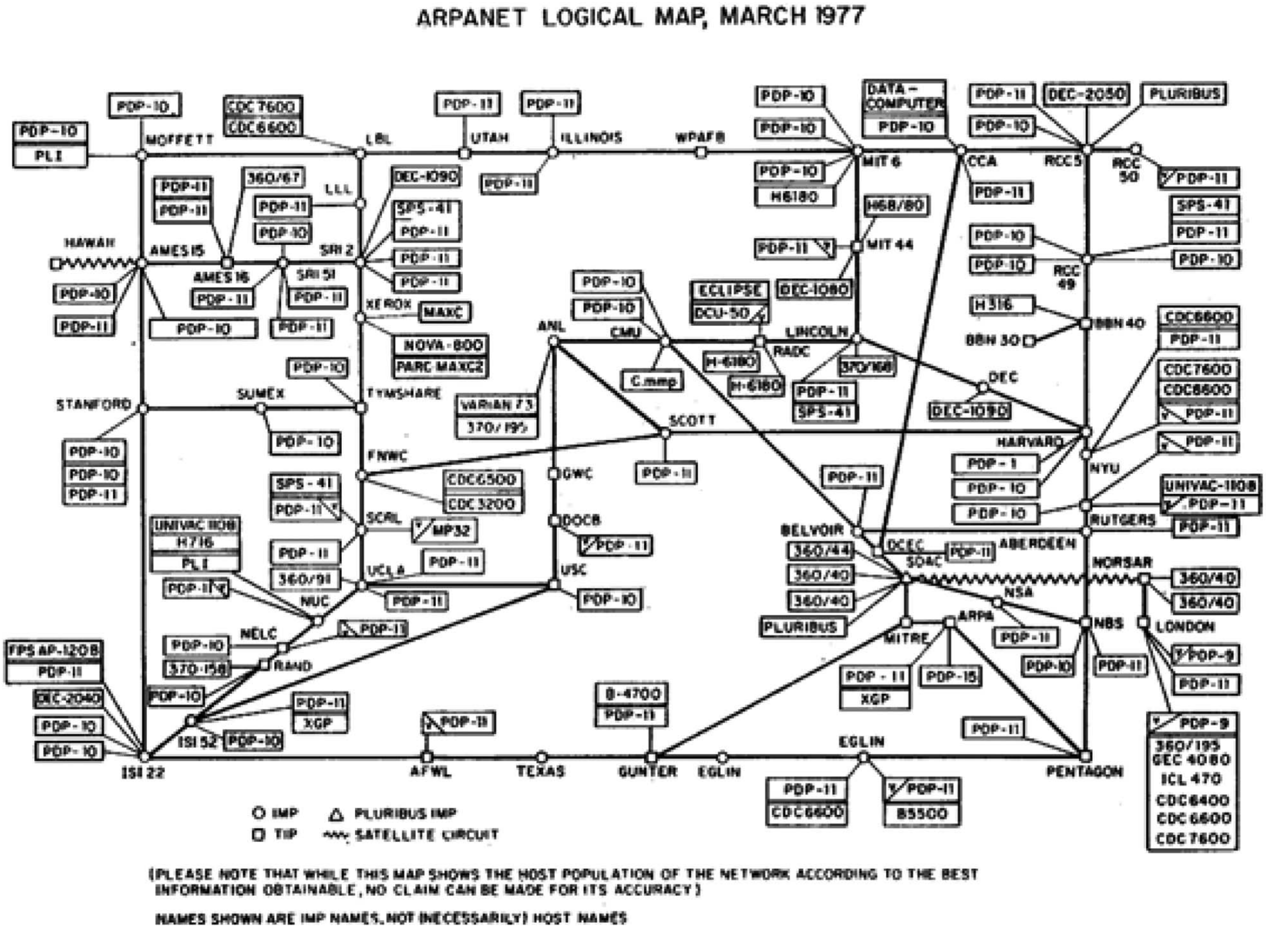

Fig. 5. ***Internet map (source: Computer History Museum).***

Lynn Conway were well aware of [some of] the limitations and hopeful that "the book" would eventually lead to their overcoming: "switching theory, not only yields the minimum number of gates to implement a logic function, but it also directly synthesizes the logic design. Unfortunately, at the present time there is no general theory which provides us with a lower bound on area, power, and delay time for the implementation of logic functions in integrated systems. [...] In the absence of a formal theory we can at best develop and illustrate alternative design methodologies [...]. We hope that the examples and techniques presented in this text will [...] stimulate others to join in the search for more definitive results" [1].

Even if the "tall–thin" engineer concept, as opposed to the "short–fat" one, was possibly brought too far, too early, and the engineer was made too tall, the "stratification" of the design implementation flow is still a challenge.

Until now, we have mostly "preserved" the gates, relying on place and route to fix "details" such as area, timing, power, etc. Unfortunately, the devil is in the "details," and fixing these "details" is more and more difficult, and leads to an infinite number of iterations and increasingly suboptimal solutions. Throwing constraints to the wolves is no longer an option; commitments taken blindly and/or too early cannot be honored, and decisions must be continuously refined and reconfirmed throughout the various steps of the flow.

Design implementation flows must evolve from traditional logic design and physical design subsystems into exploration and implementation subsystems which will share [some of] the tasks and differ by the strength. Over time, the number of shared tasks will increase as logic synthesis and place and route progressively merge.

The exploration subsystem should exploit fast synthesis, placement, and global routing engines to understand the design landscape, clean up the design constraints (which, by now, stay dirty for most of the design cycle), and perform feasibility analysis. The implementation subsystem should take over from the findings of the exploration one, and run full synthesis and place and route. Once the objective (area, timing, power, etc.) is within reach, the implementation subsystem should "preserve" placement and routing, and make heavy use of *ad hoc* "correction" techniques, by systematically changing the gates—the same footprint, different timing, power, temperature inversion points, etc. The richness of the library is fundamental: my first library, 3 μm, two routing layers, was made of 200 standard cells; today we are already seeing libraries of several thousands of standard cells, with scores of variants.

Today, design methodology and design automation progression can cure the ailments of complexity. However, there is another heritage of Mead & Conway besides keeping design and manufacturing separated and advancing design methodology to cope with the exponentially increasing complexity that is absolutely critical: education and collaboration; possibly the most important heritage.

VI. THE ROLE OF COLLABORATION AND EDUCATION

Luca Carloni said, "the heritage of Mead & Conway starts with a book that has been studied in hundreds of universities and influenced thousands of engineers" [4]. I might be wrong, but I cannot recall anything comparable to *Introduction to VLSI Systems* in the last 30 years. It is probably the right time to invest again in harvesting and educating the next generation of hardware designers and hardware design tools developers.

We are at the very first steps of the nonplanar CMOS era, and there are already several promising options for beyond CMOS; double patterning is the very first step and multipatterning will keep immersion lithography afloat as we wait for EUV. Bringing nonplanar CMOS and multipatterning from research to volume production requires a fresh look at the way we model and simulate circuits and interconnects; at the way we design our IP building blocks; and, finally, at the way we synthesize, place, route, verify, and sign off our designs, making the most out of these emerging technologies without getting overwhelmed by the new complexity they bring to us.

We are also at the very first stage of the heterogeneous systems era; thanks to increasing system integration, digital computing functions are complemented by scores of on-chip analog and mixed-signal functions, interfacing/interacting with people, environment, and other systems. Moving forward, micromechanical and nanomechanical systems, as well as biosystems, will be integrated.

The "Mead & Conway revolution" had a serendipitous start, when our industry was worth less than \$10 billion, and the Internet (Fig. 5) was connecting a handful of computers; what if the Sutherland brothers had not ignited it all?

Today, the semiconductor industry is worth more than \$300 billion, is at the heart of the \$1.4 trillion electronic industry, and the Internet has exhausted the available 2^{32} IPv4 addresses and is rapidly occupying the 2^{128} IPv6 addresses. We have gone from the Internet of computers to the Internet of things. I believe that another revolution is needed, but a more "collaborative" one.

Collaboration will be critical to overcome the challenges that lay ahead. By collaboration, I mean the willingness of all involved parties to work together, sometimes sharing the burden of debugging and improving the existing solutions, testing them again and again, accepting temporary workarounds while waiting for a new solution to become available, and always acknowledging that technology is only seldom perfect since its inception, and knowing that this joint effort will pay off.

According to Jan Rabaey's closing remarks at DATE 2013, "Mead & Conway was one of the cornerstones of the IC revolution, may the enduring ideas that resulted help enable the new wave" [4]. ■

Acknowledgment

The author is grateful to Antun Domic: it all started with their conversations on the "Design Challenges for the VLSI Chips." During the organization of the DATE 2013 special session, the many Skype video calls with Lynn Conway were critical for the author to understand and appreciate what happened during the "Mead & Conway revolution". Last but not least, the author would like to thank Alberto Sangiovanni-Vincentelli, Luca Carloni, Bernard Courtois, Hugo de Man, and Jan Rabaey for the energy and the enthusiasm they have invested in making their special session one of DATE 2013's most successful and well attended.

REFERENCES

[1] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA, USA: Addison Wesley, 1979.

[2] G. Billingsley and K. Keller, "KIC: A graphics editor for integrated circuits," Electron. Res. Lab., Univ. California Berkeley, Berkeley, CA, USA, 1981.

[3] L. Conway. [Online]. Available: www.lynnconway.com

[4] L. Carloni, B. Courtois, H. de Man, A. Domic, J. Rabaey, and A. Sangiovanni-Vincentelli, "The heritage of Mead & Conway," in *Proc. Design Autom. Test Eur. Conf. Exhibit.*, 2013, pp. 171–175.

[5] J. E. Tomayko, "Computers in spaceflight: The NASA experience," NASA History Office, 1988.

[6] Intel Corp., *Annu. Rep.*, 1971, p. 4.

[7] Intel Corp., *Annu. Rep.*, 1981, pp. 4–10.

[8] C. Mead, "VLSI and technological innovation," in *Proc. 1st Caltech Conf. VLSI*, 1979, pp. 15–27.

[9] L. Conway, "The MPC adventures: Experiences with the generation of VLSI design and implementation methodologies," in *Proc. 2nd Caltech Conf. VLSI*, 1981, pp. 5–27.

[10] SeekingAlpha, *Transcript of Intel Q1/2013 Earning Call*, 2013.

[11] L. Conway, "Reminiscences of the VLSI revolution: How a series of failures triggered a paradigm shift in digital design," IEEE Solid-State Circuits Mag., vol. 4, no. 4, pp. 8–31, Dec. 2012.

[12] C. House, "A paradigm shift was happening all around us," *IEEE Solid-State Circuits Mag.*, vol. 4, no. 4, pp. 32–35, Dec. 2012.

[13] C. H. Sequin, "Witnessing the birth of VLSI design," *IEEE Solid-State Circuits Mag.*, vol. 4, no. 4, pp. 36–39, Dec. 2012.

[14] K. Kuhn, "The ultimate CMOS device and beyond," in *Proc. IEEE Int. Electron. Devices Meeting*, 2012, Keynote Address.

[15] B. Vigna, "MEMS & sensors for the internet of things," in *Proc. Design Autom. Test Eur. Conf. Exhibit.*, 2013, Keynote Address.