

# Transcript: Panel 1: The Microprocessor at 50, Looking back and looking forward, ISCA 2021

June 14, 2021

1:30pm Eastern Time



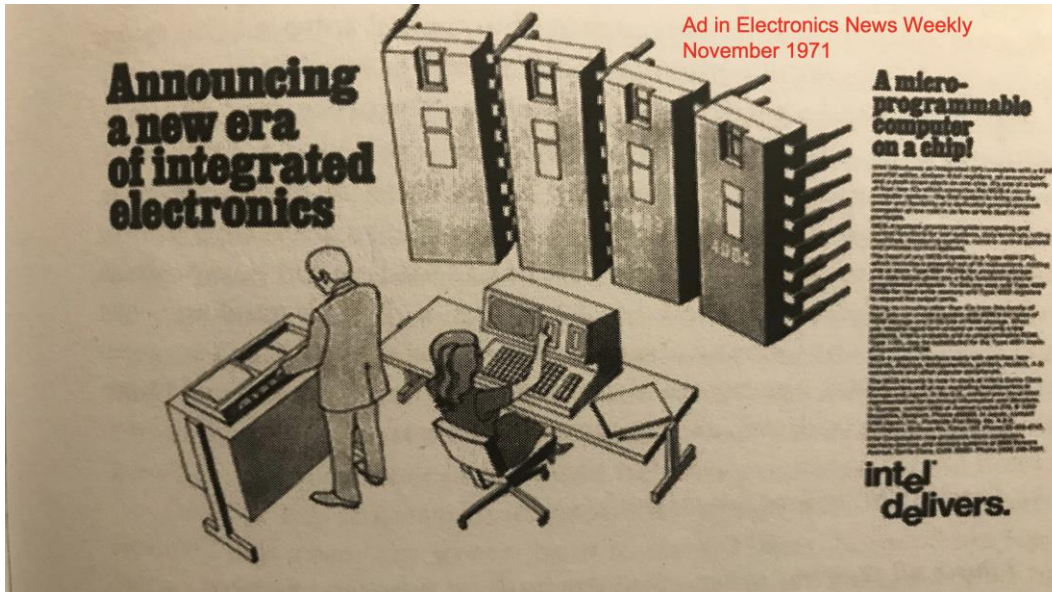
Lizy John:

Panel 1: The Microprocessor at 50, Looking back and looking forward



I am Lizy John, Professor at the University of Texas at Austin. An amazing team of panelists are gathered here to reflect on the successes of microprocessors in the last five decades of microprocessors and also look forward to the next couple of decades.






(Lizy John displayed a Slide with Intel ad in Electronics News Weekly 1971)



This is how Intel announced the 4004 in November of 1971, in a full two-page ad in the Electronics News Weekly. 2300 transistors were integrated into 4004 chip. We have come a long way from there with billions of transistors in current models of microprocessors; and to look at the successes from these five decades, this team of panelists is gathered here. They are creators: designers of original processors, creators of new ideas, founders of various microprocessor companies, an amazing team who have accomplished quite a lot. If I start describing each of their accomplishments, probably, there will be no time left in this panel for them to speak. They're all very well decorated, so I'm going to give a short introduction to all of them. So they get some time to speak, and you get some time to listen to them.

(Lizy John again displayed a Slide and introduces 5 of the panelists)

## The Microprocessor at 50: Looking Back and Looking Forward

**Federico Faggin:** designer of the first commercial microprocessor (Intel 4004), awarded National Medal of Technology and Innovation, Fairchild, Intel, Zilog, Synaptics

**John Hennessy:** co-founder of MIPS Technologies, pioneer of RISC (shared Turing award), Chairman of Alphabet, 10th President of Stanford, Author of popular arch book

**David Patterson:** led Berkeley RISC project (which became the basis for Sun SPARC), pioneer of RISC (shared Turing award), Currently at Google, Author of popular arch book

**Glenn Henry:** designed computers spanning from IBM mainframes to personal computers and custom x86 CPUs, IBM Fellow, Sr. VP Dell, President of Centaur Technology

**Kathy Papermaster:** IBM designer for 26 years, led multiple IBM projects, including the Cell Broadband Engine microprocessor, Director of Sony-Toshiba-IBM Center that designed the Cell Processor

48th Annual International Symposium on Computer Architecture

Federico Faggin, the designer of the first commercial microprocessor, the Intel 4004, he started his career in Italy and then moved to California in the 60s. He was at Fairchild prior to Intel. He founded Zilog, which made the Z80 microprocessors. He was CEO of Synaptics. He was awarded the National Medal of Technology and Innovation in 2009. He just recently published his autobiography, a book called *Silicon: From the Invention of the Microprocessor to the New Science of Consciousness*, and he describes many of the interesting stories around the time the first microprocessor was designed. Welcome, Federico, to ISCA 2021. We are so happy you are here. We congratulate you on your outstanding accomplishments and thank you for participating in this panel.

John Hennessy, the co-founder of MIPS technologies: pioneer of RISC, shared the Turing award with Dave Patterson, Chairman of Alphabet. He was the 10th president of Stanford and author of the popular architecture book co-authored with Patterson.

Dave Patterson: he led the Berkeley RISC project, which became the basis for Sun Sparc, shared Turing award with Hennessy for the pioneering RISC. He is currently at Google. He is co-author of the popular computer architecture books.

Glenn Henry: he's an IBM veteran, and he designed computers and processor chips, spanning from IBM mainframes to personal computers. He left IBM and was senior VP at Dell, prior to founding his own x86 CPU company, Centaur. Currently, he is president of Centaur Technology.

Kathy Papermaster is a semiconductor industry veteran spanning 26 years. Designer at IBM; She led multiple IBM projects, including the Cell Broadband Engine microprocessor. She originally started in Vermont in 1983. She moved to Austin in 1991 and was involved with the design and integration of PowerPC 601, PowerPC 604, PowerPC 970. She was the director of the Sony Toshiba IBM center that designed the Cell processor used in the Sony PlayStation.

[Lizy John continues to introduce the other 5 panelists]

## The Microprocessor at 50: Looking Back and Looking Forward



**Lee Smith:** co-founder of ARM, Arm Fellow, led development of software tools at Acorn and Arm, Joining from Cambridge

**Shekhar Borkar:** directed Intel microprocessor research for 34 years, former Intel Fellow, currently at Qualcomm, Started with Intel 8051 microcontrollers, *iWarp* multicomputer

**Chris Rowen:** co-founder of MIPS Technologies, *Tensilica*, *Babblelabs*, VP of Engineering for Collaboration AI at Cisco, pioneer of hardware software co-design

**J. Scott Gardner,** *consultant/analyst specializing in microprocessors and AI, VP Intrinsicity (acquired by Apple), IDT for 10 years, CEO Nanowatt Design, former Sr. Analyst Microprocessor Report*

**Lizy K. John,** Cullen Trust for Higher Education Endowed Professor at UT Austin, ISCA Program Chair, Editor-in-Chief of IEEE Micro, Fellow of the National Academy of Inventors, Fellow of IEEE and ACM

48th Annual International Symposium on Computer Architecture

Lee Smith: one of the founders of ARM. He is an ARM fellow who joined ACORN in 1983. He led the development of software tools at ACORN and ARM. Right now, he is joining us from Cambridge, UK.

Shekhar Borkar: currently at Qualcomm, but before that, he directed microprocessor research at Intel for 35 years. He is a former Intel Fellow. When he started in 1981 at Intel, he worked on the Intel 8051 microcontrollers,

and later on, iWarp multicomputer and Intel supercomputers. He has been doing research and influencing many of the processors that have come in the past three decades.

Chris Rowen: co-founder of MIPS technologies along with John Hennessy. Chris has the distinction of being an early PhD student of John Hennessy. Chris also founded companies, Tensilica and Babblelabs. Tensilica was bought by Cadence, and Babblelabs was bought by Cisco. Currently, he is VP of engineering at Cisco. He's a big pioneer of the hardware-software co-design philosophies.

Scott Gardner: he's an independent analyst specializing in microprocessors and AI. Currently, he is very active in MLCommons and in the MLPerf benchmarks for machine learning. He was at IDT for 10 years. He was VP of Intrinsity, which was acquired by Apple. He was formerly a senior analyst at Microprocessor Report, and, finally,

I am here. Once I found Scott to moderate this panel, I didn't need to be here, but as ISCA program chair, I wanted to be personally here and welcome all of these distinguished panelists to this event. ISCA is very honored that you have joined us. Without further ado, I want to hand over to Scott to get the panel going, so a warm welcome to all of you panelists to ISCA 2021.

Audience, you should send in your questions. As Scott is moderating. I will watch and monitor the questions, and as you send in your questions, I will ask them to the panelists.

**Scott Gardner:**

Great! Thanks, Lizy, for the introduction, and greetings to our esteemed panelists!. I'd also like to send a shout-out to all of the folks at ISCA. Since this is remote, we actually, I think, have a much larger crowd than a normal ISCA, and I know a lot of press have joined this panel because this is a really historic event having this group of people together. So, I just want to kind of warm things up a little bit and talk about some of the history. I have some show-and-tell, but one of the things I want to make sure everybody understands is that everybody watching and everybody on this panel, owes their career to the microprocessor. We all have great stories about when we got our first computer. Mine was a TRS-80 in 1977 with a Z80, and so I think everybody (Lee was recently telling us his stories) has their own story.

[Scott displays slide with 25<sup>th</sup> anniversary Microprocessor Forum memorabilia ]

The slide features a collage of microprocessor specifications from 1971, titled "1971-25TH ANNIVERSARY-1996 OF THE MICROPROCESSOR". The specifications include:

- 4004: 14.7 mm<sup>2</sup>, 2,300 transistors, November 1971, 10 μm CMOS, 1-layer metal
- Intel: Pentium Pro, 150 MHz, 2.5M transistors, 10 μm CMOS, 4-layer metal
- IBM: PowerPC Super Chip, 330 MHz, 10M transistors, 0.5 μm CMOS, 8-layer metal
- Motorola: PowerPC 603e, 60 MHz, 2.5M transistors, 1 μm CMOS, 4-layer metal
- IBM: PowerPC 604e, 60 MHz, 2.5M transistors, 1 μm CMOS, 4-layer metal
- IBM: PowerPC 401EP, 22 MHz, 300K transistors, 1 μm CMOS, 3-layer metal
- NEC: V851, 22 MHz, 600K transistors, 1 μm CMOS, 3-layer metal
- Alpha: 21164, 200 MHz, 10M transistors, 0.5 μm CMOS, 4-layer metal
- PK-7300LC, 200 MHz, 10M transistors, 0.5 μm CMOS, 4-layer metal
- ULTRASPARC II, 200 MHz, 10M transistors, 0.5 μm CMOS, 4-layer metal
- RS5000 Orion, 200 MHz, 10M transistors, 0.5 μm CMOS, 4-layer metal
- HITACHI: SH7708, 200 MHz, 10M transistors, 0.5 μm CMOS, 4-layer metal
- ARM: ARM7TDMI, 100 MHz, 100K transistors, 0.5 μm CMOS, 4-layer metal
- CLP87310, 100 MHz, 100K transistors, 0.5 μm CMOS, 4-layer metal

Next to the specifications is a close-up image of a 4004 microprocessor chip with the following text:

- 4004
- 14.7 mm<sup>2</sup>
- 2,300 transistors
- November 1971
- 10 μm PMOS
- 1-layer metal

To the right of the chip image is the text: "This is now a 50-year-old microprocessor!"

48th Annual International Symposium on Computer Architecture

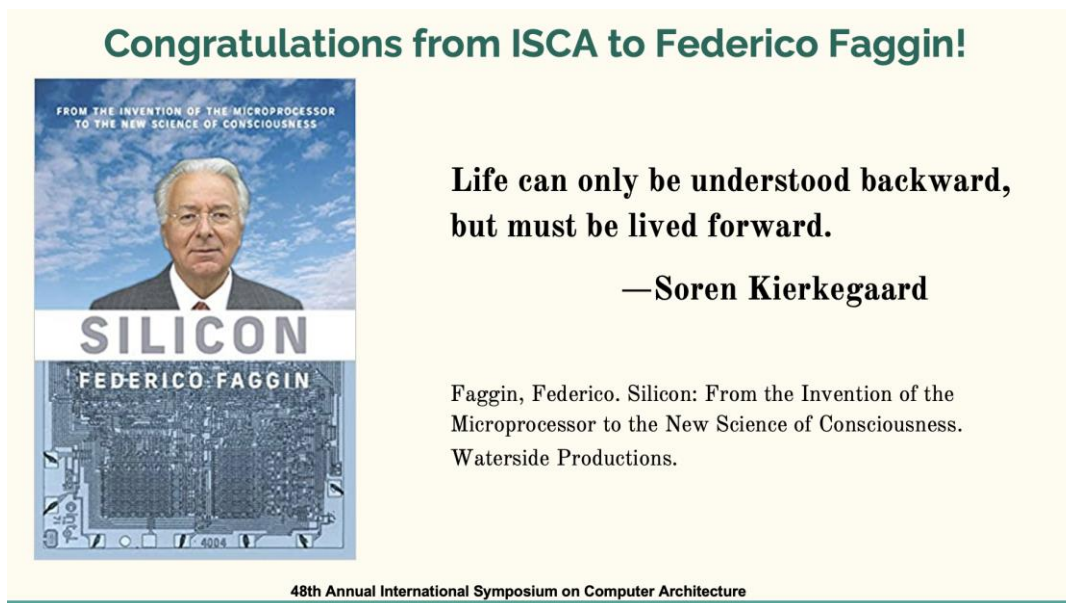
Everybody has been involved in microprocessors, so everybody watching this panel really wanted to see this interaction. So one of the things I want to point out is this is a human-interest panel. We really want to see the

panelists talking to each other. Zoom's going to make it a challenge, but I really encourage the panelists to talk to each other. So this picture: I couldn't find the physical one, but I took this picture a couple of years ago. This was the 25th anniversary of the microprocessor. So, at Microprocessor Forum, which was the annual trek everybody went to announce new processors, they celebrated the 25th anniversary, and in the upper left-hand corner of this folio is a 4004. Note that chip in there is 50 years old today!

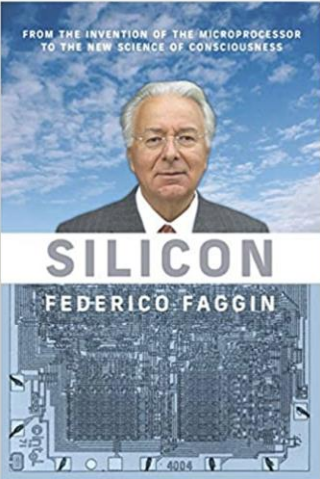
So here we are at the 50th anniversary, and we're going to celebrate that processor. One point I want to make that perhaps explains my own bio before all of these guys get going is, if you see the logo on my shirt, I was at IDT, and we actually have a chip there, the IDT MIPS R5000. So we were one of the semiconductor companies actually building the chips for MIPS. My role was probably the most fun I've had in my career. It was as a field applications engineer, so my task was to go out and talk to system designers about what microprocessor to choose, and it wasn't just the MIPS companies, because IDT also did fast logic and SRAM.

I actually had an opportunity to work with designers all over the world, because I was eventually the worldwide manager, and so all of these chips are part of my life. So I would contend, and you guys can pick a different era, but I would say the 1990s represent the coolest decade of the last 50 years, because I think that's when the most competition was occurring with the most innovation. And so, I'll be interested to hear (maybe Federico thinks it's the first decade), but it but was definitely a really fun time in the 1990s. Let's go to the next slide.

[Scott displays Slide with Federico's book's cover page and quote]



**Congratulations from ISCA to Federico Faggin!**



**Life can only be understood backward,  
but must be lived forward.**

—Soren Kierkegaard

Faggin, Federico. Silicon: From the Invention of the Microprocessor to the New Science of Consciousness. Waterside Productions.

48th Annual International Symposium on Computer Architecture

What I should do here, let's do Federico's book first. I encourage everybody to get Federico's book. I don't know if you guys can see as I get to the camera here, but Federico sent us a signed copy of this [holds up book to camera]. I got an early start reading it on Amazon, but this is his story of how he started back in Italy in World War II. He was born in 1941, and eventually made his way, as Lizy said, to Fairchild and Intel where he invented silicon gate technology. This book tells the whole story about the 4004.

This very first quote [indicating the slide]. Federico fills his book with quotes from philosophers. So, I grabbed this one which was I thought was just fascinating, because this is almost exactly the title of our panel, This fellow here is apparently a Danish existentialist philosopher. It turns out he's apparently the FIRST existentialist philosopher, but, yes, "Life can only be understood backward, but must be lived forward". What we're going to do today is, we're going to look backward 50 years, (and I'll be asking once we do some quick introductions) asking every one of the panelists to talk about some inflection point they saw in the last 50 years and then what they see going forward.

[Scott displays Hennessy Patterson book]

So the other show-and-tell item I have is pretty cool. This is the Hennessy Patterson textbook, and I had to admit that, since I left college, I had never read any other textbook cover to cover. This is actually the Industry Panel for ISCA, but this textbook from academia became a must-read for anybody doing microprocessor design. It was interesting when going through it again. There's a quote here, and the panelists might want to revisit this. We couldn't read it on camera, but this is from Steven Przybylski, one of the early MIPS folks doing memories. He says here, "RISC is defined as any computer announced after 1985". This edition of their book was done in 1990, and we can now revisit that as the definition of RISC, because that was, what, that was 30-some years ago.

[Scott displays Slide with agenda]

## The Microprocessor at 50: Looking Back and Looking Forward

1. Panelists look back at the 50 years of the microprocessor (1 hr)
2. ...Then look forward to the next 25 years... (last 0.5 hr)
3. Encourage interaction between panelists (within Zoom limits)
4. Start with each panelist giving quick intro
5. "What is the academic definition of a microprocessor"?
6. Identify one or two inflection points in last 50 years.
7. Predict future inflection points and the microprocessor @75
8. Audience Q&A based on time availability

48th Annual International Symposium on Computer Architecture

Okay, let's go to the next slide, and I will just do a little bit of the agenda here. As I mentioned, we're going to try to fit everyone in this panel, and I'm going to try to speak as little as possible to make sure the panelists get as much airtime as possible. We'll spend about the first hour looking back, and then we'll look forward. I really want to try to get the panelists to talk to each other, but we're going to start with every single panelist. They will each do about a two-minute introduction to give their long version of their bio, particularly as it relates to their involvement with microprocessors. We'll spin through everybody so that nobody has to wait very long before they get into the discussion.

I might throw out a question, "what is a microprocessor?", and I challenged John Hennessy to just tell us real quickly what would you have told your undergraduates. What made the 4004 a microprocessor? (and maybe even Dave Patterson will correct him or have a different opinion). Then, I want to go through all of the panelists and say, "Okay, it's been 50 years. Just pick those one or two inflection points that got us to where we are, and these may be negative inflection points. Maybe something went sideways, and we'd all be having flying cars if we hadn't, you know, gone this other direction". And then we'll try to get to the future and predict, "what's an inflection point that's going to get us to the microprocessor 25 years from now, and then what will be a microprocessor 25 years from now?".

So, there are no forward-looking statements. Some of these folks work for public companies; these are their personal opinions, but they're going to go on record and tell you when you come back here in 25 years; this is what I predicted the microprocessor would become. Then, as we mentioned, Lizy's going to be monitoring questions, and hopefully, we have a little bit of time for that, I think the best thing is, let's go ahead and start with Federico. Could you go ahead and tell your story in just a couple minutes, and then we'll just keep moving.

**Federico Faggin:** Very good. Thank you!! This story starts in 1968. At that time, I joined Fairchild Semiconductor, and in 1968, pretty much almost all the integrative circuits used bipolar technology. There was an up-and-coming technological (MOS - metal oxide semiconductor). They still had many problems, and I believed that was really the future because you could put a lot more transistors in a single chip, but the technology was extremely slow and unreliable. So, my first job was to actually create a new technology that removed all the limitations of metal gate MOS technology. That was 1968, and at the end of 1968, I had developed the technology and the first commercial integrated circuit using it (the Fairchild 3708), and this technology used silicon instead of metal. (Polycrystalline silicon instead of aluminum) That made possible self-aligned gates, made possible transistors that were five times faster with the same design rules of the metal gate, and also, we could put twice as many transistors in the same chip, because with another invention of mine (with the buried contact), we could connect directly, without metal, the gate with the junction. So, with that technology, now, finally, we had something that would allow us to put, in a single chip, an entire CPU. Why it's important to be in a single chip is because, in those days, the MOS technology had very poor driving characteristics. If you took a signal out of a chip and brought it back into another chip, like you would need, if you had a multi-chip microprocessor, then you would lose so much speed that, basically, it would just not be that useful. Microprocessors can only be useful if they're fast enough so that you can create all kinds of different products with the same device. And so a single chip was necessary, and the chance to do that came at Intel.

Intel had taken the technology that we had developed at Fairchild, and so Intel had the silicon gate technology, and they also had a customer that wanted a custom job. The customer was called Busicom, and Busicom had already developed an architecture that used three chips to make a CPU. But they were using read-write memory. There were shift registers, so the shift register is, you know, are serial memories, which is the only memory that was available in those days, and clearly, you wanted a RAM memory for a computer. So, Ted Hoff saw that opportunity because, at that time, Intel was developing dynamic RAM. DRAM was possible with silicon gate because silicon gate had about a hundred to a thousand times less leakage current, and a DRAM requires very low leakage current to work. So, Ted Hoff with Shima Masatoshi (Shima, the engineer of Busicom), and Stan Mazer; they improved the architecture of Busicom, and they created a four-chip architecture of which the other three chips were a ROM, a RAM, and an I/O [chip] so that we could build computers, different kinds of computers with those four chips.

But, the CPU was way beyond what anybody had ever designed before, so nobody knew how to design it at Intel. I was hired six months later after the project was sitting there, and my job was to figure out how to do it, and given my knowledge of circuit technology. Also, I had designed and built a small computer when I was at Olivetti in 1961. And I knew how to do logic and circuit design, so I had all the pieces necessary to do what was needed. That chip worked in March 1971. It was commercially sold at Busicom in that month, and then I pushed the Intel management to actually sell that chip.

In those days, it [the 4004 design] was only for the customer; it was exclusive to the [Busicom] business. To actually get out of the exclusivity contract and actually sell that chip to everybody (and that was the ad that you saw earlier of the introduction of the 4004 called MCS-4 family before chips in November of 1971). Quickly, I also led the design of the 8008, the first 8-bit microprocessor at Intel, and then I came up with the idea and led the development of the 8080, which was the first microprocessor with performance six times faster than the 8008, and that microprocessor really began to create the microprocessor market.

I then decided that it was time for me to start my own company. I wanted to make a microprocessor company. Intel was a memory company that sold microprocessors to sell more memories. So, for me, that was not acceptable, and I thought the microprocessor would be the future. So, I started Zilog. There I developed the Z80 with Shima, and that microprocessor was a very, very powerful device. In fact, I believe we are the first to use pipelining on the microprocessor—what defined RISC. We had pipelining in the four-bit ALU with which we could, we were doing 16-bit operations, but that was a CISC architecture, and that really started the whole (that plus the 8080 and the 6502) started the microprocessor market, which was really the key, the first major event in the history of microprocessors because that created a business that drove microprocessors for the following 50 years or that, let's say 40 years.

I will stop here, otherwise I'll take too much time.

**Scott Gardner:**

Yeah, that was six and a half minutes, so we are going to have to pick up our pace here, so just a couple of minutes, who's going to be able to get us back on time? John Hennessy, can you give it a shot?

**John Hennessy:**

I can get you back on time. So, I arrived at Stanford, and the first course I taught was a microprocessor lab course using 8080s and then later on from Federico's Z80s, because I remember going down to Zilog and visiting them early on when they were just a startup company. But, my background was really in compilers, and I was doing compiler research, and I realized that we could compile down to a much simpler instruction set and simply eliminate the need to do runtime interpretation through microcode, and that was kind of the way, we came at the RISC ideas. Of course, we could see that Moore's law was going to allow us to do the other things that we thought real computers should do - virtual memory, caches, all those kinds of things. That was going to happen, but I think nobody really saw what it was—that microprocessors were going to take over the entire computer industry, and they were going to do it by 1990, basically. And it happened so fast and so quickly, which is just a tribute to the incredible progress in semiconductor technology. Okay, Scott.

**Scott Gardner**

That's great, and so, Dave, I'm going to go to you because I know, I mean, you might hate this, but I just lump you guys together. It's always we talk about the "Hennessy and Patterson", and so it's like this noun that you guys are joined together, and congratulations on your Turing award! So go ahead and tell us your story about microprocessors.

**David Patterson:**

Yeah, I'm stuck with Hennessy. He's the anchor around my career. [much laughter by panelists] I just agree with everything that John says, but I actually did micro-programming tools. And basically what RISC is an instruction set that doesn't need microcode, and the prevalent reason before then you need to have an interpreter inside your hardware that did the microcode, and that that was basically the insight, and then we all believed in Moore's law like John said. So everything that was in a mainframe computer was going to necessarily be in a computer, and John and I just got working on it. When you could do a 32-bit microprocessor, and the mini-computer industry had just recently gone from 16-bit address computers to 32-bit, then we saw you could do a 32-bit address-based microprocessor. That was feasible, and then the question is, well, does that make sense to do a micro-coded interpreter in a 32-bit microprocessor, and the answer was no.

I think the rest of it was, pretty, not so surprising. It was just, I think, the thing that people don't realize: it was extraordinarily controversial. People thought we were, that we were spreading lies. Yeah, that we were going to damage the computer industry by making simpler the prevailing philosophy that you build in microcode a very high-level instruction set and that would make the software more reliable, and software takes less time, and what we were doing was going to set back the computer industry. So for at least a few years, there were a lot of people who thought we had dangerous ideas, and they tried to suppress it. I'll pass the baton on.

**Scott Gardner:**

Well, let's just keep with the Stanford theme here. Chris, so tell us what you've had to do with microprocessors in the last 50 years.



**Chris Rowen:**

Yes, so I came originally from a physics background and, of course, when fresh out of school, I went to work for Intel, which was a memory company in 1977, and gradually moved more and more towards microprocessors and then, really, applications. And in fact, looking back over the 50 years, the most important development, the most important milestones along the way, is probably not the instruction set architecture battles, but I think there is the great split, the great divide, which really became obvious starting about 25 years ago. There is the mainstream, which is mostly about how we raise the level of abstraction. We raise the convenience. We raise the speed of development, and we have higher and higher-level languages and libraries and operating systems, and cloud-based services which dominate computing today.

And, really, it's about how you make it really easy to deploy new applications, and how efficient it is, is really of second-order, and then the other side of the divide is more domain-specific. It's the really computationally intensive things, led by graphics, AI, and real-time media processing, and they're very different in their computing demands, And that's where you really care a lot about how efficiently you use energy, how efficiently you use silicon real estate, to accomplish a lot of application throughput. And so, standard architectures are about high development efficiency, and domain-specific architectures are really about how you achieve computational efficiency. The example that I would use to characterize it is, in the old days, and in domain-specific architectures, you legitimately ask questions like, what is the energy to do a 32-bit add? And you say, "Oh okay, that's a few tens of femtojoules. Isn't that great, and in this chip, I can do thousands or hundreds of thousands per cycle."

On the other hand, when you ask, "Well, what do people do for adding in the real world?" The most dramatic example is the way you do an add in the real world in the modern application domain is you say, "Hey, Siri, what's 23 plus 46?", and if you then ask, "Well, how much, how many cycles did it take to do that add?" The answer is probably billions of cycles spent doing one add compared to what you would have imagined on a GPU. My career has really followed that, going from, you know, working in the silicon, to working in RISC, to really thinking in the mid to late 90s about domain-specific architectures and creating Tensilica, which explicitly said, "Eh, instruction set architecture doesn't matter, very much."

So, over the course of the next 20 years, Tensilica brought to production through its licensees several thousand distinct instruction set supersets of the basic thing and has delivered through its partners, I think, more than 40 billion microprocessor cores. So, within that, one of those streams became very important, and in the other stream, which is the kind of the mainstream architectures dominated by x86 and ARM instruction set. Eh, not really, really important! It's the biggest part of the industry, but it represents just one part of the great divide.

**Scott Gardner:**

Okay, well, you said ISA doesn't matter. Let me go to Kathy, because IBM felt it necessary to do their own ISA, their PowerPC. You worked with some partners. So, go ahead and tell your story.

**Kathy Papermaster:**

Yeah, so I joined IBM in 1983 in Burlington. I, like some of the others, I love semiconductor physics. I joined as a circuit designer at an interesting time too, when, you know, most of the bipolar work was done in East Fishkill, and they were really the cash cow for IBM. But, in Burlington, Vermont, they had just launched. They went from NMOS to CMOS and silicon gate technology. So I love Federico's book. I love going back to that. It brought back tons of memories, of my memories. So, I was doing circuit design there, but it was for standard-cell library development, but eventually, we said, let's move south; it's too cold here.

We came to Austin, and at the time, Austin, was, you know, big, big, starting with the whole RISC processors for Consumer, and Somerset was the name of the IBM Motorola Apple design center where I started with doing the methodology work, laying the foundation. We had our own new tools, and then I was the 604 integration lead for that and its derivatives, so, you know, focusing on all of the power/performance, design-cost considerations.

From that, that launched into—I mean, that was the 90s, and again, I think— Scott, you mentioned too think the 90s was the best time, but I loved the team I was working on, the whole bit.

And then after that, after the whole Somerset and the PowerPC concluded, the Playstation 3 (the Cell) we launched, and again, I loved it. I was in the early definition phase. We took trips to Japan in a collaborative [effort], you know, just talked with the whole Sony [team]. Toshiba was involved; that transitioned to the design center. I did some other, you know, work on the side, but then finally, when it launched into manufacturing, I was the director, and again, there's a theme here. I love doing these cost migrations, you know, again faster, more cost-effective, profitable et cetera, et cetera, and so that's how I concluded my career, with the whole Playstation 3, and I've been retired the last 10 years,

This has been a lot of fun, listening to everybody here and bringing back all these memories, so thank you.

### **Scott Gardner**

Yeah, glad you could join us. Well, Glenn, are you ready to tell your story, your short story about what you have been doing these years.

### **Glenn Henry**

Yeah, well, as you can see, I've been doing this a long time. I started as an application programmer in 1963. I joined IBM in 1967. I was in the product division my whole 20 years, became an IBM Fellow, and as the old guy here, I guess I'm the guy who's got to say there's nothing new under the sun. When I joined IBM in the 70s, roughly the same time that Federico joined Intel, IBM was shipping parts that had virtual memory, virtual machine emulation, caches, multi-processing, out-of-order execution, a common API across very diverse implementations. And so, it's the silicon technology to me that's the differentiator.

I mean, other than maybe branch prediction and a few other things. Architecture has evolved, naturally, but it's the silicon that's allowed us to reduce the cost, the footprint, and the power along the way. Let me quickly finish because there's one system I want to mention in the spirit of nothing's new. One of my projects that got me to be an IBM fellow was a thing called the System 38. It had an object-oriented instruction set. It had high-level machine stuff. It had a single-level store, the 48-bit virtual address. This is in the 70s. Everything's in virtual address space. There are no files, etc.; it had a built-in relational database and the API etc.. So, there's a lot of developments independent of the silicon technology.

### **Scott Gardner**

So, you want to say anything about going to the dark side and building x86s and PCs?

### **Glenn Henry**

One of the ways to look at things in this industry to me is to look at long-term effects, and the long-term effects, since we're here to talk about the microprocessor, are two-fold. It's the silicon gate technology and the innovations that Federico had that made it realistic, and silicon gate is what allowed the size, the power, etc. to be reduced. But the other implication of what Federico did, was the 8080 which drove the microprocessor, and ultimately there's a direct line between the 8080 and the parts that we do at Centaur and at Intel and at AMD, that are the modern x86 architecture parts now. Now, I think that's a wonderful thing. Not everyone will agree.

So, for Centaur, we've been doing x86 processors for 26 years, and some of our customers are IBM, Dell, HP, Lenovo, and Samsung. So, we're a real company, and I'll thank John Hennessy, who, in a way we won't talk about, actually helped make that happen.

**Scott:**

Okay great. [Glenn has something to add] Go ahead, Glenn.

**Glenn:** I should also thank Federico, because he was on the board of IDT when we got our initial funding from IDT. We're now owned by Via technologies.

**Scott Gardner:**

Yeah, so many of us have this IDT heritage. Let's go to ARM. Meanwhile, on the other side of the pond, this little company was doing low-power processors, and you STILL ARE. You're the RISC processor that's still being very successful. That's incredible!

**Lee Smith:**

Well, kind of. I think RISC and CISC have converged hugely over the years. I'm not sure it's a very meaningful war any longer. I started my career when I was still in high school. I wrote my first program for an IBM 360 in Fortran, which was probably the formative experience. I first came across microprocessors when I was an academic at Edinburgh University. It was probably 1980 or 81, and we were looking to build a 3M machine, a megabyte, and a megapixel. We were surveying the microprocessors available and finding most of the chipsets of the era wanting in one way or another.

To cut a long story short, we eventually built a machine based on 68000. I ported the portable C compiler to it via VAX VMS. I won't go into why it was such a circuitous route, but it was necessary. At the end of it, I was pretty impressed that this thing could hold its own compiling C code with the department's shared VAX 11/780.

That was one-up in my career for believing that microprocessors were going to become real. I moved to ACORN in 1983, and they proudly presented me with the BBC machine, which was their creation, and it had a 6502 inside, and I kind of couldn't take this thing seriously. I guess my history is, I did computer-aided design tools, and then I did compilers, and the reality is 16 bits is just not enough code space for any of those things. You can do overlays on. I've used overlay managers. I've even implemented one. It's completely miserable. DOS extenders weren't much better, so, you know, for me, the real world of computing on integrated circuits began when we got 32-bit micros.

**Scott Gardner**

You had another point, Lee?

**Lee Smith**

Sorry, Scott, just to say that I was at ACORN when ACORN produced the ARM 1.

**Scott Gardner:**

Oh, okay, so how long were you between ARM and ACORN at the same, pretty much same. place? Thirty-something years?

**Lee Smith**

So yeah, seven years of ACORN, and then it's coming up thirty... Eek, I can't do the arithmetic!

**Scott Gardner**

Yeah, I just think that's astounding. So yeah, this is one of the ARM founders talking to us now. Incredible! If you're done, Lee, I'm going to Shekhar. What I'd like you to do is start talking about inflection points, and you said you had a few slides, so I'll let you go ahead and do your bio and roll right into, "Okay, here's what I think, and looking back on the last 50. Because you said you're going to be a little contentious, so I will let you go at it. Give Shekhar a little time to talk about his, you know, his past but then, if anybody on the panel would like to make a point about his contention, feel free to jump in. Let's get a conversation going.

## Shekhar Borkar

That sounds good, Scott. Thank you. So my history is not as rich as everybody else's, but I'm a physicist, and I started my graduate school in the 70s and 80s, and I used 8-bit processors as well as 16-bit. LSI-11 was the one that I used for my thesis. Then I joined Intel to work on the 8051 family of microcontrollers, and these microcontrollers now you find them in your toothbrushes. Well, in those days they were, you know, they were powerful machines. Then I switched on to high-performance computing, followed by microprocessor research, and I'm a circuit researcher at heart, and I retired from Intel a few years ago, and now I'm with Qualcomm, and I'm working in the government division.

So, with that short introduction, let me show you what I want to share with you.

[Shekhar displays slide with questions posed to panel]

## Panel Questions

Major challenges in the growth of the microprocessor?

**Unnecessary distractions: ISA wars, RISC vs CISC, etc.**

50 years, progress mostly due to technology, architecture, compilers, or applications?

**All: Technology provided transistor, architectures used/abused them, SW advances made microprocessors user friendly**

Advice for the young aspiring students - processors of the future?

**Beware of fashion and hype, research where your passion lies...**

Technological challenges as it grows to 75 and opportunities in the coming 25 years?

**It has matured, it will not grow anymore—opportunities lie in using it wisely in systems**

How will a microprocessor of 2040 look?

**Microprocessor has matured to a building block—analogue to a NAND gate in logic**

**It will look the same, it's the whole system that matters now and more so in the future...**

So, I'm going to first start with the panel questions, and the panel questions (I'm going to go through this very quickly), the first question that was asked was, major challenges in the growth of the microprocessor? As I see them all the way from the 80s to now, the major challenges really have been not technical, but unnecessary distractions, such as ISA wars. Chris mentioned ISA, "ISA doesn't matter". I agree with you, but at the same time, Chris, ecosystem matters. So, if the ecosystem doesn't matter, ISA doesn't matter.

I completely agree with you, but some of us have to make a business. 50 years? Is it because of the technology? architecture? Compilers? Or applications? and I think we are pretty much here in agreement it's all of the above. Technologies provided transistors, but if you didn't have architectures to use and abuse them, you would not know what to do with them, and finally, software advances are necessary to make them user friendly.

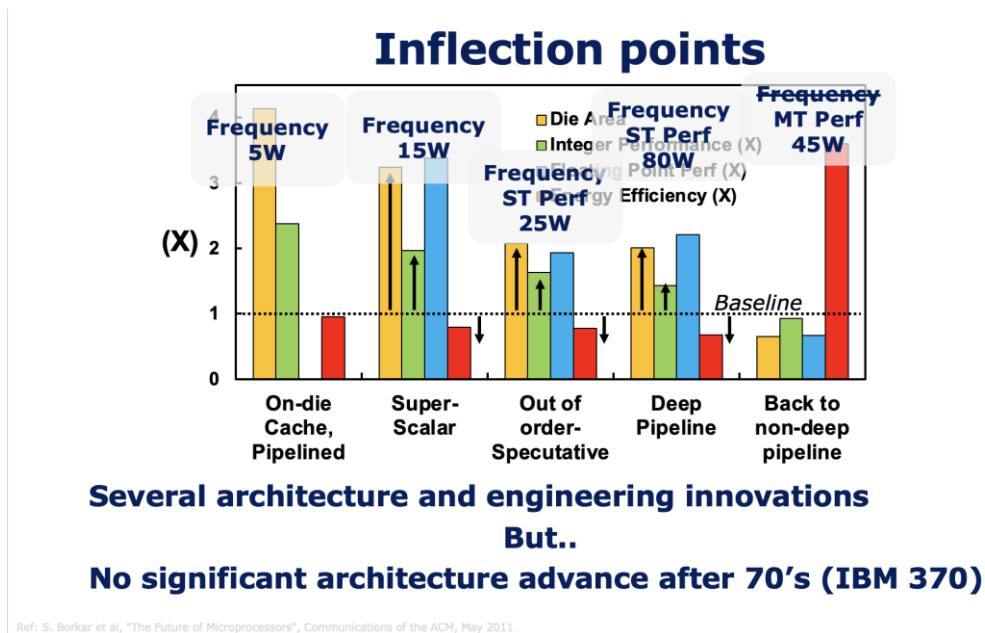
So, what's my advice to young aspiring students? Just be aware of fashion and hype. There is so much of that out there. Go do the research where your passion lies, and you will be successful.

So, what are the technological challenges as it grows to 75 and opportunities in the next 25 years? If you ask me, the microprocessor is already mature. It's not going to grow anymore. So really, the opportunities lie in how to use them wisely in a system.

So, what will the microprocessor in 2040 look like? Nothing much. It's already mature. It's just like a NAND gate. A NAND gate is mature. We don't redesign it every year. In the 70s and 80s, I designed them. I don't design them anymore. I just use them from the library.

So, a microprocessor will be a building block. It will look the same, and it's the whole system that matters even now and more so in the future. So, don't get fixated on what the microprocessor will look like, but really more think about what the system will look like. That said, let me complete my talk here in the next two or three minutes by showing what I see as the inflection points, and these are measurable inflection points as I see them over the last 30-40 years.

[Shekhar displays slide with processor performance and energy efficiency of processors for 4 decades]



So, what I'm showing you on the x-axis here is what I call the inflection points as seen by the circuit designer from the architectures. On the y-axis, I'm going to show you the die area of these inflection points, the performance, and the energy efficiency.

For example, the microprocessors before on-die caches, and before they were pipelined. When they had caches and pipeline to quadruple the amount of die area, you double the performance without changing the energy efficiency much. At that time, the frequency was the king, and the power was about five watts. Then came superscalar. This is like Pentium. So, if you look at the Pentium, compared to this, you triple the die area. You approximately double the integer performance, but, you had a loss in energy efficiency.

Then you went further to out-of-order and speculative execution. When you went there, if you look at the die area, it almost doubled. There was a performance increase, but the energy efficiency decreased. Frequency was still the king. This was the late 90s or the mid-90s. Single-thread performance was still the king. That's what everybody wanted. The power was about 25 Watts. So, everybody said, you know what? Frequency is the king.

Next inflection point, and this is, Scott, your point about a bad inflection point. Deep pipelines; pipelines 28 deep, well, yeah, the frequency increased, the die size doubled. Performance increased a little bit, but energy efficiency decreased even further. Now this is all with respect to the previous generation. So, you can think about architectural innovations being extremely inefficient in energy efficiency as we move across the inflection points. Then we said, you know what, the frequency was a king. But in the early 2000s, that is about the inflection point here. Go back to non-deep pipelines. Frequency doesn't matter. It's a multi-threaded performance, and the multi-cores. I bring it back to 45W envelope.

So, if you ask me about what we have done. There have been several architecture and engineering innovations, but really, in reality, there have been no significant architecture advance after the 70s, which is the IBM 370. Again, I'm not saying no architectural elevations, but nothing significant since then. So, thank you.

### **Scott Gardner**

Let me get some reactions here. Let's start with Dave Patterson. The bottom of our slide here says The 48th Annual International Symposium on Computer Architecture. Are we going to have to rename this conference? Do you agree with Shekhar?

### **David Patterson**

No, I don't agree with Shekhar. [Scott: "OK, let's hear it."] , John and I, in our Turing awards, said the inflection point going forward, now that Dennard scaling is over and Moore's Law's ending, is domain-specific architectures like Chris was talking about. So, an example of an architecture, and so what's happening, and this is what I think what Chris said, is the ideas that didn't make it for general-purpose computing are being resuscitated for domain-specific architectures.

So, I think, you know, Norm Jouppi gave the first talk today about Google's latest TPU, and he also gave the first talk at ISCA in 2017 about the first one, which I think will probably go down as a classic accelerator. But old ideas resuscitated, in particular, systolic arrays, so that was that kind of an interesting idea. Did it really find a place? I don't know, but because of machine learning right now, matrix multiplier is at the heart of it. That was the perfect match, basically a two-dimensional computer architecture. The systolic array and tying that to systolic arrays - a very powerful set of ideas. And more or less, you know, every time you use one of the vertical services, you're using machine learning, and most of them are running on a dedicated accelerator that is, you know, a big matrix-multiply unit that's using systolic arrays to great effect.

### **Scott Gardner**

So, David, I didn't get that question out to you guys about what is a microprocessor, so would a systolic array qualify?

### **David Patterson**

No, well no. No, I said it's, you know, I think it has, I think, what's a processor, you know? It fetches instructions and executes instructions. And, I think, a microprocessor, where the name came from, is just a processor is inside one chip. I think that's what a microprocessor is, and, you know, that's a lot easier to do with Moore's Law when there are billions of transistors than it was when there's thousands of transistors of matrix multipliers

**Scott Gardner:** But in the case of matrix-multipliers, is there enough there to say, well, this is a microprocessor or is this really just a compute unit?

**David Patterson:** Well, you know, whether the TPUs and, often GPUs, are accelerators are plugging into servers is the way they're normally done, but I mean, they do fetch their own instructions, they do make branches and conditions and stuff like that. They don't typically have virtual memory (accelerators), but that's a direction that things could go. But there seems to be, this is also, in the MIT paper, "There's Plenty of Room at the Top", is where the semiconductor is no longer lifting all boats. There's going to be opportunities for computer architects and software people to collaborate together to make big gains in important areas, and if you don't do that, you know, things aren't going to get any faster.

So, I think there's plenty of opportunity for architecture innovation in the domain-specific architecture place, because you don't have to run, you're not stuck with, you know, a million-line C++ program and trying to figure out how the hell to make that go faster. These things are written in higher-level languages. It makes it much

easier to innovate below them, and, you know, the application world is transforming. At least pieces of it are transforming away from the way they used to build programs through learning from data and machine learning. This is a force of nature. I think this will go down, you know; the big things in my career were, I guess, the microprocessor, the internet, the world wide web. Machine learning may prove to be something that's significant, and that's a revolution that's going through the whole software stack and the opportunities for architects below. There's plenty of opportunities for architects to build up: "Plenty of Room at the Top".

### **Scott Gardner**

Dave, that's great, and it's a cool topic. Let me keep things rolling. You wanted to make a quick point, Shekhar?

### **Shekhar Borkar**

Yeah, the quick point I wanted to make is, David was agreeing with me when I made the statement that the microprocessor, by itself, doesn't matter. It's a system that matters, and that's exactly what you described. Thank you.

### **David Patterson:**

The microprocessors matter, I think. The microprocessor matters. You can't have a system without a microprocessor.

### **Scott Gardner**

I got a few more people I want to bring in here. John's been nodding a lot, so let me get him to make a comment here, real quick.

### **John Hennessy**

So, I agree with Dave. I think what Shekhar's point misses is, if the software model changes, the hardware model's got to change. Because, the instruction set IS the interface between the hardware and the software, and if you think about what happened, those early microprocessors, we programmed them. We programmed them in assembly language; you couldn't use anything else. There wasn't enough memory in the machine; you had to squeeze every bit out, and then, of course, the RISC inspiration came from a high-level language viewpoint and the emergence of Unix. And this new thing, you've got to have CUDA to program GPUs, things like TensorFlow to program these other machines, and if the software model continues to change, then we ought to be willing to think about how the hardware model can change accordingly.

### **Scott Gardner**

So, let's keep rolling here. These were the two professors that teach computer architecture. Lee and then Chris. You both had some comments. I'll start with Lee.

### **Lee Smith**

Yeah, I felt that we ought to be making a distinction between the control plane and the data plane. It seems to me that in the control plane, Shekhar is absolutely right. There's nothing new. There's nowhere to go, and you

know, it's kind of, it's done. And everything you're talking about otherwise is doing more than just that general-purpose control-plane compute. It's all involved in shoveling huge quantities of data.

**Scott Gardner:**

And there's probably a control-plane processor as the host, actually looks just like every other one for the last 30-some years? [Lee: "almost certainly, I would argue"].

**Scott Gardner**

Glenn is agitating as well. You want to jump in real quick, Glenn, I promised Chris, but go ahead.

**Glenn Henry**

Well, I think the discussion of what's a microprocessor is silly. What we have are processors these days. The best microprocessor around, in my opinion, is the IBM z15. What you'll call a mainframe, but it's one chip, and it's really, really powerful. I didn't mention something, so I'll put in my special thing here. I'm a part-time professor at the Naval Postgraduate School doing research on processor security, and I think the fallacy of all this is, first of all, we sit around talking about the big challenges. How we're going to use all the silicon, right? You know, the next generation has two to four times the transistors in it, and we all invent things, Our latest processor has an AI coprocessor in it, etc., etc.

I think this is all going to change, because the risk of security in our current processor designs is very, very high, and I believe (my opinion) that the proper approach to providing bulletproof security is going to change some of the things that we've been doing in processor architecture. If we don't do something and just keep doubling the transistors every two years, I think it's going to be a real problem.

**Scott Gardner**

Yeah, just keep turning off those features you keep adding.

Chris, go ahead and make a comment but then also go ahead and give us your inflection point, so let's keep it rolling.

**Chris Rowen**

So, I think it's really important to work backwards from the application developers and, you know, a modern application developer does not think very much about instruction-set architecture. They don't even think about the OS interface or the compiler very much. They're very often working at the level of some, you know, Python or JavaScript level. They're invoking libraries, they're invoking cloud services, and how they stitch those together is centrally important to what is the distribution of work that needs to be done. A lot of those are very generic. There are trillions of lines of code out there. The average line of code doesn't get executed very much, and so we get to this very unequal distribution.

Some lines of code matter a lot, because they are very heavily used, and in those cases, architecture matters a lot, and we're going to work our way to figure out how we drive the silicon from the lines of code that matter. But, in terms of the overall impact of the IT universe, that's very much more driven by how productive the environment is. How we can create new services, which, while they may not really execute a lot, still matter a lot to the



environments in which we live. So, this leads us, inevitably, I think, to hybrid architectures where we're going to try to put together the best of both worlds, such that the average line of code is compiled and runs almost transparently, and it doesn't matter very much what it runs on, so long as it gets done. It's very much more about the ecosystem than it is about instruction, set architecture.

### **Scott Gardner**

So, do you think we'll have fully heterogeneous systems?

And heterogeneous operating systems?

### **Chris Rowen**

Sure, and you're going to have to paper over the fact that it's heterogeneous so that nobody really notices. Yeah, some parts of it are going to matter a lot, and where you're really going to push the envelope on efficiency, given that Dennard scaling and Moore's Law are not going to give us those increases in efficiency.

### **Scott Gardner**

I want to get back to Federico real quick. Go ahead. Do you have an inflection point you want to leave us with? Looking back, in the last 50 years, what would you pick?

### **Federico Faggin**

Yeah, so, to me, the important inflection points are the ones that move all the aspects of technology in a certain direction. So, for me, the first major inflection point was the personal computer and especially when IBM decided to enter the business and have a PC that was open. An open system and which, you know, created an incredible frenzy of applications, ideas and so on, and drove the market for 40 years. And then the next one was, you know, the iPhone. The iPhone was a PC plus internet.

And all this stuff that we learned in in the meantime, so that is now driving the technology directions in which the applications are moving, and so on. And now we see, of course, machine intelligence and AI and so on. Robotics that require their own specialized architecture, specialized hardware, and the technology that could be essential to these are really, the FPGAs, for example, you know, where you can fill programmable gate arrays that will allow configuring the hardware for a particular application. So that you should be gaining flexibility which is a fundamental, as I see, in the future. Someone mentioned security. I think that is fundamental also. I think that the cybercrime and all these problems that we see will get only worse, and so we really need to think through what do we need to create computers that are not hackable, and that to me is foundational.

### **Scott Gardner**

Kathy, what do you think? Let's just do the inflection point. What happened in the last 50 years?

### **Kathy Papermaster**

AActually, I'd have to add, I mean, I'm sort of surprised. I mean, I think, I definitely agree with Federico, but as an inflection point, I mean, technology kind of driving the whole thing, and, I guess you said it before earlier, but to me, just seeing CMOS in genera, And I guess that's just, you know, basic foundation there, and no one really mentioned the role of design automation along the way, and what that allowed us to do, right? So, I don't know

if you call that an inflection, but again it was something over the years, you know. Technology was driving, and I'll just add those two points, because I don't have much more to say that hasn't been covered already.

### Scott Gardner

Yeah, okay. Dave, let's go to you. Go on the record. What really changed in the last 50 years that got us to where we are?

### David Patterson

I was going to try and move on. Looking up the clock, I was going to try and move on to the predictions. Is that okay?

### Scott Gardner

Yeah, let's do that.

### David Patterson

So okay, I actually prepared a slide. [Scott: "OK, yes, go ahead."]

[Slide]

## Microprocessor at 75:

### Predict what you think must happen, not what might happen

- RISC-V Open ISA architecture will have become as significant for microprocessors as Linux is for operating systems today
  - Engineers like collaborating and working on open standards and open source designs
  - Open ISA is good enough; proprietary ISAs not significantly better than RISC-V
  - Freedom means more people can innovate in ISA, can share designs, no gatekeepers
- We've got to try to improve security via hardware
  - Ransomware is not an acceptable IT tax
  - Can't wait until we get 100% bug free, instantly updated software
- If quantum computing works, it will be for the cloud, not for the edge
  - The edge will remain important, so we need something beyond 2 nm CMOS
- 2045: "... the stored program concept is too elegant to be easily replaced. I believe future computers will be much like machines of the past, even if they are made of very different stuff. I do not think the microprocessor of ~~2020~~ 2045 will be startling to people from our time."

Since I prepared it, let me go ahead and show it. [Scott: "All right, so we're looking 25 years forward?"]. Yeah, so, what I wanted to say was, I think you can still see my screen. So, in 1995, they had the 150th anniversary of Scientific American. So, they asked me to predict microprocessors 25 years into the future. [Scott: "So, last year."] Yes, and so I was able to read what I said, and so what I said was, when I looked at a bunch of predictions from the past or the future, they were all very exotic. You know, computers would be made of biological material, we would have optical computing.

The stored-program computer is this old-fashioned idea that would go away, and so I said 25 years ago, is my radical prediction, is that the computers of 2020 will be very familiar to the people with computers today. And then, you know, speculated that we'd have, you know, more processors per chip, and, you know, the chips will get really big, and we'd also have really tiny ones, and those things had to be true. But the Scientific American people thought that was such a boring prediction that they had an insert where they said, what about, you know, single-element electrons or biological computers and- reversible logic. Remember reversible logic? So, they a side-bar in. So, I think when people predict the future, they try to do something entertaining, but I think when I try to predict the future, you try to predict, this HAS to happen. I can't imagine this not happening, and you got a better chance.

So, my bullets are, if we agree that the instruction set is not life and death, but we need one. It's going to be the control plane, and then why use proprietary instruction sets? So, I just think the open instruction-set architecture, in which RISC-V is the leading one, must become more popular. It's good. Linux was controversial 25 years ago. Companies said you can't trust the software, and open source is buggy, all this stuff, and Linux is the standard today. So I think RISC-V will be as standard as Linux is today. People like open-source stuff. This one's good enough, and lots of people innovate, and as other people have already brought up, we've got to try and improve security, and I've given up on my software colleagues figuring out a way to make bug-free code that's instantly patched, right? That's not going to happen. So, I think, and I don't think as a society, ransomware is going to be the IT tax, and we're going to be happy with that. I mean, even the president of the United States is aware of ransomware, right? I think it's up to hardware people (computer architects) to attack this.

There are people doing really interesting work now, you know, seeing the first bullet, RISC-V is an open architecture. You don't have to work for one of these companies to get it. They have industrial-strength ideas and get a whole software stack that will do it. Quantum computing, that we haven't talked about, you know, i there's a lot of excitement about it, but it's a cloud thing. It's not an edge thing, so when we're going to the edge, it's still gonna be important. I don't know if CMOS is going beyond two nanometers, but we've got to have something there, and so my prediction from 2020 for 2045 is the same as it was, more or less, in 1995, is that the stored-program concept is too elegant. It's going to look like machines of the past, at least the edge is going to look like machines of the past, so, I think if we're around in 2045, it's not going to be startling. So that's my speech.

### **Scott Gardner**

Maybe I bring in Federico, based on a point you made. We had a chance to geek out for an hour and a half or so the other night, and he said we're probably looking at a gap after about three nanometers that, you know, it's going to have to be something clever; something's got to happen, but, you know, it's pretty clear once you get transistors the size of the silicon atom that, you know, you've reached a point at which something major has to happen.

### **David Patterson**

And not only does it have to happen, but it also has to beat CMOS at three nanometers in terms of manufacturability and cost, so we could plateau for a while. You know, I always marveled that we got to design chips where the technology was moving faster than anybody, any engineer in the world. We could end up stuck like other engineers in other fields and have to build it out of the same material every year and still innovate..

### **Scott Gardner**

[Which would make you want to argue that, "architecture matters".] Go ahead, Federico.

## **Federico Faggin**

Yeah, that's what I think as well. I think that we are about to plateau. I don't see much hope below two nanometers. You know, with current technology being able to do the many things that we do with silicon; we don't do just one thing. So, then what can we get, you know, to work? Where can we get a technology that can work at room temperature and that can do better than silicon? I think, despite the fact that this is, you know, more science fiction today than real, that biology is the answer. But that biology, the answer, will come probably 30 to 40 years from now. The beginning of the answer will come 30-40 years from now. I don't think that we will have something viable that could compete with silicon anytime sooner than that.

## **Scott Gardner**

So, Shekhar, you got something controversial for the next 25-years of microprocessors?

## **Shekhar Borkar**

Yeah, I think, I mean. I'm sort of in agreement with this panel, but I'm sure that Dave will decide to disagree with me. So, that's okay. [laughter by panelists] So, the point is, really, I agree with Federico. The technology, at least, there is nothing in sight for me to see to replace CMOS in the next 15 years, because if it's in sight today, it will happen in 15 years. It's nothing in sight. So, with that said, we are engineers. We are not going to give up. We've got to keep the quote-unquote Moore's Law alive, at least in some shape or form. And it's about time now that, despite the lack of technological advances, now we are going to get really wise. i

In the past, we used architectural advances to use the transistors that were freely made available. Now is the time to think backwards and see, what are the architectural advances that I can use in the future? Especially given that the technology is not improving. So, some of the things that Chris mentioned, I agree with them, which are the things like domain-specific. Yeah, I mean, to me, when you look at the domain-specific solutions being integrated into a system, system means a lot of things to a lot of people. To me, an SOC is a system in which we have domain-specific and application processors sitting on it. So, it's that kind of system-level thinking that we need to do and not get fixated by a monolithic microprocessor.

So, I'll stop there.

## **Scott Gardner**

John Hennessy, You're going to be back in 25 years. You're going on the record. There are press here, and they're going to write this down. Professor John Hennessy said, "the microprocessor..."

## **John Hennessy**

So, I think we're going to find that machine learning is even more useful than we thought. Just lots of applications are going to get moved to that space. We're going to use massive amounts of computing resources to do training and massive amounts of data to train those machines. And then we're going to write less code, and you just have to look at Google's natural language system, right? It's one-hundredth the number of code lines, and it's more accurate than the previous system that was based on phrase-based analysis. So I think that's going to happen. I'm afraid that I agree with Federico. There may be an extended period here where we don't have a new technology, and we're just going to have to innovate without the approach we're so used to from Moore's Law. Think of it. Think of what happened when we had tubes before the invention of the transistor. It was a long steady period. Tubes did not get better much faster. Yeah, right? And we had a lot of issues with reliability and other things. We could have a similar kind of situation at the end at the tail end of Moore's Law.

**Scott Gardner**

That's cool! So, Lee, what do you think? Twenty-five years from now, you'll still be at Arm?

**Lee Smith**

No, no, no.

I have plans, and they don't involve being in ARM in 25 years. [panelists laugh] You know, I think that the panel's kind of come full circle, but it's answered its own question earlier. The way we prolong Moore's Law is by exploiting more special-purpose hardware. Now that just seems blindingly obvious. The proof point is in your pocket. You know, the baseband processor in your mobile phone contains a shedload of special purpose hardware for every radio modem your phone can do, and for the camera, and for video encoding, and probably video decode too. Arguably, the GPU that drives that giant screen is nothing but a special-purpose piece of hardware, and that's all there to get energy and power efficiency.

**Scott Gardner**

So, I'm... we're going to go Chris, Kathy, then Glenn, and then I'll turn it to Lizy.

**Lee Smith**

I want to make one more point, which is all of those functions that go on in the base station. They're all done by software on more general-purpose machines.

**Scott:** Even in the baseband?

**Lee Smith:** All that baseband stuff is done by software-defined radio. [Scott: "That's amazing"]. The reason it's that way is it allows the deployment of radio standards to be incremental and upgradable, and those base stations have to last, not two years in your pocket, but 25 years in the street.

**Scott Gardner**

Yeah, I know the panel's not about me, but that used to be science fiction. The idea that you could do software-defined radio and, you know, program the baseband to talk to any of these standards. So, Chris, go ahead. What will it look like?

**Chris Rowen**

Well, I think that it's true that Moore's Law has decelerated, that we're not going to get the energy per operation or the clock frequency increases that we got used to for a while. We still have actually a lot of headroom with respect to parallelism, and that means that any application which admits to parallelism is going to continue to thrive, particularly in the hands of computer architectures. Machine learning is an important part because it leaves room for an enormous amount of parallelism, and we'll see that at a higher level, as well, in parallelism between subsystems, some of which will be the general-purpose programming glue, and some of which will be the nodes that exploit specific domain problems.

We will see continued progress in performance overall, and we'll see progress in productivity in these high abstractions. It will be built on this heterogeneous mix of general-purpose and domain-specific architectures that will be extrapolatable pretty easily from what we see today. So, I'm an incrementalist as well. Twenty-five years from now, it is not going to look so different except that the number and variety of domain-specific subsystems

will be much longer and larger, and the general-purpose architectures will have an unbelievably long lifetime because of the ecosystem for that glue.

**Scott Gardner**

So, Kathy, you're going to come out of retirement and do a DNA computer?

**Kathy Papermaster**

No, but the technology node discussion is fascinating. It made me think of some other things, because if we really think the technology has, you know, that will be stuck, so to speak, then, you know, TSMC obviously has some, you know, lead on Intel and other fabs. Does that mean that they can catch up? I don't think TSMC would let that happen, but then at the same time, you have to think about, you know, like AMD processors, Intel, Nvidia, you know, what are they going to do? How do they differentiate themselves? In addition to, you know, the technology and some of the leads you can get through that.

So that by itself is an interesting discussion. So, I'm not quite sure it's going to be completely stuck, but I guess we'll wait and see, and other than that, I mean, I guess, you know, I'd like to just say it's, you know, it's exciting to see more and more, you know, heterogeneous compute. You know, leveraging vector computation, you know, as pioneered by Cell. Cell was difficult to program at the time. Hopefully, you know, to come and continue to evolve with that along the way, but, you know, the burden on software to utilize these accelerators, you know, hopefully, they're lowered. Lower those burdens, and maybe we'll see advantages that way as we look for the future.

**Scott Gardner**

Excellent. Well, Glenn, I'm anointing you a visionary, and then we'll let Lizy see if there are any questions that came across on the Q&A channel, and then I will be done.

**Glenn Henry**

I don't think there's any great need for more technology. The things that we talked about, the AI, cryptography, compression, what have you. They're all very regular, and you can do a lot with the technology we have. As you can see, pouring more transistors into the general-purpose processor is bad in my opinion. That's where the security leaks of today, which, by security leaks I'm including, of course, bugs. Right? The security leaks today come from them. I used to give a talk; I did it for seven years of Microprocessor Forum on the evilness of out-of-order execution. Of course, that was before we did it. [panel laughter]

The benefit you get, you know, the cost, performance and power benefits are all wrong, and we're doing that today. We sit around saying, "Well, let's see, in our next part we could use one billion transistors per core." How will we do that? What funny things will we do? "Oh we'll have a 40-stage pipeline, blah, blah, blah"., That's the wrong direction, and the general-purpose processor part of the world doesn't need more technology. The performance limit today, as we all know anyway, is the memory access, right? That's where if we need silicon, that's where we need to use it.

**Scott Gardner**

So Lizy, do you want to chime in here? You've had people who have been just pouring questions in the channel?

**Lizy John**

There are many questions in the channel, so I will ask the first one: The first DRAM 1103 was introduced a few months before microprocessor 4004 by Intel. DRAM has since stuck around for the past 50 years without much changes. What do panelists think of emerging non-volatile memories, and how would these emerging NVM impact microprocessors in the next 25 years? Who wants to take that question first...and just go ahead, yeah, David, go ahead

**David Patterson**

Me, I was just saying, it's really hard for new memory technology to win. In my career, there were all kinds of them; bubble memories at IBM, all kinds of things; you have to find a niche and win in terms of cost performance and be able to grow the volumes. So flash memory made it, you know; Well, original semiconductor replaced core; flash is a viable one, but for the new non-volatile memory technologies, many of them are promising, but there has to be a way for them to get to their volumes and get cost-effectiveness, so it's a commodity kind of industry; it's a really hard thing to do. So it'd be great if one of them makes it, but I don't know which one will or why.

**Lizy John**

Chris

**Chris Rowen**

And they've almost always use some difference in process technology. So they really only work pretty far out in the memory hierarchy, which means their interactions with the microprocessor design are sort of second-order, so they're important at the system level but not so much in direct influence in what you put on that chip.

**Lizy John**

But, you know, sometimes people say, or because the capacity is so high, they could rethink the memory hierarchy entirely, and maybe, you know, all that large capacity, allows you to design different kinds of memory hierarchies; Does anyone have an opinion about that?

**John Hennessy**

Well, I think one of the problems we're going to have is that SRAMs are starting to slow down compared to logic in the last five or ten years. It was one of the first talks in Norm Jouppi's recorded version of his talk, and that's gonna mean we get a bigger gap now. Despite the fact that we keep growing caches, we're going to see that memory wall become a problem increasingly as we go along.

**Lizy John**

Shekhar, you seem to have an opinion.

**Shekhar Borkar**

Yeah, there are three rules in memories, external memories, not integrated memories; number one is a dollar per bit, number two is a dollar per bit, and number three is a dollar per bit, so if we can beat these three rules, the only two technologies I see for the next 10 years are DRAM and Flash. We can spend two hours discussing these, but that's my bottom line.

### **Lee Smith**

And there are all also software issues with non-volatile memory to create the volume you need for that technology to breakthrough; you need portable, attractive software level models of how persistence works, and that's still a research topic, and that suggests to me that it's at least seven to ten years away from being a deployed technology and it's got to be pretty much right first time to generate the volumes that will displace or attack flash and DRAM; so yeah one can hope and I'm certainly in niches, and some of the niches might be big, but,

### **Lizy John**

Thank you.

### **Federico Faggin**

I think you see; I think it's difficult to beat one transistor. Yeah, I think yes, you know, in other words, you cannot have half a transistor per bit. That's a problem, and if we are already there with the two types of memories that are dominant in the market, so I see absolutely no way to go beyond that. There was a time where spintronics was all, you know, so full that it would create a memory that was that all the characteristics of the sum of the two but they couldn't even come close.

### **Lizy John**

Okay, there are lots of questions, so I think I'll move on to the next one. Do you feel that various fabrication companies are rushing through silicon technology nodes without extracting most out of a node before advancing to the next node? Who wants to take that?

### **John Hennessy**

That's what Intel is doing right now. It's slowing down the technology development to make the most of it before it moves to the next process.

### **Shekhar Borkar**

Yes, you have to be careful when you look at the nodes and the numbers; don't go by the numbers. If you go and look at some of the characteristics of the process technology, such as what is a contacted gate pitch, and then you will see the difference between 10 nanometers of Intel versus 7 nanometers of TSMC. Go, take a look, and then you'll find out yes, the technology has slowed down but not as it's hyped by the media.

### **Lizy John**



Okay, I will move to another question - why are we not investing more in 3D, in particular in monolithic 3D; or should we invest more in monolithic 3D? Is that the way to go?

**Shekhar:** Lizy you want me to go?

**Lizy:** Yeah, please,

**Shekhar Borkar**

So if you look at the monolithic 3D, its value proposition in logic is very slim for two reasons. One - is extracting the heat out of stacked transistors because of the power density. You are already limited by the power density. And we are going to exasperate it. The second is designing with the 3D integration of logic is so cumbersome, despite the fantastic design tools we got; so the niche where it really works is stacking regular structures like DRAMs, so DRAM stacking has been successful but not so in the logic for the reasons I just mentioned.

**David Patterson**

Yeah, flash too, right, but that's it. Yeah, power is one of the big challenges.

**Lizy John**

Another question here. Commercial planes are limited to 950 kilometres per hour. Speed limit for cars exist in almost every country. Why can't we accept an upper bound on VLSI integration?

**David Patterson**

Well, it's not. It's just like something we get to vote on.

I vote for more transistors. I don't know if that's going to work.

**Shekhar Borkar**

But my answer is very simple - because we are engineers. We don't give up, so if you give up, then just be there, and we are done okay.

**Hennessy:** Besides, who wants to limit the creativity of the next generation of application creators, right? We want them to think big.

**David Patterson**

Yeah, and we've been talking about single chips, but obviously, there's an opportunity for revolution and packaging right there. It was so much simpler just to use a bigger chip every time rather than bet on some radical packaging technology. That could be what the future looks like - is chiplets, and, you know, something that looks more like for those of us remembers TTL assembly; that could be what the future looks like and if there's a very efficient packaging technology solution out there that that could be a game-changer.

**Glenn Henry**

I hate to sound like a broken record. The issue is that we measure things wrong. Benchmark performance is not the critical issue; security is. If benchmark performance was a critical issue, it's not coming from the processor,

so I think we need to use more transistors; we need to use them more intelligently. We all know how to make a more reliable processor. We're just afraid to do it. Still, we could do it, and we could have thousands of them, for example, while today we have 100.

### **Lizy John**

There are many more questions in the Q&A channel, but, you know, good times end very quickly. This panel is about to finish; and so what I'm going to do is I will take the questions from the audience and offline send them to the panelists, and there is a way for the platform to get some answers back to those who asked but, you know, time is up. So we need to get moving. I would like to thank all of the panelists for taking time from your day and joining this panel. I thank you so much. It was a very lively discussion. Thank you so much.

Now I'm also sure many in this audience have stories about your own experience with microprocessors. So it is just, you know, by whatsoever the chance in this year, the 50th year of the microprocessor's birth, I also have another job as editor-in-chief of IEEE Micro. So IEEE Micro is going to celebrate the 50th anniversary of the microprocessor in its November issue, and anybody who has a good microprocessor story – pictures, anecdotes, send them to microp50@gmail.com, and selected stories will be printed in the Micro magazine or displayed on the IEEE Micro website.

IEEE Micro is celebrating the 50<sup>th</sup> Anniversary of the  
Microprocessor in its November Issue.

Please send your favorite Microprocessor stories,  
pictures, anecdotes, to

|  
microp50@gmail.com

Selected stories will be printed in the Micro magazine or  
displayed on IEEE Micro website.

So I would like to thank all of the panelists again and to the audience, you know, we were talking about biological computing and quantum computing earlier during the panel. There are a few more panels coming every day during ISCA, one in the evening US time, one around this time every day. There is one on biological computing; There is one on quantum computing, and there are, you know, six panels altogether. Please attend some of them.

### **Scott:**

And my thank you as well to everybody. Don't forget to order off Amazon to get Federico's book. It's actually very, very good.

**Lizy:** Thank you very much, everybody. It was great having you all for this panel. Thank You!!



Lizzy K. John



Scott Gardner



Federico Faggin



John Hennessy



David Patterson



Lee Smith

