DOI:10.1145/2909885

Derek Chiou

Interview

An Interview with Yale Patt

ACM Fellow Professor Yale Patt reflects on his career in industry and academia.

PROFESSOR YALE PATT, the Ernest Cockrell, Jr. Centennial Chair in Engineering at The University of Texas at Austin has been named the 2016 recipient of the Benjamin Franklin Medal in Computer and Cognitive Science by the Franklin Institute. Patt is a renowned computer architect, whose research has resulted in transformational changes to the nature of high-performance microprocessors, including the first complex logic gate implemented on a single piece of silicon. He has received ACM's highest honors both in computer architecture (the 1996 Eckert-Mauchly Award) and in education (the 2000 Karl V. Karlstrom Award). He is a Fellow of the ACM and the IEEE and a member of the National Academy of Engineering.

Derek Chiou, an associate professor of Electrical and Computer Engineering at The University of Texas at Austin, conducted an extensive interview of Patt, covering his formative years to his Ph.D. in 1966, his career since then, and his views on a number of issues. Presented here are excerpts from that interview; the full interview is available via the link appearing on the last page of this interview.

DEREK CHIOU: Let's start with the influences that helped shape you into who you are. I have often heard you comment on your actions as, "That's the way my mother raised me." Can you elaborate?

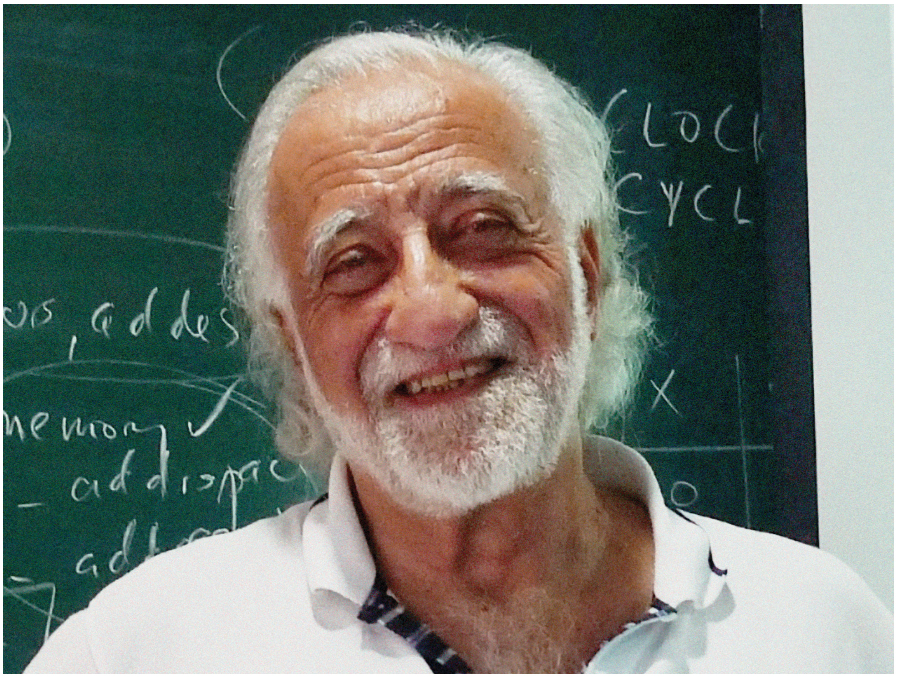

IMAGE BY XIAOHONG JIANG

Yale Patt, ACM Fellow and Ernest Cockrell, Jr. Centennial Chair Professor at The University of Texas at Austin.

YALE PATT: In my view my mother was the most incredible human being who ever lived. Born in Eastern Europe, with her parents' permission, at the age of 20, she came to America by herself. A poor immigrant, she met and married my father, also from a poor immigrant family, and they raised three children. We grew up in one of the poorer sections of Boston. Because of my mother's insistence, I was the first from that neighborhood to go to college. My brother was the second. My sister was the third.

You have often said that as far as your professional life is concerned, she taught you three important lessons.

That is absolutely correct. Almost everyone in our neighborhood quit school when they turned 16 and went to work in the Converse Rubber factory, which was maybe 100 yards from our apartment. She would have none of it. She knew that in America the road to success was education. She insisted that we stay in school and that we achieve. An A-minus was not acceptable. "Be the best that you can be."

That was the first lesson. The second lesson: "Once you do achieve, your job is to protect those who don't have the ability to protect themselves." And I have spent my life trying to do that. The third lesson is to not be afraid to take a stand that goes against the currents—to do what you think is right regardless of the flak you take. And I have certainly taken plenty of flak. Those were the three lessons that I believe made me into who I am. When I say that's the way my mother raised me, it usually has to do with one of those three principles.

What about your father?

My father was also influential—but in a much quieter way. We didn't have much money. It didn't matter. He still took us to the zoo. He took us to the beach. He took me to my first baseball game. He got me my first library card—taught me how to read. I remember us going to the library and getting my first library card at the age of five. So when I started school, I already knew how to read. That was my father's influence.

I understand there is a story about your father that involves your first marathon.

Yes, the New York City Marathon. The first time I ran it was in 1986. If you finish, they give you a medal. I gave it to my father. "Dad, this is for you." He says, "What's this?" I said, "It's a medal." "What for?" "New York City Marathon." "You won the New York City Marathon?" "No, Dad. They give you a medal if you finish the New York City Marathon." And then he looked at me in disbelief. "You mean you lost the New York City Marathon?" It was like he had raised a loser, and I realized that he too, in his quieter way, was also pushing me to achieve and to succeed.

Besides your parents there were other influences. For example, you've often said Bill Linvill was the professor who taught you how to be a professor.

Bill Linvill was incredible. He was absolutely the professor who taught me how to be a professor—that it's not about the professor, it's about the students. When he formed the new Department of Engineering Economic Systems, I asked if I could join him. "No way," he said. "You are a qualified Ph.D. candidate in EE. You will get your Ph.D. in EE, and that will open lots of doors for you. If you join me now, you will be throwing all that away, and I will not let you do that. After you graduate, if you still want to, I would love to have you." That was Bill Linvill. Do what is best for the students, not what is best for Bill Linvill.

You did your undergraduate work at Northeastern. Why Northeastern?

Northeastern was the only school I could afford financially, because of the co-op plan. Ten weeks of school, then ten weeks of work. It was a great way to put oneself through engineering school.

What do you think of co-op now?

I think it's an outstanding way to get an education. The combination of what I learned in school and what I learned on the job went a long way toward developing me as an engineer. In fact, I use that model with my Ph.D. students. Until they are ready to devote themselves full time to actually writing the dissertation, I prefer to have them spend their summers in industry. I make sure the internships are meaningful, so when they return to campus in the fall, they are worth a lot more than when they left at the beginning of the summer. The combination of what we can teach them on campus and what they can learn in industry produces Ph.D.'s who are in great demand when they finish.

I understand you almost dropped out of engineering right after your first engineering exam as a sophomore.

Yes, the freshman year was physics, math, chemistry, English, so my first engineering course came as a sophomore. I did so badly on my first exam I wasn't even going to go back and see just how badly. My buddies convinced me we should at least go to class and find out. There were three problems on the exam. I knew I got one of them. But one of them I didn't even touch, and the third one I attempted, but with not great success. It turns out I made a 40. The one I solved I got 33 points for. The one I didn't touch I got 0. And the one I tried and failed I got seven points. The professor announced that everything above a 25 was an A. I couldn't believe it. In fact, it took me awhile before I understood.

Engineering is about solving problems. You get no points for repeating what the professor put on the blackboard. The professor gives you problems you have not seen before. They have taught you what you need to solve them. It is up to you to show you can. You are not expected to get a 100, but you are expected to demonstrate you can think and can crack a problem that you had not seen before. That's what engineering education is about.

Then you went to Stanford University for graduate work. Why did you choose Stanford?

My coop job at Northeastern was in microwaves, so it seemed a natural thing to do in graduate school. And, Stanford had the best program in electromagnetics.

But you ended up in computer engineering. How did that happen?

There's a good example of how one professor can make a difference. At Stanford, in addition to your specialty, they required that you take a course in some other part of electrical engineering. I chose switching theory, which at the time we thought was fundamental to designing computers, and we recognized computers would be important in the future. The instructor was a young assistant professor named Don Epley. Epley really cared about students, made the class exciting, made the class challenging, was always excited to teach us and share what he knew. By the end of the quarter, I had shifted my program to computers and never looked back.

The rumor is you wrote your Ph.D. thesis in one day. What was that all about?

Not quite. I made the major breakthrough in one day. As you know, when you are doing research, at the end of each day, you probably don't have a lot to show for all you did that day. But you keep trying. I was having a dry spell and nothing was working. But I kept trying. I had lunch, and then I'd gone back to my cubicle. It was maybe 2:00 in the afternoon All of a sudden, everything I tried worked. The more I tried, the more it worked. I'm coming up with algorithms, and I'm proving theorems. And it's all coming together, and, my heart is racing at this point. In fact, that's what makes research worth-while—those (not often) moments when you've captured new knowledge, and you've shown what nobody else has been able to show. It's an amazing feeling. Finally I closed the loop and put the pen down. I was exhausted; it was noon the next day. I had worked from 2:00 in the afternoon all the way through the night until noon the next day, and there it was. I had a thesis!

So you wrote your thesis in one day.

No, I made the breakthrough in one day, which would not have happened if it had not been for all those other days when I kept coming up empty.

What did you do then?

I walked into my professor's office. He looked up from his work. I went to the blackboard, picked up the chalk, and started writing. I wrote for two hours straight, put down the chalk and just looked at him. He said, "Write it up and I'll sign it. You're done."

After your Ph.D., your first job was as an assistant professor at Cornell University. Did you always plan on teaching?

No. I always thought: Those who can do; those who can't, teach. I interviewed with 10 companies, and had nine offers. I was in the process of deciding when Fred Jelinek, a professor at Cornell, came into my cubicle and said, "We want to interview you at Cornell." I said, "I don't want to teach." He said, "Come interview. Maybe you'll change your mind." So there I was, this poor boy from the slums of Boston who could not have gotten into Cornell back then, being invited to maybe teach there. I couldn't turn down the opportunity to interview, so I interviewed, and I was impressed—Cornell is an excellent school.

Now I had 10 offers. After a lot of agonizing, I decided on Cornell. All my friends said, "We knew you were going to decide on Cornell because that's what you should be—a teacher." And they were right! I was very lucky. If Fred Jelinek had not stumbled into my cubicle, I may never have become a professor, and for me, it's absolutely the most fantastic way to go through life.

Why did you only spend a year there?

At the time, the U.S. was fighting a war in Vietnam. I was ordered to report to active duty in June 1967, at the end of my first year at Cornell. I actually volunteered; I just didn't know when my number would come up.

> **"The combination of what I learned in school and what I learned on the job went a long way toward developing me as an engineer."**

Your active duty started with boot camp. What was that like?

Boot camp was amazing. Not that I would want to do it again, but I am glad I did it once. It taught me a lot about the human spirit, and the capabilities of the human body that you can draw on if you have to.

What happened after boot camp?

After nine weeks of boot camp, I was assigned to the Army Research Office for the rest of my two-year commitment. I was the program manager for a new basic research program in computer science. I was also the Army's representative on a small committee that was just beginning the implementation of the initial four-node ARPANET. I knew nothing about communication theory, but I had a Ph.D. in EE, and had been a professor at Cornell, so someone thought I might be useful. In fact, it was an incredible learning experience. I had fantastic tutors: Lenny Kleinrock and Glen Culler. Lenny had enormous critical expertise in both packet switching and queueing theory. Glen was a professor at UC Santa Barbara, trained as a mathematician, but one of the best engineers I ever met. In fact, I give him a lot of the credit for actually hacking code and getting the initial network to work.

After the Army, you stayed in North Carolina, taught at NC State, then moved to San Francisco State to build their computer science program. Then you went to Berkeley. You were a visiting professor at Berkeley from 1979 to 1988. What was that like?

Berkeley was an incredible place at that time. Mike Stonebraker was doing Ingres, Sue Graham had a strong compiler group, Dick Karp and Manny Blum were doing theory, Domenico Ferrari was doing distributed UNIX, Velvel Kahan was doing IEEE Floating Point, Dave Patterson with Carlo Sequin had started the RISC project, and I and my three Ph.D. students Wen-mei Hwu, Mike Shebanow, and Steve Melvin were doing HPS. In fact, that is where HPS was born. We invented the Restricted Data Flow model, showed that you could do out-of-order execution and still maintain precise exceptions, and that you could break down complex instructions into micro-ops that could be scheduled automatically when their dependencies were resolved. We had not yet come up with the needed aggressive branch predictor, but we did lay a foundation for almost all the cutting-edge, high-performance microprocessors that followed.

You had other Ph.D. students at Berkeley as well.

Yes, I graduated six Ph.D.'s while I was at Berkeley—I guess a little unusual for a visiting professor. The other three were John Swensen, Ashok Singhal, and Chien Chen. John was into numerical methods and showed that an optimal register set should contain a couple of very fast registers when latency is the critical issue and a large number of slow registers when throughput is critical. Ashok and Chien worked on implementing Prolog, which was the focal point of the Aquarius Project, a DARPA project that Al Despain and I did together.

Then you went to Michigan. Two things stand out at Michigan: first, your research in branch prediction.

We actually did a lot of research in branch prediction during my 10 years at Michigan, but you are undoubtedly thinking of our first work, which I did with my student Tse-Yu Yeh. Tse-Yu had just spent the summer of 1990 working for Mike Shebanow at Motorola. Mike was one of my original

HPS students at Berkeley. When Tse-Yu returned to Michigan at the end of the summer, he had some ideas about branch prediction, based on his interaction with Shebanow. He and I ended up with the two-level adaptive branch predictor which we published in *Micro* in 1991. Intel was the first company to use it. When they moved from a five-stage pipeline on Pentium to a 12-stage pipeline on Pentium Pro, they could not afford the misprediction penalty they would have gotten with their Pentium branch predictor. So, they adapted ours. Since then, some variation has been used by just about everybody.

Michigan is also where you developed the freshman course.

Yes, I had wanted to teach that material to freshmen for a long time, but always ran up against a brick wall. Then in 1993, the faculty were complaining that students didn't understand pointer variables and recursion was magic. I just blurted out, "The reason they don't understand is they have no idea what's going on underneath. If we really want them to understand, then we have to start with how the computer works." I offered to do it, and the faculty said okay. Kevin Compton and I developed the freshman course, and in fall 1995, we taught it for the first time. In fall 1996, it became the required first course in computing, and we taught it to all 400 EECS freshmen.

I heard Trevor Mudge volunteered to teach it if something happened.

Trevor said he would be willing to teach the course if we gave him a book. There was no book. In fact, the course was completely different from every freshman book on the market. We started with the transistor as a wall switch. Kids have been doing wall switches since they were two years old, so it was not difficult to teach them the switch level behavior of a transistor. From wall switches we made inverters, and then NAND gates and NOR gates, followed by muxes and decoders and latches and memory, then a finite state machine, and finally a computer. They internalized the computer, bottom-up, and then wrote their first program in the machine language of the LC-2, a computer I invented for the course. Programming in 0s and 1s gets old very quickly, so we quickly moved to LC-2 assembly language.

"Engineering is about solving problems. You get no points for repeating what the professor put on the blackboard."

Since Trevor needed a textbook to teach the course in the spring, I wrote the first draft over Christmas vacation. That's why the freshman textbook was born. If Trevor hadn't insisted, who knows? There may not have been a freshman textbook. But there was no other book available because it was a complete departure from everybody else.

You ended up co-authoring the book with one of your Ph.D. students.

Yes, originally, it was going to be with Kevin Compton, but Kevin ended up not having time to do it. So I asked Sanjay Patel, one of my Ph.D. students who TA'd the course the first year we offered it. We wrote the book together, and published it as he was finishing his Ph.D.

You left Michigan in 1999 to come to Texas. Is there anything at Texas that particularly stands out?

Far and away, my students and my colleagues. I have now graduated 12 Ph.D.'s at Texas. When I came here, I brought my Michigan Ph.D. students with me. Two of them, Rob Chappell and Paul Racunas, received Michigan degrees but actually finished their research with me at UT. Two others, Mary Brown and Francis Tseng, were early enough in the Ph.D. program that it made more sense for them to transfer. Mary graduated from UT in 2005, went to IBM, rose to be one of the key architects of their Power 8 and 9 chips, and recently left IBM to join Apple. Francis got his Ph.D. in 2007, and joined Intel's design center in Hillsboro, Oregon.

With respect to my colleagues, I consider one of my biggest achievements that I was able to convince you and Mattan Erez to come to Texas. The two of you are, in a major way, responsible for building what we've got in the computer architecture group at Texas.

Six of your students are professors?

That's right. Three of them hold endowed chairs. Wen-Mei Hwu is the Sanders Chair at Illinois. Greg Ganger, one of my Michigan Ph.D.'s, holds the Jatras Chair at Carnegie Mellon, and Onur Mutlu, one of my Texas Ph.D.'s holds the Strecker chair at Carnegie-Mellon. In total, I have two at Illinois, Wen-Mei Hwu and Sanjay Patel, also a tenured full professor, two at Carnegie Mellon, Greg Ganger and Onur Mutlu, and two at Georgia Tech, Moin Qureshi, and Hyesoon Kim, both associate professors.

And a number of your students are doing great in industry too.

Yes. I already mentioned Mary Brown. Mike Shebanow has designed a number of chips over the years, including the Denali chip at HAL and the M1 at Cyrix. He was also one of the lead architects of the Fermi chip at Nvidia. Mike Butler, my first Michigan Ph.D., was responsible for the bulldozer core at AMD. Several of my students play key roles at Intel and Nvidia.

You are well known for speaking your mind on issues you care about, and have some very strong views on many things. Let's start with how you feel about the United States of America.

Quite simply, I love my country. I already mentioned that I spent two years in the Army—voluntarily. I believe everyone in the U.S. should do two years of service, and that nobody should be exempt. It's not about letting the other guy do it. It's about every one of us accepting this obligation. I believe in universal service. It does not have to be the military. It can be the Peace Corps, or Teach for America, or some other form of service.

I also believe in immigration. That's another key issue in the U.S. today. Immigration is part of the core of the

American fabric. It has contributed enormously to the greatness of America. Some people forget that unless you're a Native American we all come from immigrant stock. The Statue of Liberty says it well: "Give me your tired, your poor." It is a core value of America. I hope we never lose it.

I also believe in the Declaration of Independence as the founding document of America, and the Constitution as the codification of that document. Most important are the 10 amendments Jefferson put forward that represent the essence of America. "We hold these truths to be self-evident," that some rights are too important to leave to the will of the majority, that they are fundamental to every human being. And that's also come under siege lately. Freedom of speech, assembly, free from unlawful search and seizure, habeas corpus, the knowledge that the police can't come and pick you up and lock you up and throw the key away. Some of this seems to have gotten lost over the last few years. I remain hopeful we will return to these core values, that nothing should stand in the way of the first 10 amendments to the Constitution.

Let's talk about your research and teaching. Can you say something about how you mentor your Ph.D. students in their research?

I don't believe in carving out a problem and saying to the student, "Here's your problem. Turn the crank; solve the problem." I have a two-hour meeting every week with all my graduate students. My junior students are in the room when I push back against my senior students. Initially, they are assisting my senior students so they can follow the discussion. At some point, they identify a problem they want to work on. Maybe during one of our meetings, maybe during a summer internship, whenever. I encourage them to work on the problem. They come up with stuff, and I push back. If they get too far down a rat hole, I pull them back. But I cut them a lot of slack as I let them continue to try things. In most cases, eventually they do succeed.

Don't research-funding agencies require you to do specific kinds of research?

I don't write proposals to funding agencies. I've been lucky that my research has been supported by companies. It is true that in this current economy, money is harder to get from companies. So if any companies are reading this and would like to contribute to my research program and fund my Ph.D. students, I'll gladly accept a check. The checks from companies come as gifts, which means there is no predetermined path we are forced to travel; no deliverables we have promised. In fact, when we discover we are on the wrong path, which often happens, we can leave it. My funding has come almost exclusively from companies over the last 40 years so I don't have that problem.

There is a story about you wanting to give your students a shovel.

As I have already pointed out, most days nothing you try works out so when it is time to call it a day, you have nothing to show for all your work. So I've often thought what I should do is give my student a shovel and take him out in the backyard and say, "Dig a hole." And he would dig a hole. And I'd say, "See? You've accomplished something. You can see the hole you've dug." Because at the end of most days, you don't see anything else.

The next day, the student still doesn't see anything, so we go to the backyard again. "Now fill in the hole." So, again, he could see the results of what he did. And that's the way research goes day after day, until you make the breakthrough. All those days of no results provides the preparation so that when the idea hits you, you can run with it. And that's when the heart pounds. There is nothing like it. You've uncovered new knowledge.

Can you say something about your love for teaching?

It's the thing I love most. These kids come in, and I'm able to make a difference, to develop their foundation, to see the light go on in their eyes as they understand difficult concepts. In my classroom, I don't cover the material. That's their job. My job is to explain the tough things they can't get by themselves. I entertain questions. Even in my freshman class with 400 students, I get questions all the time. Some people say lectures are bad. Bad lectures are bad. My lectures are interactive—I'm explaining the tough nuts, and the students ask questions. And they learn.

I know you have a particular dislike for lip service instead of being real.

Being real is very important. The kids can tell whether you're spouting politically correct garbage or whether you're speaking from the depths of your soul. If you're real with them, they will cut you enormous slack so you can be politically incorrect and it doesn't matter to them because they know you're not mean spirited. They know you're real. And that's what's important.

What do you think about Texas' seven-percent law that forces the universities to admit the student if he's in the top seven percent of the high school graduating class, since many of them are really not ready for the freshman courses?

It is important to provide equal opportunity. In fact, my classroom is all about equal opportunity. I don't care what race, I don't care what religion, I don't care what gender. I welcome all students into my classroom and I try to teach them. The seven-percent law admits students who come from neighborhoods where they didn't get a proper high school preparation. And this isn't just the black or Hispanic ghettos of Houston. It's also rural Texas where white kids don't get the proper preparation. It's for anyone who is at the top of the class, but has not been prepared properly. The fact they're in the top of the class means they're probably bright. So we should give them a chance. That's what equal opportunity is all about—providing the chance. The problem is that when we welcome them to the freshman class, we then tell them we want them to graduate in four years. And that's a serious mistake because many aren't yet ready for our freshman courses. They shouldn't be put in our freshman courses.

If we're serious about providing equal opportunity for these students, then we should provide the courses to make up for their lack of preparation, and get them ready to take our freshman courses. And if that means it takes a student more than four years to graduate, then it takes more than four years to graduate. I don't care what they know coming in. What I care about is what they know when they graduate. At that point I want them to be every bit as good as the kids who came from the best highly prepared K–12 schools. We can do that if we're willing to offer the courses to get them ready for our freshman courses.

Can you say something about your Ten Commandments for good teaching?

On my website I have my Ten Commandments. For example, memorization is bad. The students in my freshman course have been rewarded all through school for their ability to memorize, whether or not they understood anything. And now they are freshman engineering students expecting to succeed by memorizing. But engineering is about thinking and problem solving, not memorizing. So I have to break them of that habit of memorizing.

There are other commandments. You should want to be in the classroom. You should know the material. You should not be afraid of interruptions. If I explain them all, this interview will go on for another two or three hours, so I should probably stop. If you want to see my Ten Commandments, they're on my website.[a]

There was an incident regarding your younger sister in a plane geometry course. What was that about?

That was a perfect example of memorization. I was visiting my parents, and my sister, who was studying plane geometry at the time, asked me to look at a proof that had been marked wrong on her exam paper. Her proof was completely correct. All of a sudden it hit me! I had gone to the same high school and in fact had the same math teacher. He absolutely did not understand geometry. But he was assigned to teach it. So what did he do? This was before PowerPoint. The night before, he would copy the proof from the textbook onto a sheet of paper. In class he would copy the proof onto the blackboard. The students would copy the proof into their notes. The night before the exam, they'd memorize the proof. On the exam he'd ask them to prove what he had put on the board. They had no idea what they were doing, but they'd memorized the proof. The result: 100% on the exam.

My sister didn't memorize proofs. She understood plane geometry. She read the theorem, and came up with a proof. It's not the proof that was in the book. But as you well know, there are many ways to prove a theorem. The teacher did not understand enough geometry to be able to recognize that even though her proof was not the proof in the book, her proof was correct. So she got a zero! Memorization!

You once told me about a colleague at Michigan who came into your office one day after class complaining he had given the worst lecture of his life.

Yes, a very senior professor. He came into my office, slammed down his sheaf of papers, "I've just given the worst lecture of my life. I'm starting my lecture, and I've got 10 pages of notes I need to get through. I get about halfway through the first page, a kid asks a question. And I think, this kid hasn't understood anything. So I made the mistake of asking the class, who else doesn't understand this? Eighty percent of their hands go up. I figured there's no point going through the remaining 9½ pages if they don't understand this basic concept. I put my notes aside, and spent the rest of the hour teaching them what they needed to understand in order for me to give today's lecture. At the end of the lecture, I've covered nothing that I had planned to cover because I spent all the time getting the students ready for today's lecture. The worst day of my life."

I said, "Wrong! The best day of your life. You probably gave them the best lecture of the semester." He said, "But I didn't cover the material." I said, "Your job is to explain the hard things so they can cover the material for themselves." He adopted this approach, and from then on, he would check regularly. And if they didn't understand, he would explain. He never got through all the material.

In fact, that's another one of my Ten Commandments. Don't worry about getting through all the material. Make sure you get through the core material, but that's usually easy to do. The problem is that back in August when you're laying out the syllabus, you figure every lecture will be brilliant, every kid will come to class wide awake, ready to learn, so everything will be fine. Then the semester begins. Reality sets in. Not all of your lectures are great. It's a reality. Not all kids come to class wide awake. It's a reality. So you can't get through everything you thought you would back in August. But you can get through the core material. So don't worry about getting through everything. And don't be afraid to be interrupted with questions. He adopted those commandments and ended up with the best teaching evaluations he had ever received.

You got your Ph.D. 50 years ago. Your ideas have made major impact on how we implement microprocessors. Your students are endowed chairs at top universities. Your students are at the top of their fields in the companies where they work. You've won just about every award there is. Isn't it time to retire?

Why would I want to retire? I love what I'm doing. I love the interaction with my graduate students in research. I enjoy consulting for companies on microarchitecture issues. Most of all, I love teaching. I get to walk into a classroom, and explain some difficult concept, and the kids learn, the lights go on in their eyes. It's fantastic. Why would I want to retire? I have been doing this now, for almost 50 years? I say I am at my mid-career point. I hope to be doing it for another 50 years. I probably won't get to do it for another 50 years. But as long as my brain is working and as long as I'm excited about walking into a classroom and teaching, I have no desire to retire.

Derek Chiou (derek@utexas.edu) is an associate professor of Electrical and Computer Engineering at The University of Texas at Austin and a partner hardware architect at Microsoft Corporation.

Watch the authors discuss their work in this exclusive *Communications* video. http://cacm.acm.org/videos/an-interview-with-yale-patt

For the full-length video, please visit https://vimeo.com/an-interview-with-yale-patt

a http://users.ece.utexas.edu/~patt/Ten.commandments