

Reminiscences of the VLSI Revolution: How a series of failures triggered a paradigm shift in digital design*

By Lynn Conway
Professor of Electrical Engineering and Computer Science, Emerita
University of Michigan, Ann Arbor



Lynn Conway at MIT, October 2008, the 30th anniversary of launching her VLSI design course there.

Preface

Innovations in science and engineering have excited me for a lifetime, as they have for many friends and colleagues. Unfortunately, our wider culture often imagines the engineering life to be one of tedium and technical drudgery, seldom witnessing the joys of such creativity.

If only I could wave a wand, I've often wished, and say "YOU CAN DO IT" to inspire young folks to dedicate their lives to such adventures. But then various friends asked me to write about my own career – a tale wherein travails, setbacks, dark days and obscurity at times seemed the theme – and I wondered who'd be inspired by such a journey, so often apparently lonely, difficult and discouraging?

However, after deeper contemplation and review, I realized that each setback in my story, each hardship, actually strengthened my skills, my perspectives, and my resolve. And when colleagues began reading the early drafts, they reacted similarly: "Wow, this is really something!" The story was authentic, real – maybe even surreal – and it actually happened.

The child who once dreamed of "making a difference," indeed made a difference after all. And with that, I'd like to inspire YOU to imagine how you too can positively impact our world. Be assured, it won't be easy, and fame may never come your way, but the satisfaction gained from a life of creative work will be immense. Trust me on this!

* Copyright © IEEE 2012. A memoir published by the IEEE; posted online with permission. Citation: Lynn Conway, "[Reminiscences of the VLSI Revolution: How a series of failures triggered a paradigm shift in digital design](#)", *IEEE Solid State Circuits Magazine*, VOL. 4, NO. 4, FALL 2012, pp. 8-31.

Childhood Fascinations

I loved listening to the radio as a child during WWII, especially to BBC broadcasts from London. Thrilled by hearing people speak from far away, I wondered how this mysterious machine worked, with all the glowing tubes and strange-looking parts inside.

My father was a chemical engineer, and he gave me *The Wonder Book of Knowledge* as one of my first ‘big books.’ From it I learned not only how to read, but also how electricity was tamed and radios were created, and that engineers did these things.

Becoming fascinated by astronomy, math, physics and electronics, and encouraged to build things that worked, I was channeled to become an engineer. Among my heroes were Charles Steinmetz and Edwin Armstrong; I knew their stories well and dreamed of doing such things.

Steinmetz pioneered methods for calculating alternating current phenomena using complex numbers, complex exponentials and vector diagrams, simplifying a highly arcane field. His books and passionate teaching launched the AC revolution, and his story carried an embedded message: Someone who faced physical challenges (he was afflicted with hunchback and hip dysplasia) or who was somehow perceived as different might become liked, even honored, if they made valuable contributions.

Edwin Armstrong pioneered the regenerative and super-regenerative circuits, the super-heterodyne radio receiver and FM radio. His visionary inventions involved elegant arrangements of simple electronic components, and helped launch a revolution in radio.

Time and Place Are Everything

Just as Steinmetz had with electrification and Armstrong with wireless communication, I found myself a student at the beginning of a technological revolution: digital computing in the early 1960s. And, I was at the right place: Columbia University’s School of Engineering and Applied Science, with its close ties to IBM, then a leading force in the emerging industry.

Along with delving into every relevant course in math, physics, electrical engineering, and computing, I also did an independent study there with Dr. Herb Schorr, just prior to his joining IBM. I must have made a good impression, for I was quickly recruited by IBM Research and in 1965 found myself at the T. J. Watson Research Center at Yorktown Heights, working on a highly proprietary and secretive supercomputer project, a project unknown even to many within the company.

The Advanced Computing Systems (ACS) project had been personally launched by IBM’s then-CEO Thomas. J. Watson, Jr., and given the mission to “go for broke” to create the most powerful scientific computer in the world. Staffed with pre-eminent IBM computing experts of the time including the legendary John Cocke, the project soon moved to what would become *Silicon Valley* [1], [2].

Herb Schorr led ACS’s architecture department, where I worked on an architectural simulation model of the evolving hardware design. The initial design for the ACS-1 exploited cache memory, instruction pre-fetch, multiple pipelined functional units, and an innovative instruction set and branch hardware for anticipating and minimizing branch disruptions in instruction flow. There was a bottleneck in instruction issuance, however, and functional units often stood idle as stalled instructions awaited results.

Gene Amdahl, already famous inside IBM for his work on System 360, along with other prominent computer architects of the day, presumed that no single-stream architecture could be found that issued, on average, more than one instruction per machine cycle [3]. Cocke questioned this presumption, but no way had been found around the bottleneck – as yet.

Unaware that this was an open research question, I took it on as a design challenge and obsessed on it for over a month. I explored varying ways to represent and issue instructions, mentally juggling all aspects of the problem simultaneously – everything from mathematical abstractions, to architectural structures, to circuit-level implementations, but to no avail.

My First Invention

In the fall of 1965, however, it suddenly beamed down to me: By holding pending instructions in a queue, and representing source and destination registers and functional units in unary positional form rather than in binary, I determined that it would be possible to scan the queue, resolve dependencies, and issue multiple instructions out-of-order (OOO), even when various entries were stalled [3].

The scheme involved not only mathematical and micro-architectural ideas, but also tricks at the logic and circuit levels, using arrays of ACS high-speed emitter-coupled logic (ECL) integrated circuits and exploiting their ‘wired-OR’ connections to scan queue-columns within machine cycle-time constraints. An ACS colleague at the time, Brian Randell, coined a perfect name for the scheme, *Dynamic Instruction Scheduling* (DIS). It was quickly incorporated into the ACS-1 design [3], [4], [5].

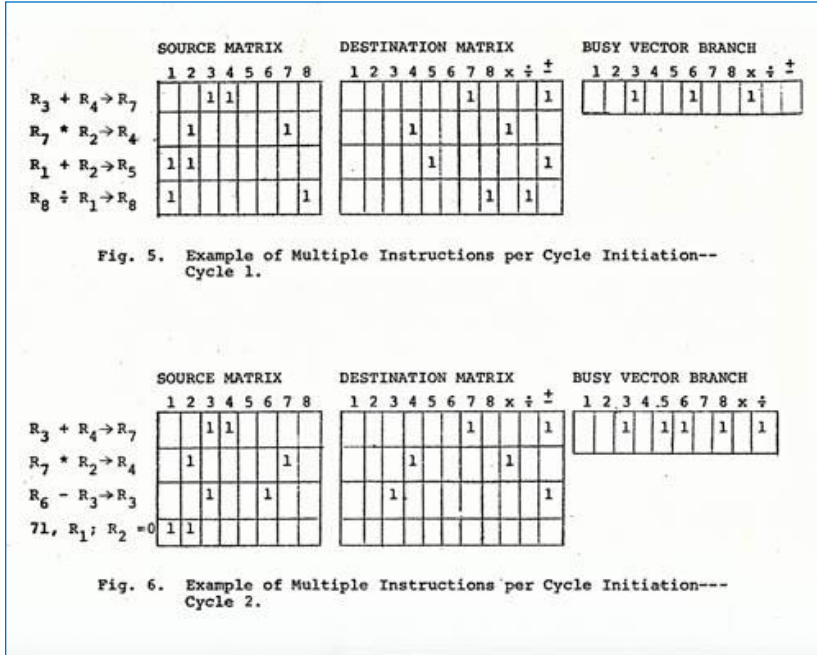


FIGURE 1: DIS Functional Diagrams.

DIS provides a sort of 'turbocharger' for pushing more instructions through a processor during each machine cycle than would otherwise be possible. Although huge regular arrays of ECL circuits were required to implement that 'turbocharger' for the ACS-1 (a moderate fraction of the main processor's total circuitry), the scheme proved simple and elegant in both function and structure, and more than doubled the machine's performance.

This was a personal Edwin Armstrong moment for me. I now knew what it felt like to invent something cool. In fact, DIS proved to be a fundamental advance in computer architecture and by a circuitous route has since become a standard fixture in modern high-performance microprocessors.

Lessons Learned

One might ask how could a shy, naïve, freshly-minted MSEE be the one to invent multiple-OOO DIS? The problem had been clear to others; why hadn't they found a solution?

The belief that it couldn't be done undoubtedly held back progress, while ethnographic observations reveal further problems: By the mid-1960s, chasms had developed between the various specialized groups working on computer architecture, logic design, circuit design, and packaging – with each specialty optimizing their work at a particular level of abstraction, and then tossing it over the wall to the next.

As a result, most computer architects lacked knowledge about the rapidly advancing ECL integrated circuitry, and couldn't envision how to reach down into and more fully exploit it. Nor could expert ECL circuit designers provide architects with the necessary circuit level hooks to resolve intractable computer architecture problems. DIS revealed that only a rethinking of the basics across all levels of abstraction could break the logjam – a lesson that deeply affected my later work in VLSI [3].

Another problem inhibiting progress was the complexity of the ACS-1's design. I realized that a rigorous overall system design methodology was required – based on a coordinated, hierarchical, multi-level computer simulation of

formalized design partitions – for there to be any hope of collective group activity to generate the sequences of internal subsystem-interface test patterns for debugging, bringing up and maintaining such a complex machine.

These realizations, along with many insights into interpersonal team behavior that I had gained from the then-recent ethnomethodological work of Harold Garfinkel, led me to design and propose a formalized design of the ACS design process, a proposal which was well-received and also strongly impacted my later thinking on VLSI design methods [3], [5], [6], [7].

My First Failed Project

In hindsight, it is now recognized that had the ACS-1 been built, it would likely have been the premier supercomputer of the era, eclipsing both the CDC 7600 and the IBM Model 91 [1]. But, that was not to be.

Instead, in 1968 Gene Amdahl proposed that the ACS-1 be replaced with a S360-compatible supercomputer, and the ACS project fell victim to the ensuing political confrontation. Declared "a failure" by IBM executive B. O. Evans, the ACS project was disbanded [8]. Apparently, neither Amdahl nor Evans nor other key IBM people had a clue about the novel DIS architectural innovations that had been made within the secretive project; the invention was shelved away and apparently lost in dusty technical reports.

Fired by IBM

At that same time in 1968, I was pioneering along another path, as well. I alerted HR at IBM that I was undertaking a gender transition to resolve a terrible existential situation I had faced since childhood. I was hoping to quietly maintain employment during that difficult period. However, the news escalated to IBM's Corporate Executive Committee (including CEO T.J. Watson, Jr.), and I was summarily fired [3].

Finding myself unemployed and in the midst of transition, I watched my contributions to ACS go down the tubes as the failed project simultaneously imploded. I grieved over this misfortune, but there was nothing I could do about it. And not surprisingly, given ACS-1's stained image within IBM, little curiosity ever arose at the company about what developments had occurred there. The DIS concepts eventually leaked out, however, and began propagating through the architecture community, the full story only beginning to emerge in recent years.

Starting All Over Again

I completed my transition and started my career all over again in early 1969, remaining right in the Bay Area. A gritty survivor, I began at the bottom of the ladder as a contract programmer, with a new identity that allowed me to work in "stealth mode". Nonetheless, it was a terrifying time. Any public outing would have killed my new career and I could have ended up unemployed, a social outcast, living on the streets.

Fortunately, after a series of rapid upward moves I was hired as a systems programmer at Memorex Corporation. On joining Memorex, I described the general nature of my computer design work at IBM to the HR department. When Memorex entered the computer business I was given responsibility for CPU architecture and design for the Memorex 30 System (MRX30), an entry-level competitor to IBM's System 3. It was now mid-1971.

Creating a TTL micro-programmed minicomputer from a blank sheet of paper, under tight time and cost constraints, was a tremendous hands-on experience. I loved the intense teamwork and gained confidence as an enthusiastic thought leader on the project. Using methods I'd developed at ACS, I quickly built a register transfer level simulator to coordinate the overall design effort. When first powered up in early 1972, the 'Memorex 7100' processor (the MRX30 manufacturing prototype, shown in Figure 2.) came up smoothly and ran code with just two minor wiring errors. It was a triumph.



FIGURE 2: The Memorex 7100.

Explosive News

Then in November 1971, Intel announced the 4004 microprocessor, followed by the 8008 in April 1972. These were blockbuster events for digital system designers and seriously grabbed my attention. I attended several intensive short courses to learn about the chips. They proved architecturally simple and easy to use.

Detailed knowledge about the underlying MOS (metal-oxide-semiconductor) digital circuitry about which I was so curious, however, was still inaccessible outside Intel (except for knowledge about the rapidly emerging application of MOSFET's in dynamic memories [9]). Did architects have to understand MOS circuits and devices to design such microprocessor chips? Did folks outside semiconductor houses have futures in computer architecture?

The future of digital design seemed to be in MOS, but I had no clue how to get into it.

My Second Failed Project

Just as we completed the MRX30 manufacturing prototype, Memorex left the computer business – a victim of monopolistic pricing moves by IBM. I was crushed and no longer saw a future there. Not only had IBM fired me, it was now stamping out many competitors that I might possibly work for!

Nonetheless, in late 1972 I asked my headhunter to open a job search and received two excellent offers: to be the architect of Fairchild Semiconductor's next microprocessor or to join Xerox at the new Palo Alto Research Center (PARC).

The Fairchild opening seemed a great opportunity, but I felt uneasy. Knowing nothing about MOS circuitry, I hesitated at the prospect of merely blocking out simple architectures that others would implement. I also had doubts about fitting into the semiconductor industry, with its famously macho disdain of women.

Xerox was different, however. A movement was underway there that promised to revolutionize computing by creating a new world of interactive personal computers and related storage devices, scanners, copiers, laser-printers and network communications. PARC was recruiting the best and brightest young talent from across the U.S. to join the effort, including a number of women scientists. A diverse and eclectic group, I'd heard of many of the 'names' already working there. I took the job at PARC in 1973.

My project was a tough one: create a compound OCR/FAX system that compressed office documents for efficient communication. It took two years of work on character-recognition algorithms, as well as the architecture, logic design, and packaging of a novel image processing system, to create the TTL prototype. The Xerox Sierra filled a full rack of circuit boards, and there was no way to then reduce it to a few LSI chips. It was clearly doomed.

My Third Failed Project

The end came in 1975 when William R. (Bert) Sutherland joined PARC as manager of the Systems Sciences Lab (SSL). Bert had led the Computer Science Division at Bolt, Beranek and Newman (BBN) and knew where he wanted his new lab to focus. He began vetting staff and projects, bringing in Wesley (Wes) Clark of LINC fame to advise him.

By then I had told Bert in confidence about my IBM work, and in an intense follow-on interview, I presented the details of Sierra and my ACS-1 innovations to both Wes and Bert. Afterwards, Wes told Bert, "This is the real thing!"

I was able to keep my job, but Sierra had to go. I was severely disheartened over yet another failed project. There was no way to know at the time, of course, that all of those failed projects had prepared and positioned me to launch a revolution in what would become known as 'VLSI design'.

Concurrent Events at Fairchild, Intel, IBM and Caltech

In 1970, Carver Mead at Caltech had coined the term "Moore's Law" for Gordon Moore's 1965 prediction [10] that chip device counts would double every two years. A specialist in device physics in addition to his teaching duties at Caltech, Mead became a high-level consultant at Intel, gaining access to vital projects and know-how there. Around this same time Mead reportedly independently invented a metal-gate PMOS circuit design for PLA-based finite-state machines, realizing that it would be easier to code logic than to draw it [11].

In 1972, Bruce Hoeneisen and Mead described MOS device scaling principles in a widely read paper, predicting that MOS field-effect-transistors (MOSFETs) would function properly at gate lengths as small as 0.25 micron, far smaller than the 10 micron gates of the time [12].

Motivated by the possibilities of scaling, Mead began teaching MOS integrated circuit design courses at Caltech, based on the dynamic-logic design methods that were rapidly evolving within several semiconductor firms to exploit the new technology – from the early work of Frank Wanlass at General Microelectronics, to that of Bob Booher at Autonetics, to that of Lee Boysel and his teams at Fairchild Semiconductor and then at Four Phase Systems, to that of Federico Faggin and others at Intel on the Intel 4004, 8008 and other early microprocessors [13], [14], [15], [16].

The latest Intel circuit design methods well exploited the new self-aligned silicon-gate fabrication technology, a concept invented in 1966 by Bower and Dill at Hughes Research [17] and by Kerwin, Klein, and Sarace at Bell Labs, and first commercialized by Faggin while at Fairchild [16]. Bright Caltech students studying these methods under Mead's guidance had no difficulty applying them to basic digital circuit design.

In 1974, IBM's Robert Dennard, inventor of the single transistor DRAM, showed that when MOSFET geometries, voltages and dopings were scaled down, gate transit times also scaled down and performance thus improved by the same factor [18]. Taken together, the density improvements predicted by Moore's Law and the performance improvements predicted by Dennard signaled a coming explosive growth in chip processing power.

Bert's brother Ivan Sutherland joined Caltech in 1974 as founding Chair of the new Computer Science Department there. Famous for his pioneering work in

computer graphics, Ivan was excited about the potential for microelectronics. He recruited Mead to join his new department, bringing in Mead's expertise in device physics and circuit design and his many connections in industry.

In '75 Ivan Sutherland, Carver Mead and Tom Everhart (then chair of EECS at U.C. Berkeley) conducted a major ARPA study of the basic limitations of microelectronics fabrication. Their ARPA report (published in '76) urgently recommended research into the system design implications of "very-large-scale integrated circuits" in light of coming advances in scaling – pointing out that no methods existed for coping with such complexity and no approaches then underway held promise of solutions [19].

Bert introduced me to Carver and Ivan that fall, and I began studying their recent work – having no idea what adventures lay ahead. Ivan soon wrote a letter to his brother Bert – a letter that has since proven to be historic – proposing that PARC and Caltech work together to attack the system complexity problem [20], [21].

The PARC/Caltech Collaboration

In early '76, the Sutherland brothers formalized a collaborative research project between Xerox PARC and Caltech. The mission: to explore ways to more easily create systems in silicon, and apply the emerging personal computing technology at PARC to the task.

At Caltech, Ivan Sutherland asked Carver Mead, and his students Jim Rowson and Dave Johannsen, to be part of the team. At PARC, Bert asked two researchers to join the team: one was Doug Fairbairn, a brilliant young computer engineer then designing Xerox's NoteTaker, the world's first portable personal computer. Bert's other invitation was to me.

Personally, I could hardly believe this reversal in fortune! I was being propelled into MOS-LSI, and was confident my experiences at ACS would give clues on how to proceed.

By now it was clear that commercially viable chips would inevitably contain several million transistors by the early 1990s. By scaling supply voltages and exploiting the coming CMOS technology, MOS circuits would become as fast as ECL but with far lower power dissipation. The capabilities of an entire ACS-1 processor could eventually be 'printed' on a single chip, and personal computers like those emerging at PARC were destined to have the power of current-day supercomputers. It also meant that my DIS invention

would inevitably come to life. These electrifying possibilities launched me into hyperdrive.

Exploration Begins

Our work began with concentrated studies, including taking a number of short intensive courses on the very latest relevant technologies in Silicon Valley. And, while Mead taught us about NMOS device physics, circuit design and fabrication processes, I shared my knowledge of computer architecture, and of multiple-abstraction-level computer-design-process design, with him.

We then waded in by building hands-on prototype chip subsystems, learning as we went along. Fairbairn and Rowson created an interactive layout system called "ICARUS" on the Xerox Alto computers, which we all used to gain design experience. Mike Tolle, Chris Carrol, Rod Masumoto, Ivan Sutherland, Dave Johannsen and Carver Mead worked on the "OM" microprocessor data path at Caltech, using symbolic layout software (ICL/ICLIC) by Ron Ayres. Ron and Ivan crafted a graphical interchange format (Caltech Intermediate Form, CIF), to circumvent the n^2 translation problem that arose when converting each design tool's output to one of many mask specs.

Our tool building and design work in that early period went well, but chip prototyping proved difficult. We could obtain masks from Silicon Valley mask makers of the time, using reticle pattern-generator code produced by ICARUS. However, wafer fabrication was quite another matter.

Engineers within semiconductor firms could get small lots of prototype chips via regular fab runs – either by stepping reticles of prototypes into a few die locations on production masks, or by substituting masks containing multiple prototype designs as one particular boatload of wafers transited the fab line. However, it was nearly impossible for outsiders to access such prototyping. Only 'writers' working for the 'printing plant' could become 'published'; i.e., only designers working for the semiconductor firms could get their chips manufactured.

Mead's contacts occasionally provided access to MOS fab for Caltech circuit designs, and he worked to gain similar access for our PARC/Caltech project. This involved extensive coordination during design and mask-making in order to meet the many requirements of the target fab line. Each line had different layout design rules, mask polarities, alignment marks, process test patterns, scribe lines and more – with all of that data communicated via detailed paperwork unique to each company. We

sometimes obtained prototype chips for our project this way. However it was a daunting activity, full of easily-derailed arcane practices, and turnaround times spanned many months.

Even so, we made great progress in 1976 as we cranked up our knowledge in MOS design and tool building – although learning more than we wished to know about what can go wrong in prototype implementation.

Meanwhile, Ivan Sutherland prepared an article for *Scientific American* about the challenge microelectronics posed to computing theory and practice. Since most of a chip’s surface was occupied by ‘wires’ (conducting pathways on the various levels) rather than ‘components’ (transistors), decades of minimization theory in logic design had become irrelevant. And by co-mingling logic and memory within regular lateral arrays of small processing structures in silicon, it was possible to save both time and energy in internal on-chip communications.

The resulting article, co-authored by Carver Mead, was a powerful statement of the challenges we faced as 1976 drew to a close [22]. The bottom line: A huge and previously-unknown territory for creative architectural innovation had opened up, and as yet there were no theories or methods to guide those explorations.

Simplification and Convergence

By late 1976, I sensed in our work a parallel to Steinmetz’s time – a time when DC technology was well established but was running out of steam – while the emerging AC concepts seemed mysterious, even to expert practitioners, who as yet had no formal theories to develop AC technology.

Steinmetz had broken the logjam by coalescing mathematical methods and design examples that enabled practicing engineers to routinely design AC electrical systems with predictable results. This starter set of knowledge was sufficient to launch the AC revolution. By applying Steinmetz’s principles, practicing engineers spawned a whole new industry.

Similarly, this seemed the right way to attack the VLSI complexity problem. Instead of visualizing an ever more complex future into which all current and evolving developments were projected, why not begin by simplifying, simplifying, simplifying? Would that not spawn something starkly simple and eminently practical instead?

This wasn’t about engineering new things; it was about the engineering of new knowledge. My key idea was to sidestep tons of accumulated vestigial practices in system architecture, logic design, circuit design and circuit layout, and replace them with a coherent but minimalist set of methods sufficient to do any digital design – restructuring the levels of abstraction themselves to be appropriate for MOS-LSI.

I theorized that if such a starter set could be composed, it would enable thousands of system designers to quickly migrate from TTL into MOS-LSI – just as I had. Most of what was needed was all around us, including the latest Intel’s MOS-LSI design lore. The challenge was to make wise decisions about what to keep, and what to toss.

Structuring a Design Methodology

With this theory in mind, I convinced Mead we should set a far more ambitious goal for the work. We should move to create a simplified methodology for designing whole systems in silicon, not just circuits – and aim it specifically at computer architects and system designers. He agreed, and in an incredibly intense period in the spring of 1977 we formulated the basics of the new methods. Happily, NMOS was perfect for this simplification.

Seen from an architect’s perspective, an NMOS chip could be visualized as a miniature 3-layer printed circuit board, with wires printed on the metal (MET), polysilicon (POLY) and diffusion (DIFF) levels, and with vias (i.e., “contacts”) connecting wiring levels where needed. As a result of the new self-aligned silicon-gate fabrication process, a MOSFET transistor was formed (and easily conceptualized) wherever a path on the POLY level crossed over a path on the DIFF level.



FIGURE 3: Lynn Conway at Xerox PARC in 1977.

But there was more. The resistance of wires was small compared to on-transistors, while off-transistors had extremely high resistance. Thus an NMOS FET could be abstracted as an almost perfect ‘bi-directional switch’ with its control gate on POLY and switch contacts on DIFF. Additionally, wiring and stray capacitances were often modest compared to gate capacitances. Thus turning a transistor on for a sufficient time and then off could charge (or discharge) the gate-capacitance of a subsequent transistor and then isolate it – dynamically storing the on (or off) state as in a Dennard dynamic RAM cell [18].

At the top level, architects composed digital systems as arrangements of interconnected registers and intervening logic, with data movement and logical sequencing controlled by state machines. Registers could now be built in NMOS as arrays of inverters, each composed as a simple pull-up/pull-down transistor pair using depletion-mode MOSFETs as loads. Data movement between registers could be controlled by pass transistors, using two-phase non-overlapping clocks to isolate the dynamically stored data. Clocking times could be calculated as simple multiples of minimum FET-gate delays. Logic functions could be crafted using simple NMOS structures placed between successive register stages. State-machines could be built using NMOS programmable logic arrays (PLAs), with registers holding state to feedback to inputs at successive machine cycles. All this could be done using simple rules of thumb for gate geometries, pullup/pulldown ratios, fan-outs, power distribution and timing.

By routing control lines perpendicular to data lines, important subsystems could be woven as regular arrays of cleverly designed NMOS cells – resurrecting long-lost non-gate-logic methods, as in symmetric networks of relay contact switches, and elevating the bi-directional ‘switch’ as a basic level of abstraction. We sketched cell topologies as stick diagrams, using blue, red and green pencils to indicate cell wiring on the MET, POLY and DIFF levels – and wherever a ‘red wire’ crossed a ‘green wire’ an FET ‘switch’ was created. Cell topologies were then geometrically expanded to form cell layouts, compacted to the degree possible under the target fab line design rules for spacings and widths.

When implemented, such designs often required far less area, time and energy to perform functions than those produced using traditional abstraction-levels and optimizations at each level – shattering years of established academic theory and industry

practice – and they were often dramatically simpler to design.

Layout design rules: The Fly in the Ointment

The stick diagrams of cell topologies contained all information necessary for laying out functionally unique cells. The layout design rules merely said what was prohibited during the compaction of geometrically expanded cell topologies towards minimal areas.

Unfortunately, MOS fabrication engineers produced large books of layout design rules unique to each new process, often running 40 pages or more. In efforts to increase yields, layout designers valiantly applied these rules, including those enabling only tiny compactions, often using arbitrary angles and curvatures to *scrunch* on-chip features down in size. Just imagine the complexity of the layouts, hand-cut into rubylith patterns for maskmaking, that resulted from such efforts!

To ease the burden for students in his earlier circuit-design classes, Mead crafted ad-hoc rules having reduced complexity by tossing low-return constraints and formulating ‘covering’ sets of rules – using line-widths, separations, extensions and overlaps somewhat larger than the minimums required for target processes. Such rules were easier to teach, apply and check, and were far better for prototype design where extreme compaction was not needed. However, such rule-crafting required expertise, judgment and close coordination with fab lines. The resulting layouts were also tied to particular processes, and had to be redone as new processes came online.

In contrast to our other successes, circuit layout seemed an intractable level of design abstraction. Questions of computational complexity also loomed: How could such complex, rapidly changing geometric layout rules be encoded, applied, and checked – given the increases in circuit density anticipated in the coming years?

Invention of Scalable Design Rules

In early 1977, I began asking myself: What is the simplest possible set of layout design rules? I found the answer in a different question: What is the maximum from among the minimum lateral line widths, separations, extensions and overlaps at all levels for a given process? Once found, I knew this one measure of process resolution could be used to limit minimum sizes for all layout features.

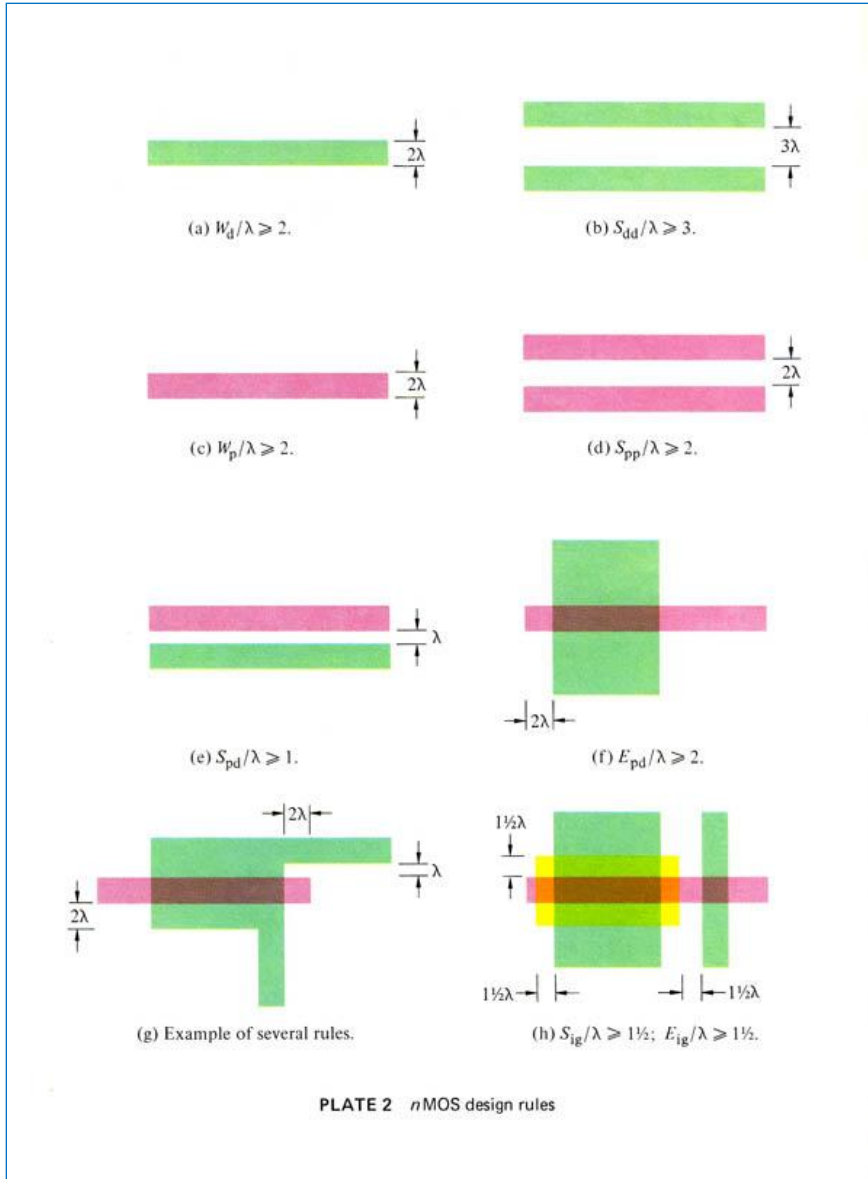


FIGURE 4a: The λ -based scalable NMOS design rules [23], [24]. (Courtesy Pearson Publishing)

The resulting, minimalist covering rules were crude and non-optimal, but they fit onto a single page – that in itself, a breakthrough. I also noticed something else: The minimalist-rules generated layouts having a timeless quality. They remained unchanged, even as the process scaled down.

Suddenly it beamed down to me: MOS design rules should not be framed as sets of lengths but as sets of ratios of lengths. Such dimensionless rules could then be scaled to any process as multiples of a basic length unit in microns, a unit I called **Lambda** (λ).

I quickly crafted an NMOS rule set to explore this idea, setting λ at one-half the maximum of minimum line-widths, separations, extensions, and overlaps. The resultant rule set was less toy-like than the minimalist rule set, and revealed the full potential of the idea.

I vividly recall seeing Mead’s jaw drop that spring morning in 1977 as I presented my strategy for λ -based rules on my whiteboard at PARC. This was it! We now had a ‘structured’ design methodology (as Mead called it) from top-to-bottom.

Of course the rules needed tweaking to gain compactness and to better anticipate scaling effects. For example, we set line widths and separations on the MET layer to 3λ , while keeping those on the POLY and DIFF layers at 2λ . Still, the rule set remained small at only two pages in length, easy to teach, learn, apply, and check (see Fig. 4).

These simplified scalable design rules had many implications. With circuit density doubling roughly every two years, why spend time on intense layout compaction? Why not compress design times by using these simpler rules,

and race to the next smaller process that much sooner? Even more importantly, scalable rules allowed cell topologies to be laid out in a timeless form – opening the door to widely-sharable, time-durable MOS cell libraries.

Adjacent subsystems could also often be abutted by designing their cells at the same pitch (extending some cells’ lateral dimensions, where needed), saving space and improving performance by eliminating wiring channels. EDA tools for generating and checking layouts were also greatly simplified and speeded-up by using rectilinear wiring on a *Lambda*-based integer grid, rather than at arbitrary angles and dimensions as in earlier practices.

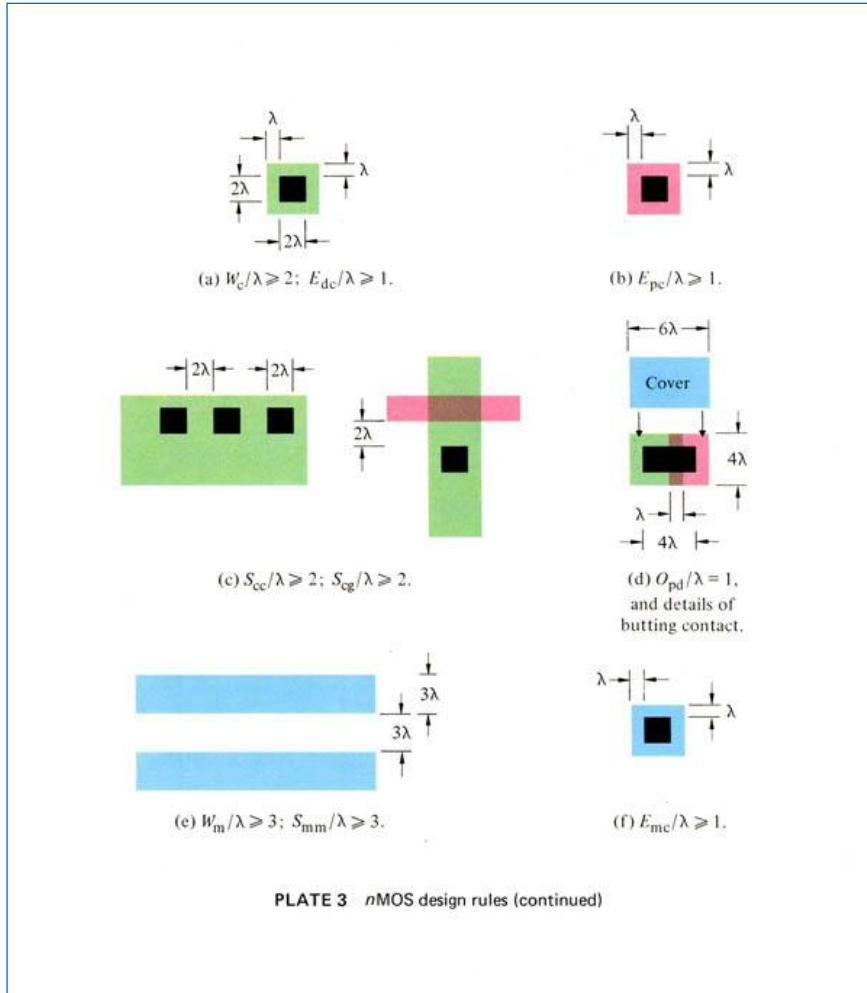


FIGURE 4b: The λ -based scalable NMOS design rules (cont.) [23], [24]. (Courtesy Pearson Publishing)

Thus the *Lambda*-based design rules played a similarly simplifying, empowering and unifying role at the knowledge-interface between VLSI designs and EDA tools, as had the self-aligned MOS gate at the knowledge-interface between LSI designs and semiconductor fabrication.

The scalable design rules opened another door, as well. Suddenly a clean separation between chip design and fabrication was possible, with extremely simple rules providing the interface.

The “Tall Thin Man”

The transparency of the new methods enabled architects to design systems from top-to-bottom, as they had in the days of relay contact switches and vacuum tubes in the 1950’s, when I was a student.

Now once again, digital circuitry could be easily envisioned and crafted, using simple rules of thumb.

No longer were extensive calculations and circuit simulations needed as in bipolar IC design. While such efforts were still needed during process development to ensure circuit function and performance, they were not needed when designing prototype circuits and layouts. So long as on-chip test patterns found that electrical parameters were within spec, our design rules of thumb worked perfectly well.

For years, ECL and TTL had imposed logic-gate and clock-edge-triggered flip-flop register abstractions onto system design – impeding top-down visualizations of alternatives for expressing architectures in silicon. Using our methods, architects could clearly visualize and instantiate their creations all the way down to the switches in silicon. It was a tremendous breakthrough!

A new world of architectural exploration opened up before us, a world I had peered into twelve years before, when inventing DIS at ACS. I sensed that

thousands of engineers could now have similar experiences as system architects by exploiting our new methods. At least, that was my theory at the time.

Meanwhile, Fairbairn’s and Rowson’s ICARUS software and Ayres’ ICL/ICLIC enabled us to input, edit, print, and visually inspect our layouts. However, these were only the beginning of a parallel revolution in EDA, as new tools evolved to support work across the restructured levels of abstraction. The scalable design rules in particular had dramatic implications for tool-building and chip prototyping.

By this time, however, signs of resistance were emerging at PARC, as critics in the competing Computer Science Lab (CSL) looked askance at what they saw as our “toy” designs and “toy” design tools. Not surprisingly, they questioned what our tiny effort could possibly bring to the huge semiconductor industry.

What we clearly needed were classy tutorial design examples, and in June 1977 Dave Johannsen set out

to rigorously apply the new methods to the design of a follow-on data path chip at Caltech. The OM2 would be completed by year-end, yielding excellent examples of subsystem design using the new methods. Unlike the OM1, the OM2 actually worked.

Early in our work Mead had coined a term – The “Tall Thin Man” – to describe system designers like Johannsen who used our exploratory methods, and the term eventually took its place in the lexicon of *Silicon Valley*. Although women engineers (including me) were excluded by Mead’s imagery, the phrase stuck, for a time.

What to do with the New Knowledge?

The rush of ideas in early 1977 led to a host of challenges. Most especially, what were we to do with the new knowledge? In response, I began evolving a tutorial to unfold and explain it all, honing a minimalist sequence of ideas sufficient for architects to visualize what a chip is and how it now might be designed.

The task was akin to revealing a medieval cathedral as composed of pointed arches, ribbed vaulting, thin walls and flying buttresses, showing how a set of basic principles were sufficient to raise such a complex structure. While doing this work, I began realizing that launching such an abstract system of knowledge by publishing bits and pieces here and there in traditional journals would be inadequate, especially when it challenged so much established practice. What to do?

The Idea of “The Book”

The die was cast in early June 1977, during a relaxed, evening team-brainstorming meeting at PARC. Thinking out loud, I launched the idea: Why not write a book about our work, and self-publish it using PARC’s Alto systems and laser printers?

If the book were comprehensive, well-written, and filled with good design examples, it would appear to reflect years of mature practice. In yet another echo of the Steinmetz story, I theorized that such a book would be taken seriously and could launch the new methods we were proposing. Mead let out a big, “Yeah!”, and Fairbairn was excited as well. So that was it. The decision had been made, and off we went.

The sophisticated computing environment at PARC gave us uncommon confidence. We could interactively create documents and designs using our

Alto systems, collaborating locally via e-mail and file-sharing, and interacting remotely with colleagues at leading universities by using the new ARPANET. Swept along by PARC’s movement to bring computer power to the individual, we had intellectual power-tools at our disposal that provided the means and the wherewithal to do unprecedented things.

As I began writing the book, my Alto became the integrating node and control-center for a wildly-expanding project and community of contributors. While I drafted explanations of the structured design methods, Mead provided input on NMOS fabrication and mask-making, Fairbairn and Rowson crafted an ICARUS tutorial, and Johannsen began documenting OM2 design examples to round out the text.

We introduced the first three chapters in the fall of 1977, interjecting them into MOS circuit design courses taught by Mead at Caltech and by Carlo Séquin at U.C. Berkeley. (Séquin had recently joined our team as a consultant at PARC). We titled those preliminary chapters *Introduction to LSI Systems*, but then paused at how to acknowledge authorship. Mead was a well-connected full professor at the time, while I was virtually unknown outside of our group. Thus even though I was the architect and principal author of the book, we listed Mead as first author – to enhance the book’s credibility [23].

Building on the feedback that came in, I prepared five full chapters for courses set to be taught the next spring. Dick Lyon, a brilliant Caltech grad and signal processing expert joined our team at PARC. (Lyon went on to invent the optical mouse, among other things.) The winter of 1977-78, Lyon and Carlo Séquin worked with computer graphics expert Robert (Bob) Sproull to refine and produce a formal description of the CIF language (CIF2.0). Johannsen also completed the OM2 in December 1977, in a much-needed early validation of the new methods.

By February 1978, I had incorporated the ICARUS tutorial, the CIF2.0 specification, and the OM2 design examples into a draft of the first five chapters, just in time for a spring semester course taught by Bob Sproull at CMU and Fred Rosenberger at Washington University. This version of the book included many color plates I had made on the new color copiers at PARC, enabling easier teaching and better mastery of the new methods [23].

Then one day, in a rush of enthusiasm, I changed the title to *Introduction to VLSI Systems*.

Bert's Challenge

By this time, Bert Sutherland had joined the EECS Department advisory committee at M.I.T., and soon after offered me a challenge: Go to M.I.T. in the fall, he said, and introduce a senior/masters-level course on this stuff. I was thrilled. We'd been testing portions of the book in various MOS circuit design courses, but this was the chance to pioneer a completely new full-fledged system design course based solely on the book.

I was also terrified. A bit shy among strangers and fearful of public speaking, I also lived in dread of being outed about my past. Up to now, I had been sheltered as a researcher in the laboratory environment at PARC, and had only recently begun to flourish as a research manager there. Teaching at M.I.T. would be quite a different matter, involving much more public visibility. It seemed beyond my reach and in my anxiety I wavered. But Bert insisted: "Lynn, you've got to do this!"

Shortly afterward, while glancing at Steinmetz's photo on my office wall, his story came back to mind, especially the impact of his teaching at Union College. It was one of the great turning points in my life: I threw caution to the wind, and went for it.

Planning the M.I.T. Course

The spring of 1978, I immersed myself in finishing the book. While I drafted Chapter 6 on the architectural level of abstraction, Charles (Chuck) Seitz at Caltech drafted Chapter 7 on self-timed systems, H. T. Kung at CMU provided material on concurrent processing for Chapter 8, and Mead drafted Chapter 9 on the physics of computation. A full draft would be ready by summer, just in time for the course [23].

I also got an important idea: If I could compress teaching of the new methods into the first half semester, students could launch design projects during the second half. If I could then organize quick-turnaround (QTA) implementation of the student projects – including layout file merging, mask file formatting and generation, mask-making, wafer fabrication, dicing, packaging and wire-bonding – I might be able to get packaged chips back to students shortly after the course ended.

I felt that the unprecedented opportunity to design your own chip would attract very bright students to the course. And their projects would, in turn, heavily test the design methods, design tools, book, course,

and quick-turnaround implementation methods. As the summer of 1978 progressed, I based the whole course plan around these ideas.

With Bert's support, I also launched a summer program for the VLSI Systems Area (my new research department at PARC), recruiting Steve Trimberger of Caltech and Rob Hon of CMU as research interns. Trimberger worked with Fairbairn on design tool development, while Hon organized mask-making and fabrication of a set of PARC designs as a multi-project chip (MPC), enhancing our experience in quick turnaround implementation during the run-up to the course.

Building on that experience, Hon and Séquin compiled *The Guide to LSI Implementation*, as a guidebook to our innovative clean interface between chip design and chip fabrication and to the logistical details of implementation. Dick Lyon created a library of critically important cells (input pads with 'lightning arrestors' for electrostatic protection, output pads with tri-state drivers, PLA cells, etc.), contributing CIF code and color plots of the cells to the guidebook. Lyon also updated ICARUS to accept and manipulate oversized CIF code files as outlines and produce a merged MPC CIF file. Rick Davies and Maureen Stone from other Xerox labs joined in the effort; in fact, the whole team pitched in to help compile the new guidebook [25].

Summer passed in a whirlwind of preparations. Before long I was packing-up boxes of freshly-minted texts and course handouts – and heading out on my 3000-mile road trip to M.I.T.

Launching the Course

Launching the course was a formidable experience, in particular because I was terrified of becoming tongue-tied in front of the students. My solution was to be massively over-prepared.

I wrote out each lecture in complete detail, including every instructional point, every drawing and every calculation. Along the way, I unfolded the fundamental concepts of electric circuit theory, electronic design, switching theory, digital logic design, and computer system design, to ensure that all students were well-grounded in every level of abstraction, independent of their background upon entering the course. I didn't see it coming at the time, but this work to avoid gaps in student comprehension would have unforeseen, far-reaching effects.

Jonathan (Jon) Allen was my faculty host for the course, and his student Glen Miranker was my TA. The class included 32 students and 9 faculty/staff auditors. Staff researcher Bill Henke built CIFTRAN, a symbolic layout tool for encoding CIF specifications, while Miranker set up a lab where students could access CIFTRAN via DEC20 terminals and plot their layouts using HP pen plotters. Meanwhile, I kept in close contact with my team at PARC, using a portable, acoustic-coupled, TI printer-terminal to transmit e-mails via the ARPANET.



Figure 5: Students at DEC-20 terminals in the MIT '78 VLSI design lab.

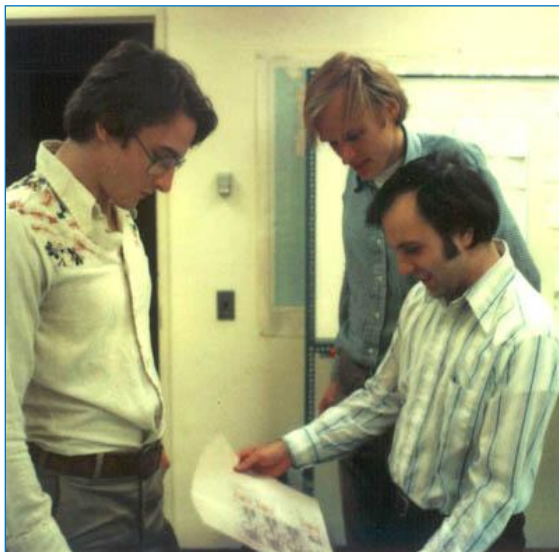


Figure 6: Students Jim Cherry and Gerald Roylance and TA Glen Miranker study a checkplot, MIT '78.

Contrary to my apprehensions, the students became tremendously excited by my teaching. They seized the opportunity to learn by doing and ran with the new knowledge. Many ambitious projects got underway and I began holding my breath, realizing if things went well, this could be a huge win.

By now, Alan Bell of BBN had joined my team at PARC. He and Dick Lyon began preparations for the QTA implementation of the projects, and everyone pulled together at both ends to coordinate things as the design cut-off date approached.

I sent the final student design files to PARC via the ARPANET on December 6, 1978. Lyon and Bell then merged the 19 projects into a multi-project-chip CIF file, converted it to Mann PG format, and had masks made by Micro Mask using their new electron beam system. In this first phase of an important collaboration with Pat Castro at Hewlett-Packard, wafers were fabricated at her Integrated Circuit Processing Lab (ICPL) at nearby HP Research using a 6-micron ($\lambda = 3\mu\text{m}$) silicon-gate NMOS process. Everything went off without a hitch, and the packaged chips were shipped back to M.I.T. on January 18, 1979 (see Fig. 7).

Although my students had only primitive EDA tools, and had resorted to hand-checking of design rules, the new methods so simplified the design work that not many errors were made, and the course led to a very exciting group of projects.

Jim Cherry, for example, designed a transformational memory system for mirroring and rotating bit-map image data, and his project worked completely correctly. Guy Steele, in an even more ambitious project, designed a complete LISP microprocessor. The processor almost worked on this first try, except for three small wiring errors. As such, it set a high mark for others to follow.

After finishing the semester at M.I.T., I took a leisurely route back to California, traveling through the South and Southwest. I knew something profound had happened in the M.I.T. course, but I only vaguely sensed where it might lead. I had also gained real confidence as a research team leader, and itched to do more. I drove on, rock music blaring on the radio, my head in the clouds, savoring the moment.

Something powerful rode along on that trip – an instructor's guidebook on how to teach such a course, in the form of hundreds of pages of carefully handwritten lecture notes [26]

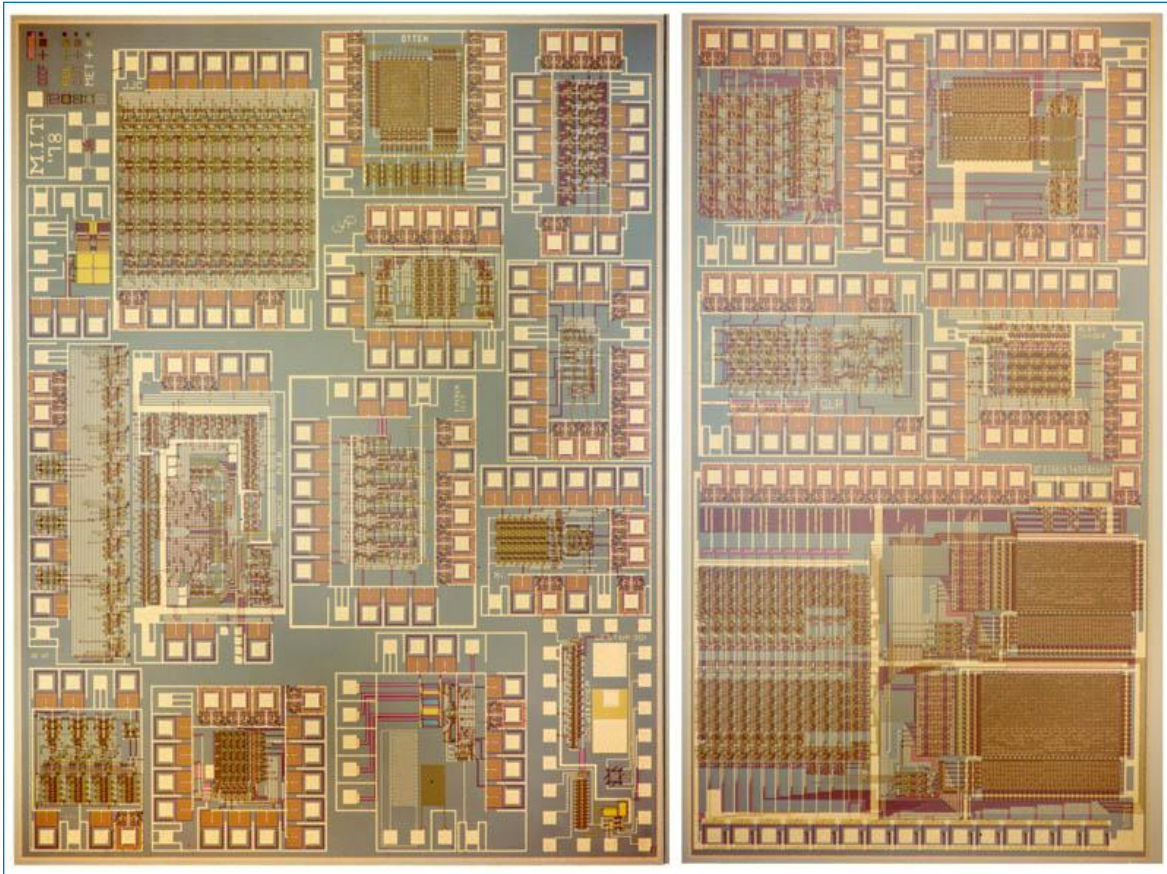


FIGURE 7: MIT '78 chip set (Melgar Photography)

Problems Arise, Pushback Begins

Mead and I had contracted with Addison-Wesley to publish the book, and in early 1979 I began the tedious task of coordinating the copy-editing, hoping to have it ready for courses slated for that fall.

Word spread quickly on the ARPANET about the M.I.T. course, especially the news about Steele's LISP microprocessor. Many professors asked how to offer similar courses, and how to lead ambitious design projects. In response, my group at PARC began to train instructors in the new methods of teaching VLSI design.

Doug Fairbairn and Dick Lyon ran an intensive short course for PARC researchers during the spring of 1979, which was videotaped. We began using those tapes as the basis for short, intensive courses at PARC for university faculty members in the summer of 1979. With the help of the PARC tapes, Mead and Ted Kehl also ran a course at the University of Washington that summer.

I also organized my M.I.T. lecture notes to create the *Instructor's Guide to VLSI System Design* and began printing copies for all those interested in teaching the course [26]. It was these notes, rather than the textbook alone, that for the first time contained the full exposition of the new design methods – unfolding a teachable, accessible, minimalist, covering set of knowledge that enabled students to quickly learn how to competently do VLSI system design.

However, we had a big problem: there was no way to implement design projects from so many universities, other than for each to arrange for their own mask and fab. We had defined a clean interface between design and fab at the layout design-file level, but the logistics of implementation were far too complex for isolated departments or design groups to handle.

I felt that unless students could learn by doing, and make things that worked, they would have merely learned a theory of design. Attacking this problem head-on, I launched work to further simplify and document the logistics in a new edition of Hon & Sequin's *Guide to LSI Implementation*, hoping to help more instructors implement their students' projects in the fall 1979 semester [25].

Mead coined the name "foundry" for any semiconductor firm that could 'print' externally generated designs created using the scalable design rules, and he began popularizing the term to lure firms into providing this type of service. Given Mead's high-level business connections, it wasn't long before folks across the industry were buzzing about his provocative term, wondering what it meant for them.

As noise spread about *Mead and Conway*, signals of serious resistance began to arise. Experts at various levels of abstraction began having allergic reactions: when seen from the viewpoint of each narrow abstraction our stuff looked far too crude and naive to possibly work.

Trouble also arose within PARC. My new research department in SSL came under increasing attack from the leaders of the Computer Science Lab (CSL), who wondered why budget and headcount were being devoted to such questionable work. They didn't seem to grasp why the freedom to improvise and playfully create things was so important when working in a new medium – whether in art or music or engineering – especially when exploring what it is possible to do.

Some in academe even began to wonder if we were nuts. "Who are these people?" they asked. To them, Mead was a device physicist making wild pronouncements on computer design, while Conway seemed some totally unknown woman tagging along as Mead's 'assistant.' Such reactions to appearances were totally understandable. Something had to be done to turn things around, but what?

Necessity is the Mother of Invention

It began as a daydream that spring of 1979, as I fantasized about the impact of large numbers of M.I.T. type VLSI design courses.

I could feel the powerful energy out there: the young faculty members hoping to stand out and get tenure, the students seeking careers in a frontier area, the folks who wanted to start companies and make their fortunes. Imagine how they'd rush to participate

in the new courses, get into VLSI and design their own chips!

Back in reality: My group had maxed out our capability when handling projects from just one school. How on earth could I scale up chip-prototyping to handle ten or more such courses?

I began doodling on my whiteboard, searching for ways to simplify the implementation process, shorten its turnaround time, and scale it up. Although we'd documented static technical interfaces in the *Guide to LSI Implementation*, many procedures needed to be charted and many questions remained about who should do what, and when. Plus we had no means to handle information flow and coordinate interactions on such a large scale.

Suddenly it struck me: What if we positioned an interactive message-handling and file-handling server that orchestrated interactions over the ARPANET? That would streamline everything, eliminate the need for constant human interactions, and bring the needed scalability.

What I envisioned was an early form of Internet commerce system, where design files could be sent to a server and packaged chips returned after implementation. From an information management point of view, it would be analogous to sending many separate magazine articles to a remote server, where they'd be coalesced into a printable mosaic and queued for magazine printing.

With such a system, we could send messages to the chip 'authors', coordinate all activity, do CIF-syntax checking and space requirement checking, and then at the design cut-off time, reel in the final projects' design files. It was clear that such a "VLSI implementation system", as we called it, could then under operator control plan die layouts for multiple multi-project chips (MPCs), merge the design files into those MPCs, and generate MEBES (Manufacturing Electron Beam Exposure System) files for mask generation.

When I excitedly revealed this idea to Mead, he went cold and said "Don't do it."

Mead worried that the event would appear to be orchestrated by DARPA and they would "take all the credit". I understood, for DARPA had ended up gaining much of the visible credit for Stoner's M16 rifle after simply running field trials and promoting the weapon, but so what? That's the way the world worked. Why let concerns about credit interfere with doing something cool?

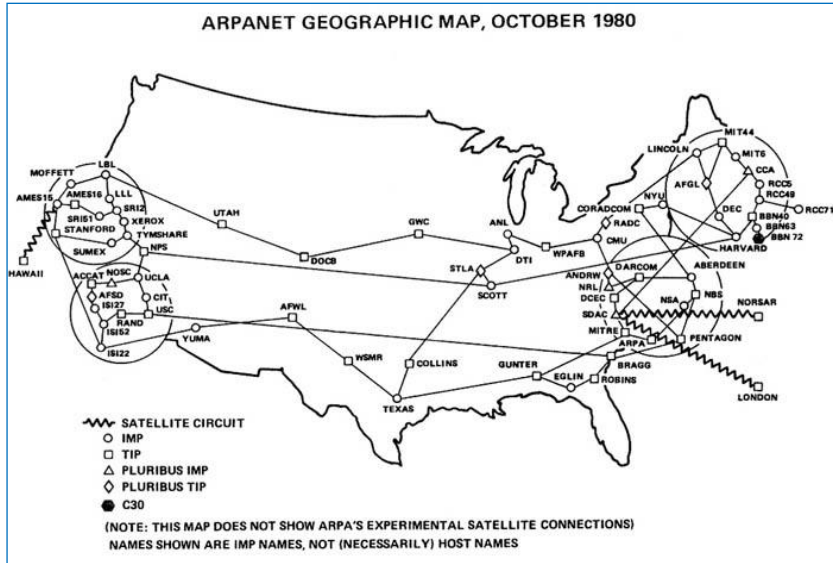


FIGURE 8: Map of the Arpanet, circa 1980.

Mead also felt that each school should connect directly with mask and fab services on its own, just as he'd been doing at Caltech, rather than fall under the control of a centralized service. I disagreed, for I thought his notion of foundry as yet undeveloped, in that it relied too much on undocumented personal expertise, lacked methods for information management, and hence lacked scalability. More importantly, it could not be widely implemented in time for courses in the fall of 1979. Uneasy collaborators from the start, these sharp differences pretty much ended our interactions.

Fortunately Bert Sutherland remained enthusiastic, and I forged ahead. We ramped up work on the implementation system, with Alan Bell and graphics expert Martin Newell developing the software. Although the software itself was conceptually straightforward, the space of possible user interactions was highly complex. It took great effort to anticipate all such interactions and formulate specially constrained key-worded messages to handle them all; Bell began making critical innovations in this area.

As summer approached, it seemed we just might be able to pull it off. By now faculty members at many universities were planning to offer the course, but we hadn't yet announced the chip implementation service. Time was running short and I had to make a decision.

With just a tinge of fear, I drafted an e-mail, complete with a huge promise to the many faculty members and many, many students out there: We at PARC would implement the chip designs from all

Mead-Conway courses offered that fall, in an ARPANET happening called "MPC79". I knew if what I was offering didn't work, I would have to go into hiding. I hesitated, suspended in the moment, then pulled the trigger and pushed "SEND".

MPC79: The Network Adventure

The summer passed in a rush.

Alan Bell and Martin Newell readied the implementation system software, while Bell, Rob Hon and I carefully crafted e-mails to send at intervals during the fall – establishing a strict timeline to

coordinate activities. Hon and Séquin completed the second edition *Guide to LSI Implementation*, which included the definition of CIF2.0 by Bob Sproull and Dick Lyon, an expanded set of PLA cells and I/O pads created by Lyon for all designers to use, along with a lot more information about implementation procedures [25].

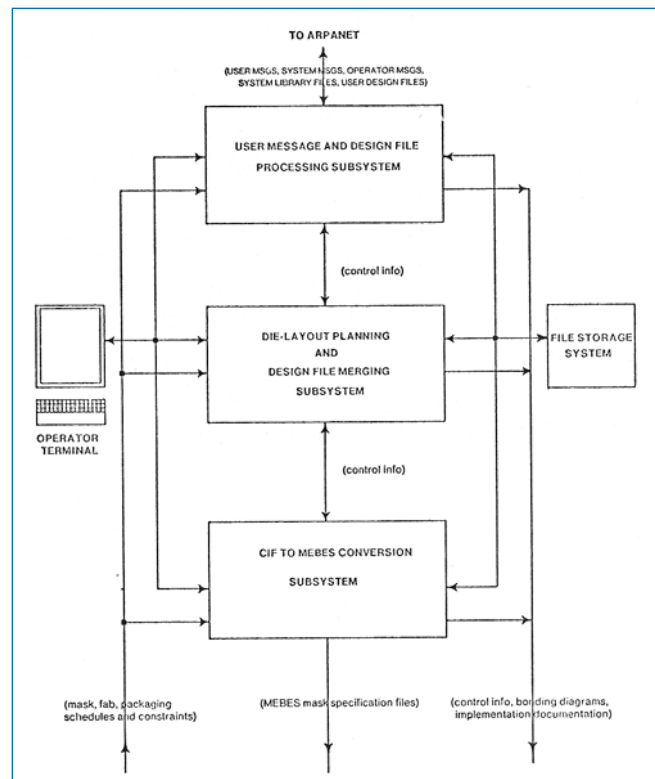


FIGURE 9: MPC79 implementation system: overview of the software.

All sort of wild things happened as we went along – some serendipitous, some funny, some scary. A young Stanford professor named Jim Clark asked if he could hang out at PARC, learn the basics of chip design and do a project for MPC79. I said sure, and helped him with some basic instruction. An expert in system architecture and computer graphics, Clark seemed a perfect adventurer to launch into VLSI. After taking Fairbairn and Lyon’s PARC videocourse, Stanford professor Forest Baskett and his Ph.D student Andreas Bechtolsheim also did projects for MPC79; they would later become famous as architects of the SUN workstation and more.

A crisis then developed. A senior academic of impeccable standing called an urgent meeting with George Pake, Director of PARC. Apparently my announcement of MPC79 seemed incomprehensible to the establishment at the time, and the academic’s school was among those threatened by the perceived infection. His message: Conway is “crazy”, the MCP79 project is unsound, and Xerox will suffer huge embarrassment unless it’s cancelled.

I could feel the apprehension in Bert’s voice as we hurried to Pake’s office, and I nearly panicked when they told me what happened. We knew the concerns were truly justified. Although the new methods had worked at M.I.T. and our computers provided powers outsiders couldn’t imagine, MPC79 was a huge gamble. However, Bert stood by me and the cloud lifted. Pake said “Not to worry. Just do it.”

The vibrant counter culture within PARC helped brace us against all doubts; it seemed everyone there was reaching for dreams. On the outside people saw a prestigious corporate lab housed in a castle-like building, high on a hill overlooking Palo Alto. It was a dignified image much like that of IBM’s lab at Yorktown Heights, i.e., one that established folks took very seriously. How could they possibly imagine what went on within PARC’s walls?

This contrast came home to roost one weekend evening, as I passed by a young Rob Hon at his Alto. In T-shirt and jeans, feet propped on a chair, using his Alto to send an important MPC79 message to the universities: “If only they knew who’s doing this,” he quipped.

Primed and bonded by our experiences during the 1978 M.I.T. course, the team was really on a roll, and an atmosphere of excitement and fun permeated our work. Everyone seemed to know what to do, no matter how novel the situation. Individuals jumped in and out, taking on creative improvisational roles as opportunities arose, much as seasoned musicians would in a fine blues and jazz band.

A huge phenomenon unfolded that fall as our coordinating messages and files surged across the ARPANET. Twelve universities participated, with courses given by Jon Allen and Lance Glasser at M.I.T., Chuck Seitz and Carver Mead at Caltech, John Newkirk and Rob Mathews at Stanford, Richard Newton and Carlo Séquin at Berkeley, Bob Sproull at CMU, John Murray at University of Colorado, Jacob Abraham at University of Illinois, Ted Kehl at University of Washington, Edward Kinnen and Gershon Kedem at University of Rochester, Vance Tyree at UCLA, Fred Rosenberger at Washington University, St. Louis, and John Nelson at USC.

All courses used the new Mead-Conway text (see Fig. 10), published just in time by Addison-Wesley [24], while faculty and TAs had access to the new *Instructors Guidebook* and the latest edition of the *Guide to LSI Implementation*, which I’d printed-up in large numbers at PARC [25], [26].

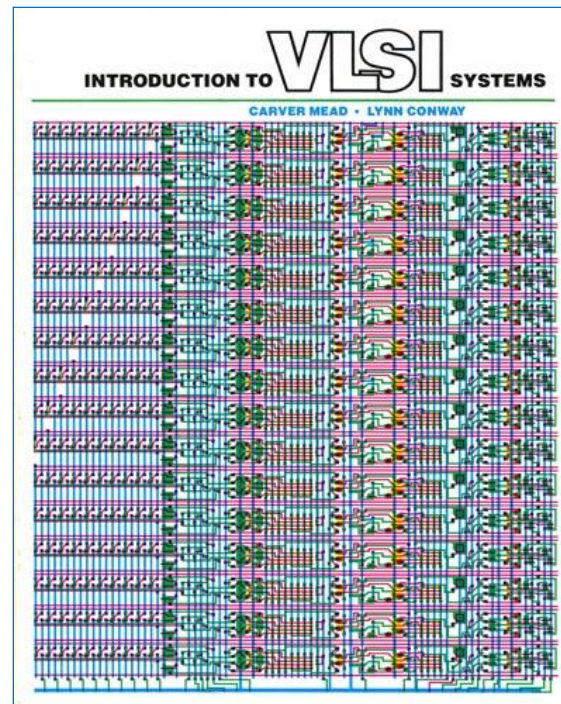


FIGURE 10: The Mead-Conway text.

All courses were synchronized with the MPC79 schedule (see Fig. 11), and most students completed projects for inclusion in MPC79. This was remarkable, as many schools were offering the course for the very first time, and design tools were being programmed as they went along. These events in the fall of 1979 escalated into a giant network adventure that climaxed as the design-cutoff time approached, and as the final rush of design files flowed through the ARPANET to PARC.

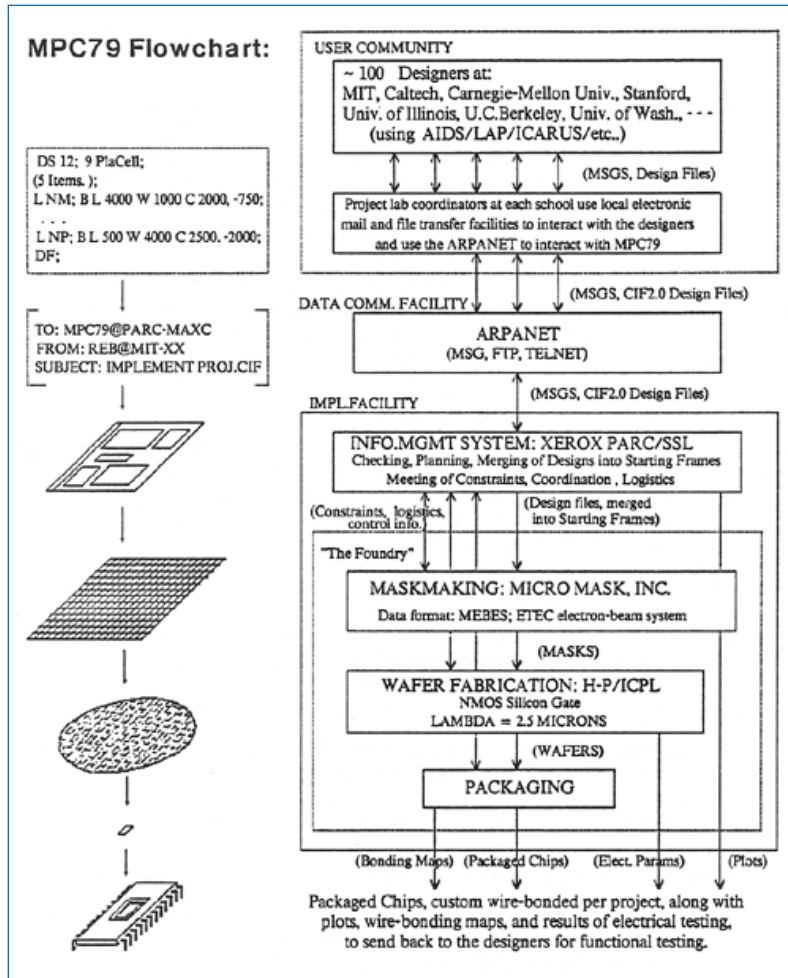


FIGURE 11: Flowchart of events for MPC79.

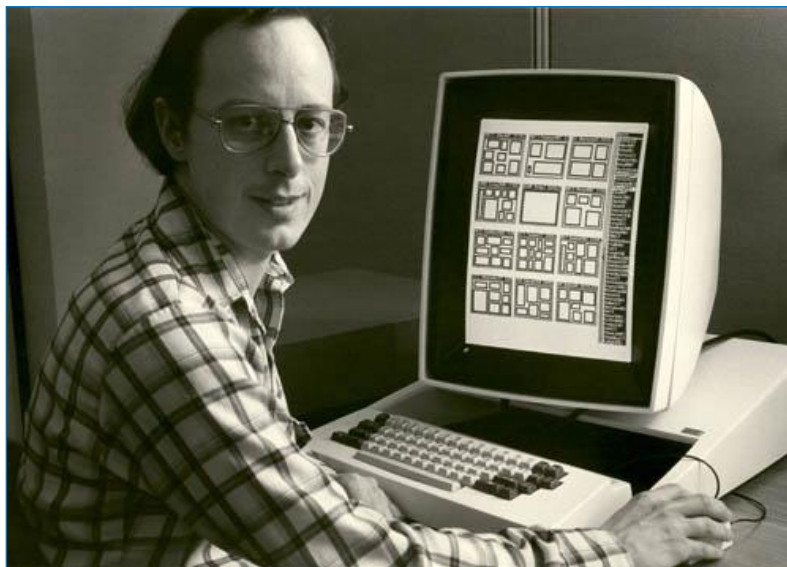


FIGURE 12: Alan Bell at PARC, completing the design-file merge for MPC79.

At 5:00 pm sharp on December 4, 1979, Alan Bell closed external interactions and began die-layout planning, file merging (see Fig. 12), and MEBES format conversions. E-beam mask-making was again done by Micro Mask, pipelined with wafer fabrication to reduce time to completion. With the support of Merrill Brooksby and Pat Castro at HP, fabrication was again provided by HP's ICPL using a 5-micron ($\lambda = 2.5\mu\text{m}$) silicon-gate NMOS process.

Meanwhile, Dick Lyon, Alan Bell, Martin Newell and I readied "Implementation Documentation" for designers, including lists of projects, die-maps, wire-bonding maps, electrical process test data, chip photos by Melgar Photographers and more. When the wafers arrived, we scribed and diced them, mounted die into 40-pin packages (enough for three per project), and wire-bonded to the individual projects within each die (see, for example, Fig. 14). Packaged chips were shipped, along with chip photos and documentation, to students and researchers at the 12 universities on Jan. 2, 1980 [27], [28].

We'd done the impossible: demonstrating that system designers could work directly in VLSI and quickly obtain prototypes at a cost in time and money equivalent to using off-the-shelf TTL.

The MPC79 chip set contained 82 design projects from 124 designers, spread across 12 die-types on two wafer sets. Astoundingly, turnaround time from design cutoff to distribution of packaged chips was only 29 days [27].



FIGURE 13: Lynn Conway, Alan Bell, Martin Newell and Dick Lyon complete the final packaging of MPC79 chips for distribution to designers.

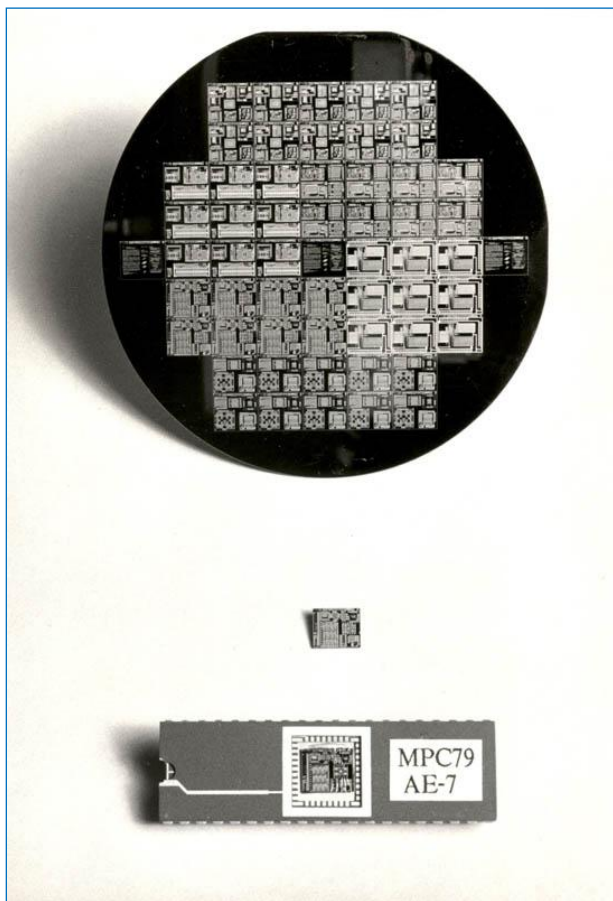
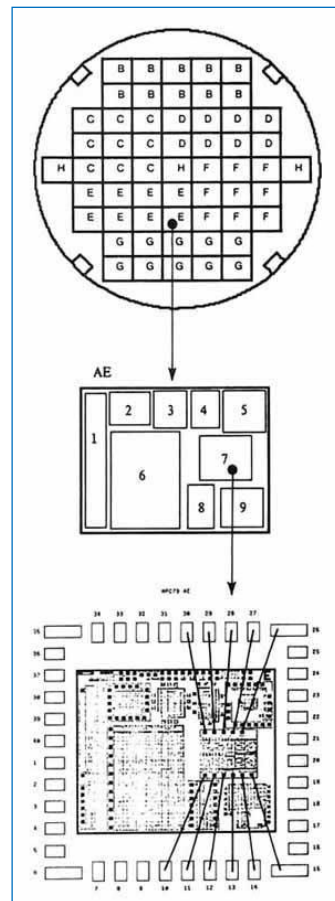


FIGURE 14: MPC79 wafer, die and packaged chip.



Importantly, these weren't just any designs, for many pushed the envelope of system architecture. Jim Clark, for instance, prototyped the Geometry Engine and went on to launch Silicon Graphics Incorporated based on that work (see Fig. 16). Guy Steele, Gerry Sussman, Jack Holloway and Alan Bell created the follow-on 'Scheme' (a dialect of LISP) microprocessor, another stunning design. Along with scores of other innovative projects, these designs signaled that an architectural gold rush was underway.

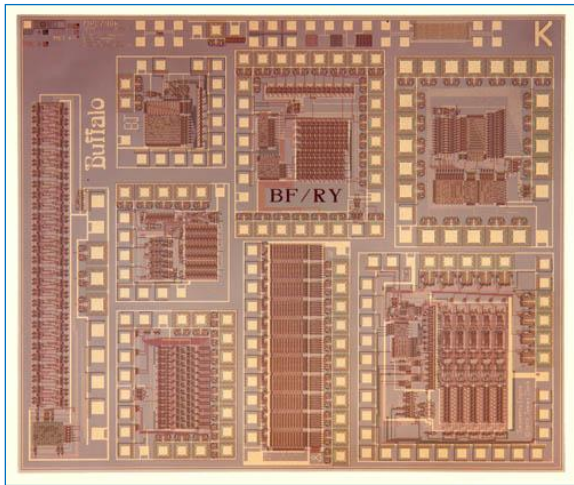


FIGURE 15: Photo of MPC79 die type BK, from Stanford University. (Melgar Photography)

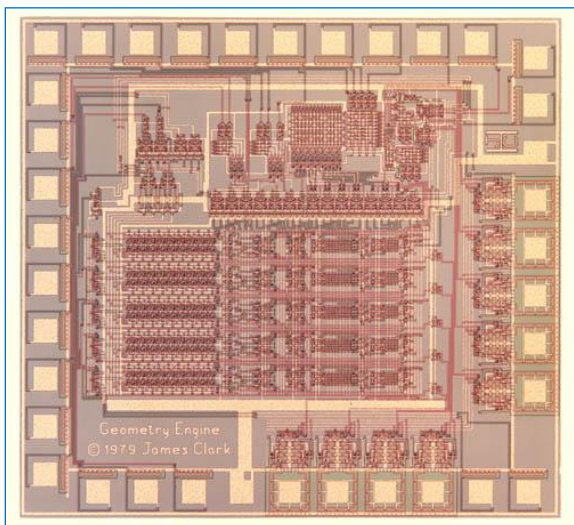


FIGURE 16: "Geometry Engine" prototype by Jim Clark of Stanford (a project on MPC79 die-type BK). (Melgar Photography)

New Media Proclaim Revolution

As engineers, our ideas are often tested by primal forces, and in the end *what works, works*. No matter how unknown the designer or how controversial the design, if a bridge stands, it stands.

MPC79 *stood*, and with it, the design methods, the instructor's guide, the book, the implementation guide, the course, and many innovative EDA tools and chip designs (see Fig. 17). To most participants it had all seemed pretty straightforward. Taking the courses for granted, most must have thought "I guess this is the way things are done in Silicon Valley." They had passed through a huge paradigm shift [29] without even knowing it, never having designed or implemented prototype chips "the old-fashioned way" – and the entire system of methods had been proven sound by the success of MPC79.

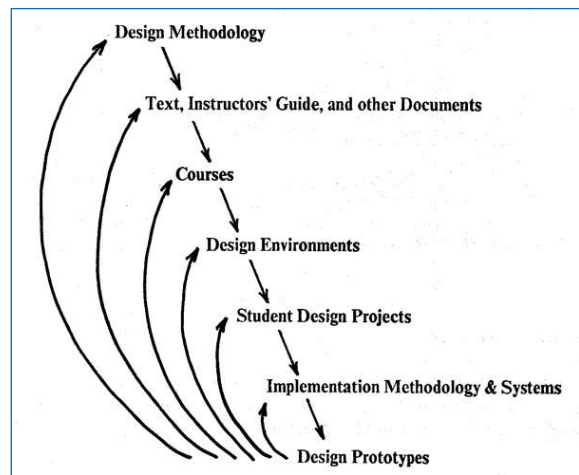


FIGURE 17: The evolution of a multi-level system of knowledge: design projects provide feedback for debugging at all levels [28].

But what about the rest of the world? MPC79 hardly seemed believable unless you were there. Like the Impressionist Movement in France, we needed our own "Salon" – a separate place for showing our works where people could stand back, grasp the thing in its entirety, and see that the new methods *stood*. Badly needed, that level of success wasn't long in coming.

Chuck Seitz had organized the first VLSI Conference at Caltech in January 1979, to provide a forum for the new VLSI systems researchers. In January 1980, a second conference was held at M.I.T., quickly bringing news of the success of MPC79 to an influential audience.

Meanwhile, during the exciting summer of 1979, Doug Fairbairn and Jim Rowson had had the idea of publishing a magazine for the emerging community of VLSI designers and tool builders, and began working on it in parallel with our work on MPC79. The first issue appeared in January 1980 (see Fig. 18), and *Lambda* (later known as *VLSI Design*, then *Integrated System Design Magazine*) soon attracted scores of technical articles about VLSI architectures, design tools and implementation methods [30]. Those articles, along with the many Melgar chip photographs it featured, made *Lambda* a potent medium for spreading the revolution [27], [28].

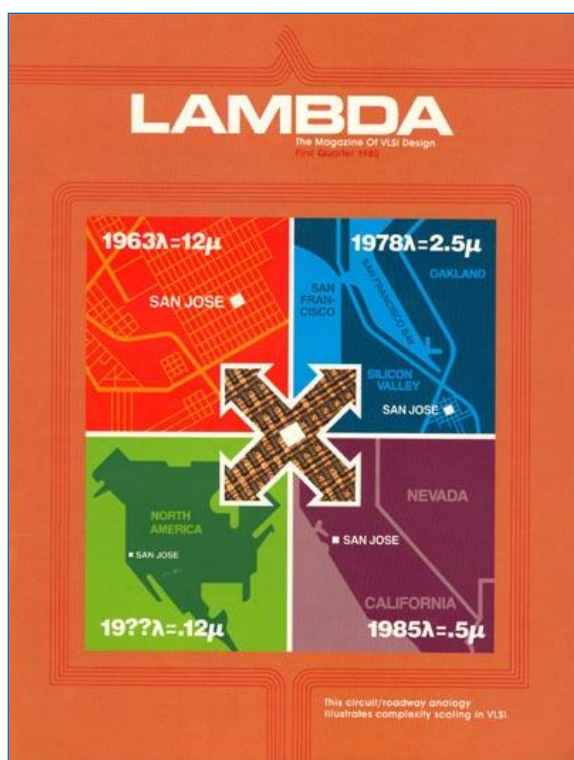


FIGURE 18: The premiere issue of *Lambda*, the Magazine of VLSI Design (1st Qtr, 1980).

In another exciting move, Fairbairn left PARC to become a founding member of VLSI Technology, Inc. (VTI), a company that pioneered VLSI ASIC design. Working with Merrill Brooksby (Manager of Corporate Design Aids at HP and by then a strong advocate of our new methods), Fairbairn also organized the videotaping of a short intensive VLSI Design Course. Fairbairn and Stanford professors

Newkirk and Mathews gave the primary lectures, with guest lectures given by Mead, Lyon, Rowson, Johannsen, Seitz and myself— along with Richard Newton of U. C. Berkeley, Jack Holloway of M.I.T and Jim Clark of Stanford. In addition to wide use within HP, the VTI videotaped courses were run at other places to ramp up their ASIC business. Meanwhile, Jon Allen ran intensive VLSI design summer courses at M.I.T., impacting design practices at DEC and other East-coast high-tech firms. Carlo Sequin also began offering intensive courses in VLSI design, as part of the Hellman Associates Tutorial Series, at many locations around the country.

Mead also began exploring opportunities to capitalize on the work. Always a charismatic personality, he generated lots of buzz among Silicon Valley venture capitalists. In 1981 Mead, along with Dave Johannsen and Ed Cheng, founded Silicon Compilers Inc. to commercialize Johannsen's work. Mead went on to start even more companies as time went by.

Perhaps the most powerful medium for spreading the new methods, however, was the ARPANET, as messages told the story of MPC79. Before long, many more schools around the country began offering Mead-Conway courses, and design tools and design files rocketed across the ARPANET into a growing community of participants, in a huge wave of disruptive technology and innovation.

Struggling to cope with these fast-moving developments, we planned yet another MPC system run in the spring of 1980. Led by Ted Strollo at PARC, the 'MPC580' project implemented 171 VLSI system design projects from 15 different universities and research organizations. It was another crashing success and a further validation of our methods and teachings. These courses generated vast numbers of large check-plots – many appearing in the hallways of EECS departments around the country – and these amazing artifacts attracted even more students to the new movement. VLSI adventurers were the new gang in town, and our graffiti were on all the walls [28]!

As courses spread to major universities all around the world, I struggled to supply startup 'care-packages' of *Instructor's Guides*, *Implementation Guides*, and *Implementation Documentation* from MPC79 and MPC580. But a bigger question began to loom: How to institutionalize the MPC implementation service, and keep it going?

The DARPA VLSI Program

Robert (Bob) Kahn and Duane Adams at DARPA had provided funding for Ivan Sutherland's Silicon Structures Project at Caltech, and with Ivan's guidance had closely followed the subsequent events. The success of the M.I.T. course in the fall of 1978 convinced them that the new Mead-Conway VLSI methods were sound. The publication of the book and success of MPC79 sealed the deal.

Kahn and Adams quickly convinced DARPA's leadership to launch a VLSI Research Program to build on the new methods, and major funding soon flowed into research on new VLSI architectures and EDA tools. Managed initially by Adams in 1980 then by Paul Losleben in 1981 and beyond, the program sponsored tens of millions of dollars in VLSI research. With this level of support, a rush of intellectual adventurers jumped into the movement.

DARPA sponsors MOSIS to Institutionalize MPC79

With DARPA support behind him, Bert Sutherland then solved another big problem: He found a home for the MPC79 technology and implementation service. In the spring of 1980 Bert, Alan Bell, Ted Strollo and I met with Keith Uncapher and Danny Cohen of USC-ISI (a major DARPA software contractor), and arranged a rapid transfer of the PARC MPC system technology and methods of operation to ISI.

ISI soon announced the new "MOSIS" service, and it began operations in early 1981. Prominent Caltech researcher Chuck Seitz later reflected that "*MOSIS represented the first period since the pioneering work of Eckert and Mauchly on the ENIAC in the late 1940s that universities and small companies had access to state-of-the-art digital technology*".

What began in MPC79 as revolutionary technology to advance the VLSI design movement became one of the earliest examples of automated internet commerce. Operating to this day, MOSIS is still housed at the USC facility in Marina del Rey, California [31].

The Paradigm Shifts

That same year, *Electronics Magazine* awarded their Award for Achievement jointly to Mead and me. The magazine's feature article about the VLSI methods, the book and the successes of M.I.T.'78 and MPC79 put the engineering community on high alert that a revolution was at hand [32].

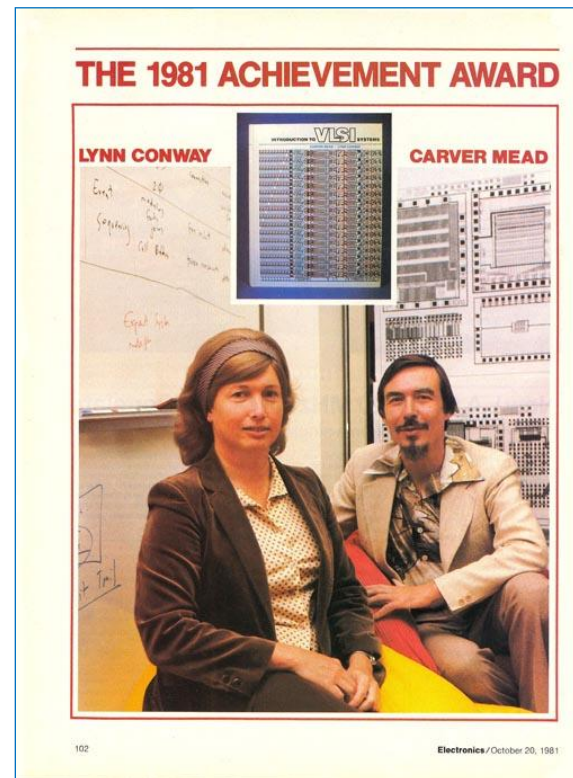


FIGURE 19: Conway and Mead receive the 1981 *Electronics* Award for Achievement.

I had now experienced my "Steinmetz moment", for within two years, 120 universities around the world were offering Mead-Conway VLSI courses, with the book translated into Japanese, Italian, French, and Russian (this last, an "unauthorized" government edition distributed among many Soviet engineers). *Introduction to VLSI Systems* eventually sold around 70,000 copies.

To provide further Mead-Conway-compatible books on key topics, Chuck Seitz and I served as series-editors of Addison-Wesley's new VLSI Systems Series – one of the first being *Principles of CMOS VLSI Design* by Neil Weste and Kamran Eshraghian.

The design-tool building to support early project labs at M.I.T., U. C. Berkeley and Caltech led to rapid evolution of tools for the Mead-Conway methods, triggering an explosion in EDA innovations. This earthquake of innovation, where teams across the globe built on each other's ideas, sharing libraries and tools, presaged and helped lay groundwork for the modern open-source software revolution.

In 1979 two M.I.T. graduate students, Chris Terman and Clark Baker, developed a pioneering set of tools, including a design rule checker, circuit extractor and static checker by Baker, and a switch-level simulator by Terman. The tools provided direct support for 'Mead-Conway design'. They immediately received widespread distribution, and began to change the way people thought about doing their design work. In particular, Baker's circuit extractor was the first time anyone had "closed the loop," making sure that the actual circuit layout implemented the intended circuit – and circuit extraction went on to become a mandatory part of most IC design processes.

During his M.I.T. Ph.D. work in 1979-1980, Randy Bryant originated new methods for switch-level simulation, and he went on to place a much-needed mathematical foundation under switch-level design. By 1983, the MOSSIM-II simulator that Bryant and his students developed (then at Caltech) was in use at Intel. At Caltech, Dave Johannsen also pioneered work on "silicon compilers" which he later commercialized with Mead. John Ousterhout and his students at U. C. Berkeley developed IC layout tools CAESAR and MAGIC, establishing an architectural foundation for many later EDA software systems – including those commercialized by VLSI Technology, Cadence, Valid Logic, Daisy, Mentor Graphics and Viewlogic. Others in the movement went on to play key roles in creating field programmable gate array (FPGA) technology and tools, such as Steve Trimberger at Xilinx.

The architectural work of Jim Clark on the Geometry Engine, and of Steele, Sussman, Holloway and Bell on the M.I.T. Scheme microprocessor gained high visibility through *Lambda* and the VLSI conferences, triggering a rush of additional brilliant young computer scientists and architects into the movement.

After attending Jon Allen's course at M.I.T. in the fall of 1979, Ron Rivest, Adi Shamir and Leonard Adelman implemented their recently invented "RSA Cipher" in VLSI using MPC79. At U.C. Berkeley, Dave Patterson and Carlo Séquin led a team that created the RISC-I and RISC-II architectures in VLSI. Carlo reports that this work was inspired in part by a private communication with John Cocke, concerning work on the 801 at IBM -- another pioneering IBM project that was "moth-balled" and only published many years later.

Similarly, at Stanford, John Hennessey, Norm Jouppi, Forest Baskett and John Gill developed the RISC-based MIPS architecture and prototyped VLSI implementations using MOSIS. At UNC, Henry Fuchs and John Poulton developed the Pixel-Planes VLSI raster graphics engine, with assistance from Al Paeth and Alan Bell at PARC.

Dick Lyon at PARC pioneered smart VLSI digital sensors based on lateral inhibition, inventing the optical mouse and implementing a VLSI prototype, and then helped Martin Haerberli and Robert Garner design a chip for Xerox's production Xerox optical mouse. Lyon also demonstrated how to create VLSI architectural methodologies for special applications, using digital signal processing as an example. Lyon and Gaetano Borriello went on to create the first single-chip Ethernet driver-receiver-encoder-decoder, exploiting Lyon's new semi-digital methods.

The collaborations between PARC and HP, Caltech and Intel, and MIT and DEC led to rapid infusions of the Mead-Conway methods into those various firms. VLSI architectural research also led to parallel VLSI processors such as the Connection Machine by Danny Hillis at M.I.T., the Cosmic Cube by Chuck Seitz at Caltech and the WARP Processor by H. T. Kung at CMU. Such research was increasingly funded by DARPA and led to many important startups, including Silicon Graphics, MIPS and Sun.

MOSIS was initially closed to those outside the U.S., triggering the launch of similar systems in other countries. DEC computer architect Craig Mudge returned to his native Australia to found the CSIRO VLSI program and AUSMPC service, and my team at PARC assisted in those efforts. Reiner Hartenstein, a professor at Technische Universität Kaiserslautern then visiting U.C. Berkeley, returned to Germany, began teaching the course, and spearheaded Germany's E.I.S. service – and he and Klaus Wölcken also began advocating for a larger European-wide service. Ole Olesen from Denmark and Christer Svensson from Sweden formed the Nordic Multi-Project Chip organization and Francois Anceau founded the Circuits Multi-Projets (CMP) service in France, led in later years by Bernard Courtois. Roger Van Overstraeten and Hugo De Man founded IMEC in Belgium, which provided a similar service (The 'EUROCHIP' service, formed in 1989, built upon these earlier efforts.)

With many researchers exploiting MPC79, MPC580 and then MOSIS, and with hundreds of bright students emerging from universities and expecting access to silicon as they had experienced in school, commercial “foundries” of various forms started up to meet the demand for manufacturing of independently designed chips.

The first was SynMOS, founded by Larry Matheny and Bob Smith in September 1980, serving as an agent/broker between design groups and mask and fab firms. Building on the knowledge generated by MCP79 and MPC580, VTI soon offered similar services, and by mid 1982 a special issue of *VLSI Design Magazine* identified 38 such companies; some were fabless firms such as SynMOS and VTI, while others were front-offices to existing fab firms.

Everything really took off as venture capital firms funded scores of entrepreneurial startups of VLSI design companies, EDA companies and foundry services – triggering the rapid evolution of what is now called the “fabless/foundry” business model, as a growing fraction of the semiconductor industry.

Some Reflections at the time

Reflecting on all this at the time, I thought back to my years at Columbia where I had minored and read widely in cultural anthropology – being particularly intrigued by processes underlying the diffusion of innovations. I realized that somewhere along the way, having recalled Everitt Rogers’ early book on the topic [33], I had mounted a meta-level exploration in ‘applied anthropology’ that ran in parallel with and guided my design of the VLSI design methods.

In my early VLSI work this involved the deliberate selection, structuring and encoding of the knowledge so as to have a good ‘impedance match’ with the culture of the targeted recipient communities, and with the simplification of that knowledge by creation and adoption of unifying open standards.

By the time of MPC79, this meta-level thrust shifted into enhancing the noticeability of the significance of the new knowledge via dramatic visible artifacts, the rapid diffusion of those artifacts (and with them the new knowledge) through cleverly augmented diffusion channels, and the provision of means for immediate exploitation of the knowledge via the new QTA implementation service – all leading to more artifacts and thus ‘gain’ in the knowledge propagation process.

The emerging internet and PC technology enabled me to operate in wholly new ways as an architect of disruptive change. Almost no one at the time could visualize what I was actually doing, thus I needed no ‘permission’ to do it and no one was power-positioned to stop it. As a corollary, few folks later understood what had really happened – much less who had done it. Participants simply slid through the resulting paradigm shift, and ran with the results.

A concise history of these unfolding events is given in the book *Funding a Revolution*, published by the National Academy Press in 1999, revealing the impact in academia and industry of the Mead-Conway design methods, the textbook, the VLSI design courses and the MOSIS infrastructure [34].

Ivan Sutherland’s challenge had been met, inventive simplifications being the key to success. Along the way we’d secured “freedom of the silicon press”, and great novels were now being written.

Along with the thrusts in personal computing at PARC and in the Valley beyond, and the vigorous entrepreneurial engineering culture they propagated, these collective events within ten years spelled doom for the domineering IBM of old. What a dramatic reversal of our mutual fortunes since that terrible time in 1968 when I was fired by IBM – a firing that could have shattered my life back in those days.

On to New Things

By 1981, the VLSI work was well on its way. Bert thought it time to move on, and I founded the Knowledge Systems Area at PARC to explore artificial intelligence and collaboration technology.

Even so, I was often asked to speak about VLSI. I gave the opening talk at the 2nd Caltech Conference on VLSI in 1981, describing the interactive meta-level research methods I had used to generate, test, validate and propagate the Mead-Conway methods [28], [35], [36]. I also keynoted IEEE Comcon Spring 1983 and the ACM/IEEE Design Automation Conference in 1984. Although reported to have given outstanding talks, as a still somewhat-reserved person I found these experiences a bit intimidating, and as the VLSI revolution went *viral* I pulled back from additional public exposure. In contrast, Mead was now in his element. Armed with top-level connections and an outgoing personality, he soared toward fame as one of the “founding fathers” of *Silicon Valley* [37].



Figure 20: Lynn Conway in her office at PARC in 1983. (Photo by Margaret Moulton, Palo Alto Weekly)

In 1983, Bob Cooper, Director of DARPA, asked me to lead the planning of a new program called Strategic Computing. The agency wanted to organize a coordinated research program in artificial intelligence, computer architecture, VLSI design and QTA prototyping to create a rich technology base for intelligent weapons systems. Reflecting on my father's leadership role in the WWII synthetic rubber program, I took the mission, planning to return to PARC after my tour. My secretive past was never an issue; I was granted a Top Secret clearance.

I'm proud of the resulting *Strategic Computing Plan*, for it quickly triggered over \$100 million in funding for important computing research. I imagine it also discouraged the Soviets, as they watched brilliant U.S. researchers reach far beyond what they could hope to achieve behind the Iron Curtain [38].

While at DARPA, I got a call from Jim Duderstadt, Dean of Engineering at the University of Michigan, asking if I'd consider a faculty position along with a position in his office as Associate Dean.

I had served on the Engineering College's National Advisory Committee, and realized that it was a time of exciting expansion at the College. *The Valley* had also become so career and money obsessed I found it hard to form good relationships there. In 1985, I took the job at Michigan and "got a life".



Figure 21: Mead and Conway receiving the Wetherill Medal at the Franklin Institute in 1985.

Confronting the Past, Coming Out, Moving On

Thirteen years later, in late 1998, I casually typed the word "superscalar" into an internet search and up popped: "ACS--The first superscalar computer?"

Professor Mark Smotherman at Clemson University had stumbled onto information about the old project, and theorized in his website that ACS was indeed the first. This had become a question of historical interest, because of the success of the Intel Pentiums and other superscalar microprocessors. Stunned, I realized the story of my involvement would come out, and that I needed to get out ahead of it.

I contacted Mark and gradually revealed my role in the project. Fortunately, I had saved all my ACS documentation including the original DIS report. I shared these with Mark and pointed him to other project veterans who might be able to find additional documents; in July 1999 Mark organized an ACS reunion at IBM Research, in Yorktown Heights, to encourage this effort (see Fig. 22). I also began posting information on my website to quietly explain my long-ago transition to my colleagues, hoping times had changed and some would understand.



Figure 22: ACS Reunion, July 29, 1999: (L-R) John Cocke, Fran Allen, Herb Schorr, and Lynn Conway. (Photo by Mark Smotherman)

Michael Hiltzik of the *L. A. Times* had earlier interviewed me while writing *Dealers of Lightning*, his definitive book about Xerox PARC. He became eager to report this further story, and his article “Through the Gender Labyrinth” ran on November 19, 2000 [39]. Since then I have interacted with thousands of other gender transitioners via the internet – expanding my website’s informational support as time went along. My website, lynnconway.com, has served as a beacon of hope for transitioners all around the world, and this work has given further meaning to my life.

During the early 2000’s, Smotherman compiled a comprehensive history of IBM-ACS in his website with the help of many ACS vets [2]. In February 2010, the Computer History Museum in Mountain view, California, hosted a special event to honor surviving veterans of the forgotten project. Around that same time I also received the IEEE Computer Society’s Computer Pioneer Award, based in part on my work on dynamic instruction scheduling (DIS) [40]. It felt wonderful to see that work, done and then lost so long ago, finally acknowledged.

Finding Closure

In reviewing my story I am struck by my good fortune of having worked at two of the greatest research outfits in computing: IBM Advanced Computing Systems in the 1960s and Xerox PARC in the 1970s. Undeniably cool ideas beamed down to researchers at those places, and creative people pulled together to really make things happen based on those ideas.

Along the way, ACS pioneered the superscalar computer architecture so important today, and the PARC/Caltech collaboration launched the VLSI Revolution. What a thrill it has been to watch our ideas become reality, ideas that have changed the world forever.

I’ve also experienced a very special personal closure: The VLSI revolution enabled my DIS invention to finally come to life, to be implemented in silicon – and while I was still around to see it happen.

What a ride it’s been!

Acknowledgements

Many people named in these reflections played vital roles in the VLSI design revolution; it was a thrill to join them on this great adventure. I especially want to acknowledge and thank [W. R. \(Bert\) Sutherland](#) [41], [42]. Without Bert’s wisdom and guidance, the Mead-Conway revolution would never have happened.

References

- [1] M. Smotherman and D. Spicer, "Historical Reflections: IBM's Single-Processor Supercomputer Efforts – Insights on the pioneering IBM Stretch and ACS projects", *Communications of the ACM*, vol. 53, no. 12, pp.28-30, December 2010.
<http://cacm.acm.org/magazines/2010/12/102128-ibms-single-processor-supercomputer-efforts/fulltext>
- [2] M. Smotherman, "IBM Advanced Computing Systems (ACS) -- 1961 – 1969", historical reconstruction website, Clemson University.
<http://www.cs.clemson.edu/~mark/acs.html>
- [3] L. Conway, "IBM-ACS: Reminiscences and Lessons Learned from a 1960's Supercomputer Project"; in: C. B. Jones, J. L. Lloyd, (Eds.), *Dependable and Historic Computing: Essays Dedicated to Brian Randell on the Occasion of his 75th Birthday*, Springer-Verlag, Berlin, 2011, pp.185-224.
<http://www.springerlink.com/content/u8952k538h128472/>
http://ai.eecs.umich.edu/people/conway/Memoirs/Lynn_Conway%27s_ACS_Reminiscences.pdf
- [4] L. Conway, B. Randell, D. Rozenberg, D. Senzig, "Dynamic Instruction Scheduling," ACS Memorandum, IBM-ACS, February 23, 1966.
<http://ai.eecs.umich.edu/people/conway/ACS/DIS/DIS.pdf>
- [5] L. Conway, "Lynn Conway's IBM-ACS Archive", *lynnconway.com*.
<http://ai.eecs.umich.edu/people/conway/ACS/Archive/ACSarchive.html>
- [6] H. Garfinkel, *Studies in Ethnomethodology*, Prentice Hall, Englewood Cliffs, N.J., 1967.
<http://en.wikipedia.org/wiki/Ethnomethodology>
<http://www.amazon.com/Studies-Ethnomethodology-Social-political-theory/dp/0745600050>
- [7] L. Conway, "The Computer Design Process: A Proposed Plan for ACS", ACS Memorandum, IBM-ACS, August 6, 1968.
<http://ai.eecs.umich.edu/people/conway/ACS/DesProc/DesignProcess.pdf>
- [8] B. O. Evans, "The Ill Fated ACS Project": pages 27-28 in Evans' memoir *The Genesis of the Mainframe*, Wilhelm G. Spruth, ed., University of Leipzig, Department of computer science, June 2010.
<http://www.informatik.uni-leipzig.de/cs/Literature/History/boevans.pdf>
- [9] L. M. Terman, "MOSFET Memory Circuits", Invited Paper, *Proceedings of the IEEE*, vol. 59, no. 7, pp. 1044-1058, July 1971.
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1450263
- [10] G. E. Moore, "Cramming more components onto integrated circuits", *Electronics*, vol. 38, no. 5, pp. 114-117, April 19, 1965.
ftp://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf
- [11] R.F. Lyon, p.e.c., March 29, 2012.
- [12] B. Hoeneisen and C. Mead, "Fundamental Limitations in Microelectronics-I. MOS Technology," *Solid State Electronics*, vol. 15, pp. 819-829, 1972.
- [13] R. Bassett, *To the Digital Age: Research Labs, Start-up Companies, and the Rise of MOS Technology*, Johns Hopkins University Press, Baltimore, 2002.
<http://www.amazon.com/Digital-Age-Research-Companies-Technology/dp/0801868092>
- [14] R. K. Booher, "MOS GP Computer," *Proceedings of the Fall Joint Computer Conference*, AFIPS '68 (Fall, part 1), San Francisco, December 9-11, 1968, pp.877-889.
<http://origin-www.computer.org/plugins/dl/pdf/proceedings/afips/1968/5072/00/50720877.pdf>
- [15] L. Boysel and J. Murphy, "Four-phase LSI logic offers new approach to computer designer", *Computer Design*, pp. 141-146, April, 1970.
http://en.wikipedia.org/wiki/Four-Phase_Systems
- [16] F. Faggin, "The Making of the First Microprocessor", *IEEE Solid-State Circuits Magazine*, vol. 1, no. 1, pp. 8-21, Winter 2009.
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4776530
- [17] R. Bower and R. Dill, "Insulated gate field effect transistors fabricated using the gate as source-drain mask", *Proceedings of the Electron Devices Meeting 1966 International*, Washington, DC, October 26-28, 1966, pp. 102–104.
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1474563
- [18] R. H. Dennard, F. H. Gaensslen, H. N. Yu, L. Rideout, E. Bassous and A. R. LeBlanc, "Design of Ion-Implanted MOSFET'S with Very Small Physical Dimensions", *IEEE Journal of Solid-State Circuits*, vol. SC-9, no. 5, pp. 256-268, October 1974.
http://www.eng.auburn.edu/~agrawvd/COURSE/E6270_Spr09/RE_AD/Design%20of%20ion-implanted%20MOSFETs%20with%20very%20small%20physical%20dimensions.pdf
- [19] I. Sutherland, C. Mead and T. E. Everhart, "Basic Limitations in Microcircuit Fabrication Technology", *ARPA Report R-1956-ARPA*, Published by Rand Corporation, Santa Monica, CA, November 1976.
<http://ai.eecs.umich.edu/people/conway/VLSI/BackgroundContext/SMErpt/SMErpt.pdf>
- [20] I. Sutherland, "The Problem: How to build digital electronic circuits from now to 1985", Letter to W. R. Sutherland describing challenges presented by advances in microelectronics and proposing the Xerox-PARC/Caltech collaboration, January 26, 1976.
<http://ai.eecs.umich.edu/people/conway/VLSI/BackgroundContext/IvanSutherlandLetter1-26-76.PDF>
- [21] L. Conway, Ed., "The VLSI Archive: An online archive of documents and artifacts from the Mead-Conway VLSI design revolution", *lynnconway.com*.
<http://ai.eecs.umich.edu/people/conway/VLSI/VLSIarchive.html>

- [22] I. Sutherland and C. Mead, "Microelectronics and Computer Science", *Scientific American*, pp. 210-228, November 1977.
<http://ai.eecs.umich.edu/people/conway/VLSI/BackgroundContext/SciAm/SM.SciAm77.pdf>
- [23] C. Mead and L. Conway, *Introduction to VLSI Systems*, Xerox PARC, laser-printed prepublication versions: Chapters 1-3, October 1977; Chapters 1-5, February 1978; Chapters 1-9, July 1978.
<http://ai.eecs.umich.edu/people/conway/VLSI/VLSIText/PP-V1/V1.pdf> (11.4mb)
<http://ai.eecs.umich.edu/people/conway/VLSI/VLSIText/PP-V2/V2.pdf> (8.9mb)
<http://ai.eecs.umich.edu/people/conway/VLSI/VLSIText/PP-V3/1s/V3.one-sided.pdf> (28.3mb)
- [24] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, Reading, MA, 1980.
<http://www.amazon.com/Introduction-Vlsi-Systems-Carver-Mead/dp/0201043580>
- [25] R. Hon and C. Sequin, *A Guide to LSI Implementation*, Xerox PARC Technical Report, 1st Ed., September 1978; 2nd Ed., January 1980.
<http://ai.eecs.umich.edu/people/conway/VLSI/ImplGuide1/ImplGuide1stEd.pdf> (5.6mb)
<http://ai.eecs.umich.edu/people/conway/VLSI/ImplGuide2/ImplGuide2ndEd.pdf> (6.9mb)
- [26] L. Conway, *The M.I.T.'78 VLSI System Design Course: A Guidebook for the Instructor of VLSI System Design*, Xerox PARC, August 12, 1979.
<http://ai.eecs.umich.edu/people/conway/VLSI/InstGuide/InstGuide.pdf> (12mb)
- [27] L. Conway, A. Bell and M.E. Newell, "MPC79: The Large-Scale Demonstration of a New Way to Create Systems in Silicon", *Lambda, the Magazine of VLSI Design*, vol.1, no. 2, pp. 10-19, Second Quarter 1980.
<http://ai.eecs.umich.edu/people/conway/VLSI/MPC79/MPC79Report.pdf>
- [28] L. Conway, "The MPC Adventures: Experiences with the Generation of VLSI Design and Implementation Methodologies", *Xerox PARC Technical Report VLSI-81-2*, January 1981.
<http://ai.eecs.umich.edu/people/conway/VLSI/MPCAdv/MPCAdv.pdf>
- [29] T. Kuhn, *The Structure of Scientific Revolutions*, 2nd Ed., University of Chicago Press, Chicago, 1970.
- [30] D. Fairbairn and J. Rowson, "From the Editors", *Lambda, the Magazine of VLSI Design*, vol. 1, no. 1, pp. 2-3, First Quarter, 1980.
<http://ai.eecs.umich.edu/people/conway/VLSI/VLSIDesMag/Lambda1Q80.pdf>
- [31] MOSIS, "The MOSIS Service – More than 50,000 designs in over 25 years of operation", mosis.com.
- [32] M. Marshall, L. Waller, and H. Wolff, "The 1981 Achievement Award: For optimal VLSI design efforts, Mead and Conway have fused device fabrication and system-level architecture", *Electronics*, vol. 54, no. 21, pp. 102-105, October 20, 1981.
<http://ai.eecs.umich.edu/people/conway/VLSI/Electronics/ElectrAchievAwd.pdf> (3.6mb)
- [33] E. M. Rogers, *Diffusion of innovations*, Free Press, New York, 1962.
http://en.wikipedia.org/wiki/Diffusion_of_innovations
http://en.wikipedia.org/wiki/Everett_Rogers
<http://www.amazon.com/Diffusion-Innovations-5th-Everett-Rogers/dp/0743222091>
- [34] Computer Science and Telecommunications Board, National Research Council, *Funding a Revolution: Government Support for Computing Research*, National Academy Press, Washington, DC, 1999, pp. 113-122.
<http://books.nap.edu/html/far/>
- [35] M. Stefik and L. Conway, "Towards the Principled Engineering of Knowledge", *AI Magazine*, vol. 3:3, pp. 4-16, Summer 1982.
<http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/374>
- [36] L. Conway, "The Design of VLSI Design Methods", *Proc. of ESSCIRC'82: Eighth European Solid-State Circuits Conference*, Brussels, September 22-24, 1982, pp. 106-117.
<http://ai.eecs.umich.edu/people/conway/VLSI/ESSCIRC82/The%20Design%20of%20VLSI%20Design%20Methods.pdf>
- [37] M. Cassidy, "Chip inventors getting their due at Hall of Fame induction", *San Jose Mercury*, April 30, 2009.
<http://www.tmcnet.com/usubmit/2009/04/30/4158801.htm>
<http://www.forbes.com/2009/05/04/computer-inventors-induction-leadership-citizenship-halloffame.html>
http://news.cnet.com/2300-1001_3-10000815.html
- [38] D. B. Davis, "Assessing the Strategic Computing Initiative", *High Technology*, vol.5, no.4, pp. 41-49, April 1985
<http://ai.eecs.umich.edu/people/conway/CSE/SCI/HighTechnology4-85.pdf>
- [39] M. A. Hiltzik, "Through the Gender Labyrinth: How a bright boy with a penchant for tinkering grew up to be one of the top women in her high-tech field", *Los Angeles Times Magazine* (Cover Story), pp. 12-17, Sunday, November 19, 2000.
<http://ai.eecs.umich.edu/people/conway/Media/Through%20the%20Gender%20Labyrinth.pdf>
- [40] IEEE Computer Society, "Lynn Conway, 2009 Computer Pioneer Award Recipient," January, 2010.
<http://www.computer.org/portal/web/awards/conway>
<http://www.youtube.com/watch?v=i4Txvjia3p0>
- [41] W. R. (Bert) Sutherland, "Management of Industrial Research: Exploring the Unknown Technical Future", *Perspective Series 2008-7*, Sun Labs, July 2008.
http://labs.oracle.com/techrep/Perspectives/PS_2008_07.pdf
- [42] W. R. (Bert) Sutherland, "Faith, Funds, & Fate: Prerequisites for the development and transfer of new technology", Presentation at the celebration of the 40th anniversary of USC-ISI, Marina del Rey, California, April 26, 2012.
http://ai.eecs.umich.edu/people/conway/VLSI/MOSIS/ISI_40th/FF_F.pdf
- [43] L. Conway, "The VLSI Archive", *Electronic Design News*, June 3, 2009.
<http://edagraffiti.com/?p=101>

About the Author

Lynn Conway (lynn@ieee.org) is Professor of Electrical Engineering and Computer Science, Emerita, at the University of Michigan.

After earning her B.S. and M.S.E.E. from Columbia University's School of Engineering and Applied Science in 1962 and 1963, Lynn joined IBM Research. There she made foundational contributions to computer architecture, including the invention of multiple-out-of-order dynamic instruction scheduling (DIS). Fired by IBM as she underwent her gender transition in 1968, Lynn started her career over again in a new identity, soon becoming a computer architect at Memorex Corporation.

Joining Xerox's Palo Alto Research Center in 1973, she invented scalable MOS design rules and highly simplified methods for silicon chip design, conceived of and became principal author of the famous Mead-Conway text, and pioneered at M.I.T. the intensive university course that taught these methods – thereby launching a world-wide revolution in VLSI system design in the late 1970's.

Lynn also invented the internet-based, rapid-chip-prototyping infrastructure institutionalized by DARPA as the MOSIS system – supporting the rapid development of thousands of chip designs, and leading to many Silicon Valley startups in the 1980's.

After serving as Assistant Director for Strategic Computing at DARPA from 1983-85, Lynn joined the University of Michigan as Professor of EECS and Associate Dean of Engineering, where she continued her distinguished career. Now retired, she lives with her engineer husband Charlie on their 23 acre homestead in rural Michigan. They've been together for 25 years.

An IEEE Fellow, Lynn holds five U.S. Patents and has received a number of professional honors for her work, including the *Electronics* Award for Achievement (1981), the Pender Award, University of Pennsylvania (1984), the Wetherill Medal of the Franklin Institute (1985), election to the National Academy of Engineering (1989), and the Computer Pioneer Award of the IEEE Computer Society (2009).



The VLSI Archive

When reflecting on the past with friends and family, we often use photo albums to trigger shared memories – memories that bind us together and reveal how we got to where we are.

However, what of our careers? Although the final products of our work may remain, mementos of our adventures along the way are often lost in the rush of events. Only too late we realize what we should have saved.

But it was different for the VLSI revolution. Perhaps it was the exciting visual artifacts, or the shared-sense that we were breaking new ground. Whatever the reasons, many participants saved original treasures from that era – research notes, chips and chip photos, even huge color check plots – storing them away for decades.

During the past few years members of the VLSI research team, along with colleagues in academia and industry, have gathered up, scanned and photographed many of those artifacts and posted them online. A work in progress, the '[VLSI Archive](#)' helps bring those exciting days back to life [21], [43].



[Lynn's Website](#)



[The VLSI Archive](#)