Oral History of Boris Babayan

Interviewed by:
Alex Bochannek

Recorded: May 16, 2012
Moscow, Russia

CHM Reference number: X6507.2012

Boris Babayan, May 16, 2012

Alex Bochannek: I'm Alex Bochannek; Curator and Senior Manager at the Computer History Museum in Mountain View, California. Today is Wednesday, May 16th and we are at Intel in Moscow, Russia to conduct the oral history with Boris Babayan. Also present in the room are Lubov Gladkikh his assistant and Yuri Merling [ph?] the videographer. Thank you both for agreeing to do this oral history for the archive at the Computer History Museum today. Let's start about talking about your childhood and would you give us your full name, where you were born, and tell us a little bit about your parents, if there are any siblings and what your childhood was like.

Boris Babayan: I was born in 1933 in Baku, Azerbaijan, now it's an independent state. My father was a technician, he was an electrical engineer, and my mother worked in the kindergarten. I finished 10 year secondary school in Baku and then moved in Moscow, where I entered the Moscow Institute of Physics and Technology. It seems to me that in Russia I definitely was the first student in computer science.

Bochannek: Now, what made you want to go to Moscow to that institute? Were you interested in technical things?

Babayan: Because I was interested in the technical education and some people told me that the Moscow Institute of Physics and Technology is a very good institute, like MIT in the United States; it was established just after World War II, specially for education in the high tech. I entered the institute in 1951; it was definitely the first group in computer science in Russia. At that time, the words computer science were not used. I entered in the specialty which was called "machine mathematics." <chuckles> So, probably, I was the first student, worldwide, I don't know. I wonder, if there was a student in computer science in the United States in 1951, or not, I don't know. Definitely, among the known Russian students I was the first.

Bochannek: Were you able to choose to go to this university? Did you have to have special accomplishments in…?

Babayan: No, we had just [entrance] exams and those who passed examination [were admitted to the institute]. [But the contest was stiff], maybe five or seven persons for [a seat].

Bochannek: So you were a very good student to be able to go to the school?

Babayan: At least, I have passed examination.

Bochannek: Did you have teachers or family members who were really influential, who were really interest…?

Babayan: No, no, no, my parents were in Baku. I arrived there alone. No, no, there wasn't any kind of help—just before entering the institute, maybe for two or three years, I have been preparing for the exams. I had the published questions in mathematics, which had been asked at the exams for the last two, three years, not even at this very institute, [but some other universities,] so I passed examination quite well and was admitted to the institute. It's very high-level institute, very good education.

Bochannek: So you arrived in Moscow?

Babayan: Yes, I arrived in Moscow and since that time I have been living in Moscow.

Bochannek: What was it like to go to Moscow? Moscow was the capital of the Soviet Union; it was the big city…

Babayan: Just because there was a very good university. Actually, I entered into the Moscow State University; at that time, this institute was a faculty of the Moscow State University. But at the first year of my studies it separated from the university. It was not located in Moscow, but in the suburb, in Dolgoprudny. It is a well-known institute. It was established by very well-known people:; Piotr Kapitsa, Lev Landau, very high level scientists, and the founder of Russian computing, Sergey Alexeyevich Lebedev.

Bochannek: What was your idea of…?

Babayan: Actually, there were two specialties. The first one was electronics and the second - machine mathematics, but because machine mathematics was very new, a very small number of people picked it; so I was among the minority. All who selected this specialty, even as the second one, became students. So, in some sense, by chance.

Bochannek: Why did you pick it?

Babayan: Because I liked mathematics, and this was the machine mathematics..

Bochannek: I like machines too.

Babayan: Yes. I had no idea of machine mathematics. Computer science did not exist at that time, it was in 1951.

Bochannek: So tell us a little bit about your studies. What did you do? What did you study? There were no computers to study with.

Babayan: Yes, actually at that time, the first computer was built in the Soviet Union. In 1949, in Kiev, Sergey Alexeyevich Lebedev built the very first MESM, Small Electronic Computer, But by today's estimate, it was just a calculator; it was a big wall of switches and lines, you know, the program was on the switches. In 1949, it already worked., And in 1951 the first Soviet computer BESM was in the debugging stage The Moscow Institute of Physics and Technology had a different way of education. The basics of mathematics and physics were lectured in the institute, , but the main education was given in the basic research organizations. At that time, Sergey Alexeyevich Lebedev had already established [a research institute] IPMCE (the Institute of Precision Mechanics and Computer Equipment) in Moscow, [with a basic chair at the Moscow Institute of Physics and Technology]. The first [Soviet] computer [BESM was in IPMCE under debugging]. It was not operational yet, but it existed and after finishing the first year, at summertime, we [the students of the basic chair] spent one month in this [research] institute continuing our education.

Bochannek: This would have been in 1952?

Babayan: Oh yes, in 1952. It was the summer of 1952. And very soon this computer started to work, but it was a vacuum tube computer, very big. . . it occupied three rooms, nevertheless, we had a chance to calculate on this computer and to communicate with Sergey Alexeyevich Lebedev; he was a very smart guy.

Bochannek: Who taught the courses?

Babayan: Sorry?

Bochannek: Who was the teacher?

Babayan: In basic mathematics and physics, quite famous high-level physicists, mathematicians. At this basic chair, Sergey Alexeyevich Lebedev lectured the main course. The people who designed this computer and were debugging it, were also our teachers. So we worked with the real designers, and little by little, we started to participate in this design, because at the second year of our education we spent one day a week at this basic chair. At the third year we spent two days and so on. At the end of the sixth year of education time, we spent in this institute a whole week

Bochannek: So you were only 19-years old when you first started working with the dozen [ph?] in the summer of 1952?

Babayan: Yes. Even 18 years old.

Bochannek: So you were 18-years old, a student and you learned how to debug and program the first…

Babayan: Yes, actually, in 1954 I delivered the first real and very important research result —it was a scientific result; it was my first scientific achievement there. You know that at that time computer had not enough hardware, it had only one execution unit. It was a completely sequential computer that issued one instruction at a time. The only possibility to improve computer architecture was to speed up arithmetic units. And my first student's work was [devoted to] developing the so-called carry-save arithmetic. I did it in 1954 at the third course, and at the fourth course in 1955. I made a presentation at the institute research conference. The first western publication of this technology appeared in 1956; I did it one year before. This technology now is basic, you know, it is used in all computers. It was my student's work. I suggested to do multiplication, division and square root extraction without carry. Usually, multiplication is a series of additions; and carries are time consuming, most time consuming. I suggested the way how to do carry-save multiplication, i.e. if you have three numbers, you are adding by each bit separately thus saving carry, not propagating it. It's the only possibility to improve the speed of computer. So, when I was 21, this was my first and a very important result research result… It is quite usual for this institute. All students worked with very high-level researchers and participated in real very serious research.

Bochannek: So when did you graduate and what was your degree?

Babayan: I graduated in 1957 and stayed at this institute, which was a basic institute for my education. I started to work there as a student and just continued to work with Sergey Alexeyevich Lebedev. There is a similar situation today. Now, I am professor of this university and head of the chair. The students [of the chair] are Intel's [interns]. Intel has a basic chair at this institute. So, we are [teaching] people there. For the projects, which we are running [at Intel], [interns] are one of the main research forces

Bochannek: So, at a later point, you completed your PhD and also Doctorate of Science?

Babayan: Yes.

Bochannek: What is the subject of your work…?

Babayan: For my PhD, the subject was design automation. At the IPMCE I built a system which helped one computer to design another computer. The subject of my doctoral [thesis] was architecture of Elbrus computers.

Bochannek: I believe those are also the subjects for which you awarded the State Prize and the Lenin Prize?

Babayan: Yes. [I was awarded] the State Prize and Lenin Prize for [the development and implementation of complex equipment] for design automation [and for multiprocessor computer system Elbrus 2].

Bochannek: So, I understood the Lenin Prize is a very high honor.

Babayan: Yes, it was the highest prize in the Soviet Union.

Bochannek: How did you feel about being awarded the Lenin Prize; did it make you proud?

Babayan: <laughs> Definitely.

Bochannek: Definitely.

Babayan: Yes.

Bochannek: Okay. So you say that you stayed on at the Institute of Precision Mechanics in 1956, ’57 when you…

Babayan: Yes, I became an employee of this institute.

Bochannek: So what was your first assignment as an employee of the institute? What was the earliest project you worked on?

Babayan: The institute designed computers, and the first computer which I took part in was M40, but I was a little bit late. I did not build the computer itself. I designed input/output devices. This computer was connected to the radar station, because it was delivered for anti-missile system. This computer used my multiplication method; it was fast multiplication and division, which I designed. So in the main computer, though I was a student, they used my results. By the way, a little bit later, probably in ’70, an American design researcher, it seems to me Robertson, yes, [designed the second basic approach]. Today arithmetic units have two basic technologies. Since that time, there have been only two basic technologies; one which I have designed and the second one, which Robertson has designed. And we met each other, exchanged opinions and so on, he was in Moscow. 1974 was the time of détente and he was able to arrive in Moscow.

Bochannek: So, at Lebedev’s institute cutting edge work, it was very advanced work.

Babayan: Yes.

Bochannek: What was the work environment like? How did people learn? How did they try out new ideas? And how much control was there by the state on assigned [ph?] projects?

Babayan: We had very good conditions; the work environment was very, very healthy. Definitely all orders were made for the military. For example, we built M40 for the first successful test of the anti-missile system; it was in 1959 in Kazakhstan. It was not so simple to have a successful anti-missile test on a vacuum-tube computer. And I can share with you some very, very funny situations. The computer reliability was very low. , Though I designed only input/output devices, a little bit later I was leading in this team and I was at the test site. The situation was as follows. The test site was near Balkhash Lake. But the missile [was launched from a place located on the Volga River]. There was a 12-minutes flight. Usually, we were preparing for test during the whole day: five hour readiness, one hour readiness five minute readiness, shoot. Everybody felt—we are in a good state. The missile started [the flight], and one minute later one vacuum tube blew up. <laughs> Terrible. I am joking, that today to find a bug is a very difficult task; in that case it was very simple. So, we were prepared, we had heated vacuum tubes. We [quickly replaced the broken one] and did all tests. The system caught and shot down the missile.,[As soon as the results were printed out, the second tube blew up]. If we had not printed out the results, all spending would have been in vain. This was a very difficult situation. So this was my first computer, in which [design] I participated.

Bochannek: You mentioned the reliability problems and with vacuum tube computers that was always the challenge but I wonder how much of this was due to component technology that were produced of low quality, low yields.

Babayan: Yes, but with vacuum tubes it is impossible to ensure reliability. After M40, my second computer was 5E92b; it was solid state computer. We were suffering very much from unreliability of M40. And for the next five years we were doing everything to increase the functional reliability, because it was the second generation technology, it was solid state already, but nobody knew if it would be reliable, so we were preparing. We built a failsafe approach. We built a computer, which had some redundant hardware, and each single intermediate or solid fault was detected with nearly 100 percent reliability. Then there was recovery and it worked. Functional reliability was extremely high. And, fortunately, the radio industry built very reliable solid state devices at that time; it was like 10 gates per chip. The computer was used in real anti-missile system around Moscow and this combination provided very high reliability. We built 10-processor computer, a big box computer definitely; having this failsafe technology it was highly reliable. We have agreed with our customers, that if eight of ten processors are working, then it’s okay. So for many years not a single one failed, we never had less than eight working processors; so the reliability was very, very high.

Bochannek: What was the relationship between the system designers and the organizations that supplied the components? Could you specify what type of transistors, what type of ICs you wanted manufactured?

Babayan: There was the Ministry of Radio Industry. The IPMCE, actually, was an institute of [the Russian] Academy of Science, but the plant that built computers belonged to the Ministry of Radio Industry. Components were built in Zelenograd under a different ministry called the Ministry of Electronic Industry. So we were working together. When Zelenograd plant designed components for chips, we helped a lot to do some specification. [We did not take part in] the process itself, but developed some electronic characteristics, logic design, we worked very, very closely together.

Bochannek: Do you recall which factor it was that built these machines?

Babayan: The factory that built machines was located in Zagorsk City, now it is called Sergiev Posad, which is close to Moscow, but the chips, solid state devices were built in Zelenograd City.

Bochannek: One of the stories about Zelenograd is about the two Americans who had defected and helped found Zelenograd and I was wondering if you had heard about that story back in the day.

Babayan: Two guys, I remember, but they did not play a big role.

Bochannek: <inaudible>.

Babayan: Sorry?

Bochannek: [Joel] Barr and [Phillip] Staros where they…

Babayan: Yes, yes, yes, yes. I don't think they played a big role—I know, I met with them, but they did not play a very big role.

Bochannek: One of the questions that sometimes comes up when you look at the American side of the story, the funding of the military for a lot of computer science, is people ask about these researchers in the 1960s who worked for the military but there is a lot of anti-military sentiment in society and I was wondering if that is something that ever came up. Did anybody ever ask the question, you know, why are we working for the military? Was there anybody who would openly ask that question in the institute?

Babayan: Oh no, no, there was no such a situation. Most money came from the military orders, and so there was no big complaint. The people were working on the technical side, and though the main order was from the military, it was a computer, civil organizations could also use it, there was a big shortage of computers. There was no market. Distribution of computers was done in the Kremlin, and <chuckles> mainly the military received them, but people usually were happy to work on the technical side.

Bochannek: So it didn't matter who the customer was; it was military or non-military?

Babayan: No, there was no such a situation. It was Soviet regime. <laughs>

Bochannek: So what was the next project you worked on after the M40 anti-missile?

Babayan: Actually, the first project was M40. The second one was a real anti-missile system. M40 was just a test. [5E92b ran the first Russian anti-missile system protecting Moscow]. The next project, Elbrus-1, was very important... It also ran anti-missile system, but many other organizations used this computer as well. For example, a nuclear establishment in Serov used our computers, the space control system used our computers, and, definitely, this computer was used in some civil organizations, but only in few.

Bochannek: So another machine that's very well-known is the BESM-6 which came before Elbrus BESM-6 machines.

Babayan: Say again.

Bochannek: The BESM-6 is also a very well-known machine and before Elbrus was earlier.

Babayan: In parallel. BESM-6 was not ordered by the military; it was some kind of a civil order, but Elbrus was ordered by the military Elbrus 1 had many innovations... I can give you the list... For example, as I told, the arithmetic unit was made by me in 1954 before western publications appeared. Elbrus-1 was the first-in-industry superscalar computer. You know superscalar program is sequential, but it is converted into parallel form at runtime. We built superscalar in 1978. The first western fabrication by Intel was [reported] in1995. We built superscalar 17 years before Intel, and it was quite fast. At that time there was a policy in the Soviet Union to copy American computers, mainly IBM and DEC computers. We built the first generation of superscalar Elbrus-1 and then the second generation, Elbrus-2. The first generation was delivered in 1978, the second generation in 1985. And in 1985, our competitor [delivered ES 1066, a computer from the Riad series (in English "Riad" means a series)]; it was a clock precise copy of IBM 3033, and it had American software. We had a competition between Elbrus 2 and ES, and [Elbrus outperformed ES by] two times. <laughs> Actually, it was a competition not with ES, but directly with IBM, because it was a clock precise copy with all IBM software. In our case, Elbrus used our software and our technology, it was solid state, our guys designed it with Zelenograd team, and Riad used our components. Elbrus-2 used gate array technology - 3000 gates per chip <laughs>.

Bochannek: So, your role in the Elbrus project was you were a system architect. Did you define the instruction set?

Babayan: My role was architecture. I was the head of all architecture and system software, but not the technology of chips; it was not my specialty, but all logic design, all debugging, everything was under my control.

Bochannek: So let's talk a little bit about software because one of the stated reasons for copying IBM machines was that the software existed.

Babayan: Actually, it was a mistake, because, first of all, it was not fair, it was stealing, very bad. Usually they stole operating systems and compiler, which did not match each other; it was a mess.

Bochannek: What was the situation like for Elbrus? How did you go about developing…?

Babayan: We developed everything ourselves including operating system. Besides it was superscalar, we built it in ’78, and we built type safety architecture. It was a real high level programming support. You know that when we started Elbrus-1, we were wondering if it was possible to support high level programming, because it was very interesting and, very important. We tried to understand what high level [language] programming meant. When we started to build Elbrus-1 in ’72, (delivered in’78) we decided to support high level language programming. First of all, it was necessary to find out what high level [language] programming meant? It definitely, did not mean to support the existing languages, like Algol or Fortran, because Algol and Fortran were built to run successfully on the existing architecture. When we build a new architecture oriented at languages, which are oriented at old computers, indirectly we are building a new architecture oriented at old computers. It is wrong. So, we rejected this approach. And then we started to think about syntax. But it was not also true, because at that time, there was such a language PL360; it was a syntax-like high level language, but it had [only simple] data types, registers. So everybody understood, and there were published articles at that time, that it was just a version of Assembler. We thought very carefully and decided that real high level languages strongly support types, thus programmers should not work with bits or instructions, but with objects and with procedures handling these objects. And we decided to support these notions. Mainly, it was necessary to support [not C like] pointer, [but real references and descriptors], and in Elbrus-1, we built architecture with type safety. The data have a few additional bits—tags, which support type safety. In Elbrus-1, we supported type safety. At that time, the languages designer number one, Niklaus Wirth, also came to the same decision that high-level languages should support type safety, but there was another question. How should we handle these types? Should these languages be dynamic by types or static by types? When we declare some variable, whether this variable should be assigned a specific type or it is dynamic? Definitely, for universality it should be dynamic, because for me, for everybody, universality means that we should be able to build an operating system using this language. Operating system is working with user codes. User code is already compiled, so it cannot be static by types. Operating system should understand that this is a pointer, this is data, after translation. We understood well that a real high level language is dynamic by types Niklaus Wirth also understood it and built a dynamic type safety language named Euler; it worked very well. At that time there were many people who liked dynamic languages like Smalltalk, LISP, dynamic by types. Euler was excellent, but provided a very low performance, because of many dynamic checks. Niklaus Wirth declared that the high level languages for current computer (computer didn’t support type safety) support type safety in language, but with static types, because static types are somehow possible to support in languages [running on that day’s computers]. What happened next in the world, except for Elbrus—there have been two possibilities till now. There are languages, like Java, with type safety, but not universal static types. It is impossible to write operating system on Java; a very low performance. And another extreme, C language; it has no type safety, because there is C pointer. So, in today’s practice, nobody knows what real universal high level language means, but we, in our team, built support hardware. We built a special language with dynamic types, type safety, EL76 in ’76, we built operating system, we built the whole system. All our customers were very happy, because it was very easy to debug programs. It was in 1978. At the beginning of ’80, Intel also tried this approach; iAPX 432 architecture was a very good approach, but implementation was extremely bad. . At that time, I told [already that in the Soviet Union] there was a policy to copy American computers, and the government and the Party sent letters to all researchers saying that it was necessary to copy, Intel, a computer named Intellect. Everybody knew that I was an advocate of type safety, high level languages. There was a commission, and I wrote a big paper against <laughs> this approach. There I told that it was a mistake, it would fail, and a little bit later it failed; so this was my forecast. At that time we already had the successful Elbrus 1. It was the beginning of ’80-’82, but we built it in ’78. So till now there is a very sad situation; nobody in the world understands what “real high level programming” means, except for our team.

Bochannek: So you talked about the programming language and the computer architecture and you also mentioned the operating system.

Babayan: Yes, all together. Look, because of absence of support from hardware, languages were corrupted, there is no real high level language in wide use; Java is not universal. C is not a high level programming language at all. Elbrus-1 was superscalar; Elbrus-2, till now, is the only case of real high level programming. Maybe, in the near future there will be another one. My estimation of iAPX 432 coincides with the analysis of Bob Colwell. Do you know Bob Colwell? Well, he built a superscalar, He was a student at that time, or a PhD student, I don't know. He wrote a paper. I have this paper. He did the analysis of iAPX 432, and wrote that the idea was brilliant, but the implementation was terrible.

Bochannek: So what was the operating system for the Elbrus machines, and was it related to anything at all?

Babayan: At that time, compatibility was not a really big problem, and it was our own operating system. By the way, type safety helps to substantially decrease complexity of designing operating system. Our operating system was multiprocessor, multi-program. It was built by 20 or 25 people.

Bochannek: What was it called? What was the name?

Babayan: Sorry?

Bochannek: What was the name of the operating system?

Babayan: Elbrus Operating System. But it was before we started to work with western companies. . Actually, I will give a piece of history. At that time, we had Elbrus 1, Elbrus 2—(Elbrus 2 was the second generation of superscalar). Elbrus 1 and 2 had two features—type safety, with high level languages support, and failsafe capability. They were reliable. But in 1985 we understood well that superscalar is a good architecture, but it has limitations. Superscalar is definitely outstanding, because instead of sequential execution, it provides parallelized execution. But it takes sequential program presentation and converts it in parallel form at runtime. And then the retirement process, retiring occurs in the original sequence, conversion is performed two times, compiler converts the highly parallel algorithm into sequential form, then superscalar does the backward transformation...terrible situation. This runtime conversion puts a limitation. It is impossible to build superscalar faster than four or six issue rate. Our superscalar was not four, or six, it was only two issue rate (Elbrus 1), but it was a real superscalar, out of order, with register renaming and everything. And we understood its limitation. We were looking for something different, and we tried a wide instruction approach. Wide instruction means explicitly parallel. It is explicitly parallel, but with static scheduling, it is not perfect. By the way, at that time, Dave Ditzel approached us.

Bochannek: All this work on Elbrus 1 and Elbrus 2, was this all still done at the Institute for Physical Mechanics?

Babayan: Yes.

Bochannek: The same building that—________ was built in?

Babayan: Yes, same building.

Bochannek: You mentioned that you had 20, 25 people on the operating system?

Babayan: No, the total number of people was more than that, but on the operating system there were 20.

Bochannek: What's the total number of people, do you remember how many people worked on the project?

Babayan: Oh, the total number was very big, I don't remember the number of people working at the Institute, a lot of people, which is impossible to count.

Bochannek: What kind of people did this work? Were they mostly people like you, who went directly from university into the Institute, to this project?

Babayan: Our team was mainly from the Institute of Physics and Technology. Like in Intel now, people working in our project are mostly—from the Moscow Institute of Physics and Technology.

Bochannek: Do you remember what was—were they mostly men, were they women, were they young, were they old?

Babayan: Mostly men, young people, definitely.

Bochannek: Mostly young people, mostly men.

Babayan: Mostly men. There were some women, but mostly men, definitely. There was not any restriction saying that only men [are hired].

Bochannek: I was just wondering if it's different than what you would expect from the American perspective, which is, like, very similar.

Babayan: Similar situation.

Bochannek: Similar situation, okay. What was the work environment like? Was it very long hours, people work on weekends, at nights, or—

Babayan: Yes. We worked hard, not because we were forced, but we were interested in the work.

Bochannek: People just loved doing the work.

Babayan: Yes. It was very exciting.

Bochannek: So you mentioned Niklaus Wirth, and some of the work that he did on programming languages. And a more general question I have is about transfer of information and transfer of technology between the west and the east. I understand that—

Babayan: Ah, before '91—

Bochannek: But you had access to journals, right, scientific journals?

Babayan: We read, but—

Bochannek: ACM, and IEEE, and all those journals?

Babayan: Yes. I don't remember specifically, but we possibly read.

Bochannek: What about conferences? Did you, or any of your team members in the '70s and the '80s, go to conferences in the west?

Babayan: It was very hard to visit, but there was some window at the beginning of '70. At that time I visited the United States. It was not so simple. [I had to get] permission, from the Kremlin, a long story. I visited Finland and then United States. I visited the Burroughs Corporation. It was a very interesting company, and we liked their approach. They used tags, but, unfortunately they did not use them for type safety. They used them for functionality, crazy situation. Then there was a mistake with iAPX 432, so they were afraid to use tags. They used other approach, which Burroughs used, but dropped it, because it was a wrong approach. They collected pointers in a separate segment. It was a wrong approach.

Bochannek: What did it feel like for you to go to the United States back then? Did it— was it a— what impression did you get from the United States?

Babayan: The first visit was very exciting, because it was a different world and so on.

Bochannek: Did it influence your thinking at all, your technical thinking, or just your day-to-day life?

Babayan: We never thought that we were behind. All the time we were well ahead of the western designers, well ahead, for many years. We never were behind.

Bochannek: At the same time, when all this innovative work on Elbrus was happening, on the other side, the ES systems were being ________

Babayan: It was the government decision. Sergey Alekseevich Lebedev, was the right person. As I know, building of copies was suggested to Lebedev. Lebedev rejected it. There was a big pressure from the government, [they wanted] to force Lebedev to join copying—he rejected. And we continued to build our own computer, because of Sergey Alekseevich Lebedev. He was a very smart guy.

Bochannek: Now you mentioned the Elbrus systems were, initially, at least, primarily for military.

Babayan: All the time for the military.

Bochannek: And there was—I believe there was another system, high end, high performance computing system that came later, that the PS 2000, was this something—

Babayan: PS, yes, well, everything was for the military. The ES system.

Bochannek: Was the Institute involved with that?

Babayan: No, a different institute [was involved]. Our institute built Elbrus. Elbrus and BESM.

Bochannek: Okay, now we're roughly in what year now? You said the Elbrus 2 was finished in the mid 1980s?

Babayan: It was finished in 1985...

Bochannek: Okay, there were a lot of changes in the Soviet Union in the late 1980s.

Babayan: No, at that time, no. Then we started to build Elbrus 3, wide instruction. In Elbrus 3 we built a new technology, binary compilation, it was a pioneering effort. I was running this project, because binary compilation helps to avoid the difficulty with compatibility. Compatibility prevents to push forward architecture design. But binary compilation helps to do this. So we decided that superscalar was okay, good, but it had a speed limitation. [To eliminate the limitation] we need a wider machine, and we developed a wide approach. To be compatible we developed the binary compilation technology. And at that time, Dave Ditzel approached us. [To be correct], it was Bill Joy who first approached us., It was the result of Mikhail Gorbachev's visit in the Silicon Valley., Gorbachev visited and asked the researchers to make a return visit. There was the return visit, T.J. Rodgers, maybe you know T. J. Rodgers? He visited Moscow. And Mr. Ross, who built SPARC compatible computer. There was a meeting in Moscow, and the organizer of this meeting asked me to make a presentation about Elbrus. After that T.J. Rodgers published an article about us in some newspaper. . . It seems to me that Bill Joy read this article, and arrived in Moscow.

Bochannek: What year was this?

Babayan: Sorry?

Bochannek: What year?

Babayan: '90, end of '90, 1990. Bill Joy said, “I would like to talk with Babayan.” And we met at some restaurant. Everybody was drinking vodka, we were speaking for four hours. He liked our approach; at that time we built Elbrus 3. We built Elbrus 3, with wide instruction, like Itanium, and fabricated it in '91. But there were changes in economic [situation in our country], the western computers were reliable, hardware technology was very poor in Elbrus 3, and we dropped it, it was impossible [to continue]. Well, later we built it as a chip, and it was wide, with security and binary compilation, but it was not superscalar.

Bochannek: Before the early '90s, when there were a lot of changes in the Soviet Union, at what point did you notice the impact of Glasnost and Perestroika? Did you notice a change in what the military was demanding?

Babayan: Until the mid '90s, probably, we had a comparatively good money supply from the government... But in '91, funding stopped. It was not very pleasant for us, but there was also a positive point. We could cooperate with the west. Two companies arrived to visit us Sun Microsystems, and Hewlett Packard. Hewlett Packard was beginning Merced at that time. We already have fabricated Elbrus 3 in Zagorsk, but we did not finish debugging, because we had no money, very bad technology, with water cooling system at that time. But there was a security feature in our computers, and our young people asked me—look, all are doing without security system, why we are doing it? I objected to them. When western computers arrived, those young people came up to me and said, that it was impossible to debug all this in western computers. Elbrus is very good for debugging. The productivity of software designing increased by ten times, we had ten times shorter debugging time, because of type safety. And trustfulness, because if the program has no interrupt, everything is correct. Plus failsafe capability, very strong trustfulness, no program mistake, no hardware fail. Unfortunately, after this economic changes, it was impossible to continue work on security. I tried to do this in Intel, too, but it was not easy to introduce, though it is very, very, very advanced technology.

Bochannek: I'm curious, with the changes and the availability of western technology in the early '90s, how did the mentality change, of the software engineers or the hardware designer? It used to be very difficult to get a computer to do your work on, and suddenly a lot of cheap PCs, presumably—

Babayan: Yes. In general, it was very positive, definitely. A lot of very good computers.

Bochannek: Did it change the way people did engineering?

Babayan: From the point of view of the design itself, all the time, we understood well that we were well ahead of everybody. Those two companies, which visited us, were interested in our design; Hewlett Packard mainly wanted to have information. They were not prepared to work with us, but Sun was ready to work. We started to work with Dave Ditzel. For three years he was giving us money. We were very happy; it was very, very, very good. If Dave Ditzel did not approach us, probably, our team would have not existed. He helped, we were very thankful to him.

Bochannek: So talk about the establishment of the Moscow Center for SPARC Technology, and the Elbrus Company. What—how did they come to be, and how did the transition from the institute for you, personally happen?

Babayan: Ah, the situation was as follows. Dave Ditzel arrived. At the end of '90, Bill Joy was here. Bill Joy was very pleased, and told me that he would ask Dave to take care. We exchanged messages for half a year and prepared a version of binary compilation compatible with SPARC, but with wide instruction. At that time, superscalar was not in the market, and we built four and eight wide instruction. Dave Ditzel also started many other projects not related to this, these were software projects. He saw a very competent team here, very good. [We got] much software work; a lot of people, maybe 300, worked for Sun; this was a very good money supply, we were very happy, because it was a terrible time in the Soviet Union, we were very happy.

Bochannek: So what was the organization?

Babayan: Organization was as follows—After these changes, IPMCE nearly died, because there were no orders, no money, but we were very healthy. We built a commercial organization. The institute was the owner of 25 percent, probably, the rest was owned by us, so it was a commercial organization. The relations with the institute were not very good, because the institute had no money, and we had a lot of money. All agreements [were made with this commercial organization, Moscow Center for SPARC Technology], and the rest of the institute people had no money. [The institute management] insisted that we must reassign the agreement to the institute—so, at some point in '95, we left IPMCE.

Bochannek: So who was the person at the institute who was unhappy with this arrangement, was it the head of the institute at the time?

Babayan: Yes. It was the head of the institute.

Bochannek: What was his name?

Babayan: Sergey Lebedev already died by this time.

Bochannek: So what was the name of the head of—

Babayan: No—I don't like to tell. So, it was very difficult to work. So we left.

Bochannek: This was in 1995?

Babayan: Ninety-five. In ‘95 we left the institute and rent some work space outside.

Bochannek: How many people?

Babayan: About 300.

Bochannek: Three hundred people all from the institute?

Bochannek: No, no, no, just from our team [the Moscow Center for SPARC Technology]. The institute was dying.

Bochannek: Okay, but all these 300 people were former employees of the institute?

Babayan: Yes, yes, yes.

Bochannek: But then joined the Moscow Center for SPARC Technology.

Babayan: Yes, yes, yes.

Bochannek: Okay.

Babayan: The Moscow Center for SPARC Technology moved [to another location]. So we were healthy, we could work very well, we had good technology, everything.

Bochannek: And who was the primary customer for this work, was it still Sun Microsystems?

Babayan: Sun Microsystems. Well, there were other companies, for example, a company who built libraries for chips. We also had [a contract with them]—but the main customer was Sun Microsystems.

Bochannek: Did the work your team did at the center become a shipping product for Sun?

Babayan: No, we just, how it calls—we worked under the statement of work.

Bochannek: They contracted with you.

Babayan: Contracted with us, and we were just doing the projects for them.

Bochannek: Okay. Now what happened to the Center for SPARC Technology? What happened there, what happened with you?

Babayan: Ah, the following. Ditzel tried to convince Sun to use our technology, wide instruction and binary compilation. But he wasn't successful, Sun rejected. Moreover, Sun appointed another guy for this project, not Ditzel. Ditzel was very upset and left Sun. After he left Sun, it was not easy for us in '95. By the way, he established Transmeta, and started the design, which research we had made for him. He was silent, he didn't tell anything—I know the reason, because he had investors. If he told the investors that he was working according to the Russian design, probably they would have been very suspicious, and he did not tell us, because we could not keep secret. So he continued to do this, but at that time, in '95, superscalar arrived, so four-eight width was not enough. At this time we dropped Elbrus 3, this big box computer. We built a chip, but it was 22 wide. But Ditzel still kept four, Transmeta Corp. made four, but superscalar posed a very strong competition. Sorry—

Bochannek: The single chip, Elbrus 3, was that the Elbrus 2000?

Babayan: It is the same thing, just a different name. So the chip of Elbrus had two clusters, and the peak issue rate of 22. It is very parallel and is still fabricated. Moscow center of SPARC Technology is still existing. In 2004 we joined Intel. Intel acquired us, but then there was restructuring inside the company and they decided not to do the second closing. Most of the people joined Intel, but some stayed [with MCST] to finish Elbrus 3. Actually, it was not an acquisition; it was hiring people.

Bochannek: So if that chip is still made today—

Babayan: Yes, they are still [working] —

Bochannek: Who is the customer?

Babayan: Mainly the military... It is not competitive, because the technology is not like Intel technology. They are fabricating on Taiwan, they use a library design. Intel is doing custom design. But the architecture is quite good, very wide, very strong, logical speed is good, but physical—

Bochannek: And is it still programmed, in one of those programming languages that goes all the way back to—

Babayan: Well they are using binary compilation, they can run x86 code, they have ported Linux, and they are using C, C++ and Linux.

Bochannek: So we talked about how you really wanted to have a programming language that takes advantage of the advanced architecture—

Babayan: But it's necessary to do hardware.

Bochannek: But now the architecture has evolved to the point where it supports all these programming languages you didn't like.

Babayan: Well the problem is as follows. Before those economic changes, we had our own operating system. When the borders were opened, we were forced to move to the existing operating systems, because of compatibility. We had to give up security, because [otherwise] we would have to build our own operating system, which is incompatible; [security was implemented with system libraries].

Bochannek: So while economically it makes sense, it sounds like you were—

Babayan: It was possible to do only by a big company, like Intel.

Bochannek: Okay.

Babayan: This was a small company, and it was impossible to do because of compatibility problem.

Bochannek: So you moved on, you left the SPARC Center. What year?

Babayan: In 2004.

Bochannek: And this is when you joined Intel?

Babayan: Intel, yeah, and till now, we are [at Intel.]

Bochannek: You're still here, okay. Now what have you worked on at Intel?

Babayan: On a new architecture, for the future.

Bochannek: So you're still doing computer architecture.

Babayan: It will be outstanding.

Bochannek: Good. Let me ask you some questions about sort of structures of decision making.

Babayan: Sorry?

Bochannek: Structures of decision making. Because it sounds to me, at the institute, because you worked on these very high end machines, and had good money supply from the military, is it—can you say that you were left alone, you could make your own decisions, your team? Or did every decision have to go to the ministry, and come back, and go—

Babayan: Formally, yes, but technically, it was our decision. There were guys in our team who ran the technological level, and I was running the logical level in software, system software. [There were disagreements between me and the director of the institute, we had very bad relations. The decision on our disagreement was to be made at the high level. Do you know Romanov, a member of the Politburo? Romanov and the Minister of Defense, Ustinov, made the decision. They fired the director of the institute.

Bochannek: What year was that?

Babayan: Sorry?

Bochannek: What year?

Babayan: So, the director was fired from the institute, and another guy [took this position]. There was also a disagreement between me and this new guy. My experience in designing computer is 50 percent of technical [work], and 50 percent of fight. All the time it's necessary [to fight], not only in the Soviet Union, in Intel as well.

Bochannek: Now the structure, you said, in practice, you, in the institute could really make your own decisions.

Babayan: Yes, yeah.

Bochannek: But you still had to deal with allocations of material.

Babayan: Material—but at that time, there was enough supply. The supply was from the military. If they decided that it was the right goal, the supply was very good. But many times, they found wrong goals, and they tried to prove that it was okay. They continued to supply, it was the waste of money.

Bochannek: Are there any stories that you recall maybe any anecdotes, any problems, where you tried to get something but you couldn't?

Babayan: There was a Party suit against me. The situation was as follows. I designed a new operating system for anti-missile and was preparing to introduce it. The director [of the institute tried to prevent me to do this]. I asked him—Why are you interfering? And he counter-questioned me—Do you think that I am a bad man? I answered—Definitely, you are. So it was the cause for the Party suit against me. But all people supported me. And the suit was stopped.

Bochannek: Lots of politics.

Babayan: The fight is quite natural for big projects everywhere. It's quite usual.

Bochannek: I had a technical question I forgot to ask earlier. One of the things that I always thought was curious about the Elbrus 2, in pictures.

Babayan: Sorry?

Bochannek: When I saw Elbrus 2 photographs, is that it still had a console with lots of lights, but it also had a display terminal. It had both.

Babayan: Oh yes both.

Bochannek: And I was curious if you could talk about, with these military machines, how people interacted with it? Did you keep in mind how people would use them, was there testing done?

Babayan: Old people know that there was a console. Elbrus 2 had a console it was necessary for the administrator. There was no possibility, for example, to switch off and switch on computer from the terminal. So, it was just the inheritance from the old [machines].

Bochannek: But you didn't do any studies of usability or you didn't bring in operators to see how they would use the machine. This is just—it's an engineering decision. It has lights or consoles. Because BESM 6, for example, has lots and lots of lights and switches.

Babayan: Yes, but Elbrus 2 also had a lot of [lights and switches].

Bochannek: But you didn't work—you did not work with people who would operate the machine and ask them what they wanted?

Babayan: No. Actually, it was the administrator who worked with console all the time. But programmers worked from displays.

Bochannek: Okay, so you have that split between—

Babayan: Yes, there were a lot of displays connected, starting with Elbrus 1.

Bochannek: Elbrus 1 already had displays.

Babayan: Yes.

Bochannek: Because it was a timeshare?

Babayan: Timeshare, yeah.

Bochannek: Okay, so let me ask you, is there any particular work that you did, that you want to talk about, before I move on to some more general questions. Is there anything that we forgot to talk about, about your career, about your work?

Babayan: First, what has impressed me, that there is a general opinion that Russia is well behind in computer science. We never were behind. All the time in our area, we were well ahead of any design. Many technologies, which we worked out many years ago, for example, binary compilation, security, are not still implemented. And what is amazing for me, only our team have experience in real hardware and high level language programming.

Bochannek: So that actually brings me to a question that I had, is, what do you feel the contributions that the work that you have done, the work your team has done, has been to computer science, computer engineering, hardware, software in general?

Babayan: Before '90—there were no connections, because the boarders [were closed]. Except for Robertson, we met with him, exchanged [some information] and so on. But after that, for example, Dave Ditzel took a lot from us, now Intel, we are doing a lot of things here. So in any measure, in computer architecture, our team is ahead of everybody.

Bochannek: What was the most exciting project you worked on?

Babayan: Here?

Bochannek: In your entire career, your most exciting—

Babayan: Well actually, in my entire career, I [was doing] only one project - improving computer architecture. All the time, from the very beginning, when I designed carry-save arithmetic—it was the only possibility to improve architecture of a strongly sequential computer, till now, I had been designing a new architecture, new technology. It's only one project.

Bochannek: We talked about operating systems earlier, we talked about programming languages.

Babayan: Ah, but for me, there is no difference. There is only one project, architecture, operating system and languages, compiler, it's only one project. It's all together.

Bochannek: What about databases and networking?

Bochannek: No, it is little bit more application-like, but operating system, compiler, language, and architecture [are a single project].

Bochannek: Is there a single project, single accomplishment that you're personally most proud of?

Babayan: I'm all the time improving.

Bochannek: Incremental, I mean—

Babayan: Yeah, yeah, yeah, I'm all the time excited of a new step.

Bochannek: Now you joined the academy, first as a corresponding member, and did—I'm sorry if I don't know this. Are you a full member of the academy, or—

Babayan: Sorry? No, I'm a corresponding member.

Bochannek: Corresponding member. Okay, what—how did these projects that you worked on, that took many decades, this constant improving, how did that influence other, through the academy, other projects?

Babayan: Academy of Science is such an organization that many people are working in. I'm not very satisfied with the organization of the Russian Academy of Science, it's questionable.

Bochannek: Let me ask you a question about things that went wrong. Are there decisions that you made, or projects you worked on, that you think you should have done differently, that were mistakes, that, in looking back, decisions you should have made differently?

Babayan: In some sense, each new step is a negation of the previous one. But, definitely, changes. For example, we built superscalar, in some sense, it's wrong, but it is some step. Wide instruction is also wrong. Today I think that what I'm doing now are the only right things. All previous works are wrong. But it's a necessary step. Well it's not wholly true, because without the previous technology, it's impossible to do what we are doing now…

Bochannek: Has your thinking about the field of computing changed? You say you always do new things, but is there fundamental change, the way you think about computing? Computers are many things today, they're—cell phones are computers, there are computers in our cars. Is the way—

Babayan: Actually, basic architecture in a phone and supercomputer, in great degree, is the same. Algorithms, definitely implementation is different, but basic things are the same. Look, x86 is in cell phone and in HPC; ARM is in cell phone, now it will be in servers and HPC—because it's implementation of algorithms.

Bochannek: Well after so many years of work, would be happy to not do it anymore, you still work in this field. What inspires you?

Babayan: I think my life is not enough to implement everything, which I am thinking about. There are a lot of things, which are possible to do.

Bochannek: So a related question to this, if a young person who maybe is thinking about going into university, or has finished a degree and thinks about going into industry, came to you and said, what advice would you give me as a young person.

Babayan: Oh, difficult question. Definitely, it depends on what this person likes, what he wants. As to me, I like my job. To the people who come, I can suggest to work with me, if they are happy. Many people are working with me, and all of them are happy. All people who are working in our team are very happy, because we are doing an outstanding project.

Bochannek: Are there any other things you would like to add, anything you'd like to say on camera while we have—

Babayan: Just I would like to say that I am happy that I am working in this very exciting area. And it's very, very good, and I think that I will do my best. There are a lot of things, which are necessary to do, to change architecture. And I am hoping that I will have enough health to finish all these things. I will be happy if this happens.

Bochannek: Well thank you very much for your time, I really appreciate it.

Babayan: Oh, thank you.

Bochannek: Thank you.

END OF INTERVIEW

