Catalyzing Computing Podcast Episode 36 – Computer Architecture with Mark D. Hill (Part 2)

The transcript below is lightly edited for readability. Listen to “Computer Architecture with Mark D. Hill (Part 2)” here.

Intro - 00:10	1
Hardware Security & Vulnerabilities - 1:01	2
Mark’s Involvement with CCC/CRA - 5:58	4
AI and the Future of Hardware - 10:38	6
Simulation of Computer Hardware - 14:00	7
Thoughts on Running Successful A Successful Organization - 18:11	9
Academia Versus Industry - 23:53	11
Future of Computing Research - 28:24	13
Outro - 30:56	15

[Intro - 00:10]

Khari: Hello, I'm your host, Khari Douglas, and welcome to Catalyzing Computing, the official podcast of the Computing Community Consortium. The Computing Community Consortium, or CCC for short, is a programmatic committee of the Computing Research Association. The mission of the CCC is to catalyze the computing research community and enable the pursuit of innovative, high-impact research.

In this episode I interview Dr. Mark D. Hill, a Professor Emeritus of Computer Sciences at the University of Wisconsin-Madison. Mark recently joined Microsoft as a Partner Hardware Architect. His research interests include parallel computer system design, memory system design, computer simulation, deterministic replay and transactional memory. He is the Chair Emeritus of the CCC Council. In this episode, Mark discusses the importance of hardware security, the impact of AI on hardware, and working in academia versus industry. Enjoy.

[Hardware Security & Vulnerabilities - 1:01]

Khari: So we're here today with Mark Hill. How are you doing today?

Mark: I'm doing very good. It's a pleasure to be here with you, even though here is cyberspace.

[Laughter]

Khari: A few years ago you wrote a blog post about the Spectre and Meltdown hardware vulnerabilities. So could you just briefly explain what those are? And then has anything important in the field of architecture changed in reaction to those exploits?

Mark: So what Spectre and Meltdown showed was, we thought we were designing computers that looked like they were executing instructions one at a time but were actually going faster because underneath we guessed at what we were going to do next so that we could get that instruction-level parallelism. We thought we completely covered our tracks when the guesses were wrong and no information would leak, but it turns out with some clever work you can actually, essentially, use that speculation to leave some breadcrumbs that another gadget can detect, and you can leak information.

This caused a big crisis, and industry has subsequently worked to mitigate this to a significant degree, but, as far as I can tell, not eliminated it. Academia is working very hard to come up with principal methods of eliminating this method of information leaking while still getting pretty good performance. Much of the formal work involves something called information flow tracking, which was originally developed for paying attention to classified data and who could see it. So, I think things are going to be better.

This is a classic example of where the abstraction of the hardware, that it just executed instructions, was not enough that underneath the implementation leaked. So, it's quite

an interesting thing, and the computers that you buy today are not completely cured of it.

Khari: So, how much security work typically goes into the design and creation of computing hardware?

Mark: I think this is a challenge. People buy computers mostly so that they are as fast as possible for the price. They may say they worry about security, but they don't as much. With hardware the principal security that has been done is to try really hard to isolate the operating system from user code and then, more recently, the operating system from the hypervisor, but it's sort of not fundamental to the design.
Part of the problem is that the incentives are wrong. If I, as a hardware vendor, made my hardware more secure but 15 percent slower, it appears more customers would be concerned with a 15 percent loss in performance than they would the improved security, which is hard to measure and know if you're really getting it. So we have a problem for security in hardware and we certainly have a similar problem for security in software.

Khari: Do you have any thoughts on how to better align those incentives to improve security?

Mark: Well, that's a complicated process, and it turns out the Computing Community Consortium, as we speak, is trying to identify possible workshops to address this specific issue. There's a hardware workshop that's in a relatively mature state, and I think there's going to be a more general one. In other fields you do things like industrial standards and even regulation. Those are heavy hammers. but we might need those hammers or we might need the threat of those hammers to get better behavior because the current situation is not working.

I blame we computer scientists. We built these computers half a century ago assuming that security wasn't that important because this is a special. And, you know, the track record of humanity is that in all domains some notion of crime eventually comes, so it was naive of us to not think of that as a first order consideration from the beginning.

[Mark's Involvement with CCC/CRA - 5:58]

Khari: Yeah, so you mentioned that the CCC is working in this area related to hardware and security. So how did you first get involved with the CCC and with CRA?

Mark: So back in about 2012, Ed Lazowska, who was one of the early people with the CCC, and others tapped me to run a sort of fast white paper with the leaders of a bunch of professional societies in computer architecture on what we should do, sort of, post Moore's Law. We wrote a white paper called *21st Century Computer Architecture*. I think the whole process was less than three months. Subsequently, NSF cited this in a funding program that was 16 million dollars a year, and in various guises ahs continued on to today. Since no good deed goes unpunished, the next year I was invited to join the CCC, and, since I thought that was a fun experience with the white paper, I joined as a CCC member and I guess I've had about an eight year arc.

Khari: So, obviously, Catalyzing Computing is the official podcast of the CCC, but for the most part, we don't really dive that much into the working of the CCC. But since you are the former chair, maybe you like to talk a little bit more about it. So what is the CCC and what does the CCC do?

Mark: Ok, so the CCC is basically a NSF funded think tank of about 20 members, mostly professors, some from industry, complemented by some really great staff in Washington, D.C., where the two are greater than the sum of the parts. Our goal is to try to identify places where computing research ought to go and investments ought to be made.

You know, finding the future is often hard, but what you can often do...William Gibson, the great science fiction writer, says the future is already here, it's not evenly distributed. So that's part of what we're doing. We get the experts together and we can articulate something and then spread it to many others, including funding agencies. This is done through workshops where we're very good at getting a short white paper out of it,

sometimes there are standalone white papers. And there are ways when we partner with CRA government affairs to bring these things to the attention of the government, and it goes back to the community and through the publishing of the papers and the fact that it's the community that participated in the workshops.

Khari: What would you say is your highlight of the time you spent with the CCC?

Mark: Well, the highlight probably has to be the 2018 AAAI/CCC 20-year roadmap for artificial intelligence. Even though I was not...I mean, I was helping to catalyze this, I was chair of the organization and I played bad cop to help get things out, but other people did more of the work from the CCC side — Liz Bradley and Ann Drobnis and others. But this was a really big deal and it has already catalyzed and is referenced in some pretty significant NSF programs. CRA government affairs shopped it around the government and I expect the biggest impact is to come.

The key trick with AI was...well AI is pretty hot in the industry, so what do we need this roadmap for? It turns out there are things from academia that can complement industry and create a sum greater than its parts. These often include things that are a longer-term focus, and they can be issues that are maybe not industry's number one concern. Social justice may not be industry's number one concern. Or maybe fairness is, maybe fairness isn't? We could address things like that, and I think it's a very nice, albeit longer than I would like, document.

Khari: Yeah, I think it's over 100 pages, but people that are interested should check that out and there will be links on the podcast webpage if you want to read more [read the full report here].

Mark: There is an executive summary that's way shorter.

[Laughter]

[AI and the Future of Hardware - 10:38]

Khari: That's true. So how do you think the proliferation of AI has impacted the hardware space?

Mark: So, artificial intelligence has the potential to change a lot in society, hopefully mostly good. The current way it's done is...the greatest successes have been in a part of machine learning — which is a part of AI — called deep neural networks. These currently analyze a tremendous amount of data with a tremendous amount of computation, and if we could do that even more effectively then machine learning could be used in even more situations. A big step to greater effectiveness was moving from regular processing cores to general purpose GPUs (Graphics Processing Unit), which did that data-level parallelism that we discussed before.

Now there are efforts afoot to do very specialized accelerators, as we've discussed before, for machine learning, such as Google's Tensor Processing Unit (TPU). I think we're going to see much more of that for deep neural networks. Then, as AI starts expanding to other things, not just deep neural networks, I think it's important enough that hardware will be developed for that.

Interestingly, there is a feedback path — we also have to design the hardware. So there are some small new efforts on trying to take machine learning and apply it back to the design and optimization of hardware to maybe exceed...the human designers instead of doing the design, they're doing the configuration of the AI to do the design. So, I think we could get a really nice synergy.

I mean, you hear all this talk about AI, you might think it's hype, but it's pretty real.

Khari: Could you say anything more about Google's tensor processing units? I understand that was a pretty big deal within architecture when that came out.

Mark: Right. So the amazing thing is that tensor processing units are basically really good at multiplying a matrix (a two dimensional array of numbers), times a vector (a one dimensional array of numbers), producing a vector, running it through something called an activation function (that's...it doesn't really matter, a little git widget), and repeating on tons of data.

We talked about ideas taking a long time, and in the 1980s there was this big thing called systolic arrays. Called so because they had, like, a heartbeat and the data, like the matrix, moved around. So these tensor processing units really are the mature outgrowth of that systolic array, just like GPUs were the mature outgrowth of the ILLIAC IV and other things from the 1960s. It can sometimes take a long time, and in this case it was not only a long time for the technology, it was much more important for there to be an application that was sufficiently specialized yet important. And machine learning has done that.

[Simulation of Computer Hardware - 14:00]

Khari: How much of your research involves the actual, like, fabrication process of chips or hardware?

Mark: I did fabrication of chips when I was a grad student, and I have largely not done it as an academic out of concern that I can't do it real enough. Other people have found that it's very useful to do it, so I wouldn't say that my approach is better. We like to have people doing different things. I use simulations and models mostly.

Khari: Ok, how does a simulator for a computer hardware system work?

Mark: Well, a simulator is a software program that is sufficiently accurate that you can load application software on this software system and it will allow the program to functionally execute and take a bunch of measurements, such as what the performance is. It allows you to study a program executing on a system that doesn't exist yet.

That's the good news. because you can then change the system pretty easily because you're just changing the simulation software. The bad news is that this runs, depending on how you do it, a hundred, a thousand, or ten thousand times slower; so you can't really run the real work, you have to do some very careful sampling. And if you do that wrong, like you just make something smaller, you may get an answer that's not correct. But it's exceedingly powerful.

Khari: Maybe this is a dumb question, but how do you know that the simulation you're working with is accurate or close enough when it doesn't exist yet?

Mark: It's not a dumb question, Khari. That's actually an excellent question. It's a very difficult problem. One thing you can do is you can configure the simulator so that it's very similar in one of its forms to an existing piece of hardware. And you can see if you relatively accurately predict what that hardware does. That's called validation.

But it is harder to know for sure when you move off to hardware that doesn't exist whether you're completely correct. One of the problems is that there is lots of hidden stuff in the simulator, so people can't just inspect all of it, and sometimes these incorrectness can hide. That's one of the reasons why I like to compliment things with models, because they can give you some detailed insights of what should happen, and if there is a divergence in what the model seems to be saying and what the simulation says, you can learn something.

For example, Gables was used in a system...there were two examples where there's a simulation compared to the model and in one case...In both cases Gables predicted better performance in a simulation, so who's wrong?

Well, it turns out, in the first case Gables made some assumptions about how parallel the workload was which wasn't completely true, so Gables was wrong but Gables could be extended. Still, you get some insight. In the second instance where Gables said a bigger number than the simulator, it was determined that there were insufficient buffers to really get the information flow that you needed. And if you fix those buffers they

match more closely, which it turns out is also an example of Little's Law. There were not enough buffers to hold the intermediate state to get the full bandwidth. So never trust one way of looking at things. If you have more than one way of looking at things, you're more likely to have a robust answer. But you can never perfectly validate a simulator of hardware that doesn't exist.

[Thoughts on Running Successful A Successful Organization - 18:11]

Khari: Good advice. Are you involved with any other organizations or computing related societies that you want to discuss?

Mark: Well, I have long been a member of ACM (Association for Computing Machinery) and IEEE (Institute of Electrical and Electronics Engineers). I was an officer for a long time with the special interest group in computer architecture or SIGARCH. And I'm old enough that I helped bring them into the World Wide Web age by setting up a webpage where people could access information before search engines happened and a monthly newsletter of web links so that people could find conferences and stuff. One of those things you don't need anymore, but they were a big step forward at the time.

[Laughter]

Khari: Yeah. So obviously, you've been in a leadership role in many different organizations, and you're a pretty good public speaker, so do you have any advice or recommendations for people that are interested in taking on a leadership role within whatever organization that they're part of?

Mark: Well, that's not an easy question. I think a couple things, one is to spend time reflecting on what you're trying to do, what your goals are, where you're trying to take the organization. But also what's really important is you can't do that much yourself. The only way to get things done is to work with other people, so you want to listen to other people, assess their strengths, you know, get people, encourage them to do what

they're good at and try to get somebody else to do what they're bad at, and make sure that you're generous with credit because it's very motivating when people get credit.

If you're running an organization, for example, you can give credit to all your people and they get credit, but you also get credit because the organization is running. So credit doesn't really divide. It multiplies in that situation. I think people don't always appreciate that. They're trying to grab the exclusive credit for themselves, and if you don't do that, you both make a lot of people happier and I think you're ultimately more successful. So it's a win-win.

Khari: Yeah, it seems like good advice for sure. So let's see, you wrote a piece for SIGARCH's blog a while back about a vision of computer architecture. And in the CCC one of the things we do a lot of is "visioning." So what do you mean by visioning and why do you think that's important?

Mark: Yeah. By the way, you didn't get the title exactly right. It was "A Vision of Computer Architecture Visioning." Anyway, it was really trying to articulate what the CCC does, and visioning is important because you want to get people to think about what the important problems are out there, so that they work on the important problems, not the easiest problems to work out.

The easiest problem to work on is, you've just written a paper, like you've just done LogTM. Well, let's do a follow-on paper on LogTM. Well, you know, that's only important if the follow-on paper is fixing a problem that people care about. Sometimes people get into the mode where they're just writing follow-on papers even though people don't care. So visioning forces you out of your box, talks to other people, and you find that future were it's unevenly distributed and you may go work on more important problems.

Khari: So, what would you say is the key to finding key problems to envision or just key problems in general to do research in?

Mark: By the way, many people think the key job of a researcher is to solve problems, but the key job of a research leader is to identify problems, which is what you're asking about. And I have found the best place to identify problems is to look for change, because if there's been no change then there's a good chance the problem is either unimportant or really hard. I'm not that smart, so I want to look for change so I can get an easy problem that has just emerged as important.

In computer architecture change happens because the applications change. Deep neural networks are way more important than they used to be. It can change because the technology changes. There's this new stuff called nonvolatile memory that is changing computer systems. Or it can change because tools change. There's things like SAT solvers, which make certain optimizations possible that were not before. So if you look for change, you may be able to identify places where, if you can do it, people will care.

You then have to complement that with...you want to make it likely that you can do something about it, otherwise it's a nice problem but you don't make progress. That's what I find using models to think about things...and I like taxonomies too or I try to divide up the space into different quadrants, like the two by two table that you mentioned in LogTM. The ultimate, by the way, taxonomy is Mendeleev's periodic table of the elements, which organized things by atomic number and caused people to say, "Wait a second. We know of no element there. Let's go look for it."

So that's a long answer to your question.

[Academia Versus Industry - 23:53]

Khari: Yeah. So as was mentioned in the intro, you're moving from academia to industry. Moving to Microsoft. What prompted this change after a career in academia?

Mark: I had 32 years at [the University of] Wisconsin at various levels of professor and then department chair. Then I was chair of the CCC, and I had the opportunity to go back to being a professor again — I was always technically a professor, but primarily a professor — and it seemed like I could do that, but let's take a new challenge. Industry is particularly interesting now for much of computer science, especially computer architecture, because with the slowing of Moore's Law, we have to find other ways to make things go faster, and that often will involve optimizations that cross many layers, which I'm relatively good at. 20 years ago, 30 years ago, it was all about optimizing your own little piece, and that's just not as exciting to me. Now we have this opportunity and the necessity to optimize across layers.

For example, systems like Microsoft Azure and other cloud computing systems have evolved from being a building full of server computers to being a computer in their own right. I mean, Luiz Barroso of Google calls this warehouse scale computing.

It's just a very exciting time, I think, to put some of my years of research experience to the test of what the real engineering constants are going forward. But it's really more a personal thing just to do something different instead of the same thing again.

Khari: So what are the tensions you've found in your career when it comes to industry and academia collaborating on the same problems?

Mark: So the tension is that industry and academia have different interests, although they have substantial overlap. I mean, industry really needs to make money, so they have to have a focus that's in the several year time frame and they have to operate within many constraints. You know, instruction set Intel processors are going to be x86. That's not a factor.

Whereas in academia you have other goals. You want to take a lot of risks. You want to look far. You want to push the envelope. You want to educate the student and get great publications. So it's a little different, but there's a lot of overlap because both sides can teach each other a lot. Industry can help you know what problems are real and

academia can study them more carefully, sometimes, than industry can when they just need to make a binary decision, not really characterize things, so there's a tension.

The other tension is with the intellectual property or patents. You have to be careful in a collaboration to figure out who owns the patents, do both own patents? That may be a little bit unfortunate, but that's U.S. law.

Khari: I noticed a presentation on your website where you mentioned that you've done three industrial sabbaticals in your career, you describe them as high costs and high reward. What were those costs and what were those rewards?

Mark: The cost is that you move, often with a family, to a different place and you devote a substantial fraction of a year to working for a company, maybe keeping your university activity going with one day a week, so it can hurt the university activity. That's the cost.

The benefit is you get embedded in the milieu that has problems, and some of those problems are very short term and are being solved. Never mind those. Some of them are just emerging. They're not yet important enough and they're being Band-Aided. And some you can intuit from the technology change. So you can come back from industry with new problems. That's the key reward. You can't come back from industry with new solutions because that's their intellectual property, but you can come back with problems. The sabbatical is a high cost way to do it, because it costs a whole year. Lesser costs ways of doing it are having industrial [inaudible] meetings or having a student go for an internship.

[Future of Computing Research - 28:24]

Khari: As we wrap up here, what do you think will be the most important area of computing research over the next 10 to 15 years?

Mark: I think it's two fold. On one hand, artificial intelligence with machine learning — maybe broader than deep neural networks, we'll see — will have a profound effect on

society, and we need to shape that and make it follow our country's values. This is a big deal. You know, people are going to say, “Oh, you know, self-driving cars. They told me they were going to have that by 2018 and it didn't happen. They're never going to happen.” This is a classic example of the Gartner hype cycle, where in the short term people think it's going to happen and then it doesn't. You reach a trough of despair and it can come back. I think that's true. AI is definitely doing a lot of things. It doesn't do everything it promises, but it will have a profound effect, so systems of supporting AI are going to be a big deal.

The other side is that when I started computers were in rooms, they were very isolated from people, and now you have computers on your body. And you can have more computers on your body going forward. You might have robots in your life. Computers are going to be interacting with people in very deep ways. For example, people may have emotional attachments to computers. I think we need to have a much deeper understanding of humans interacting with computers to make those computers more effective and serve humans in a good way.

Khari: Yeah, I think they'll definitely be important going forward. Is there anything else that we didn't cover that you want to bring up?

Mark: Well, I just want to say thank you to Khari. I mean, this is a good example of how it's great...diversity by generation and career stage is a good thing, right? I can't run a podcast. You can run a podcast. I have some thoughts. And together we produce something that we couldn't do separately. I really encourage all organizations and all collaborations to seek diversity in many different dimensions.

Khari: Sounds good. Well, thanks for taking the time to sit down with me. Have a good day.

Mark: Thanks.

[Outro - 30:56]

Khari: That's it for my interview with Mark Hill. We'll be back soon with new episodes. Until then, remember to like, subscribe, and rate us five stars wherever you get your podcast.

Learn more about the work of the CCC on our website at cra.org/ccc and find us on social media to stay up to date on all our latest activities. Until next time, peace.