Oral History of Phil Gossett

Interviewed by:
John Mashey

Recorded June 17, 2019
Mountain View, CA

CHM Reference number: X9077.2019

© 2019 Computer History Museum

Mashey: Okay. So why don’t we start with where you grew up and, you know, how you got into computing, and is that what you thought you were going to do and that sort?

Gossett: Right. So I grew up in New York City. My father was a professional cinematographer, mostly doing TV news. So he sort of specialized in riots in the New York area, late ‘60s. So if you’ve seen any riots in New York you’ve probably seen his work.

Mashey: Oh, great.

<laughter>

Gossett: My mother was a American History teacher in a high school, and she passed away when I was nine, so I, you know, that was a bummer, but anyway, I was, you know, a geek, a nerd, for as long as I can remember. I was raised as a Roman Catholic but it didn’t take. You know, I immediately saw that, you know, I wasn’t buying any of it, okay, so science was my thing. I originally wanted to be a physicist, you know, theoretical physicist, but when I went to undergraduate MIT, it became pretty obvious that it was impossible to do physics and eat, and, you know, I’m not into starvation or living on the streets or whatever. So that was, you know, fairly early in the days of popular computing. I guess the Apple II had just come out.

Mashey: When? Oh, okay. Right.

Gossett: So this was, you know, mid-to-late ‘70s, okay, and I was involved while at MIT with this funny little company called ECD, which was making-- I forget what the proper, you know, corporate name of the product was, but it was internally referred to as Spencer Love’s Giant Mind. There was this guy Spencer Love, who was the lead on it. So I worked a little bit on that. I did some font design, of all things, and a little bit of logic design, and for various complicated reasons I switched advisors. So there’re undergraduate advisors at MIT, and originally I was in the Physics Department, and when I decided that wasn’t going to work I started looking around in the Computer Science, Electrical Engineering Department, Course 6, and, you know, got connected with Ed Fredkin, who was great. He saved my bacon on multiple occasions, and in particular I was poor, I think is the right word, and I had to, you know, I had a little bit of money from my mother’s life insurance, but it wasn’t enough for four years at MIT, so I had to graduate in three years, okay, and I did, by the skin of my teeth, okay, and thank you, Ed Fredkin. I could not have done it without his help. He let me substitute courses. Turns out that graduate level courses are way less homework than undergraduate courses, so I just substituted graduate courses like math, and they were more interesting. You know, they sort of showed you the more abstract way of looking at things rather than just grinding through a bunch of algebra and trigonometry, which is, you know, exactly what I was looking for, and I did manage to graduate. So do you want me to just keep going on the timeline?

Mashey: Yeah, sure.

Gossett: Okay.

Mashey: That’s always a good way to sort of get going, yeah.

Gossett: Sure. Okay, then... So Ed Frankin was the founder of Triple-I, Information International, Incorporated, in Culver City, near L.A., and I got a job there, which didn’t pay very well. I was making $13,500 a year, which even at the time--

Mashey: And when was that?

Gossett: This was ’78, I think. So even then I think the going rate for a freshly minted undergrad from MIT was, or graduate from MIT, was 20K, okay, so I was making like two-thirds what I should’ve been. But it was fine. I was in the Motion Picture group. Gary Demos and John Whitney, Jr., and that was amazing. That was, like, you know, I got to do things that I don’t think I could’ve done anywhere else, so it was fine. It was a really interesting group, a lot of really, really smart people, and it was fun, right, I mean, what’s not to like, right? You’re doing computer graphics. I got to work on Tron, the original Tron. I designed large portions of the digital film printer, which was this amazing piece of hardware. I think it cost four million dollars, something like that. It had three optical bays, two scanners and one recorder. It would do 70 millimeter VistaVision, so 70 millimeters turned sideways, so it’s even larger, and it was 6,000-by-4,000 pixels, 36 bits per pixel, so, yeah, 12 bits per component, and it ran at-- it produced a frame every six seconds, which is really, you know, given the time and the available technology, that was quite a feat. In fact, the limiting factor was the scanners were essentially counting individual photons, okay. You know, it was so hard to get enough light through and, you know, see what you were doing at those kind of resolutions. It was a flying-spot scanner, okay, and that was a very interesting piece of hardware, which I got to work on. I also did a bunch of software for geometric modeling and that sort of thing. Anyway, that group sort of got dissolved for stupid political reasons. There was a falling out with Whitney Demos and the rest of Triple-I, so I was-- one of the things that I had to do while I was there was keep the Foonly F1 alive, okay. This was this monster of a--

Mashey: Yeah, talk about Foonly.

Gossett: Right. This was a monster machine. It was made out of discrete ECL 10K Series logic, and it was all wire-wrapped, so wire-wrap is this technology where you literally wrap a wire around these posts and--

Mashey: I remember.

Gossett: --it was-- well, okay, ECL 10K is all 50 ohm-terminated transmission lines, because we had long runs, and they were-- the logic was all in these giant panels. They were like two meters by one meter, something like that, and there were five of them, because to get enough gates together you needed that many, that much space, and it was arranged as pages. This was a technology that Triple-I had invented. So it was hinged so you could open it up like a book to get at the inner pages, and because it was 50 ohm terminated twisted pair, 50 ohms requires a very thin insulation to get the right characteristic impedance, so it would cut through the pins. You know, the <inaudible> the pins were trying to connect to would cut through the wires and short out. Excuse me. So what you had to do was first off figure out what went wrong, okay, which was challenging, and then very gingerly work through the mat of wires and find the one that was cut through and replace it, you know, without cutting through a half dozen other wires, right, in the process. So anyway, I got to be good at that. I also did a bunch of design work for them, so we had this funky vector graphics display thing that was connected to another PDP-10 and it was running the Stanford University Drawing Systems, SUDS, which is what we were using for designing logic. So I did some work for them kind of on the sly, unpaid actually, but it was fine, it was interesting. So I got to know that crew. Mostly Dave Poole. It was also Phil Petit and Jack Holloway that were the three principles. So anyway, when it became clear that the show was over on the movie group at Triple-I, and also I was really sick of Los Angeles, to tell you the truth. I was not a happy camper, and I remember the first day I flew in there. You know, it really looked like something from "Dante's Inferno," and the sky was orange, you know. There was one point when the hills above Culver City, which you could hardly see, okay, because of the smog, were burning, okay. I mean, it was literally burning, and I just, like, "I need to get out of here."

<laughter>

Gossett: So anyway, Dave Poole offered me a job at three times the pay, okay, and it was in the Bay area, which I'd visited before. In fact, shortly before this I was visiting a friend who had recently left III to come work here, and he showed me the Fry's supermarket. This wasn't the Fry's electronic stores, this was before that. So there was one aisle that had tubes of integrated circuits on one side and Twinkies and Jolt Cola on the other side, and I saw this and, you know, was like, "These--"

Mashey: Everything in engineering is--

Gossett: "These are my people."

Mashey: Yes, yes, yes.

<laughter>

Gossett: Okay. So as soon as Poole offered me the job, you know, I don't think I took two seconds to say "Yes," and flew up here and it was interesting. So Poole was a character, okay, I think, fairly legendarily. He took no nonsense from anyone. He suffered fools very poorly. There was a-- I don't know that this is true but I suspect it is. I wasn't there. But there's a story that he was trying to get money from DARPA-- ARPA, I think it was at the time, and there was an admiral who was, you know, checking him out and wanted to talk to him, and this was, again, you know, in the-- it was around 1980, I think. Maybe little bit before, and everybody smoked. I mean, a lot of people smoked, so this admiral guy takes out a cigarette and lights it up, and Poole was this fanatical anti-smoking guy, so he ripped it out of the guy's mouth, threw it on the ground and jumped up and down on it a few times, and, you know, the guy's jaw hit the floor and, you know, I don't think they got that contract, to be sure.

Mashey: Probably not, yeah.

Gossett: <laughs> So anyway, the thing I was first hired to do was get them the ability to design logic, you know, which meant having some kind of graphic system, okay. Because, you know, it was all graphical EDA at the time, which actually is a bad idea, but maybe I'll get to that later. So he wanted, Dave Poole, wanted a vector graphics design, and I talked him out of it, because vector graphics was kind of crazy at the time, and, you know, getting enough speed so that, you know, a typical CAD drawing would show up and not take 10 seconds to cycle through all the little lines. Required some pretty, pretty severe cost and it was not easy to get that kind of speed. So I talked him into a frame buffer, which was like the obvious thing to do. So I went off and designed that. That was kind of a weird experience. So since they didn't have any CAD available and any display heads available themselves, they were-- their main customer was Tymeshare on Bubb Road [yes in Cupertino], you know, so they had an office there and they had a DEC System 20, I think it was, which was their last and least performing machine, and anyway, they had some kind of display attached to that. It was almost unbelievable. Okay. It took-- I'm not exaggerating. This is literally true. It took five minutes to refresh the screen, okay. So, you know, there's this drawing system that you're trying to fly, okay.

Mashey: Sure. <laughs>

Gossett: And you can't, like, draw one wire and then wait five minutes and draw another wire and wait five minutes, so what you would do was the SUDS system had a macro capability, so you would in your mind sort of visualize what was going to happen, okay, and go as many steps forward as you possibly could completely flying blind, okay, and then hit the Refresh button and go get a cup of coffee.

Mashey: Oh, great.

Gossett: <laughs>

Mashey: Not exactly interactive graphics here.

Gossett: Not exactly interactive. So anyway, I got that to work and it was-- the graphics card we actually sold a few of in addition to using it internally. It was called Foo Vision, which is classic.

<laughter>

Gossett: Maybe I should describe the origin of Foonly, Foo, foobar, all the rest of that. This all comes from World War II. There was an expression "FUBAR," which was "F-ed up beyond all recognition." Okay. And for whatever reason the spelling changed from F-U to F double "O." I never understood why that happened, and then "foo" and "bar" became metasyntactic variables that were used mostly by the LISP community. So Foonly was kind of a riff on that. There are various stories about how the N-L-Y got added. Supposedly this was a broken parser, but I'm not sure I totally believe that. It just--

Mashey: Lives on in O'Reilly's Foo Camp?

Gossett: Yeah, yeah, indeed.

Mashey: Yeah.

Gossett: So anyway, so Foo Vision. That was the first thing I did for, you know, aside from designing some stuff while I was at Triple-I, and then the next thing was the Foonly F5, okay. So they had the F1, which was the ECL monster. There was the F2, which was a console computer, which was a very stripped down PDP-10, built out of TTL, okay, 7400 Series TTL, and then that got embellished somewhat into the F3, which was sort of their bread and butter. They sold a bunch of those to TymeshareTymeshare. They sold one to Stanford. I'll get to that in a few minutes, and, you know, it was scattered around. I think the University of Mexico ended up with one. They're scattered around various places. Anyway, TymeshareTymeshare was the main customer, so they wanted a higher-performance machine, and maybe I should diverge a little and talk about TymeshareTymeshare. So they were kind of running the western backbone of what became the internet, the ARPANET at the time, and it was all built out of PDP-10s, because they had this-- this was before TCP/IP, so there was a thing called NCP, the Network Control Program, not protocol, and it really was a program. So the packets, you know, the packet headers, had instructions in them and they were PDP-10 assembly code, okay, which is a terrible idea from a security point of view. But, you know, that wasn't foremost in their mind at the time. So anyway, you needed PDP-10s and we made PDP-10s, so that worked out. So they wanted this higher performance thing, which was dubbed the F4, which was still a TTL machine based on the F3 naturally but just higher performance, and then I'm not sure exactly why, you know, what the business argument was, but they also wanted a F5, which was a lower-end machine, something cheaper, okay. So I got in charge of that, okay. There was no architecture for it, so I created one. To make it cheap, what I did was designed it to run eight bits at a time, okay, so-- which is odd. It's the 36-bit word, so it doesn't go evenly.

Mashey: Yes. See, I remember. Yes.

Gossett: Okay. But, you know, eight was the right number for, you know, practical reasons because that's how chips came, were 4 or 8 bits. So anyway, it ran at I think 75 nanoseconds, so 13.3 megahertz, and it had a nanocycle, which was this 13 megahertz, and then you'd take some variable number of those cycles to make a micro-cycle, which was the, you know, the micro-coded speed, and anyway, it was 150 chips of, you know, 7400 Series TTL, which was impressive. I mean, impressively little, right, you know. It wasn't like you could get a microprocessor, you know, all in a chip or at least not a good one. So I got that to work. So I worked on that. Shel Kaphan, who became the CTO at Amazon, wrote the micro-code for that, so that was amazing. I knew Shel from Triple-I, actually, so we've crossed paths.

Mashey: How many people were at Foonly?

Gossett: Oh, God. So, I mean, there were the principals, you know, Poole, Petit and Holloway. There were a bunch of people, maybe a dozen. Let's see. Jeff Peters was sort of the system software guy. So I actually needed some hackery done in the system. So the system was FOONEX, which was based on TENEX. Everything's Foo. So there were some weird political and legal things, so apparently DEC screwed up on copyrights and patents for various things. So stuff became available, but some stuff didn't, okay. So, like, we couldn't use the mainline DEC operating system, but we could use TENEX, which was BB&N, I think it was, who did the first pager for the KL, I think it was, PDP-10, and for, you know, no particularly good reason none of that was patented or copyrighted, so we could get the TENEX and we could also copy the pager, okay. So the-- all of the Foonly machines were basically BB&N pagers. Yeah. So anyway, Poole was a really good engineer and a really bad businessman, okay, so he--

Mashey: I think I heard that story before.

Gossett: <laughs> Indeed. So he did a deal with Tymeshare that basically, if they didn't deliver the F4s by a certain date, Tymeshare just owned everything, okay, and that's, of course, what happened. So, you know, when that day came, we all sort of instantly became employees of Tymeshare and then, like, a week later or something Tymeshare folded and got absorbed by McDonnell Douglas. Okay. So I was briefly employed by McDonnell Douglas, okay, which was surreal. Anyway, it all just went to hell in a handbasket, as you would expect, and that was the end of that. So one of the customers for Foonly was the Stanford computer music lab, Center for Computer Research and Music and Acoustics, CCRMA, which was invariably referred to as "Karma," okay. and so they needed somebody to do-- they were trying to get a, I think it was an NSF grant, to do computer music workstations. Okay. This was a consortium with Stanford, UCSD, CMU and MIT. Okay. Anyway, you know, multiparty deals rarely go well, okay. You know, two parties is about the most that generally ever works, so this, you know, they couldn't agree on what they wanted, okay, so I was sort of hired to do this thing but it wasn't clear what this thing was. However, there were things that they needed, okay. So one thing that they really wanted was a sample rate converter, okay, so in digital audio, you know, you have things running at some sample rate and you want to convert it to some other sample rate, and that's a bunch of compute, okay. The way this would-- so anyway, I, you know, didn't know anything about this, I mean, or almost nothing about it, and so I went to one of the engineers there, a very sharp guy, Julius Smith, Julius Orion Smith, III, great name, and asked him, you know, "Well, so how does this work?" You know, "What do I need to do?" So apparently the state of the art way to do this was to sample rate convert up by a factor of about a thousand, okay, and then sample rate convert down by whatever ratio you wanted, okay, and, you know, I sort of looked at him and, you know, I think I said something like, "You've got to be kidding me." Okay. I mean, I was sort of, you know, imagining how many multiplies that was going to take and, you know, was a full rack of extremely expensive at the time chips and that just seemed insane. Okay. I mean, this can't be-- this can't be right. This can't be the optimal thing to do. So, you know, I said, "Well, can't you do, like, some kind of linear interpolation and then fix it up with some filters?" You know, I was kind of grasping at straws here, and he explained why that didn't work and anyway, we went back and forth and back and forth and eventually pretty much, you know, I don't think there was anybody who was, like, dominant on the design, but we hit on the design of having a table lookup, which was some number, you know, say, 256, times however many samples you needed for the finite impulse response, and you would-- so you'd have these different phases, okay, for going from one sample to the next in 256 increments, and that, of course, wasn't enough, enough resolution and phase space. So we linearly interpolated between those tables, and this was totally tractable. It's two multiplies, right, times however many taps, right. So we put-- well, okay. One sort of amazing or sad, depending on how you look at it, thing. We thought, "Well, let's patent this," okay. So we went to the Stanford patent office and they told us that you couldn't patent an algorithm. Okay. This is terrible, terrible advice. So we said, “Okay.” So we wrote a paper and it was published in ICASSP’84, IEEE publication, and Julius wrote the paper and I gave the talk, okay, and that’s probably the most successful thing I’ve ever done, okay[1]. So it was immediately used in sampling synthesizers. It was used in like every digital audio production ever. So unless you’ve been living under a rock for the last 30 years, you’ve almost certainly heard something that went through that algorithm. Okay. But can’t beat that. So anyway, that was a big success, but eventually, as the-- as it became clear that they weren’t going to get that NSF grant for the workstations, which was what I was nominally hired for, you know, it was, like, my job evaporated, right. So, you know, that was sad. So anyway, I, you know, easily found other work to do, so sort of went-- well, okay. So I had my own little-- well, our own little company. There were seven of us that were doing consulting work for various things, synthesizers, you know, various random electronics. So that went on for a year or two, and it just became clear that it was-- it just wasn’t much fun. It was kind of-- wasn’t fun. Excuse me. So I went around looking for work. I did some freelance stuff on-- --there was this little company called Memetics that was doing cards and, you know, accessories for Commodore Amigas, which were kind of a big deal at that period. So I did a frame buffer card for them, which was moderately successful, so that was kind of fun, and, you know, stuff like that, and then eventually, I’m not sure exactly how I got hooked into this. I think it was via the Stanford community, I got a gig with Hartmut Esslinger, who is the founder of Frogdesign, okay, which was this famous industrial design firm, and he wanted to do a home entertainment system, okay, all of his own, and that became FROX, F-R-O-X, which I guess sounds good in German, but, you know, it sounded kind of weird to me. Anyway, so I was hired to do the digital video, okay. So we wanted to do a line doubler, okay, so we had these really large projection displays, and to make that look good, you know, if you’re doing just normal NTSC with interlace, it looks kind of marginal, okay. So I came up with a line doubling algorithm and figured out how to do all the processing to extract the colors from the NTSC signal, which was weird, okay. So I designed all those chips. There are four of them, I think. They were gate arrays, I believe. So wasn’t full custom but, you know, nonetheless it was work, you know, that-- sort of the limits of what you could practically do without doing your own chips. So anyway, that was interesting. Oh, I guess before that was MIPS, right, which was where I met you. So I had a job there for about a year and a half, something like that. It was not a very happy scene, to be honest. There was this weird dichotomy. There was the chippers and the board group, okay. I never really understood why there was such a weird dichotomy there, but I was in the board group, and this was the R2000, okay, so the first commercial MIPS processor, and it didn’t have a floating point co-processor yet, okay, so that was some ways away. So they needed a floating point co-processor board, okay. So there were these Weitek chips, which were stand-alone multipliers and stand-alone adders, okay, and from this you could build floating point, although you had to do the floating point yourself or it didn’t-- it was integer, right, and you had to handle the exceptions, you know, all the denormalized underflows, all this horrible, horrible stuff. But worse than that, the way the R2K worked it had imprecise exceptions, okay, so you didn’t get to even know exactly what happened, okay. There was no way from the buses available to me to tell without a bunch of weird sleuthing what had happened. So you had to sort of deduce it by spying on the instruction stream as it went by and sort of knowing that, “Well, if the exception happens here and you’ve just seen this instruction, then it must be Tuesday,” right. So that was challenging and even worse than that, there was one thing that the chip designers did, which was the weirdest thing I’ve ever seen. I’ve

[1] [Interviewee’s note] https://ieeexplore.ieee.org/document/1172555

never seen it before or since. They actually made a pipeline go backwards, okay. So the way things work, at least at the time and the technology they were using, there were transfer gates. Okay. So these are like either the switch is closed or the switch is open and there's, like, two of them or more, and you have them in banks that are phased, non-overlapping clocks, and it kind of does this bucket brigade thing that moves the signal forward. Now, can't do this forever because there's no gain, there's no amplification, so eventually you'll run out of electrons and it'll all fall apart, but you can do this for a little while, okay. So they actually took advantage of this to go forward normally, but they could also go backwards, okay. So for certain kinds of exceptions, that was, I guess, the optimal thing to do internally. But of course, on a board, using discrete chips, you can't do that. Okay. There's nothing like that. So that caused fits. Anyway, I got the thing to work. I think I was the third engineer assigned to work on it. The previous two ran away screaming. <laughs> So I got it to work. It wasn't pretty but it worked, okay, and then left immediately, okay. I was like-- I was done with that. I heard after I'd left that some genius procurement guy decided that 15-nanosecond chips were cheaper than 12-nanosecond chips. Well, of course, we needed the 12-nanosecond chips, and, you know, anyway, they had problems with yield and then eventually somebody figured out, "Oh. Maybe this guy actually knew what he was doing." <laughs>

Mashey: Yeah. In retrospect, to be honest, we probably would've been better skipping the board and going straight to the--

Gossett: Indeed, indeed.

Mashey: --the co-processor, yeah.

Gossett: Right, right. So anyway--

Mashey: But we didn't.

Gossett: And, you know, the next version of the chip, I believe you guys fixed the imprecise exception thing.

Mashey: Yes. Well, being the operating system guy, I wanted no part of imprecise exceptions.

Gossett: <laughs> Why would anyone want imprecise exceptions, right?

Mashey: Well, I'd lived with an IBM 360-67 and never wanted to again, so yes.

Gossett: Yeah, yeah, yeah, yeah. Anyway, you know, lessons learned, so... Oh. There was one other amusing thing about that. So the board-- the chip group was mostly I think static RAM designers. Okay. That's where they came from.

Mashey: Yeah, there were microprocessor guys too, okay.

Gossett: Well, okay, okay.

Mashey: Tom Riordan and _______, yeah, but--

Gossett: Okay. Anyway, the board guys were all sort of mainframe-ish type people, okay, and anyway, there was the cache, the instruction and data caches, were external on the R2K. So the instruction cache in particular seemed very small to us, okay, and it didn't make any sense, because it was small by CISC standards, and RISC, which is a reduced instruction set, but therefore a more verbose stream of instructions, you would think you would need a larger instruction cache, right. Okay. But for whatever reason they convinced themselves that they didn't, and we argued with them, you know. "This is crazy," and I said, "Now, just--" you know, "We simulated it. Just do it." "Okay." So we just did it and sure enough, you know, the performance was atrocious.

Mashey: The first one was too small, yeah.

Gossett: Yeah, yeah, so it quadrupled it, I think.

Mashey: Immediately got doubled, right, and--

Gossett: Was it doubled or quadrupled? Yeah.

Mashey: Well, it went from 8K to 16K real quick, right, and then it got to be 32K pretty quickly after that. Right.

Gossett: Right, right. So anyway, that was just weird. It was such an odd thing because, you know, Hennessey, who's sort of famous for computer architecture textbooks, right, was the guy who was sort of calling the shots on that, and I don't know how he could've made that kind of mistake. I mean, it was just, it seemed obvious, but anyway. Got fixed. Oh. I should tell one more story. I hope you don't mind.

Mashey: Go for it. That's part of what these things are about is to get the stories.

Gossett: Okay, okay. So a certain John Mashey visited our lab once, and we had a board that was stuffed but not soldered, so this gentleman picks up the board, turns it over to look at the back. All the chips fall on the floor.

Mashey: <laughs> Oh, okay. I don't remember that.

Gossett: <laughs>

Mashey: I've forgotten that one. Okay.

<laughter>

Gossett: I think that actually happened twice. Okay. Anyway...

Mashey: I don't remember this. I don't remember this. <laughs>

Gossett: Okay. I think you were banned from the lab soon after. It's just funny. So okay. So that was pretty much MIPS, and then FROX. So FROX was really interesting. So it started out with Hartmut Esslinger, who was not a technology guy. He was an artist, right. He did industrial design. But he knew what he wanted, okay, and he wasn't a dope, and, you know, he had a clear vision and it was a reasonable vision, okay. Anyway, for whatever reason, and, you know, I don't know whose fault any of this was, there was a falling out with the investors, okay, who were really odd investors. One of them I think was an orange juice magnate or something like that. You know, was just totally non-technical money, and anyway, they fell out and fired-- the board fired Hartmut Esslinger. So they hired this very strange group of executives who I probably won't name but, you know, it's sort of irrelevant, and they didn't know what they were doing at all, and it wasn't just that they weren't technical. It was also, you know, they were dumb. You know, they just didn't-- they didn't know anything, okay. Business or technology. So anyway, they insisted on making a deadline. I'm pretty sure there were large bonuses involved, okay. I wasn't privy to this but it was fairly obvious. So, you know, they had to ship by I think it was Christmas Day, which doesn't make any sense, right. I mean, if you're going to go for the Christmas market you have to be in stores, you know, like, months in advance, right. So a Christmas Day deadline is ridiculous, and it manifestly didn't work, okay. I mean, the thing-- it just wasn't ready for primetime. It was buggy, okay, and, you know, we were telling them this. All the engineers were saying, "If you ship this, they're all going to come back and you're dead," okay. I mean, you don't get to do that twice, right, and they did it anyway, and exactly the predictable thing happened, and that was the end of FROX.

Mashey: And what was this product?

Gossett: This was a home entertainment system. So it had the digital video feeding high-end projectors. There was a digital sound system, you know, which was very fancy, and there was a controller, you know, thumb joystick type controller. All of which was cool. I mean, it was ahead of its time and, you know, it worked or at least could've worked if we just had a little more time to iron the bugs out. It was mostly, you know, there were-- it had this fairly complicated persistent object system, okay, and those are complicated, okay, and, you know, stuff has bugs until you don't, right. I mean, just takes time. So anyway, they faceplanted, and that was the end of that. So then I did some freelance work. I had my own little company, Gossett Graphics. Very imaginative name. Making video production hardware, so it was, you know, NTSC to S-video converters and, you know, stuff like that. Digital video disks. You know, that kind of thing, and, you know, that went okay. I eventually sold it off to one other company that went on and did whatever they did with it. Not for a lot of money. I mean, it was just kind of, you know, a little better than break even. But it paid the rent, so I was fine, and then right after that, that sale, I got a call from another set of, you know, friends that I'd worked with in the past who were at SGI, and they wanted to do a project with Nintendo, okay, which was sort of an odd thing, okay. I don't think that was really in the genome of SGI to do low-end things. They were the super-high end. So, you know, they were thinking about doing it using people internally, but it became obvious that this was probably not a good idea, so they hired people like me and a bunch of other people, who had done lower-end things. So anyway, that is easily the most fun and the most painless and successful project ever.

Mashey: Yeah, so talk about that.

Gossett: So--

Mashey: Was also a terrific financial deal for SGI.

Gossett: Indeed, indeed. So the financial part of it was they got royalties on not just every console that was sold but also on every cartridge, every game that was sold, and this is a goldmine, okay.

Mashey: Dave Corbin, Andy Kean special. <laughs>

Gossett: So, you know, I don't know who's responsible for this, but it was genius.

Mashey: Yeah, it was those guys. Yes.

Gossett: <laughs> Best deal ever.

Mashey: Well, there was NRE money too.

Gossett: Yeah, yeah, yeah.

Mashey: For doing that. <laughs>

Gossett: So, you know, there was no way they could lose, right?

Mashey: Yes. <laughs>

Gossett: It was the perfect deal. There's no down side and all up side. So anyway, they hired this-- well, they hired one architect who didn't last very long. I saw his work and it wasn't very inspired. Then they hired this guy, Tim Van Hook, who was very clever, and he had an outline, a sketch, I think he called it, and it was like 80 pages or something like that, that, you know, didn't go into too much nitty-gritty detail but just kind of spec'ed out what it was, how it worked, how things were connected, blah, blah, blah. So anyway, and he was mostly interested in two parts of this design, so there were two sections to it. There was the reality signal processor, was called Project Reality internally. There was the reality signal processor, the RSP, and the reality data processor or display processor, I forget which, the RDP, and the RSP was like the geometry engine, and it was also all the audio processing, okay. So depending on what you were doing you wanted different size data, okay, so it had this vector unit that would be either eigth 8-bit units, I think it was, or four 16-bit units or two 32-bit units. Something like that. I could be off by a factor of two on that, but anyway, there were some problems with that, okay. I mean, for the adds, that all makes perfect sense, but these were multiply adds, and multiplies go quadratically, right. It's a square of the size. So it didn't really tile very well, and we just ended up duplicating logic, right, you know, but that was fine. It actually still made sense, because from a programming point of view it was slick. Okay. So I didn't really have that much to do with that side of the chip. That was Mary Jo Dougherty was the overall architect of that, and she was very good, and I was sort of the architect of the RDP side, okay. So the thing you have to remember, this is like ’94, I think?

Mashey: Yeah, ’94, yeah.

Gossett: And, you know, technology was kind of just barely adequate. Okay. And, you know, the real graphics engines, the RealityEngine, I think. Yeah, yeah. Were like a million dollars, okay, so--

Mashey: Yeah, cost is no object, sing, dance.

Gossett: Yeah, yeah, yeah, yeah.

Mashey: Run multiple screens, the whole bit. Yes.

Gossett: Exactly. Exactly, and we were trying to make something that was going to retail for like 150 bucks, okay, so--

Mashey: Yeah, but with one chip.

Gossett: Yeah, yeah. So you can’t do that. I mean, you can’t do anything like that, okay. But we didn’t want it to look cheesy, okay, right, so we wanted it to look like it came from a RealityEngine, you know, be it a clearly reduced resolution, but, you know, we wanted all the fixings, so we wanted, you know, texture mapping--

Mashey: Texture mapping and the whole bit, yeah.

Gossett: --and we wanted anti-aliasing and, you know, all that good stuff. So the sort of classic ways of doing that were just way <laughs> too expensive, not even close. So what we did, so it was mostly me, but Tim Van Hook added a nice little embellishment, which I’ll get to in a sec, but, well, Tim came up with this clever way of folding the texture memory, okay, so you had mipmaps, they were called, and so power of two different resolutions and you’d sort of interpolate between two of them, depending on what level of detail you were at. This was the cheap way to do that. So he had this very clever way of folding the memory addressing so that it all, you know, optimized the use of memory, on-chip memory. But pretty much everything else was mine. So, you know, I didn’t design every gate but, you know, most of the gates, and I designed the architecture, you know, the detailed architecture. Oh, I also-- I think I designed every arithmetic element on that chip, okay, so that was one of my specialties was doing, you know, math units, and that-- we had a very limited choice of cell libraries to use for this chip, okay[2]. It was a full custom chip, but you don’t design everything from scratch. You have a cell library, right. So there was the so-called high-performance library and there was the so-called high-density library, which were really the low-density library and the low-performance library.

[2] [Interviewee’s note] https://arxiv.org/abs/quant-ph/9808061

Mashey: Library. Yeah, right. Right. Yeah, okay.

Gossett: <laughs> Okay. So the high-performance, low-density library was just-- we couldn't afford it. There was no way we could've gotten anywhere close to the target. But the high-density, low-performance library was nowhere near fast enough. Now, for a lot of the chip, this didn't really matter. You know, we weren't pushing things. Except for the arithmetic parts, okay. There you really cared about performance. So what we did was we contracted some library design company to make I think it was a half dozen carefully selected cells. Okay. So there was like a D-flipflop, a full adder. You know, a 2-1 MUX NAND gate, XOR, okay. I forget what the last one was, but anyway. They were very carefully chosen to fit the design for the arithmetic elements, so that worked great. Those knocked it out of the park. So anyway, the other things that I had to do, we couldn't quite afford the normal trilinear interpolation that you do for a mipmapped thing. So I came up with this hack, which was dubbed triangular interpolation, which split the four-element square in each of the two levels into two triangles, okay, and then you got to save a couple of multipliers by doing this, okay. Which mattered, okay. We were counting gates. So I did that, and that gave it sort of this interesting characteristic look. You know, things looked sort of hexagonal kind of. You've seen the graphics.

Mashey: I owned one of these things, so yeah.

Gossett: So, you know, had character.

Mashey: Yeah, it had character, yes.

Gossett: It didn't look bad. It just...

Mashey: No. It looked, I mean, for the time, it looked really pretty good. Yeah.

Gossett: Yeah, yeah. So there was that hack. There was-- we wanted phong shading, you know, the highlights on Mario's nose, right.

Mashey: Yes, yes, Mario's nose, yes. I had that Mario, yeah.

Gossett: So we couldn't afford to do that in the normal way, so I came up with this hack where you used the X and Y components of the half-angle normal, okay, and used that as addresses into a texture, okay, and in the texture you have a little Gaussian bump, okay, and that was the highlight, and that worked great and cost nothing, okay. It was very, very effective. It was just software to calculate the half-angle normals, and what was the-- well, okay. So there were two other really big ones. One was-- we built the first version of this. There were two version, you know, two generations of the chip. The first generation we knew wasn't going to ship. It was way too big.

Mashey: Yeah, sure, yeah.

Gossett: We weren't even trying to make the target.

Mashey: Well, let me back up a level, right. There was a-- let's see. It was a MIPS R4200, right?

Gossett: Right, right.

Mashey: Which was a bus size cut down version. Okay. Right.

Gossett: Right. That was sort of optimized for game use, and that was a separate--

Mashey: Right, and it was the graphics chip.

Gossett: That was a separate project and then there was the graphics chip.

Mashey: Yeah. Chip, yeah.

Gossett: And that was--

Mashey: And what else was in that?

Gossett: That was it.

Mashey: That was it, yeah.

Gossett: Yeah. I mean, there was some security nonsense and, you know, whatever.

Mashey: Yeah, okay. Was some RAM someplace, right?

Gossett: There—Rambus, yes, right . I'll get to that in a second, okay.

Mashey: Yeah, okay.

Gossett: So actually, let's get to Rambus now. So one of the obvious mistakes that they made was choosing Rambus. Sorry, Rambus, but it was supposed to be a cost saver. So the Takeda-san, who was the Nintendo exec who was running the project, you know, "Two yen per chip." Okay. was going on and on. Or two yen per pin, right. So he was counting pins, right, because it cost, and, you know, true enough. The unfortunate reality of this was that Rambus was eight-way multiplexed, okay, so it was getting eight data bits across every clock. Okay. Now, buses are a challenging thing under the least constrained circumstances, and if you're going eight times faster than normal, you know, I mean, duh. This is not going to go well, right. So you have to do things, and the things that you tend to do is have lots of grounds. Okay. So yeah, it had fewer single pins, but it made up for it with ground pins.

Mashey: With grounds, yeah.

Gossett: Okay. So I don't think it actually saved anything on the actual pin count, and it was a nightmare from a sort of architectural point of view, because the inputs and outputs, the data ins and outs, were shared. They were multiplexed, right. So you had to turn the corner, right, whenever you wanted to go from reading to writing, right, and we weren't allowed very much in the way of buffering on chip, as we were trying to cut die size, cut cost. So that really sucked performance. There was just no way they could be helped. But, you know, we were stuck with it, so we did the best we could, and anyway, so going back to the RDP. So there were two tricky things. One was we didn't have very much memory, okay. So even though, you know, this was, like, considered a high-end memory chip at the time, it wasn't very high-end, okay. I think it was half a megabyte, which sounds ridiculous today, but...

Mashey: Yeah, but it's right.

Gossett: Anyway...

Mashey: That was then.

Gossett: That was then. So, you know, we wanted to have a reasonable sized frame buffer, right, but we didn't really have anywhere to put it, right. So we couldn't afford to do even, you know, 8-bit per component, right, which would be a 32-bit word, you know, with some alpha. So it had to be 5-5-5. Okay. Five bits per component, and because it was a Rambus chip with parity, it was actually 9 bits times 2, so it was 18 bits. So it's 5-5-5-3, okay, so 3. What are we going to do with that, right?

Mashey: <laughs> Yeah.

Gossett: So I came up with this-- well, two things. We-- first thing we did was just implement that straight up, and it looked terrible. You had to dither the five bits, so you compute eight bits and you dither it down to five bit. The problem was that the dithering interacted with the NTSC color system. Okay. So it just looked, I mean, it probably would've caused an epileptic seizure, you know. So it was hard to look at. I mean, it was painful. So it's like, "Oh, crap. We can't ship that," right. So I came up with this dither correction filter, okay, which was cute[3]. So what you would do is you had a couple of line buffers on the way out to the display, okay, and we could just barely afford that. That was acceptable, and you used that to form a little three-by-three-pixel region, and you'd look at the color of the center pixel and then look at the color of the pixels surrounding it, and if they were, you know, above you'd add one into the LSB's and if they were below, you'd subtract one or the-- or yeah. Anyway, this was just enough to use the neighborhood to smooth thing out so that it looked right. Okay. It was surprisingly effective, okay. It sounds lame, but, you know, it worked. Then the other thing was anti-aliasing, okay. So if you do nothing, you get jaggies, okay, and it looks awful.

Mashey: Yeah. You can't have it, yeah. Right.

[3] [Interviewee's note] https://patents.google.com/patent/US5699079

Gossett: You can't have that. So we couldn't afford to do the usual thing, which was to multi-sample, so the traditional multi-sample in the aliasing algorithm is you chop each pixel up into a four-by-four subpixel region and you compute all 16 of those subpixels in the average. Okay. That works great, okay, that's fine. But, "pht," there was no way we could--

Mashey: But you couldn't afford, yeah.

Gossett: --afford that, okay. So you have the foreground color, okay, and you can compute the coverage of the subpixels that you would normally have. Now, you know, you'd sort of like to have a count of the subpixels, so you'd sort of liked to have had four bits, but we only had three, but it turns out if you just do sort of a checkerboarding of those subpixels you get eight black squares, okay, and that's enough, okay, that's good enough. Okay. So we have the foreground pixel color and we have the coverage count, okay, but we don't have the background color. So what do you do? Well, okay, so we already had-- we'd already paid for these line buffers, okay, and, you know, could only be three pixels high but it could be any number of pixels wide, okay. So if you just did the three-by-three thing and used like the-- well, basically what the algorithm was, you used the maximum and the minimum of the neighborhood, okay. Now, one of those is going to be more or less the pixel you're on and the other one's going to be the background, okay. Except you don't actually want to use the maximum and the minimum. You use the pen max and the pen-- the next to maximum and next to minimum to cover sort of corner cases, okay. That's good enough, and then if you add the max and the min and then, you know, one of which is the foreground, one of which is the background, and then subtract the foreground, which you know, because we're sitting on it, then you're left with the background, okay, and then once you have the background on the foreground, you use the coverage value to interpolate between the two and you're done. That's anti-aliasing, okay, but the eight neighborhood, you know, the square neighborhood with four corners and four vertical and horizontal, doesn't work, because the corners are square root of two further away than the other guys, okay, and that mattered. You could see it. So I came up with this thing where you do a three-by-five pixel neighborhood, okay, and you checkerboard that, okay, which leaves you with seven pixels, so there's a pixel on the center, and then there's six pixels in sort of a hexagonal arrangement. Okay.

Mashey: That's the hexagonal thing, yeah, right, okay.

Gossett: Okay. Now, it's squished, but it turns out that doesn't matter. It's an affine transform away from being, you know, the right distance, equal distances, and you can't see that, okay. So you take that six neighborhood and take the penultimate maximum and penultimate minimum, do the same thing. You know, add them together, subtract the foreground and you're left with the background, done, okay, and these two algorithms, the one for the dither correction filter and the one for the anti-aliasing, didn't really play well together, but if you were doing the anti-aliasing thing, the dither didn't matter, because you were on an edge, okay, and that overwhelmed whatever the dither pattern was, and if the coverage was full, okay, then you weren't doing anti-aliasing, so just do the dither thing. So we just switched which one you were doing based on whatever the coverage with zero or one, and the embellishment that Tim Van Hook came up with, which was very clever, I never would've thought of it, was to add transparency, okay. So this works fine if everything is opaque, but what if you have, like, a, you know, the glass on a cockpit of a fighter or whatever, right, and turns out you can just let the coverage value wrap, okay. So you just keep adding it on top of it and to do the mod-8, you know, just throw away the higher bits that are falling off the edge, and that does exactly the right thing, so that was that.[4]

Mashey: So this sounds like an example of lots of clever tricks to fit a really constrained design.

Gossett: Exactly. I mean, that's exactly what it was. You know, this was all insane at some level, and, you know, if we didn't have to do it, we wouldn't have done it, and it was a dead-end, okay. So as soon as you could afford to do multi-sample anti-aliasing, everybody did that, <laughs> right. I mean, you know, this other stuff is crazy, right. Some other crazy things about this. So we were under time pressure. You know, they wanted to ship by a particular Christmas, which meant many months before that particular Christmas, and we actually missed the original target, but they kind of knew that we probably weren't going to hit it. But the next Christmas we were like damn well going to hit, okay. So the project wasn't really staffed quick enough. So I was among the last people staffed onto it, and from then until the final, you know, tape-out, well, the chip coming back, you know, ready to go, was 18 months, which is phenomenal.

Mashey: Yeah, it's insane. Yes, yeah.

Gossett: Okay. So we had the first chip, which was done very hastily. You know, there was no effort to make it optimal. It was like twice as large as it needed to be, and it had all these problems, you know, the dither issue and some other things. There was as bug that caused it to hang after a few minutes. It's one of these things that you can't simulate, right, because, you know, real-time is real and, you know, simulating out for a couple of minutes would've taken, you know, decades of simulation, even with an army of machines slaving away. So anyway, it was very wise to do that first chip because we caught all those bugs, and fixed them, okay. So second chip, you know, the natural instinct would've been to do as little as possible, okay. Not change hardly anything. But we knew we had to change a massive amount of stuff, and there's just no way you can sort of half-change everything, okay. So we had to change everything, and, you know, the rest of SGI thought we were out of our freaking minds, but we did it, and, you know, it worked. There were a couple of things that didn't work but they were nonessential, so we just declared them non-features and, you know, moved on. So that was amazing, and just, you know, there was, you know, there was disagreements and the occasional yelling and screaming, but it was all productive. You know, it was all-- everybody was rowing in the same direction.

Mashey: So how many people were, how many designers were in there?

Gossett: Right. So there were about 25 physical design people that were doing-- well, there was the team that did the MIPS-chip, which also did the physical design for the graphics chip, and there were about 25 of them. Then the rest of the team was maybe half. No, was a little less than half software and the rest hardware, so, you know, I don't know, 14-ish, some number like that, 15, maybe, and, you know, it was sort of divvied up. There were the leads, you know, Tim Van Hook, Mary Jo Dougherty, myself, and then there were, you know, people who were designing individual units, and, you know, everybody

[4] [Interviewee's note] https://patents.google.com/patent/US5742277

was good, okay. Or great, actually. You know, everybody knew what they were doing. There was a unity of vision. You know, it was clear what we had to produce, and, you know, there were no arguments at that level. There were minor technical disagreements, but, you know, that's always going to happen, so-- and amazingly, the company let us go. So there was this-- the VP of everything, I called him, Wei Yen, who was the grand poohbah, the guy running the operation, and he was a really <laughs> amusing fellow. You know, I'd get called into his office every week and on alternating weeks he'd either chew my ass off or praise me, okay, <laughs> and pretty much awesome, all right, and, you know, it was-- it got to be kind of amusing, you know, sort of a joke, but, you know, it was disturbing at times but on the other hand it all worked and it was fine, and the guy was phenomenal. He had this blackboard in his office that was divided up into tiny little squares. He was running 80 projects, 8-0, projects, okay, so he'd have people in for half an hour, okay, and he knew everything, okay. I mean, he was totally aware and got all the technical subtleties and, you know, you didn't have to explain things to him. He's a really amazing fellow. I've never seen anything quite like-- well, maybe one thing quite like that, but I'll get to that, and anyway, it worked, it shipped, and woohoo. You know, it made the company tons of money. <laughs>

Mashey: Tons of money. Yes.

Gossett: I think they ended up spinning MIPS off, at some point, and a large part of that valuation was because of the Nintendo money, okay. So that was great. Unfortunately, SGI did not do so well thereafter, so, you know, I was looking for something to do. I really should've left and gone to something else, but, you know, I was sort of a local hero, right, when I pulled this thing off, and, you know, was just seductive to stay there. So there were three divisions, three graphics divisions. There was AGD, the Advanced Graphic Division, the Desktop Systems Division, and I forget what they were called but basically the low-end division, okay, and, you know, originally, we were going to work on the low-end division. You know, makes sense, right, as that was kind of the space I was in. So I went off and designed a somewhat embellished variant of the Nintendo architecture, and anyway, they decided they didn't want to do that. They knew what they were doing, "Go away," okay. Fine. So the Desktop Systems Division had just finished shipping, on this horrible death march, the-- I think it was called the Octane Series of--

Mashey: Well, it was Octane was the midrange, right?

Gossett: Yeah, right.

Mashey: Yeah, yeah, and Indie was the low end.

Gossett: Indie was the low end, right.

Mashey: But they'd done Octane, yeah, right.

Gossett: Right. So they had just done Octane, and they were fried, okay. They were just basket cases, and very stupidly they, the management, told the team that they would all get to be architects. Well, <laughs> that's meaningless, right? I mean, you can't-- if everybody's an architect, nobody's an architect, right. That's just silly. So anyway, I had this architecture that I'd done, you know, working on the low end, and they didn't like it either, but they didn't have anything, okay. They had no handwave of an architecture, okay. They weren't even close to what the next thing was going to be. So I was, you know, <laughs> I was invited to this conference room to give a talk on, you know, what the architecture was going to be, and they were hopping mad. They threw things at me, okay. They were hopping mad and, you know, I'm up there, "Look. What's the alternative? Do you have an alternative?" You know, "I'd love to hear it." Okay. Well, "No," and anyway, after much wailing and gnashing of teeth they sort of acquiesced and I was the architect of that. Oh, boy. So this was as SGI was plowing into the dirt, okay. They were just-- they couldn't catch a break, you know, the-- nothing was going well. The--

Mashey: And of course, buying Cray was not--

Gossett: Well, buying Cray was incredibly foolish, okay. I mean, it just made no sense at all. Their, you know, what was it called? Their version of the supercomputer was actually better than Cray, okay, and, you know, there was no reason to buy Cray, so that was an enormous distraction. They eventually fired McCracken, who had been the CEO for years and years, and hired this guy, Rick Belluzzo, "Rocket" Rick Belluzzo.

Mashey: Rocket Rick. Yes.

Gossett: <laughs> That didn't go well for-- you know, he was from, you know, PC kind of manufacturer, and just a different world, right, and then the other thing was sort of obvious. The world was moving on, so all these graphics cards, you know, that were being produced by various companies, and video, most notably, were eating their lunch. Okay, they were creeping up in performance and good enough is good enough, right. So they didn't see that coming, which I don't understand. It was obvious to all of us.

Mashey: Well, some of us saw it coming.

Gossett: Yeah, yeah, but, I mean, the--

Mashey: Some of us wanted to do a follow-on for-- you know about that, right?

Gossett: Yeah.

Mashey: Yes, okay. <laughs>

Gossett: And I mean a follow-on for the Nintendo thing would've been like the obvious thing to do, okay, and they could've owned it, because they had the street cred, you know...

Mashey: And all the patents.

Gossett: And all the patents, right, right. So, ah, God. Anyway, I'm toiling away on Odyssey, aptly named, okay. Never name a project on Greek mythology, okay. It's, like, bad karma. <laughs> Anyway, it was an Odyssey. Almost everyone died, you know, <laughs> and it took years, right. So constantly moving target. So this is something that happens in high-tech companies. They-- it's the opposite of drinking your own Kool-Aid. It's believing everybody else's press release but not your own internal projections, okay. Now, indeed, your own internal projections are probably optimistic, but so are the press releases from these other bozos, okay, so why would you believe them and not, you know, maybe discount but partially believe us, right? So there were all these announcements about how super-high performance these competing workstation-y machines were going to be. So we were constantly having to one-up ourselves and, you know. Which we did, actually. It was stupid. We should've shipped it a year earlier, right, but we did what they wanted, and it shipped and, like, it actually shipped slightly after they fired everyone. <laughs>

Mashey: Oh. Oh, good, yeah. <laughs>

Gossett: Okay. So the chip hadn't even come back yet. It'd taped out, okay. There were two chips, actually. There was a display processor and the main processor, and it taped out. It did have some minor bugs. There was this-- it had this recirculating pipeline thing, which was one of my inventions, which was a way of dealing with exceptions at the level of memory, okay. So the "page isn't available," you know, that kind of thing. So that worked, but there was a one-gate bug that caused it to not, under certain rare, very rare, circumstances, dismiss the pixels that had already been processed, that are in recirculation. So they'd just keep going around and around and around, and eventually the pipeline would fill with these dead pixels, okay. It took, you know, eight hours or so, typical use for this to happen. So nominally they could've just flushed the pipe every 10 minutes or whatever and been pretty safe, but they didn't want to do that and, you know, I can kind of see why. So anyway, one guy, you know, was last man standing, I guess, got to debug that and he did. You know, he figured that out and he fixed it and it shipped, okay. So <laughs> they changed the logo, which was also kind of stupid. They had this really nice 3D cube logo, and they changed it to this weird lowercase font, you know, just SGI.

Mashey: Yes, of their own font, yes.

Gossett: Their own font.

Mashey: I remember this all too well.

Gossett: Oh, God. I mean, if you're worrying about fonts, you know, you're dead. Okay. I mean, if that's the most important thing to you, you've already lost. <laughs> So anyway, you know, it was SGI and, you know, we were sort of drawing lines through each letter. So first they got rid of graphics, then they got-- well, first they got, yeah, first graphics, then silicon, okay, and then finally incorporated, right, I mean... <laughs> Anyway. So this was '99, I think, that they--

Mashey: Yeah, yeah, about that time, yeah.

Gossett: --closed the door, and they actually, I don't know why, but they treated us better than they treated AGD, the high-end guys. So we, you know, were told we were fired, and, you know, they processed us out and it was, you know, civil, right. I mean, it wasn't ridiculous. They-- <laughs> they brought the AGD people one by one into a room that had an internal door and an external door. Okay. And just told them, "You're fired. Out."

Mashey: Oh, great.

Gossett: Which, you know, there are people in tears, right. I mean, this is just uncivilized and pointless, right. I mean, why would you do this to anyone? Anyway, that was a very ugly scene, and, you know, anyway, so it took me think it was-- was it 24 or 48 hours, I forget, to get a new job?

Mashey: Ah.

<laughter>

Mashey: Okay. Yeah, right, yeah.

Gossett: So--

Mashey: Course a lot of graphics folks ended up over at Nvidia, again, right?

Gossett: Yeah, yeah, so that was another ugly thing. So I don't know what happened there, but they did this deal. I mean, they had them dead to rights.

Mashey: Yes, yes. Was Belluzzo.

Gossett: And for whatever reason, they decided to rather than fighting them and getting tons of money out of them, which they could easily have done, they gave them the patents, and, you know, I don't know how you could justify that. I mean, if I were a shareholder, and I guess I was, but, you know, if I were a major shareholder I would've, you know, hung them by their short and curlies, right. You know--

Mashey: Should've owned a big chunk of Nvidia.

Gossett: <laughs>

Mashey: Yeah, well...

Gossett: Anyway, it was insane. It was just completely nuts. But that was effectively the end of SGI. They did sort of limp along thereafter, and they kind of realized maybe, you know, six months to a year later that they really screwed that up, and tried to get people back and nobody was having any of it. So, you know, I mean, yes, engineering is a cost center and marketing and sales are profit centers, but you have to have a thing to market and sell, and the shelf-life of high technology is not very long. Things get old, and after a very short time they didn't have a product and "pht."

Mashey: Yeah, in that turf, yeah.

Gossett: Yeah, yeah, so... So they sort of concentrated on the processor side, I guess, you know, the computer side.

Mashey: Well, you know, the supercomputer side. The Origin 3000.

Gossett: Yeah, yeah, the Origin Series. You know, which was a fine machine. But anyway, they continued to slide slowly into the dirt and that was the end of that.

Mashey: I used to give talks internally from Clayton Christensen, you know, "Innovator's Dilemma,"--

Gossett: Oh, right. Yes, yes, yes.

Mashey: --why you need to worry about what's coming up from below. <laughs>

Gossett: Cannibalize yourself. Somebody else will do it for you if you don't, right. That's obvious, and they were perfectly positioned to do this, but anyway.

Mashey: Well, that's why some of us really regretted not doing another round of the Nintendo-like stuff as the single-chip thing for PCs, you know.

Gossett: Right, right, right. So anyway, that all imploded.

Mashey: So--

Gossett: Go--

Mashey: So then you went off...?

Gossett: So the next thing was this flaky little startup, Pulsent, which they were doing this video compression thing, and this is another lesson. They had a 5X improvement over the state of the art and they thought, "Oh, great. We can capitalize on this." Well, no, 5X isn't enough, okay. I mean, things, you know, it's 5X over what's available today. Your thing won't be available today.

Mashey: This is classic. Classic, yes.

Gossett: Okay, so by the time your thing rolls out your 5X is going to largely evaporate, okay. You know, you-- the real rule of thumb is 10X. If you're not at least 10X better than the state of the art, don't bother, and they weren't. So they discovered that the hard way, and that, you know, "poof." [ph?]

Mashey: A lesson many have discovered, yes.

Gossett: Indeed, indeed.

Mashey: And forgotten. <laughs>

Gossett: So then I'm like, "Okay. What do I do now?" So I had, you know, some savings and an idea, so this was a total change of gears, but it occurred to me that you could do this very interesting thing with radio, okay.

Mashey: Okay.

<laughter>

Gossett: So, you know, the cell phone companies are sort of legendary for being unfriendly, shall we say? Okay. And this seemed like, "I can eat their lunch," okay. So, "I can come up with some technology that'll make it possible to just commoditize them away," okay. So a friend of mine who worked at SGI on the Odyssey project, and also at Pulsent, and I decided we'd just do our own company, and do this idea. So he didn't-- he had a family and needed to, you know, pay the mortgage and all that stuff. I didn't, right, so I was willing to just coast, you know, not take any salary. He needed a salary, so he needed some money. So I had some friends, one in particular, Shel Kaphan, who I mentioned earlier, who had just left Amazon with bags of money, and he was doing seed investment. So he was very generous and sweet and, you know, thank you, Shel, and he gave us some seed money. Think it was half a million dollars. Something like that. Which was enough to keep Mike Gunter going for several years, actually, okay, and that was really-- there weren't very many other costs. I mean, there was a little bit of hardware that had to be bought, but it was negligible, and there was lawyers to be paid, which, you know. So anyway, we did this thing and we were looking for either another round of funding or some kind of exit strategy, a sale, whatever, and ultimately-- so I gave a talk at a hackers conference and Larry Page happened to be in the audience, and Google had been kind of unhappy with the cellular world, because they were a gateway for them and they were-- <laughs> they were having to pay them lots of money to get their stuff out to the market, right, so there was a desire to do an end run-around them if at all possible.

Mashey: Mm, and so when was this?

Gossett: So this was '05. Okay. So Google had just gone public. It was like a year or so after the IPO, and anyway, they-- well, actually, they started talking to us before the IPO, but it took a really long time. Google hiring, buying companies, all of that. They're very, very slow. And we were really hungry. I mean, I was eating Ramen noodles again, okay?

Mashey: That's being really hungry, that's okay, literally.

Gossett: Yeah, yeah. So it was like grad student days, right, that sort of vibe. And anyway, they did come through. So, thank you, Larry, and they hired us. Not for any serious money. I mean, it was basically break even. You know, I think I made-- for the five years that I had worked on it with no pay, I ended up with twice the minimum wage, I think it worked out to be. But eh. But you know, I did get to work at Google relatively early and their stock did quite well.

Mashey: That's not bad, yeah.

Gossett: So thank you, Larry! Okay, I mean, almost all of the money I have now is thanks to them. So anyway, you know, we did this project, and the thing we were originally doing was SISO, you know, Single-In/Single-Out. So it's one antenna on the base station, one antenna on the cellphone. And you could-- this was using the so-called high-frequencies, which is a joke. This is from three to thirty megahertz, not gigahertz, megahertz, okay? And the FCC rules on this were shockingly permissive. So you could intentionally radiate-- or this whole think about unintentional versus intentional radiators-- but you could intentionally radiate, not a lot of power, but a little power. Okay, and with enough so-called processing gain, so a very spread-spectrum, you can get some data across. And by cleverness, you could gang these up and there were some regulatory shuck-and-jive about that. You know, so you couldn't have like coordinated ganging together, but you could have uncoordinated ganging together, okay? So we were uncoordinated. And we did satisfy the letter of that law. It was legit. So anyway, we could get up to a few megabits per second per cell, which isn't a lot. And we had a range of like one or two kilometers, which isn't bad. That's like given that it's not a super high-capacity cell, you don't want it to be too big, right? And it propagated really well, because it was so low frequency. It went through pretty much everything. So Google thought it was a good idea. I thought it was a good idea. But it wasn't a good enough idea and this became clear as kind of shopped around internally. So it occurred to me that you could embellish this, and there was this interesting regulatory thing that was on the verge of happening, called the TV whitespaces. So there was this transition from analog television to digital television and digital television is much more bandwidth efficient.

Mashey: Yeah, sure, yeah.

Gossett: So they didn't need anything like as many channels as they had originally allocated. So what they wanted to do was let-- consolidate the channels into these digital channels, and the ones that weren't needed anymore would become open, free whitespace.

Mashey: Yeah, whitespace, yeah.

Gossett: So it was like WiFi. It was unlicensed but regulated. So great! This is like a spectacular chunk of spectrum, propagates really, really well. It's got all the right properties. But to exploit this, you know, you couldn't do the thing we were doing. It wasn't good enough. But because of the nature of the bandwidth, the fact that it had a large fractional bandwidth, so the, you know, the ratio of the difference between the low- and high-frequencies to the center frequency was large. It was like 30 percent, which is huge. Generally, it's tiny fractions of a percent for most cellular systems. So because of that, you could do some really cool things. And one of the things you could do-- and also because the bandwidth was somewhat higher than the stuff we were originally doing, and therefore the wavelengths were somewhat smaller, considerably smaller, you could have an array. So you could do MIMO instead of SISO[5].

Mashey: You could do MIMO. Yeah, yeah.

Gossett: And anyway, long story short, we spent a few years doing this. And just as we were about to get to the point where we could really show that it was all working, the FCC voted five nothing to let the spectrum be unlicensed. This was the day before Obama was elected. So that was like two really good days! <laughs> And then Google lost its nerve.

Mashey: Oh, no.

Gossett: And I really don't know what happened there, but they decided that they would rather make a deal with Verizon, in particular. Who promptly stabbed them in the back, as you-- any fool would have guessed, right? And this was over net neutrality. So they were going to use Verizon's system, and you know, they weren't interested in what we were doing anymore. So, you know, we continued going and that took a lot of cajoling <laughs>, but we did eventually, you know, couple/three years later, maybe it was four years later, I forget-- anyway we got the whole system up and running. We'd done some limited demos, so we had a 12-by-8, I think it was, MIMO system, which worked. And showed that, completely against intuition, as you increased the number of antennas, the capacity went up super linearly. So what happened was because you had all these extra redundant degrees of freedom, you know, they sort of covered over whatever gaps there were. So the overall system approached optimal. You know, it approached full capacity as you increased the number of antennas. So we got up to 12-by-8 and we showed, by disabling varying members of antennas, that it had this property and pretty much confirmed everything we said. And that we had working like a year after the FCC thing. So that was totally viable. And they could have changed their mind. They could have gone and petitioned the FCC. The FCC commission was like asking us, begging us to make a proposal, right? "Well, we've given you the spectrum so what are you going to do with it?"

Mashey: Oh, yeah.

Gossett: And they just didn't want to ruffle feathers or whatever? I'm not sure exactly what the argument was. But anyway, the FCC because nobody seemed to be terribly interested decided that, "Well, we're not going to just throw this spectrum away. It's perfectly good spectrum. So let's auction it off."

Mashey: Oh, yeah.

Gossett: Okay. Well, so that spectrum, you know, this is below 700 megahertz. The spectrum above 700 megahertz was successfully auctioned. Which we were actually involved in, so Google was bidding on that. Mostly to guarantee that they crossed the threshold for net neutrality requirements to kick in, which they did. And actually Verizon did very well by this. AT&T, for whatever reason, really botched that

[5] [Interviewee’s note] https://arxiv.org/abs/1605.01696

auction. You know, I can't really explain what happened there, but they ended up paying way more than Verizon paid for comparable chunks of spectrum. And Verizon just skunked them, and they were idiots. You know, but anyway, so but no, even though we'd shown that we had this super linear performance, they still weren't interested. But they did, because it was relatively cheap, you know, I think the whole project, which ran for, what was it? Seven years? Something like that. You know, I think we spent maybe 30 million dollars, including salaries and hardware. So that's roundoff error to Google, right? So anyway, that was fun. And but, unfortunately, not successful. So we did finally get it up and running. We had it in a warehouse that was somewhat shielded so we weren't screaming over the countryside. And you know, had a 32-way MIMO system. So it was asymmetrical. It was very nice. So you could have a base station that had 32 antennas. They were all coordinated. And then you could have 32 independent uncoordinated cells, or cellphones, devices. And it could keep track of them all, and individually channel signals, different signals to each of them at the same time with the same frequencies. And it could do this because it was very spread spectrum. It could do this right at the noise floor as seen by the receiving side. So it was super-efficient, and had quite long range. I think ten kilometers or something like that.

Mashey: Wow.

Gossett: Which was more than you needed. Excuse me. And because the spectrum was so nice, you didn't have to be like ridiculously high, so we were thinking we'd just rent out rooftops of tall buildings. And that reduces your deployment cost to nothing. The base stations could have been made for like a hundred thousand a pop, you know, in any kind of quantity, which is totally reasonable. We could have covered the bay area with, I guess, 40, I think it was, base stations. And you could have covered the whole country. I think I estimated, and this is maybe a little optimistic, something like half-a-billion dollars. Which is nothing.

Mashey: Which is nothing, yeah, for that.

Gossett: So a lost opportunity, but what you going to do? So anyway, that all imploded, and we were-- I mean, once we-- I was determined to prove that it worked. Because it was sort of a matter of personal honor.

Mashey: Yeah, sure.

Gossett: There were people that were accusing me and my team of fraud. And I like take that very personally.

Mashey: Yeah, yes, yeah.

Gossett: You know, the damn thing works. And we proved it. But there was nowhere to go from there, though. That was it. And then <laughs> so I was looking for something to do, and I was still at Google, which was foolish, I should have just left. But I stuck around. And eventually-- I wanted to work on Android. That seemed like a natural segue. But they weren't hiring at that time-- for whatever reason. And so Google-X was hiring for what became Google Glass.

Mashey: Hmm, yes.

Gossett: And this was shortly before the launch, but they weren't ready for prime time. They had a lot of problems, sort of across the board. One of which was their audio was atrocious. They had this bone conduction thing, well, it was supposed to be bone conduction, but as frequently happens, they did a compromise between audio quality and comfort.

Mashey: Mmm.

Gossett: So the problem with bone conduction is it needs to like be impinging on your bone, and that's actually not comfortable.

Mashey: Yeah, right.

Gossett: Duh. Okay? So what they did was they loosened it up, and sort of got the worst of all worlds. So it was uncomfortable and it didn't work well. So what it effectively was was the world's least efficient loud speaker. <laughter> So it was just sitting there vibrating making sound.

Mashey: Oh, good.

Gossett: And it was also the world's crappiest speaker. So there's a figure of merit, total harmonic distortion. And like a good audio system will be like .02 percent total harmonic distortion. And sort of a cheap very sleazy grade audio system might be .2 percent total harmonic distortion. Well, this thing was 200 percent total harmonic distortion.

Mashey: Oh, geez! <laughs>

Gossett: So the distortion was louder than the signal. Now, they weren't going for music. That would have been utterly ridiculous. They just wanted voice. So with unbelievably heavy-handed audio processing, you know, compressing the hell out of it, and just doing every trick in the book, I managed to get it to where it was intelligible. I mean, I wasn't nice. But it was intelligible. It was supposed to be private. You know, the whole argument was that because it was bone conduction the person next to you couldn't hear you. I could hear people across the room.

Mashey: Oh, great!

Gossett: Because it was a loud speaker.

Mashey: Yeah, sure, yeah.

Gossett: And it was behind your ear, so it wasn't even optimized for you, okay? And I pointed this out, repeatedly, and "Yeah, yeah, yeah." Anyway, that went on for a couple of years, I guess. And they were going to do a Version 2. There was a thing I think they called Version 2, but that was really 1.5. It was just a minor upgrade and there was supposed to be a whole new thing. And that was even worse. I mean, they fixed the speaker problem at my insistence. And there was also a problem with the microphone, because the microphone was up here, so it was very close to your nose, so all you heard was <nose-breathing>.

Mashey: Oh, good, yeah, okay.

Gossett: You know, it was like Darth Vader or something. So anyway, I talked them into-- well, I wanted them to do an array of microphones along here. So four microphones in a line, and with that I could have gotten reasonable audio quality on the microphone side. But the industrial designers, you know, were just like completely unwilling to accommodate anything, really. It wasn't just that. I mean, there was other stuff that people wanted and needed to do. But they just wouldn't have it. You know, it had to look *sleek*! Anyway, there was one microphone here, there was one microphone that was pointing in, which on people with round faces, a lot of Asians in particular, was touching their skin, so it was occluded. So it's useless. There was one that was-- there was one that was pointing backwards, which was almost useless. And then there was one that was in the speaker enclosure.

Mashey: What?!

Gossett: Okay? So you know, there was one useable microphone, which was this one. They hadn't done anything really. And anyway, long story short, it was obvious, you know, they put it together, the first version, the first implementation of V2. They got it together, a couple of units to give to Larry and Sergei, I guess. And they looked at it, and like, "This is crap." Right, you know? <laughter> "What have you been doing for the last two years? This is appalling." So they eventually shut it all down. But it got really ugly towards the end. And there was a bunch of scapegoating, and I was one of the goats. And that was that! And to be honest, I was happy to be the hell out of there. I was dying. And it was miserable. So anyway, that went the way of all things flesh, I guess. And then one of the more interesting engineers that I've seen is this very talented engineer, Jeri Ellsworth. So she's a self-taught engineer, I guess you would say, and very good. Very innovative, very, very, very sharp. So she had this really interesting augmented reality thing. It wasn't like Google Glass, it was like-- I would call it annotated reality.

Mashey: Yeah, sure, yeah.

Gossett: But this was actually augmented reality, so you'd see a virtual thing in the real world, overlaid on the real world. And it worked by this really bizarre method. They were using retroreflective material. And they had a projector up here that would shoot down on the retroreflective material and then back into your eyes. And I mean, that sounds really lame, but it actually worked really well. And at least for a certain category of, you know, use.

Mashey: What were the applications?

Gossett: Games.

Mashey: Games, okay.

Gossett: So it's sort of a-- it's like the chess game in Star Wars.

Mashey: Yes, yes.

Gossett: You know, where figures pop up, or Princess Leia, "Help me, Obi-Wan."

Mashey: Yes, yes, yes. <laughter>

Gossett: So anyway, it was cool. It worked. But again, business. They didn't have it. So I mean, they did get some funding. They got funding from Playground, which was Andy Rubin's [YES ph?] little incubator thing. That's another long story. Don't make things too cushy for startups. It's a bad idea. Okay?

Mashey: Yep, bad idea.

Gossett: You know, they were from Google, of course, so you know, Google was very plush, but Google was rolling in money. They didn't need to hit everything out of the park. They could screw a few things up, and they certainly did. But yeah, so what to say about that? Basically, they ran out of money. Long story short. A little too ambitious. They spent money on things that didn't make any sense. Like we were doing games, right, that was sort of the target. So they bought two gaming companies. Which makes no sense at all. I mean, you know, game designers are plentiful. And you could easily get them to do this, you know, on their own, you know, just speculatively, right? So that was crazy. And they didn't spend that much money, but it was a significant amount of money.

Mashey: Startups can't do that.

Gossett: You can't do that. You can't. It was a distraction, you know. And they were overly focused on game design, over getting the thing into production and cheap and, you know, all the stuff that you got to do.

Mashey: So how long were you there?

Gossett: I was there like two years.

Mashey: Two years, yeah, yeah.

Gossett: So I left before it went belly-up, and it was obviously not going to work. And you know, there was sort of nothing I could do for them. You know, it was just wasting their money really. So that was sad. That was a lost opportunity. And that's pretty much the end. So I'm doing some stuff-- I'm somewhere in a super position of retired and doing a startup. So--

Mashey: Oh, okay. <laughs> You're a glutton for punishment!

Gossett: So, yeah, yeah, exactly. So I've been noodling around with computer vision.

Mashey: Yeah, computer vision, yeah.

Gossett: And stuff like that.

Mashey: So you know, as a sort of summary in some sense, it sounds like if there's a theme that runs through this, you've been sort of all over the place in digital media.

Gossett: Yeah.

Mashey: Right?

Gossett: That's a good way to--

Mashey: In all-- you know, digital media from all different directions.

Gossett: Right, right. Yeah, so my sort of meta-observation is that you can be a hero by taking something utterly mundane in one field and applying it to another. You know, there's a lot of low-hanging fruit. You know, fields get stale. They start following the leader. They just do the same thing over and over. And if you come from like audio and start doing video, there are things you do in audio that they weren't doing in video.

Mashey: Video, yeah. Hm.

Gossett: Okay, you just do those, right? All of the things-- the other sort of unifying theme is that it's all kind of mathy. There's a lot of-- I mean, it's not like topology or anything like that, but it's all lots of linear algebra, and signal processing of various flavors. And that stuff, I mean, it's well-known at some level, but there are a lot of subtleties that aren't well-known for whatever reason. And I've-- over the course of doing all these different things, I've sort of picked those up. So people overspecialize. This, I think, is a mistake. And it's an increasing trend. You know, I see this at-- you know, I did a lot of hiring at Google. And people would be *incredibly* narrow! You know, they'd do one sub-sub-sub-specialty thing. And just be oblivious to even the things immediately around it.

Mashey: Other stuff, yeah.

Gossett: Which is like-- and people have lost track of fundamentals. You know, they maybe know how to do a particular thing, but they don't really understand why it works.

Mashey: And do you think that's-- is that an education thing?

Gossett: That's an education thing. So I mean, this is another thing. The rise of PhDs, no offense, okay?

Mashey: Yes.

Gossett: When I was in college, undergraduates made X. Well, grad-- let me rephrase that. Bachelors made X. Masters made X, plus a little. And that little was just about the delta that you'd get after you'd been in industry for the time that it takes to get a masters. So it's pretty close to a dead wash.

Mashey: Yeah, sure.

Gossett: PhDs made less.

Mashey: Made less?

Gossett: Less. Okay?

Mashey: Depends on where you went.

Gossett: In industry. Yeah, it depends on where you went. But in industry. And in high-tech industry. And the reason for this was they were viewed as being unworldly. Impractical. And there was some truth to this. Their tendency would be to overdesign things, and overthink things, and take too long to do things. And anyway, that was the case in late '70s, let's say, and it probably continued until maybe late '80s, mid-'80s, something like that. And then the wave of, well, like Sun and SGI, and all of the companies that grew from Stanford and other universities.

Mashey: Yeah, sure, yeah.

Gossett: You know, shifted that dynamic completely.

Mashey: But what shifted some places, Bell Labs.

Gossett: Yes, yes, fair enough, fair enough.

Mashey: And actually Bell Labs did this thing-- well, you had to have a masters to be a member of technical staff. And it was like about the same time. But the same salary increment sort of worked for the PhDs. Yeah, but that was Bell Labs.

Gossett: Okay, well, anyway-- yeah, yeah, but Bell Labs was like research. They weren't--

Mashey: Well, actually, no you know, real research was seven percent.

Gossett: Okay, okay, all right, all right, fair enough.

Mashey: But it was its own strange weird place, so.

Gossett: Speaking of Bell Labs, I should mention, I discussed Wei Yen [ph?] as being one of the best leaders, high tech leaders I'd worked with. The other one was Bill Coughran, who'd been the Director of Bell Labs for years, and he was hired by Google. And he was sort of this similar to Wei Yan in one way, he was the "VP of Everything." Totally different personalities. Really sweet guy. You know, he had this amazing ability to remain completely calm and happy and smiling in the middle of a hurricane, okay? <laughs> So people would be screaming at each other and just going totally ballistic, and he's like, "Calm, calm down. You know, we can work through this." And he always did. And like Wei Yan, he knew everything, okay? On any topic, you could discus some really detailed crazy-ass thing, and he would know all about it. And he would keep track of even larger numbers of different projects across Google. And you know, frankly, when he left Google, and I think that was bad. There was a noticeable change in the culture. So.

Mashey: Yeah, so that's been, like you say, an interesting career in terms of being a bunch of different turfs. Right?

Gossett: Which was totally intentional.

Mashey: And also, you know, obviously not luck, but as you well know, sometimes you work really hard and you make it work, but then something else chews it up.

Gossett: It doesn't matter. yeah, yeah.

Mashey: But you've been lucky to have a couple real winners!

Gossett: I've had a few successes. So I count three real successes.

Mashey: Okay.

Gossett: So the digital film printer.

Mashey: Yep, right.

Gossett: You know, Tron, and hundreds, if not thousands of other movies and television shows and so on. It was used from the late '70s until the early '90s. Which is remarkable for a piece of--

Mashey: That's a long time for something like that, yeah.

Gossett: So that was a success. "The Flexible Sampling-Rate Conversion Method," that paper, super influential, and like I say almost everybody on the planet has enjoyed the fruits of it. I mean, you can't ask for better than that. And then the Nintendo 64. And one of the happiest days of my life was walking into one of the Fry's around here, and they had a Nintendo 64 Demo Unit set up, and there were these kids, you know, and they were just beside themselves laughing and enjoying it immensely. And it just doesn't get any better than that.

Mashey: Yeah, yeah. Building stuff that lots of people really use is a lot of fun, yeah.

Gossett: Yeah, so anyway. You know.

Mashey: So anything we particular missed that you want to hit?

Gossett: Not that I can think of at the moment. I mean, I hope I haven't bored y'all to tears.

Mashey: Oh, no, no, no, you know, I mean, this is-- that's part of what these oral histories are for is you get different viewpoints sometimes on the same thing. Silicon, right? And it's really good to get that. So unless you got something else.

Gossett: I think that's it!

Mashey: Okay, great! Hey, thanks a lot!

Gossett: Thank you!

END OF THE INTERVIEW