Alan Kotok

3/8 Interviewed by Jamie Pearson @ MLO

AK: My response is going to be that you see everything I have collected in 30 years or 25 years at Digital so the answer is not much. I'm not a pack rat.

JP: No, my idea is just to pick your brain about exhibitry and your role in the 6's and every other machine since you've been involved in most all of them. We're classifying things by the product families; it seems the logical way to do it. What, to the observer, sets these families apart from one another? So given that, look at the PDP1, our first machine, maybe we'll start there.

AK: O.K. well the PDP1 was built prior to my arrival here so the sense I got from talking to people who were involved was that their history had come out of the TX0 and so my guess was they were trying to build, manufacture, a more modern version of that. But, well, there were noticeable differences-they always get anecdotes out of it.

When we got one of the first PDP1's at MIT it was Ken, who in his wisdom, donated it, there wasn't much software for it and we decided that a way to get an assembler, that very rudimetary thing, it was a very marginal one that was shipped was to take our TX0 assembler and rewrite it and we discussed this idea with the professor who was incharge of the lab, Jack Dennis, he said "Well, I don't want to spend much money on this, you people will be doing this for the next three months and I'll see you." And we said, "No, no we've got a deal for you. We'll do this over the weekend and it will be running on Monday morning, and if it isn't you don't have to pay us for it, and if it is, then you pay us for the weekend time." He said, "Well that seems fair." Any by golly, on Monday morning that thing was running. So the moral is the machines were similar enough that that approach could have been done in a weekend by a half a dozen people - so I think that may have been the world's fastest software project of any size.

JP: But yet, the PDP1 was very much in the TX0 tradition.

AK: Now the 1, as an architecture, didn't get carried on in any real way. There were some fatal 2 and 3 designs that I would deal with. Now, I joined the company when the PDP4 design was pretty much complete, and my first job was writing the fortran compiler for the PDP4. And the thrust there was to simplify the PDP1 design and come in it at a lower price, because the one, basic type, was $120,000 and they wanted to get it under the $100,000 market. The basic technology was marginally different as I recall, but they basically simplified the instructions and expanded the main memory size. Now that actually began the family, which went on into the -7 the -9 and the -15. I don't think we had Fortran for the PDP1 by my recollection there was some but at least not in '62 which is when I started. There was a compiler called Decal which was kind of an Algol dialect or something and it never achieved any popularity certainly in my recollection. But everyone was interested in Fortran so we decided we had to do that. In those days, at least for a company of our size, if you shipped the computer with an assembler, a debugging aid and a fortran compiler, by golly, that was software.

JP: Was the -5 under development when you joined?

AK: No, in fact, what happened was that sometime in the fall of '62 I don't know, Ken or Gordon somehow hatched the idea that we needed to yet build a lower cost machine. Gordon sat down and discussed that with me and initially said, "Do you want to work on this?" and I said "yes". And Gordon's idea then was to build a 10 bit computer, and so I basically did the instruction set architecture definition for that machine. And I don't know whose idea it was, but it was either Gordon or me, but we were going to hardware registers in the machine which accumulated things like that were a major expense items, we said well we can cut out the program counter and we just won't have one of these and we'll make an extra memory cycle in between each instruction to go to the memory read out this program counter add one to it and stuff it back. The PDP4 design had introduced this, where the memory cycle was divided that you could read a piece of information out the memory you could do something to it and stuff it right back in. In those days we had core memories, and core memories were by nature, destructive read out, that's when you read the information out, you destroyed the information so you always had to write it back. And I don't know who, again, whether it was Ben Gurley or Dick Best or someone or Gordon, had the idea that you could do something useful with this information rather than just writing it back. And so in the PDP5 we introduced the idea of using a fixed memory location, (it was a location 0 for simplicity) was the program counter. And so with 10 bits I came up the idea of having, there was a 3 bit instruction so we had 8 potential instructions and then there was an indirect bit that would give you a reference to a full 10 bit memory address about a thousand words which seems like you could do something useful with it back in those days. That was before people knew it took megabytes to do anything. I spent a little more than about a month on this project. When Gordon bounced into my office and said, "Oh, I've got this great idea, I'm going to build a big computer, I want to build a big computer." I said, "Oh, great, wonderful I like big computers." So I said, "What should I do with this PDP5 stuff?" And he said, "I've got this guy Ed Decastro, why don't you just spend a few days and explain to him what we've done and he'll carry it through." So I did that and Ed expanded the design from 10 to 12 bits. Apparently Ed decided that 1K of memory just simply wasn't going to cut it, and so they added two extra bits to the thing. But basically the architecture didn't change from what I had done back in the 10 bit version. So he built the 5. Gordon and I then started on the 6 in the Fall of '62. And I remember we had a lot of discussions about what it ought to be like and this that and the other. The model for that was the IBM 7090.

We had a big plans for this little company and we were going to get into the big time and in those days the big time meant 36 bit computers. I mean the IBM 7090 family was kind of the standard of the industry at least the scientific industry that one. And we sort of said well what are the things that most need improving over that 7090 and we said well the address space is too small for a large machine it could only address 32,000 words. And the way indexing works, you're kind of strained and restricted. And again, and I don't remember, I guess, I think we got a patent on it or something but probably a joint patent but anyway, we had these general purpose registers, we had 16 general purpose registers, and they could be referenced in the instruction in three ways it could be either accumulation, they could be addressed as memory or they could be used as index registers to offset address as part of the address calculation. That seemed like a you'd need a whole lot of programing but much simpler to use a lot fewer instructions etc. I guess another thing that was on the front of our minds then was we wanted to address what seemed to be this big new artificial intelligence market and here it was 25 years later and we're still trying to address the artificial intelligence market. But this language called LISP was just being developed then by McCarthy and friends, and we wanted to make our machine do particularly well running LISP and so we built a number of features that facilitated that. The idea of built-in stack hardware and the two addresses that you could manipulate as two separate pointers went nicely within a word, were all features that made that do quite well. And a lot of those early machines were sold to LISP researchers and stuff like that.

JP: And universities. What about the time-sharing on the 6?

AK: Well, that was another thing that we were interested in. That was a time that MIT Project MAC was just getting organized and they were looking for a machine for time sharing. We bid on that thing, and as Ken likes to say, we're lucky we lost to it because it would probably have sunk us. We really just were too small and the PDP6 hardware was, at least, at the first go around, not robust enough, for a machine of this size. So we could have just gotten sucked down there, gone broke, but we didn't. Some of the earliest time sharing systems were built on PDP1's by BB&N and MIT, and we were certainly convinced that that was a common thing and none of the smaller machines had that in mind. The 6 we built in memory protection, and things like that, and a number of features we added to make it so that we could build such a system and protect one user from the ravages of another. The PDP1 time sharing systems were not particularly robust in that sense, and also they sort of required you to have only one person's job in the machine at a time and we had a drum memory. So the 6 was the first machine to have multiple jobs in memory.

There was one that was built for Brookhaven National Lab as a experiment control system. I may be wrong, but my impression is that basically everybody used our operating system because it was by nature a multi task operating system. So whether people time shared in a traditional sense of the utility and did whatever you want or had a major real time task but still people could, somebody would log in an edit and look at the results and so forth - so my impression is that basically everybody used this time sharing system even if it wasn't primarily for that purpose.

JP; And as a follow along to the 6?

AK: Then came the 10.

JP: The PDP10.

AK: Now that got started in '65 and at least initially, we didn't really think of it in terms of a new machine but it was a new

CPU module that would fit into the same memory and all the other structure. And as we were working on that we started now reaping some of the problems from some of the flaky design on the PDP6. I guess the low point got hit when Key Data Corp. came and sent their machine back and said "you guys ain't ever going to make this work, we've had it!" There was a famous meeting up in the cafeteria on 5-4. When Ken Olsen assembled the PDP6 crew and said, "Well this has happened and this is the first computer we ever had that's been sent back." He said "At Digital Equipment, everybody get's a chance to make a mistake, one mistake, and you guys have made yours. Now go out and figure out what small computer we're going to build, because I don't want to be in this big computer business, there's nothing but trouble!" So those weren't his exact words. And so we went back to replan what we were going to do well kind of their big mistake was they made their chairman of the committee to figure out what it was we were going to do and we waved our hands around enough and basically came back with the proposal to do what we had been doing, which was to build another PDP6 replacement but we managed through some clever subsetting to have enough options so that the base price could be down, I think that the goal was this thing couldn't sell for over $150,000 for the basic system. So we had this totally emasculated thing that had 8K words of memory or something which even in those days was miniscule and we just all prayed that noone would ever buy one because then we'd have to make it work! So this got the thing approved and the design continued, and I guess we shipped it finally in '67 or something like that. I guess the 6 went out, the first 6 shipped in '64 and the first PDP10 shipped in '67. And luckily we had gotten the circuit design right this time. And my recollection is at least that we sold about twice as many the business plan called for, at twice the average price, so that seemed to come out o.k. Now those machines were all built with discrete transistor circuits, one transistor in a little can, and so forth.

JP: Wasn't there a big module that was done on the PDP6?

AK: When we started with the PDP1 the modules were about this big or so. Then the PDP4 was built with the same aluminum frame modules but Gordon had introduced the idea of a module that wasn't simply a collection of uncommitted gates but a dedicated function module. And so the register module in the 4 had one bit of each register and it was twice as long. And because these connectors were so poor on pins, it introduced the idea of having a connector on the back, and then there were connectors that ran across the back and to make this at all practical they had to just be wires that touched every module like sources of clocking and things like that. Well, that kind of worked, and we went on to the 6. For the 6, we decided to expand on this idea and we went to double height, double density and it ran about this big. And again we had the major register one that of each ???? register that was on that board. And I seem to recall that in 92 transistors or something it seem like an inordinate number of transistors in those days. I mean, now it fits on the head of a pin or something like that. And then we had all these connectors on the back and as long as you didn't touch it, it was sort of o.k. was when these middle modules failed what were all these connectors and oops you broke the connector: so that wasn't an idea destined to survive very long, and that was one of the major changes we made on the 10, was to go away from that big board thing and there was a reaction to the big board thing. Around then, the module people introduced these things that were known as the flip chip modules they were these things about yea big and a little plastic doo hickey on the back and so forth, used in the 10.

In the 10's, there were sort of two sets of designations the things that were used by engineering and the things that were sold. In the engineering sense, somewhere after the PDP6, we decided to introduce the idea of numbering the components. The cabinets that were in the processor of the PDP6 were called the 166 processor or something like that. Then, when we went to the PDP10, Dick Best had decided we were going to change from numerical to designators to alphabetical, and he had decided that "K" was "processors". "P" was reserved for something important like "card punchers" or something like that. And so his scheme was very simple, we started at A and we went on, you see, and then the number was the family at the end. So the first PDP processor was called the KA10 and despite the fact that there were any number of salesman who went out selling these things, telling customers this is really my initials backwards had absolutely nothing to do with it, it was total coincidence. So the system was sold under the name PDP10 but the processor was the KA10. Beyond that, marketing had decided that they liked the PDP10 identification, but they wanted to protect some other image and they wanted to keep the 10 around, so they came up with this brilliant idea to call the system the DECsystem 10. They would have different models. And so the machine that we were developing after we did the KA10, had a processor called the KI10, and again the marketeers had sort of gotten in and chose I rather than B, which would have been the obvious thing, because this was our first machine that used integrated circuits as the logic elements. So they called it KI10 and designated that system, the DEC system. 1060, choosing a number sort of in the middle, and then they renumbered the old PDP10 the DECsystem 1040 or something like that so sometimes you might see something referring to the DEC system 1040 that's really an old PDP10. This was all marketing doing their thing, trying to confuse the world.

So this next machine was a 1971 machine and it was the DEC system 10, we had 60 and 70 and there was, I forget the difference but anyway the processor was called the KI10. We again had smaller modules but by this time had gone to integrated circuits on the modules and that was its technology introduction. Then, about that time, we got into another one of these, "we're getting too high up into the marketplace, we've got to go back to our roots and build some low cost machines." So the engineering management came up with this project called KL10, this was going to be low cost machine. Well, that was until the engineer got a hold of it and it became yet another higher performance same price processor. All these machines, basic systems, sold in the $300K ball park. And the KL10 I guess was the first machine that we introduced microcoding in that line. I think that certainly nothing else was, I don't know if it was in any of the other DEC machines. We had a memory cache and we had all these whiz bang technological things, and then to confuse matters we decided that our venerable old operating system which was originally just called Tops, on the PDP-6, Timesharing Operating System or something like that. A good person to talk to about that, or people to talk to about that would be either like Tom Hastings or Tom Eggers. That started getting old and cranky as such things wont to do and our friends over at BB&N had developed a system called 10X and a number of us thought that that had some advantages, actually I don't think I was among those people. Our solution to this problem was to buy the rights to 10X from BB&N to hire Dan Murphy who was one of the chief developers. He's still around. And so marketing came up with this brilliant idea that we could have another family of products around this new operating system which we named Tops20: so, they came out with this family of 20 machines which were in fact no different from the 10's except for the color paint that they sprayed on them: orange.

What was really funny was several years later I would get calls from marketing people saying that they had this customer who had a Tops or PDP10 or some sort of DEC system 10, could he run Tops20? I said, "You were the people who came up with this phony distinction and now you believe your own hype!" So they so thoroughly brainwashed themselves so it was inconceiveable that you could do this. I guess initially we had to make some microcode changes or something but they then got put in the old machines any so there was no difference in the hardware although we did have cabinet differences so that was a big piece of market. Things like the 1090 and the 2060 were identical computers inside; and again they all used this processor that was known as the KL10 processor. And then along the line there, there was another machine called the 2020 that got built, a one cabinet machine and that primarily happened in the 1976 time frame and I had nothing to do with it. I was out at Berkeley at the time with a guy named Bob Reid. Not the Bob Reid who is currently around.

And I guess that's basically what was built, and then we started on another machine, and the VAX had come out in the meantime and basically Gordon decided that we didn't need these two lines of machines and it was time to cut it off.

Q: Did the 11 development parallel to the 10 development?

A: The 11, obviously from its number, started after the 10. What had happened there was that we were doing the PDP-9's and we were getting alot of market pressure from the 16-bit computer people. I don't know how much of it was particularly rational but somehow or other the style had moved to 8, multiples of 8-bit things in the computers and so it's kind of a question of, what are we going to do? What 16 -bit computer are we going to do? Because we pretty well have to do one or the other. And a group was sent off to develop something, and they did, and they came back with a proposal and it got shot down by several people who said it was unbelievably complex; we couldn't possibly do that, the customers are not going to understand it, so it won't sell. So esentially from a fresh start Gordon and a few other people came up with the 11 design, again which I had nothing to do with, but it was certainly every bit as complicated as the thing that had started out being too complicated. I would probably say that the 11 design was more innovative.

Q: Because of the Unibus?

A: Well, the Unibus was certainly an new idea that helped it, but its approach to computer architecture, I would say, was probably a fresher approach then the other guys had used, but certainly no less complex. Again I'm not a good person to ask about this.

Q: Would you say Bruce Delagi?

A: Well he's certainly a good person.

Q: After you went to Berkeley, did you get involved with the VAX engineering ?

A: Well, personally what happened was that I was involved in a project called Dolphin. We sort of got into that in '77; in '78 there was a whole lot of arguments; and by '79 the project was cancelled. I came back here to the Mill and was sort of an engineer-at-large and I messed around a lot in the network space. I was part of the gang that brought Ethernet in and so forth. I didn't design anything but I negotiated with Xerox with the group called the Interconnect Task Force that Sam Fuller had pulled together. Sam Fuller ran this office of technology or something like that. So I spent about a year or so, a year and a half maybe. We wound up in building 2. In the meantime a PDP10 project rose out of the ashes called Jupiter and it was running in Marlboro despite whatever strategy we did or didn't have. Another project, called Venus, which eventually became the 8600 was also running in Marlboro. It must have been early in '81 there was a design review that Gordon asked to have and he invited all of his friends, and the conclusion was that this project was in deep trouble. In those days, I believe Jupiter was viewed as being a wonderful project; if only Venus had been doing things like Jupiter. But, Venus was in trouble and I could sort of see what was happening and I tried to keep my head down but I didn't succeed and sometime around May or so of '81 Gordon dragged me into his office and said will you go down there and run that project, give them some technical direction? I said, "well, I'll make you a deal: I'll go and spend the month and at the end of the month I will either advise you to flush the project in which case I will have nothing further to do with it or I will recommend something that we can do with it." So I did this and my conclusion was that the basic design was fairly sound but the management practices were terrible and their design methodology was awful and the management was forcing us to meet schedules despite the fact that the design wasn't progressing. They had checked off a lot of boxes on the schedule but in fact the design wasn't done. So kind of the next thing I knew I was the management and that lasted a few months before I cried "uncle." And then Bob Glorioso came in and Bill Walton and I worked together with me being sort of technical manager and Bill Walton being sort of process development manager. It took a while, we replanned it and then we were still late on the replan somewhat but the design got completed and it eventually worked. But that took us until '84 I guess. When it looked like things were fairly stable I decided to go with the RISC machines for a while.

[BEGIN ALAN KOTOK, SIDE 3]

AK: Gordon was on the ________ the VAX idea there. I remember that the VAX group was initially located right next door to us up on the top floor of the Mill when we were working on...

BL: On 5-5?

AK: Yeah, on 5-5, it was in the early stages of that. So while we were deep into logic design they were arguing over concepts. And in fact one of the early proposals was that they were going to build a PDP-10 architecture machine. And then they got, persuaded themselves they wanted to follow on the PDP-11 concepts and do the 32 bit instead of 36, and so forth. And that's -- so anyway...

BL: Why?

AK: Why? Well, 32 bits, the multiples of 8 had become much more in vogue in the mid'70s, the early to mid-'70s.

BL: Is that because of the IBM?

AK: Well, IBM had brought out its system 360 in the mid '60s, which was 32 bit, I guess. The 360 was a 32 bit machine. And had basically persuaded the world that they brought in the usage of... well, the term "a byte" was already into usage referring to this 8 bit quantity. And the 36 bits just didn't seem to fit all this. And in the mini-computer business there was this wave of 16 bit machines and so it seemed obvious that one should continue following that path. So, at the time, we felt, that seemed like a different market and that was okay and we would continue on with this 36 bit stuff. But in '75 or so, Gordon said, - this is crazy. We really don't need two lines of computers and blah, blah, blah and we really should have only one architecture and this was really the one operation system, the one architecture, what have you, wave that came in the seventies. Now, by '75 we were starting on a succesor of the KL10. Off to the side, by the way, there was another machine alled the KS10 which was a physically noticeably smaller machine. It was marketed at the DEC System 2020 and didn't receive a lot of acceptance. Again, it was a small version of a big machine that wasn't cheap enough and was also positioned into a market that was used to a different style of machine. That was done by a different team of people than I as involved with.

BL: The biggest winner was the original KA10 and the KL10.

AK: I would say those two were the big winners.

BL: KL10 was the DEC System 10.

AK: The marketeers decided that they needed some fancier names around all this stuff. In the year of the KI10 they brought in the concept of the DEC System 10. Retroactively they called the KA10/PDP10 as the known system, it was renamed DEC System 1040 or something like that. The KI10 became known as the DEC System 1060. Now, the idea of the 20, again this was all the marketeers doing, ____ Bryan, when we decided that were really dealing with an obsolescent operating system, called TOPS 10 operating system was really creaking. We had been talking with people from Bolt, Berenek, and Newman and particularly Dan Murphy who had developed a system all Ten Ex [SPELLS TEN EX] and this seemed to have the right new directions to it. We negotiated some dean with BB&N for rights to that and we hired Dan and a few people and they recast it into what became known as Tops 20. To celebrate this the marketeers decided we needed a different machine and they came up with DECS 20.

BL: Which was the KS10.

AK: No, no, that was one of the members, okay but the family of the DEC System 20 was really the same old hardware architecture, construction architecture, painted a different color, literally. The 10 family as painted blue and the 20 family was painted orange. The new operating system came out around the time the KL10 came out. So we had the 2050 was the basic model of the KL10. You could also get it as the 1080 or something like that. They also had changed the cabinet shapes, so the same back planes could be installed in PDP10 style cabinets and painted blue or in DEC 20 style cabinets, painted orange.

BL: But ______ slightly.

AK: Slightly lower, they were rounded instead of squared and so forth. As I say, you could buy your wine in different colored bottles. What was interesting is that after a year or so of this the marketeers would come and say, - we have this customer who wants to run Tops 10 and they bought a 20. Is that okay? I'd say, - you were the people who made those mistakes in the first place, of course it's okay. The computer inside doesn't know the difference. [LAUGHING].

BL: Was there compatibility from 10 to Tops 10, Tops 20?

AK: Not very. You could get programs from one to the other but I wouldn't call them compatible.

BL: It wasn't binary compatibility as we know it.

AK: It wasn't binary compatibility in the sense that the operating system called Interface was entirely different. Now the application code, to a large extent, was binary compatible and it ran on the same instruction set. But you had to have the compiler, you had to tell the compiler. First of all we had to make the compilers a little more agile and then you had to tell the compiler which software environment to target the compiltion for. There were attempts at compatibility. It's kind of like today, a program that could run on VMS for instance, running on UNIX. If you run the UNIX and the VMS on the same computer then there is some compatibility but there are different operating system calls. There was some attempt at Tops 10 layer of interface calls to smooth this transition. Again, the person to talk to about all that is Dan Murphy. I don't recall that there was any hardware concessions to this. It was all layer software. But my memory is extremely vague and I wouldn't take much of that.

BL: Also, the marketeers started pushing it into the commercial?

AK: Right, that was the other thing, they were going to sell this 20 stuff to _____ the commerical market.

BL: Did it happen?

AK: Yeah, to some extent it did. They sold to a bunch of banks.

BL: Did requirements differ, develop configurations for the commercial _______? Did you have to redesign?

AK: I don't recall from a design hardware perspective that we did much about that. I think it was mostly a marketing and peripherals, what peripherals these people needed. They had the same old IO bus, still compatible. The KL was a very solid machine. There was a 2060 version. It had another pass of clean-up through it that managed to speed it up somewhat, 20 percent or something like that. It was only a year or so. Then there was this 2020 thing that I mentioned that sunk in there too. As I recall, in '75 I and friends were working, looking at technology for the next generation. We had been talking with Motorola and came up with this idea of a so-called, macro cell array MCA, which eventually found its way into the 11 VAX 8600. So we were working on that stuff and then I took a year off to go teach at Berkeley in '76, '76 and came back and we were working on that for a little while, for another year or so when Gordon finally prevailed to stop PDP10 development entirely.

Also at the time, the Tewkesbury VAX developers were working on, started on this project which was known as Venus. Gordon decided to make everybody equally unhappy that the Venus development was moved to Marlboro. Some of the VAX people came along and some didn't. The project successor, the KL which was known as Dolphin was cancelled. In its place the Venus project appeared using the macro cell array circuitry. About that time I decided that maybe I had been digging around this computer designing business for too long and somewhat pissed off. So I departed that scene and went over to the Mill to work initially for Bill Johnson. Bill had, at the time, the research group and a few other activities. I was just sort of general technology consultant for him and got involved in an activity known as the Interconnect Taskforce. We were wondering what to do about local area networks and things of that nature, high speed, storage connects that might be more universal and so forth and so on. I got involved with the local area network stuff and was involved with bringing Ethernet into the company. I was involved with Xerox and those people, '79 it must have been, '79, '80, that timeframe. There are other undercurrents that had been going on for many, many years that I was working with the Corporate Telecom people and was their main technical advisor, almost since I've been with the company. In '75 I was involved in the design of the DTN telephone network which a unique kind of system.

I and a New England Tel engineer sat down and designed the damn thing. So, I've always been involved in this telecommunication stuff. So I was working for Bill Johnson for a while and then Bill took over the software organization and Sam Fuller took over research. So I worked for Sam for a while. But I wasn't really in this direct research group. Again, he had maintained this, what BJ had called the office of technology and we were more general engineering gadfly type of people.

BL: This was before Bob Taylor?

AK: This is long before Bob Taylor and Butler and company. So, about 1981 I guess it was, spring of '81, Gordon calls for a design review, a full dress design review for this Venus project which had been chugging along for several years, seemingly not doing exceptionally well. They were continuing to slip schedules. Gordon wanted to get to the bottom of what was going on. And so he invited me to come along to this design review. What was revealed was that they were not only ready to build the prototype, ______ was quote complete unquote except there were hundreds of loose ends which the plan was, they would be worked out in the debugging phase. We said, - no, no, this is not going to work here. So Gordon scratched his head for a month or so on what to do about this, assorted meetings. I could read the handwriting on the wall. What happened was that eventually I got this call from Gordon saying, - go down there and do something about this. So I said, - I'll tell you what, I will spend a month in Marlboro, this was where this was happening, this was in June of '81, and at the end of the month I will report back to you and will do one of two things: either I will tell you, I think I know what to do and am going to do it or that I think you ought to cancel this project, in which case, I will have nothing further to do with it. You can do whatever you want about it. So, during the month what I concluded was that, rightly or wrongly, that the basic engineers know what the hell they were doing and that basically the management had pushed them into a micro organized schedule around these MCA chips and that because of the flow requirements through Motorola to process these chips they had to have chips done on a very rigid schedule. Despite the fact that the design wasn't done, the management put so much pressure on the engineers to declare it done, that they would merely collect a list of thing that were not right, hoping to fix them in the next one but of course now the schedule was even tighter so this list of not working things had really grown without limit. So what we decided to do was to reset, go back, cancel this prototype business, to get the design right because it seemed like the fundamentals were right and furthermore, Gordon insisted strongly and correctly that we do very thorough simulation of this thing because it was much too complicated to debug into operation. So, a whole simulation program got put into place and it was done at various levels, logic level and macro instruction, micro instruction simulators and so forth. I guess it now took sometime in '84 or something to actually ship the machine.

BL: At that time, I guess he pulled the management out(?) on Glorioso. He used to be in R&D, right?

AK: Right. The micro events of what happened there were that, it was shortly after July or something like that, when I became quote, - chief architect - unquote, the existing management disappeared and I not only was chief architect I was chief everything. After several weeks of that I said, - I think you got the wrong guy here. I have never been the manager of a 40 engineer group. I'm in over my head on this. I really need some mangagement talent. So, there was an engineering manager who I had known quite well, been with the company for a long time and we became joint managers of this thing. Then, the VP who had been running this whose image also got tarnished and disappeared and Glorioso. Glorioso showed up as the group manager. So in a span of three or four months, now we were working for Glorioso. I returned to doing more technical things.

BL: What were the real challenges once you reset?

AK: The first challenge was that there had not been a total agreement on how the clocking of this computer worked. And the _____ in that machine referred to various boxes. We had the I box, the E box, the F box, N box which were the four major sub-systems in instruction execution ______ and so forth, instruction phase that basically got the instructions out, execution of instructions and then there was specialized floating point unit and a memory sub-system. The technical leaders of each of these boxes had agreed to disagree on how the clocking system worked and as I looked at that I said, - I hate to tell you guys but this computer ain't ever going to work. So I annointed a fellow who I have worked with before named Paul Leone as chief in charge of the clock. I had read this management book, somewhere along here, I picked up an MBA along the way as well ...

[END OF SIDE 3].

AK: ... read some management book about an interesting technique, which turned out to work pretty well, which was that I would get all these people in the office, every time we had a disagreement and I would say, you people have a week to come back and tell me how we're going to do it, and if you don't, I'm going to tell you how to do it. And since they would much rather figure it out themselves than have me tell them how to do it. Because the chances were that my idea wasn't going to be any good anyway, it would just be an idea, that a lot of these issues got resolved. So we finally got a common clocking (?) system that could be accepted, except for the fact that the floating point box people insisted that they were going to run their box at double speed. And so they, we accommodated them there. They had to be synchronized into the correct phase whenever they interacted with the rest of the system. So that was one major thing that had to get sorted out. We had a lot of work to do on microcode that the simulation was a very challenging phase. And we were gathering computing cycles on every machine that we could find in the company to run this stuff, to run hours and hours and hours, a million to one simulation ratios, things like that. But eventually we did get a, simulated it, and had I think a reasonable debugging phase with the actual hardware. Also we decided that we would need the ability to replace an IC chip with an emulator for it, so we built some software that would in essence automatically design a screen circuit emulator for these fance macrocell chips. And we had this rack that you couldn't, these umbilical cords that you could plug in this monster umbilical cord to where one of these big chips would plug in and have the logic externally. But we also decided we couldn't depend on those too much because they were slow in keeping the signal integrity was difficult because you had to slow the machine way down. So we kind of got through that and as things seemed to be clearing up and my usefulness to the design was finding (?) off, I decided that risk machines were the wave of the future, and gathered a small group to start working with risk machines in the context of how to do this in the way that will be compatible with VAX.

BL: Let me go back a second. There was a lot of continuity between the KL-10 team and the Venus.

AK: Yes.

BL: What was the role of Jupiter in that?

AK: Well, Jupiter surfaced, I don't know how long, maybe six months to a year after the conclusion that we didn't want to build any more PDP-10s. There were customers saying you can't do this to us, and so forth. And one of the old PDP-10 managers persuaded somebody that it's time to do another one. And he hired a design leader from Itel, I forget who it was, who had been building IBM clones and so forth. This guy had some pretty exciting ideas of how to do this. It was a pipeline machine, which we had never done before. And he started up this group that was in the same office area as Venus. And I sort of would occasionally get called in to see how they were doing, and oftentimes it wasn't very good. And they were good at handling how it was going to work, but not too good at actually producing it to practice. And so with various bumps along the way, it had its demise. And there was an effort to go -- these characters who had been building the IBM adapters for the PDP-10 on the West Coast, they were also designing their own, they decided since DEC was going out of the PDP-10 business, they would meet this need. Systems concepts. And we went out and met with them to see, maybe we would license their design or something like that, with Jupiter going down the tubes. But the guy, the president of that company, who I've known for 30 years or 25 years, no one has ever accused of being a businessman. And he basically had too much pride to give DEC the rights to his thing, and thought then, since they were so small, they never made a success of that either. But the DEC negotiating team as it was, you know, the lawyers and the like, it was crazy. [LAUGHTER] So that was an unfortunate sort of ....

BL: When the risk project came out, why did you think that was going to go?

AK: Well, there was a lot of talk about risk machines.

This was in the '84 time frame. And we had the idea that we could do a risk machine that would in fact emulate VAX code faster than VAX could actually do it. And we were, we had written some beginnings of some translators that would translate the VAX code, and we had fleshed out the architecture and so forth, and were working out circuit ideas, and along the way we needed a new type of RAM chip that would be a synchronous-clocked high-speed RAM that had existed previously. In fact, we had a patent on it. And at least from our point of view, from my point of view, what happened was that Dave Cutler, who was living out in the West Coast in Seattle and was proposing various VAXes or something like that, decided that he wanted to work on these machines. And so he told Jack Smith that unless he had responsibility for these machines, he was quitting. Now he did this several times, and this was kind of the last time he got away with it. And Jack said, Right, Dave. And all of a sudden we didn't have a job. And we said to Jack, did you know what we were doing? Well, mnmnm ahmmm, mmmmmm, uhmmmmm, not really. Well, okay. So we sort of lost our corporate position in some sense. And Glorioso said, well, you know, charge ahead anyway. And I said, well, I'm not going to -- I once concluded I was to get out of this processing business, it's too busy around this company. So it's time to make this decision again. So I guess it was in fall of '85 I got offered a job of chief architect for Storage Systems. And so I said, goodbye Bob, I'm going over to Storage Systems and I was there for ____________, that was the summer of '85. Labor Day of '85, 4-1/2 years would've been early '90, which was indeed when I came over here. So having been there for 4-1/2 years, I got word that the telecom group was going to establish an engineering development --

BL: With your experience in doing net already --

[BOTH TALK AT ONCE]

AK: It seemed like the obvious thing to do, and I knew BJ (?) from his past, and so I took this job as telecom technical. So this is the product organization, a selling organization, not an internal ____________. So here I am. And there was a bunch of stuff to the story.

BL: I did hear the same story around from everyone in the company that when Cutler took over that it was just like huh? But what was going on?

AK: Well, of course what happens, and subsequently [BOTH TALK AT ONCE] was that Cutler didn't seem to be doing sufficiently and this MIPS opportunity came along, and we made the decision for better or for worse, to jump onto MIPS. And then after a year on that, the Hudson people kept saying, well, we have this, they were working on the background on this Alpha thing. And many of the same concepts that we had worked on back in '85 had resurrected themselves. And we had this VAX compatible thing with code translation and so on, and so, so technically I feel vindicated, although somehow or other I got cut off at the knees at the time, when maybe, who knows, it might have put us in a much better position a lot sooner than where we are now. But this is life.

[BL: This is the end of the Alan Kotok interview.]

AK: You want to start with my arrival here or something?

BL: Yeah, when did you?

AK: Okay, in 1962, in June of '62. So, it's almost 30 years now.

BL: How did you fall into the Mill?

AK: Well, Ken in his wisdom gave a PDP-1, one of the early PDP-1s to MIT in I guess it was the fall of '61, summer of'61 and I happened to work for the lab as a student then, came into possession of this machine. I along with a number of other people, students, ____ programming staff. I started to get to know what DEC was doing and it seemed interesting. I was going to graduate that next year, '62 and Ted had been admitted to the graduate school and was lookng for a summer job. And at some point, Gordon Bell wandered through. I Knew Gordon when he was a graduate student who did a project for the same lab a couple of years previously. So Gordon said, - what are you dong this summer? And I said, - I don't know. I had been working for Western Electric doing research down in New Jersey for a couple of years. I said, - I don't know, probably go back there. He said, - why don't you come out and work for us? So I said, - that sounds like fun. He said, - you're hired. Call

Personnel and tell them you're hired. So, I did. And so I came out in the summer to do programming and at that time there was a very small programming department, consisted of one guy named Dit Morris, Harrison R, quote DIT, unquote, Morris, the 3rd. He ran the programming group and there were like three of us or something because Digital, at the time, didn't know that programming that a computer company did. The computer that was being completed at the time was the PDP4 and they needed a Four Tran Compiler and that was admitted to be something that a manufacturer should probably supply was a compiler, an assembler, debugging program of some sort. So I wound up writing, doing a lot of work on this PDP4 Tran Compiler. In the fall I said, - I got to go back to school. But I arranged my schedule so that I had two days a week free. So Tuesday and Thursday I would come out to the mill and pursue endeavors for DEC part time. Then, at the end of the calendar year, Gordon said, - well, why don't we just put you on full time and you can still have time to go to school. So I said, - well, that's seems fair. So in the spring, I was working three days a week and going to school two. My Masters program kind of petered off into somewhere. Finally, after about three or four years, pressure from my mother and Gordon and my thesis advisor built up on me and I actually finished the degree. Anyway, basically I got officially, we did some, after the fact, things so if you look in the Personnel records, they think I started in August, which was a compromise of part time and summer work, but I was actually here in June of '62. I worked on this programming project for a while and then Gordon came running in one day and said, - oh, you've got to build a smaller computer than the PDP4. In those days, of course, the obvious thing to do was to call it the PDP5. Gordon said, - why don't we make this a ~~tended(2~~) 10 bit computer of all things, make things smaller, to get the ____ down. He said,- why don't you work on that? So, basically I roughed out the architecture for this machine, into the fall of '62. And then Gordon came running in, as he was wont to do. He's sort of this idea generator type person. He said, - time to build a big computer, we need a big computer. MIT Project MAC had just been formed. They were looking for a large time sharing system and Gordon felt that there was no reason that we shouldn't built this for them. So we started on what became the PDP6 and Gordon said, - why don't you hand off this PDP5 stuff to somebody else. And the guy in question went on to other things, named Ed DeCastro. So he took that over and somewhere in that process expanded from 10 ato 12 bits because 10 was too small. In those days characters were represented in 6 bits. So, at least you got two characters in the wordage, made memory a lot more useful. So I went on to work with Gordon on the PDP6 and he and I basically designed that machine from the end of '62 to the end of '63. It took about a year.

INT: How was it like working with Gordon?

AK: Well I found it very exciting. I thought we always got along well. I think that when I noticed people getting into trouble was when his reputation became so, sort of strong, his power became so strong and the fact that he was a volatile type person. He literally, I saw him jump up on the table and jump up and down, that people would tend to cower. I remember telling people in the early days that my job was to filter Gordon Bell's bad ideas. He threw out lots and lots of ideas and reall expected people to say, - no Gordon, that's the stupidest thing I've ever heard. You get taken aback and you could explain to them, - you'd say, - oh yeah, you're right. And go on to the next thing. I noticed, as years went by, people stopped doing this to him because they were afraid. He was too big a person to tell him he was full of shit. But I didn't have that problem. So I enjoyed that. Then this PDP6 project, I guess we delivered the first machine sometime in '64 early or in the middle of '64.

BL: I'm interested in the development of VAX. It seemed very important, the first time sharing, the seeds of the

10, large machines. How did you start designing these? You had never designed before the time sharing system.

AK: It's hard to recall. I think the process was, - in those days the word size of a computer was not something that everyone had settled on, 32 bit computers hadn't taken over the world, so part of the problem was deciding the size of the machine was very strongly dependent on the size of the word. We said, - we need a bigger machine here and we tried all kinds of numbers. We were guided by the, at the time, the IBM 70-ID(?) and its friends were around. That was the going thing in big computers. We settled in on the 36 _ thing. IBM's family that started about in the mid fifties called the 704 vacuum tube machine. Then they had the 709 which was a vacuum tube macine and then the 7090 was their transistorized version. There was a 7094 and so forth, that whole series of machines were common architecture essentially, a 36 bit machine. We were very focused on memory addressing, the ability to address a lot of memory. We persuaded ourselves that memories were going to grow rapidly.

BL: How? Why?

AK: It seemed that there were just a lot of programs that were straining against the limits of the memory in those days. We had in mind that we wanted this machine to last for a few years, that all of the various memory extension packs that had been proposed, were all very cumbersome. So we wanted to have direct addressing of memory, to what seemed at the time, a very large amount of memory. Now, this very large amount of memory was a quarter million words which is roughly like one megabite of memory in today's _____, which doesn't sound like a lot but in those days that was a huge amount of memory. Word addressing was the way things were done in those days. So, it took 18 bits to address a quarter million words and two of those addresses fit in memory. We were also at the time rather enamoured of this programming language that John McCarthy was promoting called, Lisp. It was very important in List to have two addresses in amptitutde, in Lisp was called the CAR CDR. were the two pointers out of the Word. So creating List structure was a 2 pointer structure was important. We decided 32 bits didn't have big enough pointers and so we'd gone to 36 bits. Then we thought about what instructions we could put in this computer. The desire to do list type things influenced that somewhat, the ability to extract addresses. Stacks or as they were called, in those days, push-down lists seemed like a good programming concept and we put in instructions specifically to deal with those, which no other _____ had before. The other thing was that we were both quite familiar with the TX0, which was this Lincoln Lab machine that Ken and others had been working on back in the early fifties. The lab that I worked at, at MIT that had come into possession of the PDP1 was the lab that had taken over the TH0 from Lincoln Lab in 19, - I guess in the end of '57 or something, - this machine had been moved from Lincoln Lab to the Cambridge campus. So, if you want to go back that far, I was a very active member of the MIT Model Railroad Club, which you probably have heard allusion to. One of the members of that club when I joined in '58 was Jack Dennis who was a professor of some repute at MIT and Jack at the time was a graduate student and was doing some thesis project of something on this TX0. So he brought a number of the club members over to see this machine and since it was very lightly used the management there didn't seem to object to our using this computer. It was really a rather powerful computer of its day. That osmosed into a part time job with that lab. So I did what today would be called software support for that lab along with a number of other under graduates. So then they came into possession of the PDP1.

BL: You had been familiar with the PXC1(?).

AK: The particular thing was, I'm not sure if I have the right word, I think it was called, - micro programming. Now, this was not the same usage of the word as it's used today. What it meant was that there were instructions which were decoded in little groups, each of which performed a specific function. There was a couple bits that said, - decode these bits to mean, move this or that internal register, here or there and this one, another bit, caused an arthitematic operation to occur and other bit would clear some register or something like that. PX0, one of the instructions, it was an 18 bit computer, all 18 bits were decoded in this fashion. It didn't have an address and so you could cause all kinds of interregister manipulations to occur with this instruction. It seemed like a good idea because it was very lost cost to implement, didn't take a lot of gates because the functions were decoded from individual bits or pairs of bits in the instruction. When we went to the PDP6, we wanted a lot of instructions, a large reportary of instructions, to do all kinds of things and few instructions. This was the beginning of what became the, quote, CISC wave of computers. Obviously, to us, it seemed like you would want to use the fewest number of instructions since they each had a fixed cost in time, not precisely fixed but they were dominated by the memory references. We wanted to have very powerful instructions where in one instruction you could cause a whole sequence of events that seemed useful to occur. The design tools, the sophistication of the design, is the packaging of the larger unit, made it very expensive to have all kinds of instruction decoding logic. We combined the idea of this micro coding, as we called it, we had expanded the instruction, the word to 36 bits. We now had enough bits to devote to different fields in the instruction. We set it up so there eight major categories of instructions. Within those various fields were decoded simply to do particular operation. For instance, one of the fields had to do with the destination, ___ memory that would go to a register. This was uniformly decoded throughout all the instructions. The number of gates, electronic gates in this PDP6 was, I don't remember but it was into the thousands or something like that, including all the arithmetic and everything, which was relatively, comparatively low, which was certainly low compared to say the IBM machines. I would say that what we had done was, we had imported some good ideas that allowed us to minimize things. The machine was two cabinets worth of logic and each one of the circuit boards, there was 25 of them across the rack or something like that, they were five and a quarter inches high, about 400 of these modules and many of the modules were very simple modules in the sense that a module would have half a dozen _______. We couldn't do the whole thing that way. So some of these were structural purpose modules. For instance, one module had one bit of the major registers in the machine. They were deeper and they stuck out in back. They had to put connectors on the back. Most of the things were made out of standard deck modules. So, I don't know what the actual gate count of the machine is, probably in the order of five thousand or something like that. With them we had something like 365 different instructions for the machine. It seemed to strike the fancy of every computer hacker who came in contact with it.

BL: It was an elegant solution?

AK: Yes. Now, the problem was the capabilities of Digital and circuit designers who designed these gates, mechanical designers who designed the innerconnection systems, to build something of that size, which was four or five sizes larger than the previous machines, was somewhat lacking. One of the early machines were sold to a company in Cambridge called, - Key Data - which was a Charles W. _____ Associates, spin-off, subsidiary. Our salesman was deliriously happy, he sold them a very large configuration. It never quite worked right because there were just too many of them, too many flaky electrical signal problems, cables that had to _______. There was static discharges from the line printer, paper would, line printer was like a _______ machine, sparks would fly in the line printer and the memory over here would die and things like that. So, this was a great embarrassment. They sent the machine back. This was in

'65 or so. In the meantime we had started working on a successor to this. Ken Olsen called this group together in the cafeteria in the mill and said, - well, at Digital everyone gets a chance to make a mistake and you people made yours. I always knew the big computers were a bad idea. So you really have to be thinking about doing something else. Well, what Gordon Bell thought about was that it's time to leave. And he went off to Carnegie Mellon. At this piont with Gordon leaving, I was sort of then chief technical person through this whole large computer line. So we scratched our heads for a while and came up with a solution that preserved all the work we had done by coming up with a - we persuaded the management that we could make this thing small enough by having a loss leader configuration that was small enough to meet the guidelines that ______ _____ which I believe was like a 150 thousand dollars. Ken said, we don't want to sell anything for more than 150 thousand dollars. So we put together this proposal where we pulled out the floating point, pulled out this and that. By this time numbers had gone by and we were up to PDP10. We had an 8K word memory version of this thing. We knew none of the software would run but you could develop some stuff and then run a dedicated application, that was the theory anyway. We kept our fingers crossed that no one would actually buy this thing. But Oxford University actually did. We said, - you know, you're going to have to give them another ____ _____. We gave them the extra cave(?) memory. Luckily no one else bought the script ____ machine.

INT: Tell me how the 10 differed? It sounds like the 6 was a classic design engineers machine, beautiful from a design point. But it was more like a mechanical engineers nightmare.

AK: What we had decided to do between the 6 and 10 was, (see p. 14-15) there were a couple transmitter, having a new kind of logic; one was the development of a new module-typology which, although the name in fact was a misnomer, was called the flip chip modules. This was a grand idea to use some sort of a thick film ceramic substrate technology which would be, the circuits would be plated on to these things and then the chips would be flipped over, these big, big chips, silicon chips, they were ceramic, half inch or a quarter inch, plated circuits on them and the the idea was that you flipped this thing over and bond it to the printed circuit board but it never worked. The trade name got picked up and referred to a new line of circuit modules that were about this size.

BL: Two by three inches instead of five and a quarter by whatever it was.

AK: And along with it we went to Gardmer Denver wire wrapped back planes instead of hand soldered connectors. The style of circuit, the actual circuit design for the gates had evolved. So, in essence, we had gotten to a much better electrical design and a much better mechanical design. The PDP10 was a reimplementation of the same PDP6 architecture in a new style of circuits that were two or three faster as well as being a much solider design. It was, in fact, a very solid design.

BL: Tell me more about the circuit designed ____ ____. What were the evolutions at that point in electrical design.

AK: The original PDP6 circuits, remember that I am not and was not the circuit designer, I designed the architecture and logic but the original circuits were a style of circuit that was brought over from Lincoln Lab by Dick Best and tended to be, what I call, resistor capacitor ...

[END OF SIDE A].

AK: ... coupled transistor logic, which used these reactive components, if you will, as speed enhancers but the effect of all of this was that circuitory performed differently with respect to clock rate(?). The beam of the circuits were ______ _______ proportional speed to capacitors in the past.

BL: So the speed ____ the clock rate?

AK: Yes, so that there were also a lot of delay lines, electronic delay lines. It was very soft focus logic, if you will. It evolves with the TX0. The circuitery in the PDP10 was what we called, diode transistor logic. It was pretty much DC coupled throughout. It was still discrete transistor logic, integrate circuits were beginning of the available ___ but they weren't fast [UNINTELLIGIBLE] whatever it was, I don't remember the combination, the weighing of the factors. But in any case we had this discrete dio transistor logic. I can't think of all the names of the people who helped develop it but in essence, they presented that stuff.

BL: That came from Lincoln Lab.

AK: No, no, no, this second generation of logic was developed entirely within Digital in '65, '66 timeframe. My feeling is that prior to that, it was very heavily Lincoln Lab influenced but by this time enough of a core of people had had enough time to work internally to develop a new style of logic.

BL: One facet of this is that it wasn't affected by an increase in the _____ _____.

AK: It was very stable in the face of [INAUDIBLE].

BL: So you wouldn't have as many timng problems.

AK: So the timing problems were fewer.

BL: And you wouldn't have any delay problems, maybe signal problems ____ __ _____.

AK: This is not to say that one could ignore these problems but they certainly diminished. We also had somewhat better designing tools. We were at this point using computers to aid the design. We had some software which would count up the _____. We were using the PDP6s to do today would be called CAD. We didn't have simulation or anything in those days.

BL: It sounds there was a type of a modeling.

AK: A certain part of the design could be checked by software, which of course, we had to write that too. Most of what we did was focused around the wireless, 134 because the object of all the design software was to get to something that you could punch a deck of IBM cards that would run, the ____ ___ ___ ___ ________. This was called the wire ~~less~~ list. We did all the design besides paper drawing but we would then encode that including all the signal names and the connections and so forth. We had developed this printed wire list.

BL: On a grid system?

AK: There was a line printer listing of all the different signals and what the sources and _____ were and an analysis could be made to see that there weren't violations in the total amount of wire. [INAUDIBLE]. So a certain amount of analysis could be done with the encoding of the design into this computerized form. We didn't have any automatic _____ tools so we would position, arrange the modules, assign them to particular slots in the _____. There was some ______ that would then develop the control sequences to the wiring machines. now you got to put the wire here and run it here and then down here and so forth.

BL: I think Allen Kent told a story about one fellow who went to wire a machine with double ___ boards.

AK: I don't recall but he was very heavily involved in that phase of the ______.

BL: The machine wouldn't do the double sized board so somebody invented a robot arm that somehow fixed the problem

AK: I don't recall this but it sounds very plausible. We certainly have recollections of straining the limits of what that machine would do.

BL: This is the ultimate of the technology before IC. So you had figured out how to overcome the PDP6 signal integrity problems which caused all the errors through this new technology and through the evolution of circuit design. Was the logic design essentially the same?

AK: No, it really wasn't. We had taken a different approach to the caulking. The PDP6, the timing was done entirely with the delay line things.

BL: How did you calculate the delay lines?

AK: We tried to compute how long various events would take due to the rules that the circuit designers had given us and then we would try that. Sometimes it wouldn't work and have the length of the _____, change the ____ on the delay line. It wasn't very scientific. The PDP10, we didn't yet go to what we call the synchronous clock system, where something would just beat(?) up time and you would do everything. We still used a flexible type of system but we combined many, many of the operations to call occur on standard time _____. But you wouldn't always go through the same sequence, there was a steering mechanism for this timing thing. So on certain kinds of instructions we didn't need to do certain operations, by-pass a whle set of timing. [INAUDIBLE]. ... kind of thing where if an event was going to occur it would in general always occur at a particular signal. We have these big flow charts which show execution time 1 and then all the different instructions, what they would do, so the design process then came to be a designing the control logic to say when this time signal came along which particular register transmitters would actually occur.

BL: But the architecture stayed the same so there were the same number of instructions.

AK: Same number, well we didn't resist the opportunity to add a few but all the old programs were basically run. In general from the 6 to the 10th to that family, there were always upward compatible.

BL: You're talking about differences in logic and design.

AK: Well only that a lot more of the logic design focused around setting up conditions that would be ready in time for some pulse to go by rather than the old style which had a lot to do with pulse steering and so forth. When all is said and done, the operations all had to net out to the same result. In some way you had to be able to match one to the other but it really was a style of designing.

BL: As you worked in this design did it come out as elegant as you thought it at the 6?

AK: I think we were persuaded and there was certainly more elegance, it ran a lot faster and it worked essentially all the time. That wasn't to say that we still didn't have new incidence of static electricity. This same Oxford University computer which we gave a free upgrade to. Word came back from England that this machine would run and overnight it would crash. They would beam it to run some computation, in the morning it would be dead. And this would happen over and over again. A support guy would come in from the U.S. and he would spend weeks there. They couldn't figure it out. Finally I had to go there. So I spent the best part of a week looking at this machine. I was pretty much convinced that there had to be some sort of a static electricty problem because nothing else would seem to explain the fact that the machine would wind up, sufficiently incoherent that it couldn't have gotten there through any simple errors. I was about at the end of my wits too as to just what to do about this. In fact I had to come home at night(?). Saturday morning I went into the lab to check the machine and it was still running. And I went out to the console for some reason, at the time we always shipped a very stylized chair with all of these computers, armless chair that had a sweeping back, anyway this chair was sitting in front of the machine. The console was over here and there was a teletype machine. I tripped over the foot of the chair, the chair was rolling off and hit the teletype and as I'm looking at the machine, the machine freezes and I said, - ah ha and the ______ of that machine in that case was _______, wheels on the chair. It would discharge through the teletype frame and the wires then went into the logic and sparked it out, that was about it. Having discovered that, it was a simple matter to replace the chair, replace the carpet.

BL: Somebody must have moved the chair around.

AK: Well, we figured what happened was, that the cleaning people came in and to vacuum under the computer they pushed the chair around and it would hit something. They didn't know from beans that they were doing anything. We then sent our advisories to the field on carpets with no threads in them and careful about good straps on teletypes, discharge signal cable. So, I guess that machine finally got delivered in '67. We were doing so well we went on to an integrated circuit version.

BL: You mentioned before that aside from this stripped down version that it was designed to use larger configuration. As this hit the market, it was delivered, what kind of configurations was it being used for, very large time sharing?

AK: Not really as I recall. Time sharing was the way of life there but not generally lots of people. We were still mostly focused on engineering labs and scientific uses, small usage operating system got to the point where [INAUDIBLE]. Memory boxes which initial machines, the 6 I guess, 16K, 16 boxes. I guess we got to 64K. I don't think a lot of those PDP10s, the original PDP10s were maxed out on memory. MIT [INAUDIBLE] third party which was a full quarter megabite [INAUDIBLE]. It seemed like an unbelievable amount of memory at the time. [INAUDIBLE]. He had been working on logic simulations for dozens of years now and then another person who is involved, some software, some hardware is Tom Edgars(?). He's currently [INAUDIBLE] ... since 1964 or so. He and Dave _____ pop up in the history news conference. So they were on this PDP10 final development from the early days. A guy who did a lot of the circuitry stuff, I don't know if he's still floating around, his name is Russ Doane.

BL: He seems to have done everything.

AK: He was one of the original circuit designers in the PDP6 and the PDP10 days. Where do you want to go from here?

BL: [UNINTELLIGIBLE].

AK: From the perspective of the number of machines we had projected to sell, I think we sold two or three times as many. I think one would have to count it as a success. But on the other hand, the whole line of large machines never really made a hell of a lot of money for the company over the years. As time went on, it probably succeeded in losing more money than it ever made. But it was certainly sufficiently a success to insure the future of that line for ten years or something like that. Certainly the original PDP10 got to be known at a lot of universities as the machine to have in computer science community.

BL: I've heard that and why?

AK: Well, the price was reasonable. You got a lot of machine for the money. At least in the computer science area, people tended to program a lot of machine items and so forth. We had a very elegant construction set. The competition, IBM, was always viewed in universities with some disdain and compared to scientific data systems, there were a few other machines that were around. Anyway that went on for a couple more generations, there was a machine called the KI10. KA10 was the processor of the original PBTD(?)_10. KI10, by this point we were designing things so you could upgrade memories and processors independently, so people who had bought the old PDP6 memory could still run on PDP10. We maintained compatibility with the memory bus and the IO bus for quite a number of years.

BL: What [INAUDIBLE].

AK: Well it was a proprietary bus, there were two buses that emanated from the machine. There was one called the memory bus and there was the IO bus. These were unique to that family of machines. There were a few people, a few third parties who built some peripherals for those. We never wanted to build an IBM tape adapter because [INAUDIBLE].

BL: The IBM were standard.

AK: We had this group who kept saying, - we're going build you a tape drive and a ______ and they never did. The customers really wanted these IBM drives. There was a little company that is still in business doing _____ down in the West Coast called Systems Concepts which built a PDP10 to a IBM channel adapter. It kept them in business for quite a number of years. We don't do this but if you want [INAUDIBLE]. It's called Systems Concepts. So the KI10, I guess the technical success, I think the feeling was that the cost incentive didn't improve enough for its time.

BL: The KA10?

AK: The KI10 was part of the KA10. KI10 was the integrated circuit. It was built again using the clip chip style module. Instead of having screet(?) circuits it had integrated circuits.

BL: So it was the same logic.

AK: No, no, of course we got the _____. My memory doesn't serve me well.

BL: [INAUDIBLE].

AK: Yes it did. I don't recall the numbers. My recollection is that it wasn't really a screamer from a performance standpoint. It didn't save any significant amount of money at that time. From an overall market standpoint, it didn't do exceptionally well, which caused for another time to go back to smaller machines, lower cost machines. The next project became known as the KL10, where engineering minds, people liked to use these letters to indicate something someone could relate to, not chosen out of the alphabet. And L stood for low cost. Go to the front office again and tell them, no, no, 'cause the KI had gotten higher, 450 thousand(?) dollars. So, the KL project was intended to bring the cost back down again. I guess it did to some extent but that's where we really have to use micro coding. [INAUDIBLE]. .... as a consequence of the programs stored in a micro program memory. And this micro program was in fact loadable [INAUDIBLE]. So, again, the design now was really different because now you were designing machines micro program execution and didn't really know specifically what the instructions were doing. They were dispatched out of the instruction. And that machine did very well again too.

BL: Did KL lower cost(?)? High performance?

AK: Much higher, noticeably higher performance. This now takes us into, this was being done in the early seventies and [INAUDIBLE].

BL: Meanwhile Gordon had ...

AK: .. the idea of the VAX.

BL: Not quite.

AK: Well the VAX development was overlapping the tail end of the _____.

BL: So Gordon had drawn up the 10s after playing with the 6s.

AK: He lost interest and was involved in the original PDP11 developed ____.

[END OF SIDE B].