Title	A Personal Journey Through the Information Revolution
Author(s)	Carver Mead
Language	English
Event title	The 2022 Kyoto Prize Commemorative Lecture
Publisher	Inamori Foundation
Issue Date	01/31/2024
Start page	1
End page	13
URL	https://www.kyotoprize.org/wp-content/uploads/2024/01/2022_mead_en.pdf

URL for Japanese translation: <u>https://www.kyotoprize.org/wp-content/uploads/2024/01/2022\_mead\_jp.pdf</u>

# The 2022 Kyoto Prize Commemorative Lecture Carver Mead A Personal Journey Through the Information Revolution

## Growing up in the Mountain

This is where I grew up (Fig. 1). My father worked in this power plant here (building in the center of Fig. 1), and we lived in this little house, right there (upper left from the power plant), a little group of houses for the people that worked in the power plant. Up here on the hill (upper right in Fig. 1), I went to school.

The power plant itself was a high technology thing when I was growing up. There were these great big hydroelectric generators (Fig. 2), which were impressive things, and this control room, up here (upper left in Fig. 2), full of amazing instruments for watching over those big machines.

The school I went to was this one (Fig. 3), was a one room schoolhouse. One teacher, there she is, there (rightmost in the last row). She taught all of students from first to eighth grade. It was a great way to go to school. I'm next to the teacher. When I graduated from that school in eighth grade, I moved away to town and went to high school.





Fig. 1

Fig. 2



Fig. 3

## Undergraduate at California Institute of Technology (Caltech)

I was admitted to Caltech in 1952. This is what Caltech looked like then (Fig. 4). I went through the usual Caltech undergraduate courses. Some of them were absolutely off-scale fantastic. I learned things I didn't even know were there. Some courses were just hard because they were required courses and they were hard.

I remember one course when I was a junior taught by Charlie Wilts, and it was a great course on electronics. One of the things we learned was how to make electrical analogs of mechanical systems. We learned, on this device (Fig. 5). This is, I think, the world's largest analog computer at the time. Of course, when we did our little projects, we would just use small bits of what was there—inductors, capacitors and operational amplifiers. I liked what I was learning so much, I decided to stay for grad school.





Fig. 5

Fig. 4

# **Thesis on Bipolar Transistors**

I did a thesis on the transistors we had then, which were bipolar transistors. I studied how they behaved as digital switches. A bipolar transistor (shown in the left circuit diagram in Fig. 6), when you turn it on, the current rises (as shown in the center graph) because the minority carriers are rising in value. When the current gets to a certain point (at  $t = t_r$  in the center graph) in Fig. 6), this collector (shown as a right upward line next to the rectangular in the left graph) gets to ground potential and won't go any further, but the minority carriers in the base keep building up. So, when you try to turn the device off at this point in time here (at start point in *t*-axis in the right graph of Fig. 6), nothing happens. And you wait, and you wait, and you wait. And then things start to happen (at the point in time after  $t_s$ ), and by the time you're out here (after another  $t_d$ ), the device is switched. So, I concluded that these devices were not going to be what you'd really want for making digital logic. But it's what we had so it's what we used. We all did.



## **Electron Tunneling**

I was just finishing my thesis on this topic and joined the Caltech faculty in 1959. We had a seminar by a visitor from Japan, named Leo Esaki (Fig. 7). He had just invented a device that worked by electron tunneling, no minority carriers. I was very jazzed by that because I figured that I'd had enough minority carriers for a lifetime. So, I started studying quantum phenomena of electron tunneling.

It was fantastic. The way it works is this: Let's talk about a specific case. We're going to make a capacitor, a thin layer of insulator, say, a very thin layer of mica or something like that. We'll evaporate metal on both sides. So, we've made a capacitor. In the metal electrodes, the electrons make a big collective wave function (schematically shown in Fig. 8). It's like having one great big macroscopic electron. When it propagates, it carries current with it. Of course, that's what makes a metal a conductor. But when it hits a surface of an insulator, the wave function can't propagate in the insulator, and it dies out exponentially with distance (described in the top figure). So, if the insulator is thick, the wave just bounces off the insulator and doesn't go anywhere (center). But if you make a thin insulator, the wave function hasn't died out all the way and so a little bit of it gets through and propagates in the other metal (bottom).

So, the capacitor is a leaky capacitor, and it gets more leaky as you make the insulator thinner. Well, that's interesting. So, our little group was measuring insulators and semiconductors and how electrons worked in them and got up to where we were good friends with the electron wave function. We worked with insulators as thin as 30 Angstroms (3 nm) (Fig. 9). It was a wonderful time.



#### **Encounter with Gordon Moore and Integrated Circuits**

At this time I was consulting for Gordon Moore. He had come by Caltech in 1960, and I'd got to know him. They'd formed this company called Fairchild Semiconductor, which was making really neat transistors. One day, I went for my consulting day at Fairchild and Gordon said, "Carver, I have something for you." And he gave me this little display (Fig. 10).

These objects are silicon wafers. They're one inch in diameter. That was the standard in the 1950s and early 1960s. These are wafers at various steps in the process of making the first commercial integrated circuit. This is a historic thing. First, you have a wafer (upper left in Fig. 10), then you polish it—there used to be a polished wafer there (upper right), but I needed it for an experiment once so I didn't get it back. Then you oxidize the wafer (second row, left) and then you cut holes in the oxide where you want things to be N-type, and you implant N-type impurities in the silicon (second row, right). And then you oxidize it again and cut other holes in it and you put P-type impurities in (third row, left). We're still building a bipolar transistor here, so that makes the base of the bipolar transistor. Then you oxidize it again and cut holes in the oxide, again, and then implant N-type impurities to make the emitter of the transistor (third row, right). So now, we have a complete NPN transistor. Then you oxidize it one final time and you cut holes again where you want to make contacts (bottom, left). And you put metal on top the whole thing, and then you pattern the metal in these little patterns here (bottom, right), which you can't make out very well. That's the integrated circuit.

That integrated circuit is the simplest one here (upper second from left in Fig. 11). Figure 11

is a picture Gordon gave me in 1965. This was the transistor that they were making in 1959 (upper left). This is the first commercial integrated circuit (upper second from left). As time went on, they figured out better things to do with their integrated circuits. And to do better things, more complicated digital functions, you needed more transistors. So, they found ways to cram more transistors onto a given chip. By 1965, they had done all those on the picture, and Gordon made a plot.



## **Question from Gordon Moore**

The black dots in Fig. 12 are the log of the number of transistors on the chip as a function of the year. This was the one that appeared in his famous 1965 paper on what we now call Moore's Law. And then he drew this very bold dashed line and he kept adding integrated circuits and their transistor count as the years went by. Well, this was a fascinating phenomenon. In this period he said, "Carver, you're working on electron tunneling, and that happens when things get very small—won't that limit how small you can make a transistor and have it still work?" I said, "Yes, certainly will." He said, "Well, how small is that?" "Well," I said, "That's complicated question, but one thing you do know, and that is, that you can make insulators down to 30 angstroms, " which you could see on our tunneling plot. We had done that and measured them. Their current leaked away a little bit, but wasn't bad.

That's not going to stop you using transistors with oxides that are 30 angstrom thick. Well, in those days, the oxides people are working with are a thousand angstrom thick. So, that's a factor of 30 in linear dimension, factor of a thousand in density. That's a long way from 1968 or so, where we were having this discussion. That was a year I got invited to give a talk at a thing called a Device Research Conference. It was a great little meeting. Every year, the IEEE put on this thing and invited the people doing leading edge work in solid state devices. We'd all get together and hear the latest things people were doing and argue with each other—it was great. They had asked me to give a talk. So, I decided I would talk about Gordon's problem, "how are these transistors going to behave as you make them smaller?"



#### **Scaling Law**

Figure 13 is a slide I used at that talk. Now, there are a lot of ways you can make transistors smaller. I just chose the simplest possible way. We have a transistor here (schematic illustration on top in Fig. 13). This is a different transistor than the bipolar ones. It has N-type regions (shaded regions on the left and right). The left is the source and the right is the drain, they're called. And in between, there's this region whose potential is controlled by this gate element up above (shaded rectangular in the center of the illustration), made of some kind of metal. So, it's a metal-oxide-silicon device, MOS, we're still living with them. The simplest way to scale them down and make them smaller is make all those dimensions smaller by the same factor. Makes things easy. Let's make them smaller by a factor of two. Where I had one transistor on a piece of silicon, I'll now have four. I've scaled the voltage down so the electric fields stay the same. That means that the electrons going from the source to the drain will be going at the same speed. But they only have half as far to go. So, the transistor will be twice as fast. So, on a given piece of silicon, I get eight times the computation. That's amazing. Well, a lot of people were saying that the power was going to go through the roof so you'd melt the silicon if you did that. Well, no. If you scale the voltage down like this, the power per unit area stays the same.

So in fact, for the same power and the same piece of silicon, you get computation that goes like the cube of the scaling factor. It's fantastic! Unbelievable! And of course, there was a lot of yelling and screaming there. One of the people over there that was not yelling and screaming, but thinking about it was Robert Dennard, the guy that invented the random access memory that we all use in our cell phones and personal computers. Dynamic RAM was his invention. He's another of the Kyoto Prize laureates. He was thinking about this. Next year he gave a talk where he'd worked the thing a little more carefully and basically agreed with my conclusion. So, now there were two of us, one from industry and one from academia, that believed that this kind of scaling could go quite a ways and it wouldn't violate any laws of physics. There were all kinds of papers on all the horrible things that were going to happen if you made things too small.

Here I was, back in 1968, and I was saying you could make things a thousand times more compact and they wouldn't violate any laws of physics and they'd dissipate the same power per unit Silicon area (Fig. 14). On the way home from that conference, I had a gut feeling that my life had changed forever, because a problem was no longer, "How do you make better transistors?" It was, "How could you ever make anything with 10 million moving parts and have it work?" No way!



Fig. 13



# **Intel Setup and Their First Commercial Microprocessor**

The year 1968 was a watershed year in a lot of ways. I had been consulting for Gordon Moore (left, Fig. 15) and Bob Noyce (right, Fig. 15) since 1960. We were good friends by then. So, Gordon got in touch with me and he said, "Hey, we've left Fairchild. Would you like to join us as a consultant in our new little organization we're putting together to do integrated circuits?" Well, I couldn't say no to Gordon, so I became badge number five at NM Electronics. A while later Andy Grove joined us and he came up with the name Intel. As you know, it's become one of the great companies in the world. This was the start of that. Well, as a consultant, I got to see how they were designing and making masks for their integrated circuits.

Figure 16 is one of the circuits, probably the most famous circuit in the history of Intel—the 4004 microprocessor-first commercial microprocessor and subject of the 1997 Kyoto Prize. But it's just a little 4-bit microprocessor. It's not a very capable thing. And if you look at it, it's a hideously complex pattern. So, how do you design a thing like that, and in particular, how do you make a pattern and get it to go together to actually make the function you want? Non-trivial.





Fig. 15 Gordon Moore and Robert Novce at Intel in 1970 by IntelFreePress is licensed under CC BY-SA-2.0

4004/4004-composite-photo.jpg, CC-BY-SA-3.0

# How They Designed Chips in Those Days

So, I watched how they did it. First of all, someone would make a functional description of

what you wanted the thing to do. Then they would make a logic description of how the functions inside it related to each other. Then someone would take that, a different person, and make a transistor diagram of an electric circuit that did that logic function and was connected in those ways. And then someone else would take a drafting machine and would draw this on a big piece of Mylar, a huge diagram of every element in the layout of these integrated circuits. All you have is a composite diagram of what the various layers should be. But now we need a mask master for each layer so we can pattern each layer in sequence. And that was done in this way:

You took the master diagram, which is on this whole light box here, in this picture (Fig. 17), and you put on top of it a thing called Rubylith. Rubylith is a piece of Mylar with a very thin, transparent red coating on it. And the coating is thin enough that you can cut it with a razor blade and cut the outline of places you want to not have the red coating. And then in this diagram, the razor blade is guided by the very precise thing which allowed motion in either X and Y on a very particular coordinate—It's called a coordinatograph—a big, expensive thing on this big, expensive light table. And then, once you've cut the outline of the thing you want to take away, a different person comes along with a pair of tweezers and peels out the red stuff carefully so it doesn't disturb the red stuff that's supposed to stay there. This is insane! It's hard enough to do this for the integrated circuits with a few thousand transistors. And we're going to be making millions. It's nuts! There had to be a better way. So, I had to do this. I absolutely had to make my own integrated circuit because this was the future. And there's no way I could do it this way. But it turned out, there is a better way. And there was, at the time, a better way.



Fig. 17

## New Methodology—Computer Pattern Generation

Figure 18 is a little drawing of a device called a Gerber plotter. It consisted of a light head, which had a light source in it, that had apertures in front of it that could be focused down here on a sensitive photographic material (rectangular stage at the bottom of Fig. 18). The position of the focus can be moved in X and in Y, and all three things, the light beam shape and intensity, and the motion, X and Y, were controlled by a computer, untouched by human hands. This is a way that scales. I can write a computer program to generate these pattern generator files, and I can make arbitrarily complicated, arbitrarily precise, masters for the chip geometry. This was a huge aha for me. There was a way.

These are the kind of plots (Fig. 19), this is the metal mask for a small test chip. This is 250 times scale of the final chip geometry. Well, now, I had a way of generating the masters for making integrated circuits. And then there were mask houses that would take those and turn them into masks that could be used by the wafer fabrication system, which we called The Fab.

But when you look at the design of something like the 4004 (Fig. 16), to write a program that generates that geometry, it's just as bad as drawing it by hand. It's a horrible thing. So, I don't know yet how to even think about designing a thing that's going to have millions of parts in it.

The more I studied these things, the more I realized that the complexity is not in the transistors, the complexity is at how things are hooked up. And that complicated mess you see is because signals have to get from this transistor to that transistor over there, and they have to get there somehow, and they can't interfere with all the other... It's a mess. So, why don't we think about the flow of information first and make an overall plan for how the information is going to flow in this system. And then if the information has to interact with other information, then where is it going to come from and where is it going to go to? Let's make sure they're hooked up.

And then we'll design the blocks that do the functions on that information so they fit with the communications—the wiring. Well, that was another big aha.



Fig. 18

#### **Mead's First Chip**

Figure 20 is my first chip and it came from that line of thought. What it is, is there are two logic arrays here (left picture in Fig. 20). The first one (left array) takes the inputs, which come in, these (below the left array) are drivers for the inputs, and they drive lines vertically through the left array. The horizontal lines there can be hooked up to make an arbitrary AND function of those vertical inputs.

Those lines run over into the second array (right array) and they have vertical lines which can compute an arbitrary OR function of those AND functions. So now, by just putting contacts in the right places on this array, I can create an arbitrary logic function. Now I can take some of the outputs here (lower left of the right array) and feed them back around to the inputs so the little system can compute the next state it wants to go into from the last state and the information that's stored in this memory (bottom of the left picture). That makes it a general sequential logic machine. This was a tremendous invention. It was so good that three of us had done it independently, one at Texas Instruments, one at Hewlett Packard, and me.

I had a freshman collaborator at the time (Steve Colley) who looked at this thing and made a computer program that simulated what the system would do from just the little 'ones' on these tables here (right picture in Fig. 20). I wrote the code, of course, to make this geometry, this finite state machine with memory.

I took the code to my favorite job shop here in the Los Angeles area. The fellow there made Gerber masters for me. I took the Gerber masters up to MicroMask, which was a mask house that made masks for Intel. They made masks that were suitable for the Intel fab. I had two former students, Gerry Parker, and Ted Jenkins at Intel at the time, and they were willing to run my chips through as an engineering run. And so, I got chips back.

One Saturday morning, I bonded up my chips in this little package here, there it is (small black thing on the white square in the center of Fig. 21), and it's hooked up directly to this little fluorescent display (just to the right of the chip package on the picture), and it's doing exactly what the simulation did! So, this is a physical object that came from a conceptual definition. That's what integrated circuits are. And I had a way of doing that that didn't require a lot of hand drawing and didn't require peeling stuff out of Rubylith. It was a disciplined way of doing design that would scale to millions.





Fig. 21



#### **Multi-Project Chip**

Of course, the students got wind of the fact that this thing actually worked. So, they wanted to learn how to do it. Here's the first class (Fig. 22), we actually had eight students in the first class. I had a semiconductor device course at the time, EE281. So, I just redefined what semiconductor devices were, they're just more complicated than you thought, and taught students how to do Chip Design.

What I taught them was exactly what I had learned in the last three years figuring out how to design my own chip, that you can actually get a conceptual picture of each of the process, steps, and what it does in terms of the structure of the geometry, and what that does in terms of the electrical circuit diagram, and what that does in terms of logic. If you have those conceptual pictures in your head, you can hold the whole process in your head. One person can do this whole thing. I had done it. I think the students are smarter than I am so they'll certainly be able to do it. And they were.

This is the first class chip here (leftmost chip, Fig. 23). Of course, you're not going to make a different wafer run for every student so you make a multi-project chip. Each of these projects here (on the chips) is a student project. The students wrote the code that generated their geometry for their project. And then I wrote the code that would put the projects down on the master chip. And then we got together, like you saw on that picture, and made sure that nobody shorted out anybody else's project.

And then I went down to my friend at the job shop and got Gerber masters for the process and took them up. And MicroMask made mask for me and either Gerry or Ted ran them through the Intel fab. By January in 1972, we had chips and I put them in packages. And then each student got to bond their own project up to the pins on the package because they knew where their project pads were and where they wanted them on the package. Some were better with bonders than

others and some had a harder time getting them working than others, but they all worked. Nobody could tell those students that you couldn't do it, because they had gone from their concept to a physical object that realized that concept. This multi-project chip method of actually getting people their own chip in the first quarter became a thing we did every year.

I was learning how to teach this course so I got better at it. And the students—of course, I had teaching assistants from the prior year, so they knew a lot about this. So, the course was getting better and I was explaining things better. This is 1975, you can see that up here (third one from the left in Fig. 23). By then, there are some graduate students who are doing theses on very capable integrated circuits that were quite unique. By 1976, you can see this big tall one here (on the rightmost multi-project chip), these are bit slices out of pieces for a microprocessor. That was the thesis project of Dave Johannsen.



Fig. 22



# **Example of New Paradigm of Complex Chip Design**

Here he is with a plot of a 16-bit microprocessor, which was state-of-the art at the time (Fig. 24). This layout was generated 100% by his silicon compiler. That's the name we gave the program that would take a high level description, like how many bits in a word and what functions you wanted to have communicating. It arranged them all. That's what the compiler did.

It would arrange what functions you wanted, and they're all connected by a bus. So, it was as a nice regular wiring arrangement. And then there were logic arrays, like the one I showed you before, that gave the functions their personality. So, you could create a wide range of microprocessors just by putting in a high level description. This became the textbook example for microprocessor design from there on, and this methodology is with us to this day.



Fig. 24

## Writing Book with Lynn Conway: Introduction to VLSI Systems

About this time, I got invited to give a talk at Xerox Palo Alto, which was, in the day, a very hot place in computer science. And I talked about what we were doing here. After the talk, I met this lady named Lynn Conway, who was a very bright gal. We had a long talk and she said, "Carver, you've got to write a book about this." I had a lot on my plate in those days, and I said, "If you really think so, why don't you become a co-author and we'll write it together." So, we did.

These are three versions (Fig. 25): This (left) is a 1978 pre-print, 1979 pre-print (center), and the actual book came out in 1980 (right).



#### How was Multi-Project Chip Concept Spread

This is how it went (Fig. 26). These are out of the preface. We started working in 1977, had the first three chapters done by the fall so I could use them in the course. Carlo Séquin, up in Berkeley, taught the course on his own. By 1978, that pre-print that you saw, that Xerox underwrote, was used in four different places, and in fact, all around the country and around the world. But the big thing happened in 1979: Lynn Conway had built up a really good group of people, a lot of whom were from here, and had a very competent arrangement there for designing integrated circuits and handling the logistics.

She put on this thing she called MPC79, where she invited a whole bunch of universities all over the world to contribute chip designs. She got permission from ARPA to use the ARPANET (it

wasn't an internet then), and submit their designs that way. Pat Castro, one of my old colleagues from Fairchild days, was then in charge of the Hewlett Packard fab, and they volunteered to do the fabrication. Chips were shipped back in January, just like we'd been doing at Caltech. So, this internationalized the course that came out of here. And, as they say, the rest is history.



Fig. 26

# **Closing—More Connected World**

Now, you've all seen this picture of the earth at night (Fig. 27). I want to remind you that every person in every one of those spots of light is potentially connected with any other person in any other spot of light. We live in a vastly connected world, and that's totally different from when I got to Caltech in 1952. And it's even different from when I joined the faculty in 1959.

My view is that a more connected world is the first prerequisite to a more enlightened world. I've been privileged to be able to contribute to that evolution process down through the years. On behalf of my students, my collaborators, colleagues, and the people all over the world that made this great transformation possible, I'm deeply, deeply honored to accept this year's Kyoto Prize in Electronics.



Fig. 27

NASA Earth Observatory/NOAA NGDC