Oral History of Feng-Hsiung Hsu

Interviewed by:
Dag Spicer

Recorded: February 14, 2005
Mountain View, California

Total Running Time: 58:59

CHM Reference number: X3112.2005

Q: Or should I, or how would you…?

Feng-hisung Hsu: Feng-Hsiung.

Q: Feng-Hsiung?

Feng-hisung Hsu: You can call me C.B.

Q: C.B., okay. Well, I'll say, we're here with Dr. Feng-Hsiung Hsu--and we'll just call him C.B. for now, which stands for "Crazy Bird," I think according to your book.

Feng-hisung Hsu: Crazy Bird, yes.

Q: Thanks very much for coming today. I really appreciate it.

Feng-hisung Hsu: You're welcome.

Q: It's February 14th, 2005--Valentine's Day. We're in Mountain View, California, at the Computer History Museum, and we're here to talk about the Deep Blue Project, which you led at IBM, with a team of very hardworking people, many of whom came from Carnegie Mellon, and worked with you. I want to start at a very high level, a meta-level, with you, Feng-hisung Hsu, and have you describe why computer chess is such an interesting problem for computer scientists, and why you decided to tackle that.

Feng-hisung Hsu: It depends on which area of computer science you are in. In my case, I'm interested in it mostly because, in the end, you find that speed matters [in solving this problem]. And when I started, I was a hardware designer, so I saw what we could do is just push hardware. But I think originally that wasn't the case.

Originally people were interested in computer science mainly because of their interest in human intelligence. And a lot of the people who are involved in this are AI[Artificial Intelligence] people, or cognitive scientists, and who are thinking that if [a] computer can play chess as good as the best human then you might have penetrated the essence of intelligence--at least that's the starting point. I think, as time goes on--as time went on--people found that the computer became stronger when it had greater speed, greater search speed. And that's when I got interested, because as time… you interviewed Ken Thompson earlier. Ken and his cohorts at Bell Labs built a machine called Belle. Belle was the first special-purpose machine that played as strongly as the National Master Level. I was a graduate student at the time and the first time I saw that, [I thought] well, it's pretty fast, but what happens if you get it 1,000 times faster? <Inaudible> communicating with each other, with the other students… maybe that would be very… we don't know whether that would be successful, but that would be very interesting. And somewhere along the line I just figured out--maybe someday--I realized maybe I can build such a machine. So I decided to go for it.

Q: Were there other people who thought that you could convert--I think it was Kasparov who said you'd managed to 'convert quantity into quality.' Am I on the right track in asking that question, where at a certain speed you get...?

Feng-hisung Hsu: That's sort of the--people, once they find that a computer [could] play better when they got them to run faster, people started conjecturing that speed may be part of the problem, part of the scenario. I think it was the Chess 4.5 program, [whose author] David Slate made the suggestion that once you give the computer some knowledge… when they searched deeper, they started to evolve new knowledge from that, because the fact is, when you search deeper, you start seeing things that's from <inaudible>. Like, for example, you don't even know that capturing a piece is good. Once you see more [?], start realizing a pin is good, and trapping a piece is good, because you actually manage to win pieces… you sort of can *derive* knowledge. And the more chess knowledge you put in… I found the deeper the search becomes, you start getting greater effects.

Q: Is it fair to call this emergent intelligence? That the strategies of the new machine… it's not thinking but it can reason, in a sense, or show signs of reasoning, once it reaches a critical….

Feng-hisung Hsu: Once you've gotten the computation. It's like doing the search of a space. Even if you don't know how to find things, if you search a large enough space, you can find things. And, in that case, you-…on the first glance, it looks like it's intelligence. It's not quite so clear, this [?] time, because you're doing things very stupidly, but you manage to do the things.

Q: Well that's one of the--I don't know if I'd call it criticism--but one of the comments on Deep Blue is that it's throwing raw horsepower at the problem. But it's more subtle than that, I think, you would argue.

Feng-hisung Hsu: Well, yes and no. Our starting point [was that] we don't care about intelligence. We really care about just computational speed--what happens, you just put in speed. And it's interesting. For people doing massive computation, that's how you can speed things up, [and it's] a very interesting problem by itself. But, at the end, when we actually tried to beat Kasparov, we realized something: that you really need to put the intelligence in [as well]. You need to put the chess knowledge in. Because then the compounding effect, the fact that search was going to enhance your knowledge, become even more pronounced.

Q: Right. Let's move on to that… [to] the Grandmasters and the various chess experts that you brought into the project before the various matches.

Feng-hisung Hsu: To teach the computer.

Q: To teach the computer. How did that work?

Feng-hisung Hsu: It actually played a very significant role in the second match. After the first match, we realized the computer just didn't have enough chess knowledge, and we needed to put chess knowledge in. And probably we just went ahead and just looked at all the books and just made changes. In other words, we looked at the design and saw what we can do and made changes.

But then we brought the Grandmaster back in. And the Grandmaster played games against the machine daily. And each day he played what he called take-back chess--that is, to try to trick the computer by allowing himself to take back moves, and then eventually find the bad move that can be replayed. And then from that process, we realized what kinds of weakness the computer has.

And in one specific instance … I happened to design the new chip at the time. And one day, I think Joe Hoane was the guy who came by to say "there's something you must see." And he brought me to see the Grandmaster and the Grandmaster showed me the scenario about what happens when you have an open file. Even a beginning chess book will tell you that if you have an open file, you want to put your rook on it. And you want to double-up your rook on it if you can. The problem if you do that is usually that [you] also need to trade up rooks. And then the part where you occupy the file doesn't really mean anything. And so, at that time, who was Grandmaster? Joel Benjamin. Joel told us that that's not actually- not necessarily the best way of doing it.

Because sometimes the ideal thing is you keep the option open on file--you have the option but the opponent doesn't have it. You pile your pieces on it, and then you open up when you have… when it's most opportune to you. And that was like, whoa, there's this idea. And chess people don't know usually talk about that. And it actually has… when you start trying to play with the computer and you realize that that is actually an important concept, from a computer point of view. So we just put in the hardware that said "if you have that option, you actually have a bonus--you just give a bonus." And that actually has very big, big effects.

Actually, in game two of the 2nd match, if you look at the match, Deep Blue just lined up the pieces on one file that wasn't open yet. It was just keeping it open, that option open there. And then Kasparov was forced to just defend, defend, defend. And in the end, that game could have been drawn. Kasparov knew how to play. But the whole process, because he was being forced to be defend the entire period, there was no hope because the computer was holding the card when you got to open it up. And that really psychologically drained him.

Q: Interesting. Now, you mention in your book that many of the evaluation functions came from Benjamin--some of the features. Can you give us an idea of how many variables are in this function--how many things you look at?

Feng-hisung Hsu: We listed, when we wrote it in the software. It's something like 3,000 terms. But these terms are actually very convoluted terms. They are actually basic parameters that are [asking] how much an opponent's piece would be worth if you went deeper, went further ahead? That's the parameter actually involved.

Actually, you just parameterize the whole thing as curves and mathematical concepts. A lot of the concepts came from Joel Benjamin. But some of them actually came from… we were trying to match the computer's moves against the Grandmaster's move. And we found that a certain thing that's normally, that is defined in [the] computer chess literature, doesn't really match that well. And we started looking deeper into the situation. We found out there are actually more shades of gray to a specific concept. The concept actually can be divided even further. So you start… actually, normally, there's one parameter. We end up with more than one parameter just to describe one specific concept.

Q: Fascinating. And so the chips that you designed were mainly hardware assists, to this evaluation function. Is that correct?

Feng-hisung Hsu: It actually is a full evaluation function. But it can be adjusted by software--it's a complete evaluation function, by itself. It can be controlled by software to adjust the weight[ing of its variables], because we [didn't initially] know what the right weight was.

Q: Right. And you mentioned too in your book that the ability for the software evaluation function to tune itself was critical to the project. Can you talk a bit about that and how the machine got better as a result of that?

Feng-hisung Hsu: Actually, it wasn't doing much tuning by itself. The tuning by itself… actually we used humans … in the process. But we actually, we tried some automatic tuning. But that only basically gave a hint about what was wrong. Like I mentioned earlier about looking on open files, there was actually one term that when we tried to match against Grandmaster play really performed very badly because it almost had completed frat[ricide]… because there's no signal. You would just use the original terms, that's what people have been using. And once you start looking into it, as we are, oh- even if you can control files, it also depends where they can penetrate. So you decide where your penetration capability into that full file is. So you start to get into a lot of more detail coming out.

Q: I see.

Feng-hisung Hsu: And then we just did a lot of things that you do--human adjustment. We just had some ideas… that a new concept was required. The tuning process told us a new concept was required. In one specific case, we actually used automatic tuning to adjust things and that term actually would have had a bad effect.

So we see it as one… right before the match, we used a different tuning process to suggest that all the King's safety should be jacked up. So we jacked up the King safety terms. And what we did… we didn't realize that we jacked it up so high because we originally designed it with certain wrenches [?] it's one-day's wrench, two-day's wrench. [?] You can jack it up so high [that] once it gets sufficiently bad, they all get saturated, at one fixed value. So it doesn't distinguish between very bad versus extremely bad. And so in Game One, because of this, [?] just give a point in front of his King because you open up the other side's King. It's unclear what was the right thing to do. It really looked bad but somehow it also annoyed Kasparov, because his King, he couldn't save. So he thought--he actually complained about it after the match, saying that it's so conveniently---at the time that's… I really got things <inaudible>.

Q: Yes, there's a psychological effect.

Feng-hisung Hsu: Yes, psychological.

Q: Now, one of the things that you mentioned was that the Deep Blue Project was not really man versus machine, but it was man as performer versus man as toolmaker. And I know that people involved with the project have—as have probably computer scientists in general--much more realistic expectations of what the machine is versus, say, the general public, which thinks it's some kind of transcendental giant brain. Can you talk a bit about that, the perception versus the reality, and what it meant to IBM maybe?

Feng-hisung Hsu: Yes. Well, IBM, during the whole match was pretty careful that it didn't say 'this is artificial intelligence'. We also emphasized this is not artificial intelligence, because that wasn't the intent. We would like to finish this job, to beat Kasparov, beat the world champion, whoever he is. But we know for sure we're not building an intelligent machine. We're just building something that does one computation really well. You can say it's intelligent, but it's a very limited domain. It only does something well in one specific domain. It doesn't know other things. Okay? I think actually in one… Kasparov actually once played a match against Anand, right at the World Trade Center… assume that we were playing Kasparov at the World Trade Center, and 9/11 hit. Kasparov would've run like hell. We would've run like hell. Deep Blue would just sit there … computing.

Q: That's a good analogy, right. Now there's an interesting quote from Kasparov--Here I just found it--which shows that even he was not immune to this sort of anthropomorphizing of Deep Blue. Here it is: Kasparov's quote, from your book: "They have succeeded in converting quantity into quality. In certain kinds of position, it sees so deeply that it plays like God." Do you have any comment on that?

Feng-hisung Hsu: I think Kasparov, at that point, actually was referring to the fact that Deep Blue has nowadays more solved the computers or serious ones anyway [?]. They contain so-called endgame databases, or table bases. That means that when you have very few pieces left, the computer can know precisely what's the shortest way to mate. And for a human playing against a database like that, or a table base like that, people sometimes have no idea [how to] find the shortest distance to mate. And there's some--Ken Thompson actually did some experiments, awhile back--when he first built the database. He gave the database, that actually supposed you can win in 50 moves to the stronger side. And he played against a Grandmaster. The Grandmaster couldn't finish off against Belle in 50 moves because the machine always knew what the longest defense was and humans don't know what results in the shortest wins. So as time progresses, it just gets worse and worse [for humans].

Q: Yes, and those endgame databases produce very puzzling results for human players, I think.

Feng-hisung Hsu: Yes.

Q: Because they see very deeply--and so you think that's what he meant, when you get near the endgame.

Feng-hisung Hsu: I think that's part of what he said. And also there are things that when it's purely technical, Deep Blue could have played much better than he did. After the first match, he came to our lab. He was thinking about one specific move, in one of the games. At that time, he was thinking about sacrificing his bishop for a pawn. But he worried about it because Deep Blue is technically stronger than he was. So he was worried, what would happen if he did that? So he played in our lab against the machine. And the machine came out with a lot of computation that he didn't realize over the board. So he was really happy that he didn't do that--didn't sacrifice the pieces. So I think that's part of what he said, as well.

Q: Now "anti-computer" chess. Can you tell us a bit about what that means when players play it?

Feng-hisung Hsu: In general, people talk about "anti-computer" chess, meaning you play in a specific style… [something] that may not be the optimal move--you don't play the optimum move from a pure chess point of view as purely from [the] win-draw point of view. You play moves deliberately with the idea [that] the opponent is going to play worse than you are. And, in this case, the opponent is the computer.

The computer is unknown <inaudible> an entire position well. Typically he'll say, closed position. That's what Kasparov played in game two of the 2nd match. In that case, unfortunately for Kasparov, people happened to know how to play that particular closed position. So he got beat bad. Then there are cases that the computer cannot play, as when you have attacked the computer has the chance for an attack. Except that attack requires you to understand that you play at a material deficit, but you are playing for an attack. You have attack potential. But you are suffering the material deficit in order to get an attack.

Humans can handle that because humans know the only thing important is not the material--it's to win. So it's the main opponent. The computer typically, because of the way the evaluation function got defined, would decide to get the material first, because the king safety return is not high enough. And that's what Kasparov did in game six of the 2nd match. Kasparov didn't really play a position that… where Deep Blue sacrificed a piece, for a pawn.

Now [the] consensus from [the] human community--chess player community—is that that is a winning position. But why? Because the black player cannot defend. But if you look at the literature… right before the match, there actually was computer chess literature describing somebody who was playing exactly the same position as black against the computer and managed to beat the computer handily.

And even after the match, some International Master tried the same position against all the commercial computers, and they all took a beating on them. Because the commercial computers at that time didn't realize… didn't [appreciate] the importance of king safety in that position. So the first thing they do is to grab a pawn back, and then give up the attack. Now you are down. You have a piece for two points. You are down a pawn. You have no attack. You lose. That was the end of that situation. In the case that we were playing in game six, Deep Blue realized the King safety return is so high. His king safety return was so high that he wished he'd keep on attacking. And we actually saw that exact position right before the match, like two months beforehand. And we had the Grandmaster look at the position. What I did is, the first thing, they let Deep Blue file up and see what Deep Blue's evaluation function was.

Deep Blue thought it was ahead. In fact, during the game, I think it was thinking it was a point ahead--even though it was down two points, it thought he was a point ahead. It has enough compensation from king safety for three points, and it would just keep on attacking. And if you look at the- Kasparov's expression during that match, that particular game, initially he was kind of surprised. Some people say he was in panic- that's what Seirawan said. But it is from a point perspective; it looked like he was just really surprised that Deep Blue actually took the pawn, with the piece. And after that, initially he wasn't feeling that… it seemed he was just taking it…okay, just see how it goes. But I think after a few more moves, he realized Deep Blue is not just taking pawns, and he realized he's in trouble--because this attack just kept on coming.

Q: Right. Interesting. So I want to just wrap up the discussion of Deep Blue a bit, and then we're going to go back to the antecedents to Deep Blue. This became a real cultural phenomenon and there were things on David Letterman and the Tonight Show and so on. And I didn't see any of those, but I can imagine they were sort of poking fun at the computer, as being a person. Kasparov had a very emotional sort of reaction at the end there. And I think at the end he was complaining a bit of whether it was fair or not. And most of the chess people that….

Feng-hisung Hsu: I have a tape with all those.

Q: The highlights.

Feng-hisung Hsu: Yes.

Q: Yes. Oh, that's great. I think we got one from IBM.

Feng-hisung Hsu: Yes--that's probably the same source.

Q: Thank you. And it was the knight sacrifice, I think, that he made--move 7--the h6, where he realized that he'd lost the game at that point, and that's why it was a very short game.

Feng-hisung Hsu: Well, I don't think he realized he had lost. He realized…he just make a gamble. And it turned out very bad for him, because at that point, even up to the knight's sack [?], all the other computers were losing. I think even the 1996 version of Deep Blue would lose to him. At that point, the king safety return is not high enough. Only the <inaudible>.

Q: So that's what you were talking about earlier.

Feng-hisung Hsu: Yes, that was what I was talking about.

Q: I see, okay. And so do you consider this over as far as…? I don't know if Kasparov does.

Feng-hisung Hsu: Well, it's a done deal. So, from the press point of view, it's not that interesting. So that has already reduced the possibility of a new match. And you look at what [has] happened in these last few years. Kasparov and Kramnik, even though they played against as a PC-class team, they can't beat them. The writing's just on the wall. Had Deep Blue continued, Deep Blue would have just been a small pocketsized machine now, because you can just redo the whole chip--these are all new single chips. And that kind of machine probably can keep Kasparov out and still beat him.

Q: Wow. Now I don't know if you've heard of Marshall McLuhan, the media theorist, but he has a great quote, which is often used, which is: "We shape our tools and they in turn shape us." I'm wondering what your thoughts are on how people now use and train on computer programs, and how that has affected the way people play chess.

Feng-hisung Hsu: I know for sure that that didn't make sense on it. [?] But I've been reading some of the recent articles on how the younger players play chess. And apparently they are more technical than people used to be. They're more willing to play moves that people previously would consider insane--because the computer taught them that you can play that [particular move]. And somehow they manage to survive. The other thing that shows up is that opening preparations become tremendous--that is, because you can use the computer to do also analysis. And sometimes, nowadays, you can have a Grandmaster play against some purely unknown, say a 16-year-old, who happened to study the guy's games, and happened to find an innovation against his opening--and they got beat. The Grandmaster got beat. And that happens.

Q: That is amazing, really amazing. That's one of the offshoots of this whole project that I thought was really interesting. And so can you give us a final thought, just to wrap up Deep Blue, on what it was like to win the Fredkin Prize, for example, which is sort of a prize... besides the IBM publicity… this was really your colleagues way of saying, "hey, way to go, you did that." Can you tell us a bit about that, and how it felt?

Feng-hisung Hsu: At the time, when we beat Kasparov, you could sense that the general public wasn't really that happy. In fact, at the match, when we--in one game, we beat… I think either that or draw--we just tied the match. We came on board, came to this auditorium and we got 'boos.' Okay?

But the Fredkin Prize was quite a different experience. We went there and all the guys that there were--AI people--were really interested in this thing. So we actually got a standing ovation. That was quite a change!

Q: Quite the opposite.

Feng-hisung Hsu: Yes.

Q: Yes. So the Deep Blue audience felt that humanity was kind of lost?

Feng-hisung Hsu: Yes, they didn't see the match as we saw it. We saw it as just people in a different role, playing against each other. And they saw it as 'man versus machine.' But it wasn't that clear-cut, because always, both sides always had men behind it. We had different timelines developed.

Q: Exactly. And I think that's the most profound thing that people don't realize is, yes, it's a computer, but it embeds so much human knowledge.

Feng-hisung Hsu: Yes.

Q: Okay. Well, let's take a break there, just for a couple of minutes.

<CREW talk>

Q: So, we talked about Deep Blue, and now I just want to chat about how you got there. So we won't start with your childhood or anything. But how about we start at CMU, and do you want to take us from when you started graduate work there, or earlier, if you think it's appropriate. And maybe take us up to the start of Deep Blue. I know it was a very interesting time, with lots of colorful personalities.

Feng-hisung Hsu: I was sort of interested in computer chess, even when I was in college. And I think during my sophomore year, I found the book, "Chess Skill in Man and Machine," as you have here.

Q: Oh yes. This one here. [Peter Frey's classic 1977 work].

Feng-hisung Hsu: That was a great book. And then there, there was this guy talking about if he had a chance to get involved with a team that could build a machine to be the world champion, he would be very interested. At that time I was saying, 'well it doesn't look anything will happen anytime soon.' So I didn't really think too much about it. Anyway, when I came to the [United] States I came here to study how to design chips--and that was sort of my interest at the time. And when I got to the [United States... we had to spend two years in military service. So, at the time, I just lost track of what was happening.

Q: This is in Taiwan?

Feng-hisung Hsu: In Taiwan. And then when I came to the [United] States, at Carnegie Mellon… was a place where actually there had been computer chess resources from the early days. So there were lots of computer chess books there. And one of them was “Mastering Computer Chess.” And then there was one article in there that talked about Belle. And [Belle designer Ken Thompson] came down to talk and wrote an article on Belle’s design. And then there’s another article, next to it, saying, oh, what happens if make Belle play faster? So here had Belle play 8-ply against 7-ply, 6-ply against 9-ply, etcetera to see… and he plotted a curve. [He determined that you got] about 200 points for each additional ply--which [translates into] a factor of 600 times in speed, if you do brute force searching.

So, that’s intriguing because Belle was playing at about 8-ply--brute force--any player at the National Master level, which was about, say, 800 points from the world champion. If you take each ply as 200 points, that’s 4-ply. A factor of 1,000- that’s 6? 4 times. So my office mate and I were discussing this. We saw interesting chess, in general… we just happened to have a group of people interested in chess. So we were just talking about it and we were saying, ‘well, it would be interesting if Belle--we can get Belle [to run] 1,000 times faster--just tie it all with a bunch of Belle and see how you can do it, write it--1,000 times.

We looked at the curve. It looked like it might beat the world champion. But somehow all of us—for some reason, we all thought that [alone] is not going to do it. We just said, okay, that’s going to be a very interesting machine because it’s going to be one hell of a technical wiz [i.e. accomplishment]. But I don’t think that it will beat the world champion. That’s how we were talking to each other. That was when we just got into the school. And I think by the 3rd year, I just finished all my other requirements, so I started looking for thesis topics. I found one. I was thinking about doing--it might make some sense, and it made some sense, at that time. I was thinking about building the printer controller chips that allow you to make nice Chinese printing, all the ancient characters, you get more complicated--and that’s kind of interesting. And there should be a market for that. So I was interested in money. And so that was what I was planning to do.

At that time, there was a team at Carnegie Mellon building a machine called Hitech. And they got their chip working, after several years of work. They were building all these machines. And Hitech [was hosted on] a Sun-4 [workstation], that is- they didn’t build a hardware evaluation function yet, at that time. So they were thinking about building a hardware evaluation function. And they were thinking about building it in the same way as they built the chess chip they were doing. The chess chip design was kind of strange, especially if you come from a VLSI design background. They couldn't fit an entire, say, Belle design onto a single chip. So they opted for a strange idea--that is, you just divide the chess board into 64 squares and you have one chip per square. That's a very expensive proposition because [it results in a] 64 times increase [in complexity] vs. what you can do using single chips. But they managed to get it working. And it worked with reasonable speed, that is, at that time, because it was searching roughly a comparable speed to Belle--respectable.

Q: 64 chips.

Feng-hisung Hsu: The real number is actually 80 chips, because they needed an additional one to attend to castling and things like that. But that's beside the point. It's basically a 64-chip solution. Anyhow, I was looking for my thesis topic. So I was also preparing it. But, on the surface, it looked like I was just not doing anything. So Ken Arnott [?] and Hans Berliner saw me as not doing anything. He approached me saying, well, we're thinking about doing this evaluation function. Could you help us? I looked at him and said, well, this is not my thesis topic, but if it's only going to cost me a few months, during the summer vacation or something, why not? It's going to be for fun.

So I start looking at it, and I actually did a paper design of it. It's reasonably complicated, but not so bad--if I can do it within a month or two, it's not that bad. The thing is, because you do it, you devise a thing—an evaluation function--divide it into 64 chips, you had to actually add the whole thing together. And you had to have an add-on tree or something to add… to combine everything together. You have two options. You can either do it externally, which allows you to do it faster, because you can use bipolar chips, which are faster. Or you can put the things into the chips you have. But then you suffer from the fact that your speed is going to suffer because, at that time, the chip wasn't… the transistors on those… on most chips, weren't that fast. So it's going to be pretty slow. So the conclusion I had was, oh, you had to do it externally-you had to do an external add-on and things like that. The Hitech team wasn't too happy about hearing that, because they can be slow. So, they weren't happy with the design.

Q: Is that, in a sense, revealing an architectural flaw?

Feng-hisung Hsu: I think that was really a flaw in that specific case because it has to do--actually do a combination operation, and that's divided up, and you had to do off-chip communication [which] was very expensive, at least at that time. It's better at this date, but even now you still pay a penalty, for that. So neither side was happy. I wasn't too happy with their response, and I'm not going to spend that much time, beyond what I think is reasonable for me. And they were not happy. So it's a no-win situation. So basically, it wasn't going to happen. And I was sort of frustrated.

I started re-reading the Belle paper, and I started looking at it. Wait a second. Is it really true you cannot fit the Belle chips—at least the move generator part--which is what they implemented--on a single chip… is it really true? The first reaction was, yes, it's not possible, if we use the Belle design as is. But you don't have to do that. Once I really thought about it, there are actual tricks you can play. [For example,] Belle uses a so-called disabled stack--that's using memory to store what previous moves have been searched, and you eliminate them from the searching process. Used memory is one way. But, actually, when you do it in chips, in VLSI, you have no other option. You can re-compute things. And in this case, re-computation, actually it's dirt cheap. So you do it--you can custom design things. And instead of having a large stack, which is…can cost you thousands of transistors, you can do it within like 20 transistors per square. And that's only eight [inaudible]. This is doable.

Okay, there are other problems- like the <inaudible> other way to solve the chips. So I started looking at it… wait a second, this is doable. And I started re-thinking what they tried to do with the evaluation function. They were doing just hard division. There's another way of doing it--do pipelining. You just can… you can get the thing smaller, by a factor of X, or whatever, depending on how many pipeline… how many stages [in the pipeline]--at least theoretically a factor of eight anyway. Once I was thinking about it that way--wait a second. The chip they had was operating on something like 5 microns per moves. So if I get a fast enough pipeline cycle, I can do it on a single chip--do what they originally they wanted to play, wanted to do.

Q: 5 microseconds, you mean? [I.e. not 'microns' as stated above]

Feng-hisung Hsu: Their chip was operating at about 5 microseconds per move. So that's the cycle time you can get, probably 100 nanoseconds or something. Then that's like 50 cycles… that's more than enough.

Q: What was the device technology—n-MOS, you said?

Feng-hisung Hsu: They were using 4 microns, at most.

Q: Right. And this was about 1980?

Feng-hisung Hsu: This was 1985… in the ancient [days]. So you start looking at it that way. It looked like you could do a move generator on a single chip. You probably could do an evaluation function--at least a simple evaluation function--on a single chip as well. So, what the hell, you want to do 64 chips? So I wrote a paper--just a note to Hans Berliner saying, 'well, I don't think 64 chips is the way to go. Here I have some idea how to do this and there is--oh, by the way, if you are interested in doing the single chip version, I might be interested. Because it still leaves me reasonable time.'

And Hans blew up, saying okay--initially; you told me you have to do this computation. Now you can do it in the single chips. You are just wasting my time. At that time I got pissed at him. So it was just a no-go. And Hans just gave up--I don't want to deal with it.

Q: Now, he was not your supervisor?

Feng-hisung Hsu: He wasn't my advisor. My advisor was H. T. Kung. And so H. T. told me this: "Well, you just pissed off somebody in the department. You better write what you are thinking about, write it down--because otherwise, you have a problem." So I just wrote out a technical report. And as I started writing the technical report, I started realizing, wait a second. I can do this. I can put the whole thing on the chips, given time. Do I want to do it? I spent a month on vacation in Taiwan and I started thinking, thinking, thinking--and finally made a decision- say, "if I don't do this I might regret it. Because that's one shot at making history. And you don't get the chance to do that very often." So I just went ahead.

Q: Do you remember where you were?

Feng-hisung Hsu: Oh, yeah. I just made up my mind to <inaudible>.

Q: On your vacation…?

Feng-hisung Hsu: So I told my… I told H.T. [Kung] that I have a thing about doing it. H.T. said, "Okay, you really want to do it, go ahead, but you've got to do it fast because there's already another team doing things, right? If you don't do it fast, there's a problem too." So at that time I made the decision--an arbitrary decision--that when Ken Thompson and Joe Condon built Belle, they did it in six months. Well, I can build a whole machine in six months. I can build a single chip in six months, right? I just went ahead and just… just did it. The whole thing just fit right into a 40-pin package, just barely [laughs]. It's just like a two-micron level or something on one side.

Q: This is a MOSIS project?

Feng-hisung Hsu: This is a MOSIS process. It was done in three micron CMOS.

Q: Okay.

Feng-hisung Hsu: So that was the chip that's used in Chip Test, Deep Thought and Deep Thought 2 for the next ten years. That was the core of the stronger mission for the next ten years or so.

Q: Deep Thought, Deep Thought 2.

Feng-hisung Hsu: Yes, and Chip Test.

Q: And CT, yes.

Feng-hisung Hsu: Which you're going to have a copy.

Q: Great. Okay, and so tell us a bit more about what made this chip interesting in architectural terms. I know you worked really hard to get this done in terms of layout and device geometry and stuff.

Feng-hisung Hsu: Well architecturally, it's almost identical to Belle. There are certain refinements to make it smaller. The reduction following refinement was a factor of two to three. That allowed you to actually fit it into the technology of the time, but it's basically the same chip as the Belle move generator, essentially based on a single chip. Belle searched about 160,000 nodes per second and this chip, theoretically, could search up two million nodes per second. So that's a huge jump in speed and also Belle has 4 boards, here you had a single chip. But then, because of Deep Blue--by the time we did Deep Blue--we realized we had to do something different. They're just slightly different chips.

Q: Tell us a bit about the tournaments and how the various machines that this chip went into got better and better and faster and how you gained confidence in this goal--

Feng-hisung Hsu: Okay. It was the old chip, the first thing we did-- First, we actually, got I think Thomas Anantharaman to actually write a program for the chips and then we got ambitious and decided to enter it into the ACM Tournament, even though it was only seven weeks away. We managed to get a board built, that's the chip that [?] and Chip Test, at the first event, we did okay. We broke even. We finished half and half. We lost half, won half, which is respectable, considering we only had seven weeks to prepare. But we weren't running at the full speed of the hardware. We were searching, maybe, about 50,000 nodes per second. Respectable, but not at the top, and the next year, we got all of the microcode on the hardware fixed up. So we're searching about, I think somewhere between 300,000 to 400,000 nodes. I don't know precisely what was the number.

So we were above, at least a factor of two to three times faster than the closest competitor and in that year, we just wiped them all out, we just won. Between '86 and '87, '86 is the first time we played—'87. We also found the idea of single extension, and actually this came from the '86 match [where we discovered] that some other people had problems with the fact that we were forcing a line they didn't realize that they are losing.

So that was a big change. After we won the ACM Tournament, H.T., my advisor, started thinking that these kids make some sense, okay, and he gave us like $5,000 to be able to make the next machines and that became known as Deep Thought. Deep Thought and Chip Test. Prior to that, Chip Test only played against humans up to that point and then the next year, in May-- actually, I think it was the long weekend in May?

Q: Memorial Day?

Feng-hisung Hsu: Memorial Day, yes. During the Memorial Day weekend, Chip Test and Deep Thought played the first time against human players in a tournament in Pittsburgh. Deep Thought finished second in the match, in the tournament with a performance at 2551. That was the highest ever by a computer, and Chip Test did okay. He also finished over 2500. So that was the first time a machine actually played that high.

Q: Now there's a great little chunk in your book where you describe Vivek Rao playing against Chip Test, where he's reading a magazine <laughing> I think. Let me get this straight.

Feng-hisung Hsu: No, it wasn't really that. Vivek was more serious than that.

Q: No, I'm thinking of someone else. Let me get this straight.

Feng-hisung Hsu: You're thinking about - what's his name?

Q: Ivanoff [ph?]

Feng-hisung Hsu: Ivanoff [ph?], yes. Igor Ivanoff [ph?], yes.

Q: Right. So he's reading a magazine during the early part of the game against Deep Thought.

Feng-hisung Hsu: In the U.S. Open, yes.

Q: So he had no respect. By move 19, he put down his magazine and he had seen something he didn't like. By move 29, he resigned. So it's kind of interesting. That's a very interesting little paragraph.

Feng-hisung Hsu: In some sense, Deep Thought was a sneak attack on the chess world. Prior to that, the only well known machine was Hitech and even Ivanoff had played against Hitech before and actually, I saw the video [of that] game that he played against Hitech. He pretty much just beat all attacks and won. It's just like routine to him. It's just purely simple with him. So when we play against Igor in the U.S. Open, we were actually kind of worried because we saw what he did to Hitech, but nobody really was expecting that Deep Thought was that strong.

Q: I mean there are two chess teams at CMU during this time, right?

Feng-hisung Hsu: Yes.

Q: As you mentioned, you didn't invite each other to your parties and so on. I understand.

Feng-hisung Hsu: Actually, we did invite each other to parties.

Q: Maybe now, or--?

Feng-hisung Hsu: No, even back in those days. When different teams had parties, it just is for the whole department.

Q: Oh, I see. Yes.

Feng-hisung Hsu: And Berliner was gracious enough. He actually invited us to his home for a party as well. Initially, we [were] actually pretty close. We actually were [?] even though we were competitors, we were still sort of working together.

Q: So did Hitech and Chip Test or that line ever play against each other?

Feng-hisung Hsu: There were lots of test games played and the results of that I cannot tell.

Q: You mean you can't--

Feng-hisung Hsu: Berliner didn't want us to say. So we honor that request... we're not going to do it, to say anything about it.

Q: I understand. Okay. Let's see now. So can you tell us a bit about the developments that took you to IBM? These machines... you said that they used the same basic chip architecturally. Even physically was it the same chip, or did the clock rate get faster?

Feng-hisung Hsu: You mean from Chip Test to Deep Thought to Deep Thought 2?

Q: Yes.

Feng-hisung Hsu: It's all the same chip, from, more or less, the same batch. I had 21 working chips besides the one that's in Chip Test and Deep Thought and that's how those 21 chips were used to build Deep Thought 2. So Deep Thought 2 started with 21 working processors and they eventually started dying.

Q: Oh, yes. With time. Well tell us about the name, first of all. Deep Thought, how did that...?

Feng-hisung Hsu: Well, Deep Thought… at the time I was a fan of [the television series] *The Hitchhiker's Guide to the Galaxy*, and in the *The Hitchhiker's Guide to the Galaxy* there was this alien race that built a machine called "Deep Thought" to compute the answer to life, the universe, and everything and the answer was 42… and that [computer] was Deep Thought. We figured that, okay, the machine can be world [chess] champion. It's certainly worth the name Deep Thought.

Q: That's a great story, and so as these machines-- Take us now--

Feng-hisung Hsu: But I couldn't use the name in IBM.

Q: You couldn't use Deep Thought. You know what I heard too?

Feng-hisung Hsu: There were other reasons.

Q: Yes, I heard there were some interesting substitutions in the name there. Well, you did a dissertation, of course, and so that was on this project then?

Feng-hisung Hsu: Yes. That was part of it … on this project, but partly on the idea of parallelizing alpha-beta search. Actually, the title was "An algorithmic study of Alpha Beta…" or something. I don't remember the title anymore.

Q: That seems to happen a lot - people forget their dissertations with time. Now a lot of people-- I think maybe it was you that mentioned that Slate and Atkins Chess program influenced you to some degree. Is that right?

Feng-hisung Hsu: I think you can say the Chess 4.X series was the prototype of all of the chess machines afterwards. Belle was basically based on that idea. Deep Blue and Deep Thought were more or less along the same line. There were some new refinements that we added… a lot of select extensions, which weren't there in the early stage. We also had some ideas that were new, but [we used] the basic design that they hit on, that is, it's probably easier to just search everything instead of trying to do artificial pruning.

Of course, that's when people come out with different ideas saying, "You do a nominal pruning," which is an algorithmic way of doing pruning, which is much easier. I think the main lesson from Slate and Atkins was it’s better to do things algorithmically than doing it heuristically as far as search paradigm goes.

Q: Okay. Can you explain that a little bit?

Feng-hisung Hsu: Search is such a complicated situation [?] things. Prior to Slate and Atkins with Chess 4.X, they actually had a machine called Chess 3.5 or something and that one actually was doing the so-called selective search. They do selective pruning. They had some rules about when you’re going to cull out the moves. The problem is that it’s very hard to come up with new rules as good, especially doing it *ad hoc* okay? And Slate and Atkins, they did one thing. They just cut out all of those rules. They just did it and searched everything. Okay.

Later on, people found out, you also do algorithmic pruning. It can produce fairly good results instead of doing it in this *ad hoc* way of doing things because it’s a no end situation. It just keeps on going, going, going and you do algorithmic [searches] and you can actually reason about them and see what you are missing and you can try to solve the problem as you go and, at least, that’s what the dominant way of chess programs is these days, even the commercial ones, except they do a lot of pruning. We didn’t do it in Deep Blue, though.

Q: Now tell us a bit about the transition from CMU to IBM and how that happened for you.

Feng-hisung Hsu: It was a culture shock. We were in the university and people tend to work longer hours and they’re just dedicated, they just keep on doing and the IBM people have families. They work 9 to 5 and initially that took some getting used to. But now, I have a family. I understand it perfectly <laughing>.

Q: Yes. Right. Right. I’m interested in how IBM supported the project initially because it seemed like they supported it more as success seemed imminent and I know you had to fight for your supercomputer in a way. It was on the shop floor. It was going to a customer or something and you kind of had to compete for that in a way to get resources.

Feng-hisung Hsu: Well, I think for the second match we didn’t have much problem. For the first match, it required a little bit more work but we got it so there’s no complaint on our part anyway.

Q: One thing that I hear a lot is--and I think Murray Campbell wrote on an article on the IBM Web site about this—is that this had application to things like oil exploration among other things. Can you explain that because I'm not sure I understand how this relates to those other things.

Feng-hisung Hsu: Definitely not the chess chip we see here because that's specifically for playing chess, but the basic design idea [is right]; that is, you can combine general purpose computational power [from a] special purpose supercomputer with special purpose processing that just deals with the core of the computation and you can do that very fast. The general purpose computer—the RS/6000—gives you the flexibility and the other one gives you the real computational power and sometimes, that's a fairly good fit. For chess, that happened to be the case.

Q: Okay. So you take the kind of critical loops, if you like, and do them with hardware…

Feng-hisung Hsu: That is the general idea and this idea, obviously, applies to a lot of other things, but they are all somewhat specialized applications. They are not general applications.

Q: Right. Let's see. I'm going to go back to just the end here and ask you some general questions, a few cosmic questions.

Feng-hisung Hsu: Cosmic.

Q: Yes, well large, you know. Okay. So in one point in the book you say, you know, the project's over and you said, "Oh, I've been working on this for 12 years," and you tell yourself, "Get a life, you're free." So what was that like when you reached Everest? How did you decide what to do next and how did you feel?

Feng-hisung Hsu: Well, in my case, I went back to my roots. I began my college as an EE major--electrical engineering. When I went to the States, it was purely by accident that I became a computer scientist. So deep down inside I'm really an engineer. I really like to build things.

Q: Is it the hardware side?

Feng-hisung Hsu: On the hardware side. I'm more towards the hardware side and you start looking at things that, actually-- especially during the whole process of designing Deep Blue. It was really a hard struggle. You design things that are really complicated and I figured that well, there are things less complicated but still equally interesting and that don't take as long, and that's where I decided to spend my time.

Q: It seemed like the tools were very rudimentary at times when you were working…

Feng-hisung Hsu: Yes. These days, well first, you wouldn't necessarily actually build a chip directly. Most likely, you might want to build Deep Blue again today, or do it for some other game like Shogi or Go anyway, but just an example. I'd probably start with something like a PGA [Programmable Gate Array] because you can actually implement the whole hardware right there including the software and you can run it, and that allows you to get faster feedback. You have a much faster process loop.

In the case when we designed Deep Blue, the whole thing had to be… you saw I designed the whole thing and simulated it in software. That's like 1,000 times or maybe a million times slower than you can run it. And a PGA would be slower than, say, you tried to do a full-custom chip But it's like, at the most, a factor of ten to 100, which is much faster. You can get much faster feedback and that is how people would probably do it today.

Q: I remember you mentioning that if you had to go through some of these game trees as a simulation, it would be running for months when you were developing your chip. So you can't do that.

Feng-hisung Hsu: Yes. That was extremely painful.

Q: All right. Tell us what you're doing now. Are you working at Microsoft Research Asia?

Feng-hisung Hsu: Yes. I am working mostly in the mobile area. We're doing some hardware work, and we're doing some software work. The details probably shouldn't be mentioned.

Q: Yes, that's all you can say for now. Okay. Just two more questions. What do you think is the hot area of computing, or the next sort of Big Blue-ish kind of project, if there is one?

Feng-hisung Hsu: Well, I'm sort of getting away from those so-called big projects. I'm more towards things that I think are relevant. I think mobile computing is going to be with us. Things get smaller because of Moore's Law. Then we've got to go up. I think the FCC will free up some additional bandwidth further up and I think there's an auctioned band called 60 gigahertz and there's 70 gigahertz available there [as well].

So the bandwidth is going to go up. There are a lot of things when you start, as things get smaller, when they get mobile, there's new things coming up and I think mobile represents another trend. That's the convenience from the user perspective. You look at the PC market. In some sense, we are reaching the saturation point. People are already satisfied with the performance, but people want more convenience. Mobility is one way of getting – of making computing more ubiquitous. It's not saying the only way, but mobility is certainly a big thing.

Q: Well that brings up an interesting point, which is how does it feel when you have, say… I don't know if it's true but say a Palm Pilot could be running Deep Junior and, I don't know if it's as powerful as the Deep Blue, but it's not unreasonable to expect there are machines small enough to fit on your desk that are as powerful as Deep Blue. How does that affect you?

Feng-hisung Hsu: Well, in terms of computation power, Deep Blue, actually, is not easy to replicate because the chess knowledge we have in there actually requires really heavy computations. It's not clear you need that kind of chess knowledge. I just found a recent example from the matches we've seen where it looked like humans had problems even though you don't have all of the chess knowledge in there.

The fact is the thing that searches deep enough [creates the illusion of chess knowledge]... and that seemed to cause a profound effect… humans had problems dealing with it. So I don't think the speed of Deep Blue is easily replicable now. To create a machine that can beat Kasparov may not require things as complicated as Deep Blue.

Q: That's a better question. Yes.

Feng-hisung Hsu: That, we don't know. You may want to build Deep Blue again. Deep Blue is a single chip today. You can put the whole Deep Blue on a single chip today.

Q: Including the supercomputer part?

Feng-hisung Hsu: You don't need a supercomputer in that case because everything fits on the chips though you still need the memories and so on. The core computations can be all [done] on the single chip.

Q: So it would be your hardware assist chips plus a bunch of functional units like execution units?

Feng-hisung Hsu: All of the execution units could be all in there on that one chip. In that case, yes, you could have a... well, actually, on a Palm Pilot-size pocket PC, we could say envision a Compact Flash plug-in that could beat the pants off of Kasparov.

Q: That's pretty amazing. Okay. My last question for you is do you have any predictions for AI, or computing?

Feng-hisung Hsu: Not really.

Q: Really dangerous predictions.

Feng-hisung Hsu: It's a dangerous thing to do. I know.

Q: Okay. Anything you want to leave us with? Any thoughts?

Feng-hisung Hsu: Well, I think there will be lots of other youngsters coming into things. I think one thing probably worth saying to them is [to] do things that make you happy. That's usually a good thing because that means you're serious about it, you can be passionate about it and you can do great things.

Q: That's wonderful. Thanks very much.

Feng-hisung Hsu: Okay.

Q: Great. Great interview.