Reminiscences of computer architecture and computer design at Control Data Corporation
CBI OH 321

Interviewee: Toth, Dolan
Interviewee: Steiner, Kent
Interviewee: Specker, Wayne
Interviewee: Rowan, Tom
Interviewee: Resnick, Dave
Interviewee: Pavlov, Mike
Interviewee: Pagelkopf, Don
Interviewee: Moe, Robert
Interviewee: Lincoln, Neil R.
Interviewee: Krueger, Larry
Interviewee: Krohn, Howard
Interviewee: Kort, Raymon
Interviewee: Hutson, Maurice
Interviewee: Hawley, Charles L.
Interviewee: Grinna, Dennis
Interviewee: Control Data Corporation
Interviewee: Bhend, Bill
Interviewee: Bergmanis, Maris
Interviewee: Alexander, Curt
Interviewer: Neil R. Lincoln

Repository: Charles Babbage Institute, University of Minnesota, Minneapolis
Description: Transcript, 210 pp.

Abstract: Organized discussion moderated by Neil R. Lincoln with eighteen Control Data Corporation (CDC) engineers on computer architecture and design at CDC. Engineers include: Robert Moe, Wayne Specker, Dennis Grinna, Tom Rowan, Maurice Hutson, Curt Alexander, Don Pagelkopf, Maris Bergmanis, Dolan Toth, Chuck Hawley, Larry Krueger, Mike Pavlov, Dave Resnick, Howard Krohn, Bill Bhend, Kent Steiner, Raymon Kort, and Lincoln.

Citation: Reminiscences of computer architecture and computer design at Control Data Corporation, OH 321. Oral history interview moderated by Neil R. Lincoln, May and September 1975. Charles Babbage Institute, University of Minnesota, Minneapolis.

I'd like to start out by going around the room and having everyone introduce themselves to the tape recorder and I'd like you to introduce yourself in your relation to any of the projects up there on the blackboard, and one other thing--I would like your definition, individually, of computer architecture, and computer design. Why don't we start with Bob Moe.

BOB MOE:

I started with the company in 1960 and the first couple of years was spent as a checkout engineer on the 1604. In 1962 I started in the design moved over to the design rather, of the 6600 and moved to Chippewa in 1962. I stayed at Chippewa through 1968 and returned to Arden Hills in the 7000 development division. I was part of the development efforts on the 6600 and the 7600. From 1970 to 1974 I visited Europe and in 1974 I returned to the STAR 100 Development Division, currently as Program and Systems Manager in that division. I am not sure how to answer your last question, Neil. With no chance to think about it ahead of time I think architecture is the unique assembly of, in Control Data language, functional units, memory and control, and peripherals. Does that make sense? Your other question was what, design, Neil? Well, that's interesting. I think most of us that were pretty new at the computer business, 15 years ago, didn't know much about design because we didn't know much about computers. So, I've always felt that logic design as opposed to circuit design (and that kind of thing) was a pretty rudimentary technique of taking building blocks, design rules and implementing an architecture.

WAYNE SPECKER:

From that list of computers I built the "Perform Algorithm" type box for the 3600 that did polynomial evaluations. Later on I worked on the design of the main control section of the 3500 and then the instruction unit and virtual addressing for the STAR 65. Then there was the NPL line and I guess I worked on that for a year and a half. That never got past the initial specification writing phase. I guess I was working on the storage access control and memory organization for NPL.

NEIL: You and Curt did a little logic design on the NPL didn't you? It seems that there was not an inconsequential effort invested in that.

WAYNE: yes, we made an investigation into what was called the fresh operand file, which everyone now calls cache memory. That was just before IBM announced their version.

Architecture, well there's hardware architecture and then there is (in my opinion) a concept of overall architecture. I have heard people talke about both. I presume you are talking primarily hardware and not about operating systems, which do go along with system architecture.

There are even certain distinctions within hardware architecture. I think we've included the instruction set, the memory addressing system including the virtual addressing system, and the central memory structure (number of ports); I guess we should also include the use of multiprocessors and other configuration aspects and also I guess a little bit of how you start the thing up and provide some sort of diagnostic interface.

Logic Design is a cut below that. You're trying to build the architecture for a given performance level for a given cost. Putting the components together whether they be chips or discretes or LSI.

DENNIS GRINNA:

I started with the company in 1961 on 160's in checkout and transferred over to the Chippewa lab also in 1962 and spent the full 12 plus years there in checkout. I started out, of course, as a junior engineer and didn't really get involved in design or systems work or anything approaching that until later years. Presently I am working in the Aerospace Division as a systems designer. My concept of architecture is pretty much as has been defined already. It would be the overall concept of the interrelationship between an instruction repertoire and the hardware. Architecture deals with both sides of those topics a little bit; but more on a concept level. Then we have the logic designer ; and I would add another interface in the form of a systems engineer. The logic designer would be responsible for implementing that concept in the hardware.

TOM ROWAN:

I started at Control Data in 1962 just about the time that the 924 was getting completed and I was in product line management so that I got involved in the 3600, 3400, 1700; you left out the 3200 and 3300 on your list and also the 3500. I was involved in the planning side of the 6600 and 7600 and finally on STAR. I think they have said enough about architecture and design and I am not going to add any more.

MAURICE HUTSON:

I started in 1964. I came right out of school and trained on the 3400, of which CDC got the last one built about the time I was ready to start. Being the least valuable in that group I was transferred over to 6000 Development when they needed people at the time the 6600 came to Arden Hills. I was in that group until 1968. Then I came over to the STAR project. At that time the architectural design was done. I am not sure if I can define that, but I was in on both designs of the STAR 100 (laughter). I think the difference between architectural design and logic design really is the division of labor. Usually you have a management group that more or less formulates the architecture. Purely because it is physically too much work for them to do all the way to the end, they create block diagrams that vaguely define the machine. It is then the logic designer that actually creates a finished product.

CURT ALEXANDER:

I started with the company in 1961 and worked in checkout in the field on the 1604 and 160's and finally 3600's. The first design I was involved in was the 1700 processor. After that I worked on the 3500 main control design and from that went into some of the paper design work on the MPL; and from that moved to what was then called the PL50 which was a smaller version of STAR 100. We went to Canada in 1969 because we ran out of money and finished it (as the STAR65) up there. In 1974 I came back and am now working on, or beginning to work on IPL. My definition of architecture is the definition of the features of the whole system including the features of the central processor I/O and other features of the whole system and how they interrelate. I think that the designer then takes over from there and once you begin to

reduce that to any kind of block diagrams or eventually to logic that would be called design work which is really a process of reducing architecture to some kind of implementation.

DON PAGELKOPF:

I joined the company in 1960, just like Bob Moe, in 1604 checkout. I was the EIC on 3600 S/N 1. I moved on as a design engineer, or as project engineer of the 3400, the 6400, the EM1 and then I moved into technical management. There I was associated with 6000 and 7000 and now STAR 100. I think Maurice came close to my definition of architecture because I think we all have a feeling of what that means. In my mind architecture has to be done by one or two or a very few number of people who get inputs. Whereas design again, must be done by many. One of the aspects of design that has not been emphasized so far is really what it's all about. The designer produces documents from which computers can be manufactured. All too often I find one of the problems we have on projects is that people forget that we have to document for manufacturing folks because that's the name of the business we are in. That's what I would conclude.

MARIS BERGMANIS:

I joined the company in 1961 at which time I was part of the 160 department under Don Malcolm; then I was assigned to the Z program (it was known later as 3200) which eventually evolved into 3300 and after that I moved on to the 3500. For the 3200 I designed the BCD unit, (not the BDP, but the BCD unit) and the floating point unit on the 3500. After that I was kind of an addition to the NPL project, looking at emulation possibilities for 6000 and 3000 and IBM machines. From there the logical step at the change of the program was to go to PL50 which thereafter became known as STAR 65. I went to Canada, came back and now I am part of the IPL program at Arden Hills, laying out the design plans for the P3 and P4 of the IPL product line. My interpretation of architecture is probably closer to Curt's, probably because we have been working together for so long. In my mind, it is a definition of the major functions of hardware and software both and the relationships between the various phases. It includes all the I/O, the operating system, the maintenance features and the hardware. You cannot separate them, I believe, any more. Logic design is actually taking those basic concepts (that are defined, I agree with Don, by a few people--they can't be defined by a committee) and actually putting that into logic circuitry that can later be manufactured.

DOLAN TOTH:

I've been with the company since 1959. I worked first on logic and circuits and printed circuits that eventually went into the 3600 and the 3200 machines. I had some association with 160, and 924. I got involved with STAR in the very early days and had an input into its design and architecture even though I am unable to define exactly what I mean by either of those terms. To me architecture is taking a basic set of requirements and fitting components around these requirements in order to come up with a machine that will satisfy the cost performance criteria that are desired. Design then would be an actual detailed implementation of these rather gross ideas into a final kind of design.

CHUCK HAWLEY:

I started in 1958. I worked on 1604 and the 1607 (which was a tape unit for the 1604) and after that I did the 924, 3600 and 3400. I was associated with the 6400 group and then EM1 and STAR 100. The definition of architecture: There are two definitions; one of them architecture, one of architectural design. Architecture is something that defines functional modules and interfaces between them. Architectural design is the process of making trade-offs to decide what functions should be where and what the size of the interfaces should be. To talk about logic design I think we really have to divide it into two areas; one is the design of logic and the other is the implementation of that logic in hardware. What I mean by that is when we talk about the design of logic we might be talking about having a register, having connections between registers, in order to accomplish the function that is required. There is a further detailed level of logic design which now designs a register out of circuitry, partitions this among different boards, figures out wiring and so forth. I think there are two very separate, distinct types of things. Quite often the same person will do both but they are two different types of work.

LARRY KRUGER:

I am on the STAR 100 design team. I spent some time babysitting the STAR 1B for a few years, both the hardware and microcode. I'm responsible for the STAR microcode now. I don't think I can add much more to the definition of architecture since I have not done any architectural design at all. I know one thing from being on the STAR project; the architecture pretty much better solve the problems of the programmer or we are not going to sell any machines. That would be part of the architectural concern if I were to start a system design. We have problems to solve for the user, and if he can't come in an do that we are not going to sell anything to him, no matter how good you put functional units together and how detailed your design is or how well it works.

MIKE PAVLOV:

I've been with the company since 1968. Since yesterday I have been attending the corporate technology committee meeting and I haven't been to the office since Tuesday which means I may not still be at Control Data. [Laughter] I have never been actually involved in the design of machines but have remained strictly on the software side so I won't give any architectural things other than to note that clearly architecture is how you put the skins on the machine and build it so that it stands up and other cosmetics on the outside. Architects obviously build houses and things like that. Design, of course, is doing the structure to fit the joists and things together so the cabinets have something to hang on to. Don't construe that as a programmer's view of architecture.

CHUCK HAWLEY:

One thing I didn't say when I was giving my definition of architecture. I didn't call attention to the fact, if you note very carefully, I did not say hardware. Architecture applies to software, hardware or anything else that you want to concern yourself about.

DAVE RESNICK:

I have been with the company since 1968 working on 7000 and STAR and I am now doing some LSI work. As far as architecture is concerned I agree mostly with Chuck, it is a definition of a bunch of blocks and how they interact; what the function of the blocks is, based entirely on what you're doing, software, hardware or anything else.

HOWARD KROHN:

I've been with the company since 1963. I worked on numerous Aerospace projects and in 1970 I became associated with STAR. Right now I am working on the LSI project at ADL, designing tools which can be used to build future LSI computers. My concept of architecture is a little different. I think architecture has kind of evolved over the years. Earlier there was no concept of architecture when the 6000 was designed. We had a specific circuit technology and were going to design a set of instructions that will make a fast machine and use the circuit technologies to the best advantage. I think right now that architecture is still somewhat driven by circuit technology and it will always be that way. You come up with new circuit technology and you try to evolve an architecture around that. This is really happening now with the IPL line. It tends to forget about the circuit aspect but I think it's still there and is still driving it.

BILL BHEND:

I've been with the company since 1961. I started out in Field Service and went into 1604 checkout and I don't know if Don remembers me or not but I worked on the 3400 prototype for a few months. I went over to Chippewa Falls in 1964, worked on the tail end of the 6600 development, went through the 7600 effort and when Seymour left took over the fixed point and floating point design for the 8600. As far as I'm concerned architecture is the building blocks of the computer system and logic design, I guess, is deciding what goes into the building blocks as far as the hardware is concerned.

KENT STEINER:

I've been with the company since 1964. I've worked on the lower 3000 line of computers in computer checkout and also on the 1700. From there I came and worked on the design of STAR. My concept of architecture is basically just the total system. The total computer system to do the job; it's not just hardware. I don't think of hardware as much as I think of the instructions or the spec that you try to implement to perform this particular architecture that you're trying to get at. I look at logic designers basically as divorced from that.

RAYMOND KORT:

I've been with the company since 1961. I worked in checkout for 1604 and 3600. I've worked in computer design since then. I worked on the 3800 a little bit, then the STAR 1-B and the STAR 100. My concept of architecture is a vehicle or means to perform a job for whatever you want to do.

NEIL LINCOLN:

This is the last chance you're all going to have to pontificate. One thing about having a microphone in a bunch of strangers is they tend to get very philosophical. We now have everybody's voice pretty well identified so that when we try to make the transcripts we'll be able to tell who's who. I've asked you to come to talk about the aspects of the projects that you've been associated with and it looks to me like there is sufficient overlap among you that we can piece together some of the information about project history, technical innovations and lessons learned. If there is other information you'd like to dig out please intervene and let's ask questions. I'd like

this discussion to be more interactive rather than have one person speaking. I don't know how far back we can take this historical search. I did put the 160A on the board. What I'm looking for is a little bit of background about that project. If the people who were here when the 1604 was conceived could kind of describe the history of the project, how it came to be, how and why, we could start there. Why did we end up with 1604, why did it end up looking as it does. If we can't get at that maybe we can take the next one down the line. Most of us ought to be able to have some pithy things to say about things like the 7000, 8600. Should we start with the 160A. Do we have enough talent here to drag the 160A through this?

TOTH: I think I may be able to fill in a little bit on the 160A. Actually the 1604 preceded the 160A in design and that came about after a meeting {and this is all heresay} with those who were then in the company {numbering 15 or 20} in which most of the management people were not in favor of beginning development and construction of a big computer until we had a contract for one. However, there was one person in the beginning who felt otherwise and that was Seymour. I think he was able to convince them that they needed to begin to build some hardware and so the 1604 then was started. The success of the company hinged on the successful development of that computer. Then it is reported {and I can't say that this actually happened} that Seymour was sick one day and came in the next day with the design of the 160 computer. It was then put together also. I can't remember which actually was working on the floor first, the 1604 or the 160.

HAWLEY: I can maybe add a little light to what was going on. Preceding both of them there was a machine called "Big Character" which was a small computer which was built as a test vehicle for the memory stack for the 1604. This was to prove out the memory drive circuitry, sense amps, etc. for the 1604. This huge computer which was about half a memory rack full of cards, had a paper tape reader and punch and one mag tape unit. We used this to do our mechanized design work for the 1604. The design of the "Big Character" machine was kind of what led then to the 160. The story was told that Seymour was sick for a day and came back with the design but what he had done was take the "Big Character" design, cleaned it up a little bit, changed it from the 16K 6-bit memory to a 4K 12-bit memory and came up with the design for the 160.

NEIL: Who did the "Big Character"?

HAWLEY: I'm not sure if it was Seymour or Carl Kohler. They were both involved.

NEIL: Ok, now the 1604 came about how? There must have been a requirement.

HAWLEY: I'm not actually sure the design started on that before we got a contract or after. The first one was for Monterey, the naval post-graduate school. At the time I joined the company in the fall of 1958 the company did have a contract for that.

TOTH: I'm reasonably confident that the design was started before having had a contract.

HAWLEY: Jim Thornton started with the company in June of 1958 and there may have been some designs previous to that but I think he scrapped them and started over at that time. This was about the same time as Control Data got the contract.

NEIL: The conception of the machine basically was due to what? What was Control Data doing at that time. They must have been making money doing something.

HAWLEY: They had built some displays for FAA and built a drum system that Perkins had worked on.

NEIL: Ok, but the basic reason for starting the company was to go out and build machines for Captain Wolf at Monterey really, wasn't it, even though we didn't have a contract.

HAWLEY: We didn't have a contract for quite a while after the company was formed.

TOTH: My recollection was we had no contract and the most likely first customer was not Monterey but NSA.

PAGELKOPF: Was that computer called "CLIPPIN"?

HAWLEY: That came about a year later.

TOTH: They were anticipating {with mostly hope I think} a contract with NSA rather than with any other customer.

NEIL: Ok, so the 1604 was then defined, there appeared to be a customer; if we could make a machine with that performance we'd be able to sell it. What was the rationale for building the 160?

TOTH: The 160 was built I guess because there might be a market for a small computer and indeed there was.

NEIL: You mean we were building to inventory?

MOE: Was the delivery date of the first 1604 in the fall of 1959?

TOTH: I believe so.

HAWLEY: It was about January 1960.

NEIL: Ok, so the 160 was born and you did sell it.

HAWLEY: The thing that really pushed the 160 was the contract from National Cash to make a device for NCR to use as a programmed accounting machine. CDC made the 160 in a brown cabinet, it was called the 310.

MOE: I believe the 160 was completed definitely later by a significant amount of time than the 1604. Anybody have a clue as to when the first 160 was built and sold?

HAWLEY: It was pretty close to the same time as the 1604.

NEIL: So it wasn't designed as a satellite machine for the 1604 or anything like that. Some people think that was the reason for creating the 160.

HAWLEY: No, but that was one of the first uses to which it was put. In other words, once it was available then it was hooked up.

TOTH: I venture a guess that the concept of the satelliting as it was called, attaching the 160 to the 1604, came about the summer of 1960.

NEIL: Here at CDC or at the customer's site?

TOTH: At Control Data, in fact we copyrighted that name Satellite.

NEIL: What did it consist of, how did you tie them together?

HAWLEY: They accessed a common tape unit. {Laughter}

NEIL: I see.

HAWLEY: I mean the tape unit was designed so that it could be accessed by both computers.

KORT: The first one didn't work.

NEIL: That was our first two-by-two controller.

KORT: I worked on the tape unit that came back for a little while, it never worked.

HAWLEY: It used the Ampex FR300 and it was so far from meeting specs as to be unused.

NEIL: Is that the one that had the tape loop on the side in the shape of two diamonds or triangles?

PAGELKOPF: That's the one that you put the sensing unit on the "sense tape wrapped on upper capstan".

HAWLEY: It made lots of long skinny tape.

NEIL: That was before MYLAR tape. Any other light that can be shed?

TOTH: The 160A came about because some of the customers didn't feel that 4000 words was enough memory. We redesigned it for 8000 words of memory.

HAWLEY: They also speeded up the memory in time.

TOTH: I don't remember that.

BHEND: They increased the I/O performance too.

HAWLEY: The original memory was 6.4 microsecond core which was the same as the 1604 and then the updated version got down to 2.4.

NEIL: Ok, the 160A had a buffer channel then.

HAWLEY: That went along with the extra memory.

TOTH: There were some additional instructions put in which made a super-set of the 160.

SPECKER: They added a real interrupt.

NEIL: How many 160's did we build?

TOTH: I think our contract with NCR was for 50 if I recall. I believe that all 50 were delivered. There were well over 150 of the 160, 160A.

HAWLEY: Between the two of them I think there were over 300.

NEIL: There had to that many, they seemed to proliferate like rabbits. It seemed like any place you'd go in the country in about 1964 or 1965 you would run across a 160 in somebody's back room. What technology did we own? Did we originate new ideas in the building of 1604 and 160A; obviously not tape units.

PAGELKOPF: How about Germanium transistors? That was the key wasn't it?

HAWLEY: The drift transistor.

PAGELKOPF: And then the Mesa.

HAWLEY: That was quite a bit later.

TOTH: That was not Control Data technology.

PAGELKOPF: That was the key that unlocked them.

NEIL: Was anybody else using the technology, were we able to exploit it?

TOTH: I really don't think we were ahead of anybody else.

MOE: Dolan, is there any truth to the folklore that Univac was actually developing a machine very much like the 1604 under Seymour or with Mullaney and Seymour? But they chose not to market it?

ROWAN: I think he's talking about NTDS.

TOTH: I don't know if there's a straightforward answer to that. Seymour had developed over at Univac the NTDS computer, the military number of which I do not recall. But it was not all that similar to the 1604, it was a 30-bit computer, the 1604 turned out to be 48-bit.

NEIL: I guess the folklore I heard was that the general overall political climate at Univac at the time was responsible. Seymour and whoever else working on the NTDS had come to some conclusions about some machine design and that there were, whether they were Seymour's or somebody else's, version, proposals for a 48-bit machine for Univac which was turned down, not necessarily the 1604.

HAWLEY: I was working on a project at Univac which was to redo the 1103 out of a switch core logic similer to what the BOGART machine had been built out of. Pete Zimmer, Howard Sheckels, and Chuck Cooper and Bob Schmidt were in on that.

TOTH: Do you remember the name of that project?

HAWLEY: It was called the X301. We got fairly well along in that project. It would probably not have been a successful one because it used the switch core technology but things got cancelled pretty much because Philadelphia had been working on be LARC for Livermore and they were sucking up all the money and so everybody's budget was getting slashed and there was nothing left for St. Paul.

NEIL: As an outsider to the company at the time I heard that the LARC was responsible for the creation of CDC.

HAWLEY: LARC was going so far in the hole that it soaked up all R&D dollars and there wasn't anything left over to start anything new. In fact all budgets were slashed in St. Paul.

ROWAN: They just retained military products.

HAWLEY: Any commercial work was pretty well cut back.

PAVLOV: Along the line of Folklore I have heard, since you brought it up, concerning the naming of the 1604, I'd like to explore that with you. Is it true that the name 1604 was derived from 501 Park Avenue being added to Univac 1103? {Laughter}

HAWLEY: It was quite popular at the time that this was the origin of 1604.

TOTH: We've never been able to substantiate it. However, there's still lots of people who believe it.

HAWLEY: The official word promulgated by Mullaney, Crey and Thornton was that 1604 stood for 16K central memory and 4 tape units. That was the original design goal for the 1604. And then the 1604 went to a 2-bank memory so it ended up with 32K, but it started out as a 16K machine.

ROWAN: One other thing, the 1604, you asked about why it was designed. It did at that time compete very favorably with IBM's big seller, the 7090. In fact, it gave a lower price and higher performance and that's why it did start selling so well and it did cut into IBM's 7090 business.

NEIL: How is it that we managed to build a machine with higher performance and lower price than the 7090 which was a very tight evolution from 701, 704 and 709?

ROWAN: There were a lot of vacuum tubes in 7090 yet as far as I know.

NEIL: No. The 7090 was all transistors with one exception. I believe the first core drive was delivered with tubes.

MOE: You'd be amazed what you can do with less than 100 people. (Laughter)

HAWLEY: I think the low overhead was the reason. We did not have a lot of overhead. I can tell a story about logic design, or two stories, on the 1604. After we got the main part of the machine running, in other words, could reference memory and read instructions, etc., Jim hadn't finished designing the arithmetic yet so he and Les Davis would sit there and they would go through a sequence and they would plug in a couple of cards and see what time the pulses came out and add a couple of more. If it worked then they would write that down. That is how they designed the multiply and divide sequences.

NEIL: Now you can understand why Jim wasn't too upset 4 years ago when we told him the condition of STAR.

HAWLEY: Another story about Seymour and the design of the console typewriter logic said that he'd come in and work at night and he'd do his own wiring and everything and the next day Ed Reagan would come in and trace through the wires and write down what he had done. Talk about documentation!

NEIL: I'd like to get back to technology for a minute, to take some of the components of the machine. The memory technology came from where? Was it literally stuff that we borrowed from our Univac experience or did we do anything innovative?

HAWLEY: I think Bob Kesl was largely responsible for memory technology and I think it was pretty much new design. It was pushing technology--4 wire 50 mil core.

NEIL: What other parts of the machine do you consider innovative in terms of design?

TOTH: To me the most innovative thing about the machine was the fact that it had a split memory cycle which I had not been familiar with on any other machine that I had seen before. If the addresses worked out right, you could get two memory references in one cycle. I'm sorry I used the wrong word, I meant banking. There were two phases instead of a larger number, however.

NEIL: I can't think of, at that time, any machine that had "banking" or phasing".

TOTH: I don't know of any.

NEIL: That was two banks, right, odd and even addressed.

TOTH: It was quite possible that there would be a bank conflict; then you would wait out the time, but on the average it picked up at least 33%.

NEIL: Anything else that was innovative in 1604?

MOE: It had a 6 microseconds or 12 microseconds cycle?

HAWLEY: 6.4 microseconds.

PAGELKOPF: There's a couple of things we learned from the 1604 and one of them was that when you interrupted a machine you always had to wind up storing off all the, what did we call them, B boxes? When you returned to the program you had to restore those and we quickly saw that rather than program that we could make that an automatic function and put it in the hardware and call it an exchange package which was then implemented on the 6600 and again on the following machines. The other thing the 1604 had that I remember is a 48-bit wide data channel. It was channel 7. What was the name of that channel? It was never used.

BHEND: Wasn't it just called high speed channel?

KORT: High speed channel, yes, that's right. (Laughter)

PAGELKOPF: It was just another one of these wide band channels that someone was going to use sometime that I never did see wind up with much application.

HAWLEY: The only use I ever knew of was that the Annelex printer was connected on that channel and you sent out line by line the print image. In other words, you sent out around 120 bits which was 2 ½ words or something like that. You got an interrupt from the printer which meant that all the A's are coming by now and you've got to quick send out all the bits to drive all the hammers for the A's.

NEIL: There was a use made of that channel. It was hooked to an analog computer made by Raytheon Corporation and there were two systems installed with that and that 48-bit channel was necessary for that damn thing. They made themselves a digital analog system and one of those was installed at SDC for a while. It was a lot of hardware, in fact, the analog was bigger then the 1604 to my recollection.

HAWLEY: The 1604 for the satellite center too used the channel 7.

NEIL: If somebody were to ask about architectural concepts of 1604 by the varied definitions of architecture what would you say is the major architectural concept of 1604.

HAWLEY: I don't really think there were any new architectural concepts in the 1604. It was conventional architecture.

TOTH: It had a large word length for that time frame.

HAWLEY: But that's not architecture.

TOTH: I consider that architecture.

NEIL: Ok, you had B boxes as Univac had B boxes.

HAWLEY: Univac did not.

PAGELKOPF: They used memory locations for index registers.

HAWLEY: Univac at that time did not have index registers in hardware.

TOTH: The military computer did.

HAWLEY: The 1103 did not have.

NEIL: So there wasn't really any architectural innovation. The instruction set was conventional. The only thing we really did was improve on good use of design.

HAWLEY: There were some design innovations, such as phased memory.

TOTH: Circuit design. There was a circuit that I couldn't begin to draw any more but which had feedback causing the transistor never to saturate and therefore made it very fast.

NEIL: Was that Seymour's design?

TOTH: I don't know if that was Seymour's or Bob Kesl's.

NEIL: What did that thing cost?

HAWLEY: It sold for about a million bucks.

MOE: Project cost, I think I remember, ran about a quarter of a million dollar range. The hardware system cost.

NEIL: How about software?

MOE: We never said there was such a thing.

HAWLEY: We had the COOP monitor.

NEIL: When you shipped that first one what did you have, a loader?

HAWLEY: It had my little program that played music.

MOE: There were a few diagnostics. What we did for system test prior to shipment is run a bunch of programs, FORTRAN type programs.

PAGELKOPF: GFAB and GARFAB.

MOE: A whole bunch of bullshit that you'd write in that punched a tape and give it lots of FORTRAN type statements for COPY and other I/O.

NEIL: Who wrote the FORTRAN compiler?

MOE: Seymour wrote it.

NEIL: In fact it sounds like the very same compiler that has survived to today on the 6600.

HAWLEY: There was a pseudo-assembly language called MAP that Seymour had written that allowed you to address blocks of data by a name and a relative offset within the block.

MOE: All diagnostics were written in machine language.

PAGELKOPF: Probably another feature worth talking about arose when I went to 1604 school probably about when serial 15 was on the floor, which meant that several of them were already shipped. So I didn't go to school with you, Bob. You remember that was about six months later. Several machines were already out the door. Jim Benson was the instructor and the class found, I guess in the period of 8 weeks, we must have found no less than 15 logic design errors. When we'd find them in class, going through the sequences Jim Benson then would go upstairs to one of the machines and about 1 out of 3 of the errors would have been caught and changed, but about 2 out of 3 of them would really be bugs. What this meant was that somebody else must have found them too and fixed them but they didn't tell anybody else about it. So you can see how some of our people were trained in documentation. I'm hitting on that again as you can see. {Laughter} That is just the way it was. A lot of the ECOs were written on the back of old lunch bags and carried around to the next guy's room to him. And you'd say "Here's one I pulled out today, Charlie." We got the job done but we had some problems.

HUTSON: Would Univac have a good reason for not building that machine? I'm not saying they did have a chance. Was it strictly money or was it too close to one of their product lines already?

HAWLEY: No, I think it was budget problems.

TOTH: Univac, furthermore, had a philosophy unannounced perhaps, of not building computers to inventory. They would be glad to build a computer to order but were very reluctant to build computers to inventory.

NEIL: And Control Data was willing to do this. Did we build more than one 1604 right off the bat?

TOTH: We weren't terribly interested in building for inventory either but Seymour felt it was better than sitting on his hands waiting for a contract to come in. So the guys were out borrowing transistors from various vendors to get prototypes built up.

KRUEGER: Were they going to give them back or what?

TOTH: No, sometimes they were samples.

NEIL: Now I know how Seymour got the idea of borrowing Fairchild chips for CRAY-1. Was there anything else in the 1604 that might be relevant what every body might be interested in.

TOTH: It might be interesting to state that the company had no desire to get into software at all. In fact, each announcement was that we would not get involved in software, we would only build hardware. Let the customer do his own software. It didn't take many months before that got turned around.

NEIL: Part of the problem probably was shortly after the first 7090 was delivered IBM began to sock people with IBSYS which was really the first grandiose operating system.

HAWLEY: I think the users group which was called COOP was the instigator of all the software.

NEIL: You mean as far as forcing you into software? Anything else on the 1604, any questions? We are all trying to uncover primordial history here. What was next?

TOTH: In that same regard there was another machine built called the 1604G, G standing for government. It had 13 bits, some kind of crazy abortion when it came out.

PAGELKOPF: Parity bit?

HAWLEY: No. They had to have an extra bit which had extra op codes which could use that.

NEIL: That has to be the offshoot of the Fieldata computers. It sounds like the same thing they did with the 7090. They made a 37-bit word that added strange features to the hardware.

HAWLEY: Chuck Pallace was the lead man in that.

ROWAN: The 160G was built specifically because they got a contract with NASA, at the Cape. To satisfy the requirements they just modified the 160A to fit those requirements and it became the 160G and they packaged it entirely different.

KROHN: There was actually a switch on the console that allowed you to work in A mode or G mode. In G mode you had an extended instruction set and more indexing for memory.

PAGELKOPF: Didn't that go in the submarine too?

MOE: No.

KROHN: It was used for checkout of the Apollo spacecraft.

TOTH: You're thinking of the Polaris computer.

HAWLEY: That's one we didn't have on the list. The Polaris.

KROHN: It was one of the first attempts at having many channels in the memory for I/O. They had a memory box that had 8 channels and an I/O.

NEIL: The 160G you mean?

KROHN: Yes. It had three separate boxes, one for computer, one for memory, one for I/O.

NEIL: The 160A had a buffer. What was the difference between that and the buffer channel?

KROHN: It had a buffer channel and a normal channel too.

HAWLEY: There were extra instructions that you could get with this extra bit.

KROHN: You could tie two computers together if you wanted to. It had a cable so you could tie two compute modules together and have a transfer between memories. In case one crashed you'd still have a backup.

NEIL: Was it any good?

KROHN: Made a profit.

NEIL: It was good.

MOE: I think another thing happened on the 1604 which we apparently might have missed. Long after Seymour quit working on it and started on other machines they added the 77 instruction which doubled the God Damn repertoire and started all the fancy things over again. I don't remember what the 77 did but it was weird.

NEIL: That wasn't the "perform algorithm" instruction was it?

CURT: The sidewise ADD/Multiply.

HAWLEY: It got the pop count and bit merge too.

PAGELKOPF: That was my serial 30.

TOTH: Bob, was that the machine we built for IDA {Institute for Defense Analysis}? For Princeton?

PAGELKOPF: No, that one didn't go there. It went to NSA. Princeton thought up the 77 order.

HAWLEY: They had the customer engineers design it out in the field. They bought spare parts and had the customer engineers do the logic design, incorporate it into the machine and when they got it all done they said why don't you make this standard.

KROHN: The 160G was really a 3000 class machine. It used 3000 cards

NEIL: So it wasn't a 160A. Looks like it was on its way to being a PPU.

KORT: You know there was a 1604A too.

NEIL: What was the 1604A?

KORT: It had asynchronous channels. Some interrupt channels were included.

NEIL: Now what were the requirements for that? Did we have a customer that wanted it or did we just think it would be a "nice thing".

KORT: That was the OPCON computer.

BHEND: There was a 1604B also.

NEIL: Sometimes a machine gets built because you sit down and you say here's a customer market and then you have to build and sell it. Sometimes we've had to upgrade our products in order to hang in there with the competition. So in my mind the 1604A could have come along either because the government says we'll buy it if you paint it green or because we needed to upgrade.

HAWLEY: The 1604A wasn't really a new computer, it was really an ECO on the 1604.

BHEND: That was the same as the 1604B.

HAWLEY: That was an upgrade in the I/O rates.

NEIL: It was built then as a specific requirement and not so we could stay in competition. It had to meet the requirement of some contractor who had told us they needed it.

HAWLEY: I think it came out of the OPCON project.

NEIL: Ok, now were you still selling 1604s after this time? And therefore, we didn't sell the 1604B as a standard product?

HAWLEY: I think we switched over to the 1604B in order not to make two kinds.

NEIL: So the 1604 we sold after that would have been the 1604B.

PAGEL: Yes. It had a different kind of transistor in it even.

HAWLEY: I don't think that is related to the 1604B development though it came about at the same time. When they went from the DARLINGTON "DRIFTS" to the "SINGLE MESA". The Polaris computer, to fill in where it is, kind of overlapped the 1604 a little bit. It was built to go on a submarine so it was built on narrow chassis so it would fit through hatches and it was basically a 24-bit 1604 without floating point.

NEIL: That means it had pretty much the 1604 instruction set.

HAWLEY: Without floating point and 24 bits. The 924 was an offshoot of the Polaris thing to make a commercial model of that.

NEIL: How reliable was that Polaris thing?

HAWLEY: Very.

NEIL: How big was it actually?

TOTH: As I recall, it was as long as that wall, ten feet maybe.

NEIL: To go on from the 1604 I'm hoping that we can get to the point where we get more direct knowledge rather than heresay about how things came about. What happened after 1604 and Polaris?

HAWLEY: Ok, we thought about making an equivalent commercial version of the Polaris machine which was a complete redesign of it. We sold this to Lockheed for their satellite network. We got the contract for that, built the machine and delivered it and it turned out they changed their mind and so it kind of crawled along and finally later I guess we sold about 20 or 30 of them, and a big chunk later on to the Navy.

NEIL: What was that called?

MOE: The 924

NEIL: Why did they change their mind, did they want a 7090?

HAWLEY: I think that they didn't have people to program it or something like that, or a shift in emphasis on what they were doing. We got the contract in January of 61' and we delivered by Labor day of that same year. We got it out there and we went through acceptance and they accepted it and as far as I know it sat there and was never run.

TOTH: It's almost unfair to say that we sold it. They came in and begged us to build it. We really didn't want to do it. We weren't really awfully anxious to build it but they were so anxious to get it we agreed to do it.

NEIL: Why couldn't you have just sold them the Polaris?

HAWLEY: The cost and other factors. They didn't have a submarine out there at Lockheed.

NEIL: So after the 924 then what? By this time we've always made

money on these machines, I assume?

TOTH: For the 160, 1604, I think that's true.

HAWLEY: The salesmen were all complaining that 1604 was running out of steam.

NEIL: This was in what year?

HAWLEY: 1961.

TOTH: By this time LARC had been delivered to Livermore and was semioperational.

MOE: The design of the 6600 started in fall of 1960 and in fact terminated in the summer of 1961 because the circuit chosen would not meet the specifications they wanted, speedwise. At that time Fairchild came in with a transistor that supposedly would so they continued, starting over with that technology in 1961.

ROWAN: The circuit that was rejected for the 6600 was the one used to design the 3600.

MOE: That's right, although slowed down.

NEIL: So the design of the 6600 commenced before the 3600?

MOE: Definitely.

NEIL: So if the circuit had worked in the 6600 we may not ever have built the 3600?

HAWLEY: That's right.

NEIL: The 3600 was gone ahead with, why?

HAWLEY: Seymour at that time was all alone and they didn't know when they were going to get anything and we needed something to sell. They needed something specifically to follow on after the 1604 because this is what the salesmen wanted to sell. In other words they wanted an improved 1604.

NEIL: And did they tell you what improvements they wanted?

HAWLEY: Yes. Chuck Cassel and I and Max Goldberg and Ken Thiede sat down and went over all kinds of letters and stuff that we'd gotten from salesmen and from the programming group in Sunnyvale (from Clair Miller)with all the crazy things that they wanted and the four of us hacked that down into what we felt could be built and came up with the 3600.

NEIL: Ok, and all of this came about because Seymour is in deep gloom. When did they move to Chippewa?

MOE: July of 1962. He had started over again with the new circuit.

TOTH: Perhaps it was an experiment on his part at this time. There was no assurance that the project would be successful and there was no customer at this point either. I recall going to one, what did he call those meetings, planning meetings . It was small, not larger than this group here, but that included all the wheels in the company and Seymour was asked how much of the equipment he had should be considered inventory and how much of it should be called scrap. He says well since we have about a 50-50 chance, I figure, of making it we'll call half of it scrap and all the other half inventory. That way we can only be half wrong.

NEIL: Had he exposed anybody to what his design idea was? It was obviously not compatible with the 1604.

MOE: His criteria were a 1 microsecond memory and 100 nanosecond minor cycle in 1960 and at that time our 1604 wasn't that bad with a memory at 6 microseconds and a 200 nanosecond clock.

HAWLEY: There wasn't a minor cycle. There was asynchronous logic.

PAGELKOPF: That was a register to register transfer in 200 nanoseconds.

KORT: Back in 1960 and 1961 was he already working with the idea of 10 PPU's?

HAWLEY: The group was pretty well closed for anybody finding out what was going on.

NEI: Who was in the group?

MOE: At that time it was Seymour and Jim and Les and Paul Christianson.

KORT: So that when you say he was talking about a 100 nanosecond minor cycle in 1960.

MOE: That was his design requirement.

KORT: If he was talking about minor cycles in 1960 he must have been talking about that kind of an architecture.

MOE: He wasn't going with asynchronous logic, he was going with synchronous logic.

HAWLEY: He was going with asynchronous logic?

TOTH: Oh, I guess I don't understand.

NEIL: I don't either.

HAWLEY: He wasn't having a clock at each stage of logic. He'd start off and later sample the output.

MOE: You're right but he was certainly thinking of minor cycles as opposed to clocks.

MOE: As opposed to sequences and things like that. He wrote a paper of 1957 describing that.

NEIL: Seymour wrote a paper?

MOE: Yes, for IEEE, a computer conference or something. Basically describing the architecture of the 6600. That's the earliest documented evidence I've heard of.

NEIL: That's before he even satellited the 160A on 1604.

MOE: That's before even Control Data.

NEIL: Seymour has always made such strong representations about his anti-professional society thing.

ROWAN: He has written s few things though.

MOE: Certainly in 1960 what he was trying to do was markedly different from anything he had done before. That's true, it was pretty wild. His speed was pretty fast, the parallelism was the first approach to that as far as he was concerned.

NEIL: In other words in 1960 he is looking at a machine not only in PPU's but also has multiple functional units and this high speed circuitry. And a lot of memory?

MOE: At that time, yes! But you'll have to ask Jim about that but I'm sure that the memory was one thing, high speed parallel function units another. I'm not sure when the PPU concept developed.

NEIL: Now who at this time made the decision to then proceed with 3600? Was it based on not only Seymour's evaluation but also on the high risk nature of what Seymour was doing?

HAWLEY: I think that's true. I remember Casell and I and Dolan working on Bob Kish who was in charge of the computer division at the time to get the permission to go ahead with the 3600.

NEIL: So then it became essentially a parallel effort at this point. Was there anything in communication back and forth between the projects?

TOTH: Very little, actually.

ROWAN: The 3600 was a much shorter term project.

NEIL: Well, did it look like it at the time?

MOE: Yes, because he was starting with the circuits that had been designed and they could slow them down and make sure they were reliable.

HAWLEY: And also we were building on the 1604 architecture. We weren't building something new.

KORT: Where did the memory come from?

HAWLEY: The memory was a new technology. It was developed by memory our group and in conjunction with "F-ERROXCUBE" and it was a one half micro-second memory which was something new for us. The memory was all in one chunk which we bought from the Vendor. It wasn't spread out with a chassis having 6 bits and another chassis having 6 bits. It was a 16K memory stack.

NEIL: Ok, now we have first hand people here with 3600 background. You essentially based your design on the 1604. What technological innovations were there?

HAWLEY: There were some architectural changes from the 1604. We split the memory off into a separate box with an interface. There was a multiple port entry to the memory with its own scanner and access control, in each memory bank. The I/O was separate, almost programmed I/O channels that could run short I/O programs. So this was an architectural change from the 1604.

NEIL: Now that sounds like the 7090 data channel. Was that stolen from them or was that your own invention?

HAWLEY: Somewhat.

NEIL: But it's not identical. The concept is there but you did it considerably different than they did. Anybody here work on that? Anybody know why he made the decision to go that way?

HAWLEY: I did it.

NEIL: Why didn't you do it the way IBM did on the 7090?

HAWLEY: Well the thing we were trying to do was get modularity and we made the channels and channel controller and memory such that we had very rigidly defined interfaces so we could connect these together. We could configure multiple computer systems with multiple I/O channels, etc., just by plugging in the cables.

TOTH: There is also another reason. That is, at the time we did not have a cross-licensing agreement with IBM.

SPECKER: That's really a pretty significant architectural departure.

CURT: That's right, that's where we started down the road to having a box with a computer in it and boxes with memory and could have 3 or 4 or 2 or 1, or boxes with I/O units and you could have optional channel.

HAWLEY: The I/O did not go through the CPU? I/O had its own separate path to memory and the only thing between the I/O and CPU was the control the initiation lines and function lines.

NEIL: The reason for this was you were trying to achieve modularity?

HAWLEY: Yes.

NEIL: Who thought up this idea?

HAWLEY: I did.

SPECKER: Did you sell any multiple main frame 3600?

HAWLEY: We did.

NEIL: Very early in the development?

TOTH: Yes, we put one down at the "CAPE .

HAWLEY: At the Eastern test range.

MOE: When was the first delivery on the 3600?

HAWLEY: It was in June of 1963, to Livermore. It was supposed to go to Michigan State. But Livermore said sorry Michigan State, we want that one, hence, our priority which made Michigan State very unhappy.

NEIL: Did they ever get one?

TOTH: I think they did.

NEIL: Now when you set out to build the 3600 you established what? For example, Seymour had established in 1960 certain ground rules, the 100 ns clock with a one microsecond memory. Had you established some kind of performance levels as your objective that you were trying to achieve? What were they?

HAWLEY: We had settled on the 1.5 microsecond memory and we had set the master clock rate because of what the circuits would be - 62.5 microsecond.

NEIL: Where would this "peg" the 3600 in comparison with the 1604 performance?

PAGELKOPF: Over 3 to 1 just based on the clock.

HAWLEY: I think it was about 4 to 1. The magic number was 4.

NEIL: How about cost?

HAWLEY: I remembered Dolan screamed when we found out when we got done that we had the $850,000 in inventory.

TOTH: The worst part was we couldn't account for where it went.

KROHN: How big were these machines? How many gates or inverters did they have? Say 3 gates on a card?

HAWLEY: 2 gates.

KROHN: 40 card slots.

KORT: Some cards only had one. There were about 6000 cards total.

NEIL: Now when you started the 3600 with the idea of the separate I/O channels, had the idea of satelliting (more integral satelliting) come along, that is selling 3600 systems with 160A front ends.

HAWLEY: Oh yes, that was done commonly on 1604's by then. I was automatically supposed that we would on the 3600.

KORT: It was done different on 3600. It was connected through a separate box.

HAWLEY: At that time we built a satellite coupler which allowed direct connection between machines not through the tape unit as on the 1604.

PAGELKOPF: The boxes were called 3681, 3682, and what was the other one?

TOTH: That was the first machine we built where we attempted to build an entire system from the ground up with a computer and all peripheral controllers and peripheral devices associated with it. Seymour did not want to get involved with that so he built couplers (6681's) that coupled with a lot of that stuff.

PAGEL: That was the data channel converter.

NEIL: You mean the 6000 to 3000 interface?

HAWLEY: Seymour didn't do that. Seymour on the 6600's had direct driven peripherals from PPu's.

NEIL: The Original 6000 tape units were like that, I know.

HAWLEY: The card reader and punch and everything else was all the same way.

HUTSON: It didn't last very long though.

NEIL: Wait a minute, the one inch tape drives had a controller on them but they were a strange type of controller.

MOE: The controller was in the chassis on the mainframe.

HAWLEY: When Don and Jim started rolling on the 6400 they built up these channel converters and started using standard peripherals rather than the special peripherals.

SPECKER: That was just an economic thing. They didn't want two different lines of peripherals.

NEIL: The first 6600 we delivered, was delivered with the 6000 mag-tape units which were horribly unreliable to my recollection. In fact, CERN had a bunch of them.

MOE: We delivered 12 - 24 tape units with most of the first ones, half ½ inch and ½ one inch. We had the disk files, the Control Data card reader, Control Data card punch.

NEIL: The first delivery ofthe 3600 was when?

MOE: In June of 1963.

PAGELHOPF: 606 tape units came from Control Data too.

NEIL: So we are building our own tape units then. How long was that project? You started actually doing the work on the 3600 in 1960?

MOE: 1961.

NEIL: I mean the initial work you were talking about you know, in terms of deciding on architecture?

TOTH: It couldn't have been before 1961.

HAWLEY: After I got back from installing the 924, Chuck Cassell and I started working on it.

NEIL: So two years elapsed then you're talking about from the time you started working on the idea and when the finished product was delivered with the peripherals on it. The decision at that time was to build several or was CDC still building to order?

HAWLEY: We had several orders by the time we delivered that first one but there was that case of Livermore pulling their priority in order to get the first one which had been sold to Michigan State.

KROHN: Did Livermore have a 1604?

MOE: Yes.

NEIL: I don't remember that. Where the hell was that one located? The 1604?

HAWLEY: I don't know, but the 3600 was in a trailer.

NEIL: Yes, I know where the 3600 was but I had never seen 1604 at Livermore.

MOE: Probably near the "STRETCH" or the LARC.

NEIL: No, the LARC and STRETCH filled the entire real estate and the rest were 7090's.

PAGELHOPF: I never saw it either and I was on tour there once.

NEIL: I was wondering what motivated them to be so high fallutin' about getting the 3600.

HAWLEY: They were running out of compute power.

NEIL: Well, see that means the 3600 was damn competitive with 7094.

TOTH: Yes, that was the market it was supposed to hit.

HAWLEY: It would beat the "STRETCH".

NEIL: For a lot less money and more reliability. So the motivation for the 3600 was that it was to meet the demand for replacement of the 1604 and I presume a sales force pressure.

HAWLEY: The 7094 was the competition.

NEIL: So the architectural changes are obvious except for the ones involving modularity. So why modularize? Just so you could announce it as a new "buzz" word? Did that seem like a good idea? Why?

HAWLEY: For one thing, it was a way to break the design up so that we could build it easier and also the requirements were for more memory and this was one way of having optionally different types of memory. You could just plug in extra modules.

NEIL: So what was the range of memory you could have?

HAWLEY: It could go from 32K up to 8 times that. I don't think we ever sold any with 8 cabinets of memory.

PAGEL: 3 is the most I remember.

HAWLEY: I think we've gone to 5.

HUTSON: The 6000 was probably a salesman type thing to get your foot in the door with a stripped down model and you could always sell more after they proved they could do great things if they got more memory. In the 6000 they had a 16K version which I don't think they ever sold to anybody, that I remember ever getting in the door.

NEIL: I don't remember anybody ever quoting 16K-6000 in the beginning, 65K was the smallest configuration.

HUTSON: Well, there was no such thing to begin with as a ½ or ¼ memory.

NEIL: So the 3600 then evolved through 1963. Seymour was working on his thing. He moved to Chippewa in 1962 and when did we deliver the first 6600?

MOE: September of 1964.

NEIL: Seymour wasn't too far behind as far as developing the 6600 in parallel with the 3600.

HAWLEY: A period of about a year and three months, 15 months.

NEIL: So what were the relative prices for the 3600 and 6600?

HAWLEY: I think 3600 sold for a million and a half.

SPECKER: Systems were like 3 million versus 7 million.

NEIL: The first 6000 systems sold for 7 million?

GRINNA: A little less than that - about 6 million.

NEIL: That's with 65K memory?

MOE: I really doubt that. Seymour had no intention of making small memory systems.

GRINNA: We never had a 65K system, S/N #5 was designed with 65K but before it went out the customer changed their mind.

NEIL: I have a question that I should put off until later, but if I don't ask it now I'll forget it. When I first heard about the 6600 they were quoting 300K memories. Why were they never able to build the 300K memory?

MOE: The addressing never allowed for 300K.

HAWLEY: I've never heard that.

NEIL: If I don't have a brochure that states it, I'm sure some salesman stated that while making a presentation to the government.

TOTH: He just slipped a decimal point.

HUTSON: They only had 18 bit registers for addressing. You couldn't change that.

GRINNA: This I think, was typical of the 66, 76, 86. We'd go out to Livermore and make a presentation and they would say - that's fine except we gotta have more memory than that. The salesmen would come back and ask and, yes, it could be expanded or we could do this and this and this to the machine. The salemen would then go out and tell the customer it could be done - someday. But it would be a major rework that no one ever started.

NEIL: That's what I'm trying to tell you, that's where the confusion came in. 18 bits was enough to address 300K memories.

PAGEL: Yes, but there was a sign bit there that would get all tangled up in all the address arithmetic.

NEIL: That's probably it. But you know I remember these guys saying we have got an 18 bit address in there and thus have 300K memory.

MOE: Ok, they didn't understand. No, I never heard anything talked about anything but 131K memory.

NEIL: So what happened after the 3600 was delivered? They went right on to the 3800? You went on to the 3800 after you finished the 3600?

PAGELKOPF: 3400 was next.

HAWLEY: 3400 came next, that's right.

TOTH: Well, simultaneously with the development of the 3600 was the 3200 for a low-end 32 bit machine.

NEIL: Oh, now we're talking about product lines.

PAGELKOPF: There wasn't any simultaneous 3200 project really, the 3400 was more in parallel than that.

HAWLEY: 3200 ran on and on and on.

NEIL: You mean, it was a lower end machine but it took longer to build?

TOTH: There were several competing proposals for computers that were exact. I don't remember what they all were.

MARIS: I believe Ray Allard and Mike Schumacker worked on those.

HAWLEY: The one that ended up was the "Z", which stood for Peter Zinner.

MARIS: Yes, but that was a code name before they knew 3200 wasn't assigned yet.

HAWLEY: There was a "Z" and a couple of other letters that were competing and the "Z" won out and became a machine.

NEIL: Now wait a minute, I have to understand something. How did the organization, first of all, stack up? When this whole thing started

we had Seymour and Jim and a little tiny grop of people designing a computer called the 1604, right? So I got a little box with a bunch of computer designers in it - architects and designers. SCray & Co. This small organization does the architecting and designing of the 1604 but immediately you've got a need for all those other goodies here. The 924, the Polaris and obviously SCray & Co., didn't do any of that right?

MOE: None of that.

NEIL: So another group forms over here consisting of government systems for GSD and who is in that box? Who's the architect designing the military stuff?

KROHN: Doug Johnson and Chuck Pallace.

NEIL: Now we get to 1960 and Seymour decides he's going to build the 6600, and you have the 3600 group over here.

MOE: No, now in 1960 you have Seymour over there and then 501 PARK Design Engineering Group which was making 1604's and doing everything else and then the government systems group over in STRUTWARE. I think it's better if you just draw buildings. (laughter)

TOTH: That's what he means by architecture.

HAWLEY: We've got another couple of groups in here that are building up in parallel with the 1604 and a group that's building a machine called Clippin, which is basically a 1604 plus a couple of other boxes about the same size as the 1604 with classified stuff in them.

NEIL: For NSA?

HAWLEY: Yes, for NSA. Serial 2 1604 was part of ClipPin, but there's a lot of extra hardware, so a design group was built up and doing that. Later on there's this OPCON contract for the Navy that builds up a design group.

NEIL: Did they design computers or did they design systems?

HAWLEY: Systems. Well, there's also special hardware boxes and stuff to bé designed so there's different groups building up with people that have expertise in design.

MOE: Now then it's blank for six years for me 'cause I don't know what happened except what year.

NEIL: What I'm getting at is you mentioned that now that you have the 3600 under way there was a 3200 and there were competing proposals, and what I'm trying to figure out is who in the hell, or where was the competition coming from? What was the origin of competing proposals? You had a market requirement set by sales people I assume that you needed a low end machine. Is that right? And you then had some designers, architects, engineers or pseudo architects and engineers who were proposing then alternatives for that machine. Where did they come from?

SPECKER: That was just kind of a room full of programmers, wasn't it?

HAWLEY: It came from the group who did the 160 and 160A.

MARIS: Don Malcom's group.

NEIL: He had built a cadre of people who were considered logic designers and so they were making a proposal for the 3200.

MARIS: Right.

NEIL: Who else was proposing designs for the 3200? You said there were competing proposals.

HAWLEY: I don't remember who.

NEIL: Obviously, Seymour Crey wasn't doing it, you guys weren't doing it.

MARIS: No, I think there were different instruction sets proposed that were incompatible with the 3600. Because this was now a 24 bit machine and not compatible with the 3600 in any way. We were hashing over different instruction sets and features to put into it.

HAWLEY: I think maybe then it was in that same group that there were competing proposals.

MARIS: The same group of guys were proposing alternatives.

NEIL: So basically then there's only one group doing the 3200 after all.

TOTH: We kept getting "help" from marketing. Tom Rowan was in a marketing group at the time. We kept getting help from him (and help in quotation marks). They would come in and demand certain things and we would quietly turn them away and go back to our work.

HAWLEY: At this time there was a department that was called Dolan's Zoo.

TOTH: I don't think it was called that.

HAWLEY: Because of all the crazy people he had working for him.

TOTH: Be careful - some of these guys are sitting here.

HAWLEY: It consisted of Chuck Casell, myself, Russ Van Krevelen, Lynn Gallup & Don Malcom who were the main animals in the zoo, and Sam Slais.

SPECKER: But I think it's significant that there was a recognition of a need for a product line, but there wasn't full recognition that all machines should be compatible.

TOTH: I think that's very significant.

NEIL: And they really didn't say there had to be. That this new beast could be anything.

SPECKER: They did make a midstream decision on the 3400 to slip it into compatibility.

NEIL: Compatable with what, the 3200?

HAWLEY: The 3400 was always compatible with the 3600.

PAGELKOPF: Well, to be truthful the 3400 was a bummer because it wasn't exactly compatible. There was enough changes to make it not compatible and there was a discussion about this many times. Even though I worked on the 3400 and naturally I like it cause it's got my blood in it, we should have never built it. We should, in fact, have put out a 150 nanosecond crystal and plugged into the 3600 and cleaned up the 3600 to kind of cheapen it a little bit and get our performance difference that way. Control Data would have made a heck of a lot more money.

HAWLEY: The idea behind the 3400 was that the salesmen wanted a cheap 3600.

TOTH: We always get sucked into that, and I guess we always will.

PAGELKOPF: Since we are looking for architectural ideas and we touched on this one, I'll touch on one other one. Marketing also said, at least through bosses, that we had to have an optional floating point unit, again a very bum decision. We should have made that optional floating point unit out of 2 wires. You know...delete 2 and add 2....to put the thing in or out, because from a total cost to the company we could have built it in with a 2 wire change cheaper than it was to add the receivers and transmitters and cables, the extra chassis, and whatnot to actually have an optional unit. I've gone down this argument several times since (against Manny Otis as a matter of fact) when we wanted a cheaper, slower 7600. We wanted cheaper, slower everything at one time or another. It used to be every Fall we would get into this argument. Well, if you want a cheap, slow machine we know how to fix it - just add two cards. We finally wised up architecture-wise and design-wise to make a 6200. We added 3 cards which put in a certain amount of clock delay on the exit of every instruction and EUREKA we've got a 6200. (laughter) Prints are real easy to make, you know, the diagnostics all run and so does the software. Plus you have the feature now that when the 6200 comes back in for retrofit you can have one little technician off in the corner converting them into the higher speed 6400's by merely changing those wires and knocking off the emblem and putting in a 4 instead of a 2.

NEIL: It's true. I believe the CYBER 70 project plans specifically got performance changes just by changing the master oscillator to change the performance of the machine and to call it different names.

KRUGER: The question is why do that? The obvious question is why make a cheaper and slower machine? Cheaper, I can see. But why do something ridiculous like delaying the RNI time?

HAWLEY: If you make it cheaper and it isn't slower you aren't going to sell any of the more expensive ones.

KRUGER: But that applies, Don, to what you said about the 6200. You made it slower but not cheaper.

PAGELKOPF: Oh yes, the price to the customer was less. Don't worry about cost.

HAWLEY: We're talking about prices.

PAGELKOPF: In addition, I'm also worried about long-term cost. Because no matter what kind of box you build you have to document it, you have to make diagnostics for it, etc., etc., etc. at a fantastic cost to this company not only in the year that you develop it but in the ensuing years you have to support it with continuation engineering. If nothing more the amount of parts you have in the combined bill of materials start eating away at you after awhile.

KRUGER: But is there an actual reason for slowing down a machine? What I'm thinking about is STAR. I could go into the microcode and add a 1 microsecond delay at the beginning of each vector instruction. Is that going to reduce the price?

HAWLEY: You have already. (laughter)

KRUGER: That's not my fault. Seriously, could something like that happen to the STAR 100?

PAGELKOPF: In about 2 years, we're not ready for it right now.

NEIL: Let's review the early 6000 days. They pegged the price for the 6600 based on what they expected to sell, the number they expect to sell. We got 6400 serial 1 in Palo Alto and I can't remember what the price was, it was pretty high but the price performance ratio was competitive. After they had been in the business awhile the 1108's came out and Univac started hitting us pretty hard with software and everything else and the Univac price/performance ratio looked pretty good.

Somebody said, well let's look at how much it cost us to build the remember what it was, but the cost and price didn't seem to be related. Do you know what it was selling for?

PAGELKOPF: A million.

NEIL: It was unbelievable but somebody said the thing to do or the options available to the salesmen were limited. Sometimes you can discount (like the salesmen can cut prices in some cases) but not for a GSA proposal. You don't do that. You have to go in with a base price of a machine as you quoted in the GSA catalog. So the argument was, ok, we'll go in and cut the price of the 6400, but you can't do that cause you already sold 30 at the old price to government customers and you would have to refund that price and all sorts of penalties. What do you do - you call it a 6200 and slow the clock down because we already absorbed the 6400 development costs, we can now sell it for half a million and still make a helluva good profit.

HAWLEY: The only time you can do that though is after you've gotten enough manufacturing experience to where the manufacturing cost has dropped down due to the learning curve.

NEIL: It's unfortunate in such a high risk business that you can't start out being able to predict that well. For example I think, somebody correct me, I recall that the expected number of 6600's, (hot prospects and everything else) was about 6 or 7 so they pegged their breakeven point at 6. The company priced them accordingly and look how many we've built now! Well, back to the 3400, 3200 and 3600.

HAWLEY: The 3200 started, I think, a little bit later than the 3600.

MARIS: The 3600 was in checkout already when the 3200 started.

TOTH: I question that.

HAWLEY: I think it was a little bit behind.

TOTH: It might have been a little bit behind but they were really parallel developments.

MARIS: I just remembered that there were circuit problems with the 3600 and we didn't have the design done on the 3200 yet. We went back and changed the prints but we didn't have to change the design.

TOTH: I also remember that it took longer to do the design of the 3200 than it did the 3600. I always attribute that to the fact that we had more designers on the 3200.

PAGEL: I would counter that by saying that it had that nasty 6 bit character garbage in it which always is a mess. It is nothing but end cases. That's my experience. Design is a snap except for the end cases. You spend 9/10ths of your time working on the end cases.

TOTH: But there was also a 6 bit character in the 3600.

PAGEL: No, there were no BCD adds or other operations, none of that stuff was in the 3600.

TOTH: Ok, but the 3200 didn't have BCD arithmetic to begin with.

PAGEL: Did it come later?

KORT: It was optional, along with floating point. The reason the 3200, I think, took long to design was that it was a new computer. The 3600 was still the design of the processor, it was still an extension of the 1604. You could go in there and see the 1604 inside of the 3600. It's all in there. But now you have a really new machine, a new architecture and you've got to in there and design and solve problems you haven't solved before.

NEIL: What was the new aspect of the architecture?

KORT: A new instruction set.

NEIL: Why, what was the rational for that? Does Ray Allard have to answer that question?

MOE: No, no, I doubt that. I think that in 1962 or 61 there was definitely a split in direction within Control Data Corporation probably directed mostly by competition as far as the instruction repertoires were concerned. Seymour just did not want to change his and wouldn't but others were and there was great pressure to have these exotic instruction repertoires. I think that was competition driven.

HAWLEY: I don't think there was an exotic repertoire as much as that the 3200 was supposed to be a business processing machine as opposed to a scientific computer.

MOE: Ah, that might mean the same thing as far as words are concerned.

HAWLEY: I think those are better words, and that they more clearly describe what the pressures were.

MARIS: Was it in any way driven by IBM 360 work at that time?

TOTH: There was some feedback from the marketing people on what they thought the new product line was going to be from IBM but it was very poor.

HAWLEY: We didn't know about the 360 in 1963.

NEIL: But you did have 1401, 1440 competition and the 7040, 7070 and 7080 were in the field in those days as BDP type machines, with exotic instruction sets when compared to the 7090. The 3200 then was specced from which the designers had to operate. Who did that?

HAWLEY: Malcolm,with some help from Marketing.

NEIL: At this point in time there wasn't an attempt to have compatability between the 32 and 34, and or, 3600?

HAWLEY: There were some of us working on the 3600 who would say, "Hey, why don't you remain compatible"- but we were ignored.

PAGEL: I had a feeling at that point in time that the 3600 and 3400 people might just as well be working for some company, ACME computer company with the 3200, 3300 team working for GOLD BOND computer company and in Chippewa they were working for some other outfit called Control Data for all practical purposes. Zero communication, in fact, it was negative.

NEIL: Why?

PAGEL: I don't know. Personal feuds and competition.

KORT: The 160 wasn't compatible with the 1604 so why should that bother you?

NEIL: Compatability doesn't concern me as much as the lack of communication. Again as an outsider looking at Univac Corporation, it just didn't make any sense back in 57 and 58 to see that God damn company with the small amount of computer design talent that existed in the world in those days, with designers out there and designers here who refused to use each others circuits or even read each others papers or even talk to each other about what they were doing. It didn't seem like it made any sense to use all those resources in that way. And I'm specifically going back to the Univac 1, the early years at Univac with the 1101 and the 1103. I'm not even talking about the logic design or instruction set, but basic things like why not use the same size chassis. I found out that the guys working on the LARC came from a BDP area, down in Philadelphia, and most of them had never even spoken to 1100 designers or hadn't even the foggiest notion what an 1100 looked like and here they were building a scientific computer - the LARC. I feel that's why the LARC really died.

TOTH:I'll tell you a worse story than that about LARC, I don't know if you know it but the St. Paul part of Univac put out a proposal for LARC and then Eckkart went out to Livermore and found out what we were proposing and how much and went back and proposed something on top of that which they accepted which became LARC. He put enough additional performance in it so that essentially he had to wait for the state of the transistor art to catch up with him before even meet that performance.

NEIL: Whose idea was the integral I/O processor rather than a separate sattelite?

TOTH: That was Philadelphia's idea. St. Paul had no input into it at all. It's a good idea, it's not bad. But in the process they lost somewhere between 20 or 30 million dollars which was at that time an astronomical amount.

NEIL: It was about 7 million dollars per machine for 3 machines.

TOTH: It was published policy of the corporation not to have different divisions of the corporation compete on bids for the same program.

NEIL: Ok, let's go back to Control Data. Was there any attempt at all to promote communication between here and Chippewa or the 3000, 3200 projects?

HAWLEY: From Chippewa, Seymour pretty much said leave us alone.

MOE: And from a decision standpoint, from a business standpoint I'm sure that Freon Cooling, very expensive complex modules and very different memory modules, made it obvious to anybody in Minneapolis that that expensive hardware would not permit you to make a low cost 3200, 3400 when everybody else was "air-cooled" and that kind of technology permitted you to get your chassis costs and those kinds of things down. That wasn't technology, that was obvious applicable for low cost products.

HAWLEY: Communications were pretty much cut off from the Chippewa end. Seymour said don't call us, we'll call you.

PAGELKOPF: Freon cooling and parity were the two big "hard spots" between Seymour and the rest of us.

TOTH: In spite of that, for example, if you went to Chippewa you could tour the place. If you wanted to know, needed to know, or had the right to know you could find out what was going on.

NEIL: How about between the 3200, 34, 36?

TOTH: That was a personality problem.

HAWLEY: That was open battle. The two sides were called Kasell and Malcolm.

TOTH: They were the two project engineers on the 3600 and 3200.

NEIL: So they had incompatible instruction sets.

TOTH: They had incompatible personalities too.

NEIL: How about compatibility for circuit design?

HAWLEY: They did use common circuits.

PURCELL: But they'd use different ground rules

NEIL: Ok there was no cooperation. How about documentation?

HAWLEY: They used the same mechanized design routine on the machines except I had a continual running fight with the people who were running them on the 3200 cause they wouldn't do them the way I told them they should cause I had written the programs. I was continually having to go into files and clean up garbage and straighten them out and they never would do it the way we had intended the programs to be run.

NEIL: It sounds like instead of attacking technology problems today we ought to be attacking the "WE/THEY" syndrome. We'd save more money for the company than by inventing a new circuit.

PAGEL: Isn't a certain amount of that natural, look at STAR 65 and STAR 100. There's a normal competition there.

NEIL: Oh, I think we got along pretty well.

TOTH: Look, we've got a plug-in module for the STAR 100 which is different from the one on the STAR 65. Ten years from now somebody will be sitting over in that chair saying didn't you guys communicate?

PAGEL: We did, but the STAR 100 people wouldn't listen. (laughter)

CURT: There was one-way communication - like shouting down a well.

NEIL: That was all a very valid argument because the question was always which of us was the dog and which was the tail and who was wagging what. But you know it changed back and forth. But to get ahead of our story, there was a time when the STAR 65 looked to some people that it was going to be the big seller, for price/performance reasons, thus STAR 100 was supposed to compromise on some issues. And then when it became obvious that the 65 was not going to be pre-eminent then the STAR 100 stock went up and the STAR 65 group had to compromise.

HAWLEY: Looking back at the 3200 3600 situation, I think one of the problems was that the 3200 group was essentially a small computer group trying to build a large computer. I think it's easier for a large computer designer to build a small computer rather than the other way around because I think they ran into a lot of pitfalls on the matter of size and design that caused them a lot of trouble, caused a lot of headaches.

MARIS: You mention pitfalls. I think they had a very unrealistic schedule set up.

TOTH: Now I want to get your impression of an impression I have, to see if it corresponds all with my own as the manager of the thing, it seems to me that there were too many people involved in the design and, therefore, the inter-communication between people slowed the development down. The interfaces between the devices, in other words, slowed down the development because it took so much time and effort to communicate those interfaces. I have the feeling that if we had had half the number of designers we would have cut those interfaces down by a factor of 2 and we would have got the design done sooner.

MARIS: I think that's probably true.

KENT: How many designers were there on the 3200?

TOTH: I don't remember exactly but it was in the neighborhood of 6 plus or minus 1, I would guess.

NEIL: How many designers were there for the 3600?

TOTH: There were about 3 or 4 and that included a lot of peripheral stuff.

NEIL: How many actual designers were working on the 6600?

MOE: About 3, Seymour did the PPU the ten PPU's, he did the memory redesign, and he did the disk.

NEIL: What do you mean the memory redesign?

MOE: Well, Paul Christianson started with the memory and started with the disks and Seymour redid both of those. Jim did the CPU, Les Davis did the multiply unit and implemented the design of that. Seymour then basically designed the I/O, he did the electronic design for the memory and he did the disk controller for the Bryant disks.

NEIL: That design began from the time they restarted with the new circuits in 1961, and they delivered in 1964.

MOE: Right. Ok, and they delivered in 1964. So really it was a three year design development.

HAWLEY: On the 3600 I did the memory and I/O channels. Sam Slais did the tape controller, Ron Hintz and Stan Aschenbrauner did the CPU. Sid Anderson did the card reader/punch.

SPECKER: The 3200 suffered from outgrowing that chassis so fast, I think. Understand that cost a lot of time since they kept trying to reduce the card count to fit it in.

MARIS: They had trouble fitting it in and they had problems with the interrupt system. They had to redo that. They had channel problems, they had to redo the channel, designing over and over again because initially I think it wasn't well defined. The instruction set kept growing.

TOTH: My point is that things don't get too well defined when there are too many people involved.

GRINNA: This goes back to our definition of architecture. It sounds like there wasn't a clear definition before the project was started.

NEIL: There wasn't a strong control over that.

HAWLEY: On the 3600 before we started design I had worked up and written down on paper what all of the interfaces were between different modules and got approval from Chuck Casell on that before we started on design.

MARIS: That was not done on the 3200 because instructions changed. Parts of the machine were designed, but instructions kept coming in, additional instructions, additional features.

HAWLEY: I do think one of the problems of the 3200 was that it wasn't separated into separate units with a clean interface.

TOTH: That's right.

HAWLEY: It was all in one box, in other words, you had a whole bunch of designers, all designing in the same box, in different sections of the same box and there wasn't any clear defined interface. There were always some extra wires going across.

NEIL: Sounds like the STAR STREAM unit, only we had 11 designers in that box!

CURT: Was there on the 3200 a head guy who understood the whole machine?

TOTH: Presumably, Don Malcom.

HAWLEY: He was too busy "managing".

NEIL: On the 3600, was there some person who really had the total technical comprehension of the project? Seymour obviously did on the 6600

PAGELKOPF: On the 3600 I do see one flaw on the way that program was run. I was working on the checkout floor and many times it would be like 11:00 at night and Bob Schmuck and I would be working together and we'd say, well everything works up to here, now what the hell's supposed to happen. So we'd get the prints and for two hours we try to figure out what was supposed to happen next. What I'm getting to is we didn't have written down for the checkout people a sequence. When you're in this instruction this was supposed to happen at time 200 this or that was supposed to happen at time 200. We had that from the 1604 days. That was the one big problem I saw with the 3600.

HAWLEY: You didn't have to start with on 1604. That came much later.

PAGELHOPF: It came as a result of the people in training who wrote it because there wasn't one.

HAWLEY: This is the disadvantage of having a small crew of designers, you get a real good design but the documenation lags.

NEIL: How about the documentation from Chippewa:

GRINNA: It lagged. (laughter)

HUTSON: It consisted of a stack of green books about this high. It had the transistors and the wire tabs in it...that was it. That was the entire machine, however.

MOE: The diagrams gave you all the wire tabs, if you knew the circuit you knew every God damn electrical component in it.

HAWLEY: There was no description of what's supposed to happen.

HUTSON: Once again it was the training classes that had projects of trying to draw sequence diagrams and block diagrams of the 6600. I'm sure that anything that exists now originated out of the first training classes.

KRUGER: Chuck, when you said what happens when you have a small group of designers is that really true or is it just the habits of the designers? I really wonder if that's because you have few designers as opposed to a few good designers. If you're going to do a design it can't be that difficult to write down what you're doing.

RESNICK: I don't know of any designer who operates like that.

SPECKER: Seymour did a little bit better on the 7600. He wrote that massive sequence document.

NEIL: I think that's how he conceived the machine.

MOE: No, no the document came after the design was complete.

MARIS: The really crucial thing was that Seymour was the key designer as well as the architect. He understood the whole thing and kept it under control. On the 3400 there was no such thing. Pieces were designed without control, engineers left and we would have to start all over again.

NEIL: Is there a fundamental question then about the concept of a "key designer" which is simply ifshould have a key designer who makes sure this all fits together. The 3200 did not. As Maris said this led to redesign. That was not true of the 3600, I assume. How about the 3400?

HAWLEY: Don did most of the design on that.

PAGEL: That particular machine had several advantages. To start with, Chuck had sketched out almost all of it. We had known proven circuits in the 3600 with experience on those circuits.

NEIL: How did you cheapen it?

PAGEL: We took out the clocks and put in the delay line. I think as a result of that, we only reduced the hardware by about 25% in the main frame but the one thing we got to do in the 3400 then was to take time out to write all the sequences before 1 piece of control logic was designed. Now we started off by knowing what the adder looked like, what the main registers looked like and sketched up the rest you know, so you knew exactly what you had to work with. Every operation that was going to take place was written down before the control design started. As a result we had one designer at any given moment, when Chuck finished what he was doing and turned it over to me, I became the only designer. I had working for me two customer engineers and a writer and the writer kept busy all the time. That really turned me on to the fact that you could give the checkout guy a printed copy of what supposed to happen on a prototype. Oh yes, we had our changes too, but it was a significant departure from checkout on the 3600 prototype.

NEIL: When did the first 3200 get delivered?

PAGEL: I don't know.

KORT: But there's one thing in what you say, Don, when you look at the 3400 and the 3600 the scheduling was different.

PAGEL: The first day I was on the job Hawley said we were 16 weeks behind, get going. That's what the PERT chart said.

KORT: Yes, but still the pressures are different, right? On an initial machine you're fighting different problems.

PAGEL: Well, Ray, I don't know. As far as I'm concerned before you design you've got to get organized and this stuff about "well I'm a designer now so I can't be bothered to document", I just never got along with that viewpoint.

KORT: I'm not arguing that point but that's the way it's always been. It seems like as soon as anybody wants to build something around here they want the hardware on the floor next week.

PAGEL: I think Dolan was touching on the right problem. You can't have too many people too early also you have to zero in on what you want. I think we're seeing this with the 3200. If you've got a design that keeps changing and the goal posts keep moving along. What can you write down about that? Whatever you did yesterday is gone tomorrow you know that's not good either.

NEIL: The reason they kept changing was it because they were getting more marketing inputs or because Malcom was not controlling the design and holding fast on changes.

MARIS: The reason behind it was that there were better ideas evolving about how the instructions should work.

NEIL: After you had already committed to design.

MARIS: Yes.

NEIL: But that's the project manager's responsibility to prevent that! The 3200 then was delivered some time after the 3600?

TOTH: I think so, I can't recall the order of events anymore.

NEIL: How about the 3400?

PAGEL: Not to answer your question, but I want to reflect on something first. As I see it now, we see that Seymour in the 6600 and the 3400, 6400,those three machines didn't receive any "help" from marketing. Isn't that interesting. The 3200 got all sorts of help. The STAR 65 and PL50 boy,did we get help on that one.

TOTH: Did you notice that I put "help" in quotes?

NEIL: The 1604 didn't get any help from marketing, did it?

PAGEL: You can always improve a design. I don't care who did it you can improve it. I think the lesson we're learning here is you have to have enough faith in the team you put together, say we'll do it and then go do it. That's what Ray is saying.

RAY: Yes, once you get ready to start, that's it. New ideas can go to the next line.

KRUGER: What you just said about Seymour now, this is way before my time, but he had no "help" from marketing yet he was able to realize one. Was there a market or did his machines help establish one?

PURCELL: They established it.

KRUGER: What I'm getting to is the little article by Jim Thornter where he said "there's a market that will be developed by STAR (it's not there now).

NEIL: I understand from "the folklore" that Seymour was aiming at a market.

GRINNA: Seymour had a real gut feel for market. He wouldn't talk about it in terms of whether there's market for this machine. He knew customers and he knew himself where the market was he was shooting for but I don't think I've ever heard him say that.

TOTH: I think in my recollection Seymour really thought the market for the 6600 encompassed about 6 machines and the market beyond that was the one Chuck Purcell alluded to that was actually developed.

NEIL: Nevertheless, Seymour did have a conception of FORTRAN when he was building the 6600. He didn't build a machine and then adapt FORTRAN to it.

TOTH: I remember Seymour saying he didn't know if FORTRAN could be written for the 6600.

PURCELL: Although he was fresh from the FORTRAN project for the 1604, he did some very smart things.

NEIL: Looking at theoretical things....on the 6000 compared to the 1604 which improved the FORTRAN ability, forgetting the problem of multiple functional units, and registers.

TOTH: This was very early before anybody tried writing a FORTRAN for the 6600, he didn't know if one could be written.

MOE: His advantage I think was that if he said, gosh, if I can make the 6600 ten times as fast as the 1604, I can probably sell it. That's not a difficult judgement and that's about what he did.

NEIL: I have never asked him straight out"well, now come on you must have had some great principles in mind when you built the 6600." If you were to go out there and say to Seymour"you must have had some overall concept of the system before you designed it,"(with that pontifical tone of voice) he'd likely look at you with distaste and answer, "Well, no!" That would end the conversation. But you take a look at the machine, take a look at the Chippewa operating systems put on it, 10 PPUs don't buy you more computational speed, but they do buy you more system performance at a time when operating systems were getting monsterous. The 360 was announced in 1964, we delivered the first 6600 in 1964, but we delivered it with the Chippewa operating system on it. The 360 was the super "by God" operating system machine.

PURCELL: The committee operating system machine.

NEIL: Well, the point is that all the things they were talking about being able to do on the 360, time sharing and all that Mickey Mouse, CDC was already doing well on the 6600. What I would like to know, and I would like to ask him sometime, how much did he have in the back of his mind; what the system's philosophy would be for that machine; the operating system and functional philosophy before he built the damn thing.

PURCELL: I would say that some early papers that Ray Allard worked on system structure on the 6600 software, could really allow you to get some insight as to what Seymour had on his mind cause Allard literally sat at Seymour's feet writing down the things that Seymour told him about how to do the system.

NEIL: Was that Allard or McDougall?

PURC ELL: Allard wrote down quite a bit of it. At least I got my stuff from Allard.

NEIL: Are those the ones with the artfully drawn diagrams?

PURCELL: No that's Casell's stuff.

NEIL: No, the stuff I'm talking about I got from McDougall and I thought that McDougall worked under Allard and I was wondering if that's where it cam e from.

TOTH: I think McDougall's input was on ECS.

NEIL: No, when the decision was made to kill SIPROS, I believe McDougall and Dave Young were sent to Chippewa for some period of time. I don't know how long they were out there.

PURCELL: This was 2 years before McDougall got involved. When Allard was down at the Fulton Bag Company.

GRINNA: I was going to comment on something about the par allel from the 8600 as far as Seymour's systems concept. I have no direct comments on the 6600 and even the 7600 but on the 8600 he had designed in some instructions and some special features to facilitate multi-programming and he had a feeling that something ought to be done with them but he didn't have any idea how to do it. Or at least he wouldn't admit to any idea of how to do it. He just had a feel of what could be done, put in some features so that some smart programmer somewhere could do something with it.

NEIL: But he really only put the features in if they fit the hardware.

GRINNA: Well that's true, but the hardware was designed somewhat around those features, some of them were kind of basic to the whole concept.

NEIL: You mean like the interlock bit?

GRINNA: The interlock register and other things.

NEIL: So what you're saying is he probably put the 10 PPUs on the machine thinking they'd be a good deal but at the time he did it he really didn't have a definite plan in mind for their use?

GRINNA: I don't think he knew exactly what the operating system was going to look like. He hadn't sat down and flowcharted it or anything like that, but he thought, well you know, we should be able to do something with this.

HUTSON: There were other things in the machine too, like the relative zero, and field length, to divide up memory space for control.

PURCELL: On the 1604 he had experimented with the 160 as a peripheral processor. He had some idea what kind of power a little machine would have against the support of the larger central processor system.

NEIL: Well, I'm going back up to the 3400 and 3600. We're kind of chronologically working our way through this although we do hop about. Anyone else have any observations of the 3200 or 3400. The 3400 came out how much later than the 3600?

HAWLEY: It started kind of before 3600 was delivered but it started essentially after the design on the 3600 was done.

PAGEL: It went out in the following spring to a joint computer conference in Washington, D.C. whatever month that was.

NEIL: That would have been 1964.

PAGEL: It was less than a year.

NEIL: That would have been '64.

PURCELL: Don, on the 3400 you were burned by one stupid decision and that was to save some money in the hardware design of 3400 by cutting out some of those extra instruction facilities. It saved all kinds of hardware money, maybe $100. The software didn't work on the 3600 so it would have been much better to simulate at great expense the capabilities of all the instructions of 3600 then the software would have run.

PAGEL: You missed my earlier comment. I wouldn't have done it all.

NEIL: O.K., how did 3300 show up in all this?

PURCELL: It was the 3200E, an improved 3200.

NEIL: Did we sell many 3200's?

HAWLEY: I think the 3300 basically took the option packages from the 3200 and integrated them into the main frame.

MARIS: It didn't have paging, but it did have the segmentation of programs.

HAWLEY: O.K., the 3800 did have virtual memory.

SPECKER: At least it had a paging option.

PURCELL: It had BCD option earlier and then it went to a BDP option.

NEIL: Was the 3300 a complete redesign?

HAWLEY: Yes, it was completely redesigned.

MARIS: Most of the logic was the same as the 3200. It was repackaged so it would fit the chassis and the ground rules.

NEIL: But there was a motivation for the 3300, why build it?

TOTH: Business data processing.

NEIL: But we had 3200's.

TOTH: They weren't adequate business data processing machines.

NEIL: What was wrong with them?

TOTH: I don't believe they had such wonderful things as move instructions.

MARIS: They had a lot more memory, the 3300 had banks and you could get 8 times the memory.

NEIL: Was it speeded up?

MARIS: No, I don't think so. Just 8 times the memory.

NEIL: So the marketing people said we can't sell many 3200's. We need a BDP machine.

HAWLEY: 3200 was also severely suffering from reliability problems. I'm not sure why. The 3300 solved that problem.

KORT: It was a problem I think.

HAWLEY: No, I think some of the design ground rules had been stretched too far on the 3200. They followed the ground rules better on 3300.

HUTSON: It had a reputation problem then, too, didn't it? As far as selling it was concerned, we needed a new number on it.

SPECKER: We recognized that it had the highest potential market. They were going to sell a lot of them so they decided to build another one because they thought it was a good market to be in.

NEIL: By this time than everyone recognized you were going to have a low end divergence. It's not going to look like any other machine. Who had the 3300 project?

SPECKER: Dick Day. At that point CDC decided to make a line of the computer.

NEIL: 3150, was that part of the decision too?

PURCELL: That came a little later.

SPECKER: 3500 came along a little later, the 3100 came first after the 3300.

NEIL: What was the 3100? A cheap 3300?

MARIS: A slowed down clock. It didn't have the options for paging.

CURT: Incidentally, over the years since then there have been people over in Plymouth putting all those options back on the 3100.

NEIL: Once you've decided to have a line the 3150 is easy to explain... just tuck another machine in the middle.

MARIS: It had a stand up console, the I/O was cheapened so they went to memory for every address. The I/O controller was completely redone.

PAGEL: That can be described as kind of productizing stuff, no new architecture and no new design.

MARIS: It was a process of trying to fit the price and the performance gaps in an offering.

NEIL: Who was responsible for the design of that console, for the 3600/3800 our "STAR TREK" console?

PURCELL: Mostly Russ VanKrevelen, wasn't it?

PAGEL: No, that's Polivka.

NEIL: Wait a minute, before Polivka got hold of it someone must have decided that they wanted all that data displayed to look at. I've never seen a console with more garbage on it.

KURT: You've seen the 1604 console. It's smaller than the 1604 console to start out with....physically smaller at least and there's really not all that much more on the 3600 console displayer.

NEIL: I guess cause it's spread out so nice.

HAWLEY: We put out all the I/O channel displays and there are a lot of individual bit lights. As far as registers and the like they were the same.

CURT: Customers want it that way. More impressive.

PAGEL: It's like Chrysler Auto fenders at that time. People needed to be impressed by all that chrome.

NEIL: The LARC had one that looked like a theatre organ. This is the first one I saw that was that big. IBM built one where you had a barroom rail along the side and you pulled yourself along the machine in your swivel chair.

MARIS: Wasn't that when glass came into the picture. Glass doors.

HAWLEY: Glass doors. On the 3600?

KORT: I went to one customer and the first thing he wanted to see was his orange chair.

NEIL: Was that on the 3600?

HAWLEY: Yes, it was.

NEIL: O.K., to wrap up the 3600, why the 3800 and why did it come so soon after the 3600?

PURCELL: Virtual memory.

HAWLEY: It really wasn't all that soon after the 3600. There was a bit of a gap in there before it was actually started.

PURCELL: We had problems with that big memory. The first memory module. The oil cooled memory.

HAWLEY: The 3800 had a new memory unit, it never worked and finally got scrapped.

NEIL: Why build it?

HAWLEY: This was after 3600 had been going. We had sold 30 or 40. Again the cry came from marketing "hey, we're running out of steam, got to soup her up".

NEIL: What kind of improvement could you predict at that time?

HAWLEY: The things that went into the 3800 then to speed it up were a faster memory, and look ahead for instructions.

NEIL: Now Seymour had already gotten his look ahead running on the 6600?

HAWLEY: What "look ahead"?

PURCELL: There was another channel from memory to the instruction decoder that wasn't shared for data.

HAWLEY: We put in the extra pipe to memory for the instruction trunk. We then decoded the instructions, did the indexing, etc. to get it ready for execution while the previous instruction was being processed.

NEIL: Was there any other technology or architectural innovations in the 3800?

HAWLEY: The option was put in for virtual memory.

NEIL: So that really primarily the only changes. How about 3400, was there any new technology or architectural ideas there?

HAWLEY: The changes were not architecture but included technology. We went to delay line sequencing rather than chains of flip-flops.

NEIL: Was that a good idea?

PAGEL: Very reliable.

NEIL: Compared with flip-flops?

PAGEL: It is definitely Slower.

TOTH: It's not a new idea. It had been used in other computers. It's cheap and reliable as opposed to fast.

HAWLEY: We had done this on the memory of the 3600 and then we carried it over to CPU. The 6600 is kind of this way where you have pulses that aren't like a timing chain for a given operation time. You have a universal clock that comes out at a certain time.

NEIL: Are there any other hidden computer developments?

PURCELL: Did we talk about the 1700 because that came after.

NEIL: Where did 1700 fit in?

CURT: Timewise? It was before the 3300. The intent was to build a machine for process control and it was defined, the features in it, were defined with that intent in mind. In other words, the people at LaJolla thought they were doing process control computers. They wanted some features in a small machine they couldn't get anywhere else.

NEIL: They didn't like the 160A?

CURT: It wasn't like the 160A at all. The 1700 was the architectural response to that problem and technically was trying to take the 6000 technology and air cool it, take the 6000 package, spread it out a little bit and blow air through it and took the 6000 stack and expanded it to 18 bits and expanded it physically so they could blow air through it.

HAWLEY: At this time 6600 was having many troubles with their memory. So the cycle time at the 1700 was lengthened out to 1.1 microsecond so they could get a little more margin.

NEIL: The instruction set was driven by LaJolla?

CURT: Yes, we were the engineering people who were writing the spec and designing instruction, but they were the people making the decision as to whether this was a good idea or not. The moving force was coming from LaJolla, as a matter of fact, at that time it was a very firm management decision that there wouldn't be any software written, it would <u>never</u> be marketed as a standard product.

HUTSON: Didn't it have environmental requirements like you'd have to sit beside a blast furnace or something like that? Did it ever meet those specifications?

CURT: Well, as far as there was in the memory design circuitry for temperature compensation it did.

HAWLEY: I remember when we ran it through the environmental labs.

CURT: That went through fairly well but after that we had trouble with that memory for years.

MOE: It was probably the most successful unfunded project we've ever had, however, but it sure took a lot of people and a lot of unfunded projects to work on it. It was clearly a case of like you said, "It really wasn't very well funded", it never had been projected as a standard product. Like all the other ones that we were going to sell

500 a week of, but never made, never sold. That poor guy just sat in the background with salesmen selling it wherever they could and pushing in a little bit of money doing it on other projects to keep it rolling and get it done.

NEIL: How many designers did you have on that?

CURT: There was one designer on the processor and one designer that did some logic for memory and the thing that we called buffer and channel, another box for the channel.

NEIL: Any innovations?

CURT: No. No architectural innovations.

PURCELL: It had the 8 bit byte It was an innovation for Control Data.

CURT: We thought at the time anyway that the interrupt scheme was an innovation for us. The priority interrupt scheme could be programmed for the interrupts.

STEINER: That was our first go at hexidecimal wasn't it?

CURT: Yes.

STEINER: In fact, there for a while we were using U,V,W,X,Y and Z. I remember that. As soon as I learned that then we switched to be compatible with IBM you know, A,B,C,D,E, and F.

HAWLEY: That came from the G-15.

NEIL: That was the old Bendix-G-15 notation. That was an interesting thought because IBM had hexadecimal on the 650 back in the early days. Harry Huskey who designed the G-15 said, "It's much more natural to add one to Z and get 10 than it is to add 1 to F and get 10". Is the 1700 development in the time frame of 6600 being shipped?

MOE: Yes.

HUTSON: The 1700 checkout crew was the old 3400 checkout crew.

NEIL: That's half of a 3400.

HUTSON: It was about the time that the 3400 got turned off, the 1700 took off. A little later than that actually.

NEIL: What were the major innovations that Chippewa felt was the most significant on the 6600?

GRINNA: The peripheral processor was probably the biggest innovation and having access to the memory from a PPU. The parallel functional units.

PURCELL: And a high band width memory, too.

GRINNA: I think the peripheral processors and that system concept was probably the biggest thing.

PURCELL: The 32-way interlace allowed that high speed access.

GRINNA: That's true, it made it practical for the PPU to access memory.

PURCELL: That's the important fact, too. It was an integrated concept. It wasn't a bunch of diverse ideas.

MOE: Independent functional units I think were the biggest CPU change.

GRINNA: Without interleaved memory, there again, you wouldn't be able to call up instructions to initiate operations in the parallel functional units.

HUTSON: How about the instruction stack? Was that new?

MOE: Yes.

HAWLEY: The register file was new. Arithmetic was register to register and not register to memory.

PURCELL: And that led to the exchange/jump package.

TOTH: Another thing related to I/O and that's the twelve channels, cross-barred.

MOE: Accessible by 10 PPU's. The barrel itself, 10 memories sharing one arithmetic unit.

HUTSON: How about the memory hopper in the CPU?

MOE: Oh yeah, that plus the storage of instructions that shouldn't have been issued or couldn't issue but did. And all that jazz. But of course the mechanical technology, the packaging was far superior to anything, much denser than anything we had thought of before. The year before in the 3600 there were 64 transistors per module as opposed to 6 so that technique was spectacular and the great cooling.

NEIL: Whose idea was the cooling?

MOE: Probably Dean Rausch. Oh, I don't know. I'm sure it was Seymour's idea.

PURCELL: It's fair to say Rausch was responsible for that.

GRINNA: Certainly, for the design. Seymour couldn't have even come up with a concept for his machine without counting on that kind of density.

PURCELL: Seymour said "Get rid of the heat" and Rausch turned to freon cooling.

BHEND: Wasn't that clock scheme unique at that time?

MOE: Did any machine have that kind of clock scheme? I can't say, I have no idea.

HAWLEY: What about the clock scheme?

BHEND: They had 100 nanosecond clock that was broken down into 4 -

25 nanoseconds phases.

HUTSON: Actually that's a fiction too. Because there were actually an infinite number of clocks like the STAR.

BHEND: There are a lot of things that were learned on the 6600 though that changed on the 7600.

HUTSON: I think one of the things that had been in the 6600 that nobody ever really worried about before was wire lengths.

MOE: Yes sir.

HUTSON: You suddenly found out that you spent a lot of time in the back panel and you were often missing clocks.

MOE: Before that you could run the wire around the block and it didn't matter.

NEIL: Did anyone read the article in EDP Design magazine, the interview with Seymour. This is the EDP magazine supposedly for electrical engineers. It's the most banal interview I've ever heard from anybody and the only quote they could attribute to Seymour goes, "One of the things we discovered with this new machine technology is that we really have to measure all the wires to make sure the signals all apply at the same time!" (laughter) It's the only quote in the whole article.

BHEND: Another thing on the 6600. Serial 1 was done and on the floor and the transistor art hadn't caught up with us yet. We were replacing individual transistors in certain paths with faster ones that were just coming out to meet the necessary speeds.

HUTSON: What was really neat about that was the fact that in chassis 1 the PPU's worked with the slower transistor. When the art did catch up, everything else worked, but chassis 1 didn't anymore. We ended up having to put a lot of long wires in chassis 1.

MOE: The transistor art moved extremely fast.

PAGEL: That was the brink. Well, from speccing it in 1961 and not making it until late '64 when 4 machines are on the floor to then going wild and exceeding it to a tremendous amount within the next six months made a very interesting problem.

PURCELL: They put more gold in the dope.

NEIL: That same thing is true of the register file on STAR.

HUTSON: Serial 6 6600 had in the read flag registers three different stages of transistors. Over in the center of the machine it had the slower ones and as you got towards the outer edge of the chassis it had medium speed one and at the outer edge you had fast ones and you had to make sure you used them that way or it wouldn't work. It was that tight at that time.

NEIL: That sounds something like this problem we were discussing last Friday. We have a problem with skew from the clock fanout now in the LSI logic. There's one whole gate-time skew because of wire lengths from the clock driver to the driver clips.

KROHN: Four adjacent chips.

NEIL: Four adjacent chips is one whole gate time skew.

KROHN: Which are separated by 2 inches which will give you askew of 600 picoseconds which is 20% of your clock width.

MOE: Both on the 6600 and 7600 there were no oscilloscopes available to see what was going on. And the oscilloscope manufacturers weren't terribly interested. They really would come out about a year after we needed something.

HAWLEY: I remember when we first put a sampling scope on the 3600. We were horrified at all the junk we saw on the trace. Speaking of technology, clock systems are something that have changed a lot as we've gone into different machines. On the 1604 the clock was a whole bunch of individual oscillators that were just all tied together, all oscillated together.

TOTH: Just like a power distribution system. They locked onto some common frequency, and nobody knew exactly what clock might overrride but they had to be cut fairly close otherwise, there will be some glitching due to overlap.

HAWLEY: You tuned each clock individually and tied them all together and if they weren't close enough you set up an oscillation on the wires the connected them and you get some real screwy patterns on the scope. On the 3600 we went to a master clock with a multi-rank fanout where each rank could be tuned to guarantee that all final ranks were in phase.

NEIL: Since we are running out of time today, let me remind you that our intent for future sessions is to review the more intimate projects, the ones we all really have first hand knowledge about. I'm not so concerned about 1604 unless it's important. I'm particularly concentrating on the 6600, 3500, the 1700 if necessary, STAR, 8600, 7600 era and hope we can talk in great detail. I think I wrote down design approach and initial design. The reason I wrote those was that generally you start out with design approach and then you start designing and you start modifying the design approach cause it doesn't work or something. The session after that was scheduled to discuss actual design and this meant getting at the real design in the areas that we think are worth examining. Like talking about the design problems that arise in the design of the PPU structure on the 6000, 6600 or 7600. But, I would like to skip ahead briefly. What about the 6800?

MOE: Well, the 6800 was quite simple. It was 4 times as fast as the 6600.

NEIL: In exactly the same architecture?

PURCELL: Twice as many PPU's.

MOE: No.

PURCELL: Headquarters thought this.

MOE: Well, that's fine. So the initial design started with the 6600. We started the 6800 1965, or late '64 and early '65 and Seymour chose then some guidelines. It was to be 4 times faster than the 6600 with his nonsaturating transistor which is the same thing we used on the 6600. Clearly the 250 nanosecond memory was required. So those were the design requirements, there was the circuit that we had reasonable confidence in such that the circuitry could easily support it. That's what started it. Seymour took off on the design of the CPU and actually told me to design the PPU. He said, "You can have 125 modules to replace the 10 PPU chassis" and he went on with the CPU. Not more than six months after I had been working on, I had it down to about 180 modules. He "SCrayed" me again. He said, "No, no; we're going to use separate PPU's." So he designed an independent PPU (to keep the folklore going) on a weekend. And there we implemented that. Now, I have to go back to the 6600 because the 6600 was implemented similarly. The 10 PPU chassis was built at Strutware, moved to Chippewa. Garner McCrossen and the software people were responsible for checking it out and keeping it running and then that's what Seymour worked with quite a bit. He verified the circuitry and verified the memory. Now for the 7600 he did the same thing in reality. The first thing he designed was a single PPU on a refrigeration chassis about the size of a 211 or 217 console with independent refrigeration and he built several of those.

GRINNA: Several versions, several circuits.

MOE: I don't know what day it became the 7600 but I do believe for sure the PPU design was one of the contributing factors to the 7600. I can't say because I don't remember whether he had large core memory in the 6800. Obviously, he didn't. I believe it was just 4 times performance. So in addition to the PPU then he added large core memory.

NEIL: Was that a result of our ECS experience with the 6600?

MOE: I think in "real-time" large core memory and ECS were concurrent projects.

PURCELL: They were no more than three months apart. Seymour was talking with Jim an awful lot at that time, particularly about long-wire memory design.

MOE: So the 6800 was, in fact or reality, contractually commute as a result of the 6000. Any 6800 contract or initial intent for. The machine, I would say arose in 1965. In '66, I guess, we went full steam ahead on the 7600 CPU. Interestingly enough, on the 6600 Seymour designed the PPU's and the memory and so on. On the 7600, Seymour designed the entire CPU& the PPU. Les Davis designed the large core memory.

BHEND: Les and Harry.

MOE: Seymour never touched a scope probe to the CPU. The only thing he ever worked on were the PPU's.

BHEND: He handed everyone a particular functional unit to verify.

MOE: All Seymour did was a complete paper design in less than a year and never looked at it, never implemented it. In fact, when you think about it, Seymour didn't know the CPU in the 6600 either cause Jim did that. It's incredible what Seymour did with the 7600. He designed the entire thing on paper and simplified it. He made an exceptional simplification over the 6600.

PURCELL: There were lots less ranks of registers.

MOE: And a really controlled clock since we had learned that was a problem.

GRINNA: He also brought in Bob Horton for documentation kind of early in the project and that helped us when we shipped it to Minneapolis. We didn't have near the problems that we had with 6600 which didn't have any documentation at all.

GRINNA: It wasn't so much that he brought Bob in, but that he "permitted" him in. That is a general commentary, I think, as far as Seymour's attitude towards documentation and or any additional people on the project; whether it be documentation or marketing or just any other inputs. I think Don mentioned something earlier at the time that the 3600 and 6600 split off, that 6600 was a high risk project and Seymour was not willing to say that we will deliver a machine. It was identified as 50% scrap. And I think he followed that philosophy pretty much all the way through even on the 8600. He was never willing to tell marketing that we will deliver this machine. He did that intentionally, but it worked out quite well in some ways because there were not a lot of people trying to come in and provide inputs It was considered a high risk and there was no desire to put a lot of extra money into the project as far as documenting it or marketing it or anything like that, until the risk became less. I think that's contributed a lot to the success of the Chippewa group.

HAWLEY: I think that one thing that needs to be said though is that Seymour considered it a high risk, but Headquarters considered it a sure thing.

GRINNA: That's because anything that Seymour did was considered a sure thing, but would never permit them to act as though it were a sure thing.

NEIL: Wasn't that really part of the difficulty he had with Headquarters over the 8600? They needed another sure thing quickly.

MOE: Significant though on the 7600 in documentation, Don. I think anybody that likes to design or thinks they can design should see the book Seymour did write on the 7600 documentation was written (chronologically now) after the design was complete on paper but long before there was any hardware that existed and it's a CPU documentation.

PAGEL: I think that's good.

MOE: It's unbelievable for anybody who needs to work at that level for people who want to work on a machine or have to learn it because that was the extent of the documentation (in addition to the wire tabs) but that kind of documentation was something incredible for engineering.

MARIS: How did Seymour define his instruction sets? What was his method of working?

MOE: He designed one. O.K., that's another point I wanted to make. He didn't accept a whole lot of help and I don't know if he accepted any. My experience (having been on the other side of a brick wall) was that he communicated on the telephone, with several people and I can't even say who. I'm sure he communicated with Jim a lot, I don't know who else. That would be his afternoon's entertainment, to communicate on the telephone. I don't know who he got his inputs from but he sure as hell didn't take them from anybody that wanted to give them.

PURCELL: Seymour and Jim did discuss a 64 bit word size for the 6600 mainly because the STRETCH machine was 64 bit and Seymour finally decided that he didn't know what those extra 4 bits were good for and he couldn't count them on his hands very well or something..it wasn't octal. And so he wiped out those extra 4 bits. I think that decision changed the history of commuting.

BHEND: I think he got his inputs from Livermore. I think that's what affected the decisions he made on how to change his instruction sets. He talked with people at Livermore and once he made up his mind that was going to be the instruction set he wouldn't let anybody else change it. I don't think he was influenced by marketing at all on his instruction sets.

NEIL: We still aren't quite done with architecture or at least how we got where we got. We left off with a kind of hybrid discussion of the 7600, 6600 and that leaves to my mind the MPL, which we should visit briefly, and it leaves the STAR 1B, STAR 65, STAR 100, the 7600, 8600 and anything else in that time period. Probably most of you don't remember how we ended up. We did talk a great deal about the fact that the 7600 documentation that Seymour did was probably the first time in history of the company that something like that was done and it turned out to be very useful, I gather. I just would like to stop at the 7600 for a minute and ask if anybody's got any other observations or comments about it. We know why we built it? It came out of the 6800? We kind of got locked into it. We know why, or do we have to ask Seymour, why it was changed from the 6600. For example, why did we go to the very limited access to the central memory, why did we go to something like independent PPUs? Anyone shed any light on that?

GRINNA: I don't remember what we said last time. I think the biggest thing was the cleanup, and there was quite a bit of study done at the close of the 6600 project. I think maybe Bill could comment on this better than anybody because at the wrapup on the 6600 Bill spent quite a bit of time studying the usage of some of these features that were in the 6600 and how much utility they really had.

BHEND: They wanted to find out just how important it was to have things in the computer like the "stunt box" for example. If you could take a program and go through it and change the program so that it would run faster or as fast, without having things that they had in the 6600 to implement high speed execution. They wanted to find out if they could simplify the control basically. That was what they were looking for. I took several programs that we had run on the machine and went through the code and changed it so that there wouldn't be delays waiting for something from memory or waiting for an answer to be put into an X register. We found out that if we did that there were lots of ways that we could make control much simpler on the 7600. Dennis, didn't Dave's report on the 8600, mention why on the 7600 we decided to use processors and large core, or why we went away from it?

GRINNA: Or why on the 8600 we went back to some 6600 concepts.

BHEND: But I think they wanted processors mainly for I/O use. They felt they could hook up a bunch of equipment to separate processors and let them run independently of the CPUs.

NEIL: Let's take the things that come to mind. The cross-bar for the PPUs in the I/O channels which was a thing that was completely absent from the 7600.

GRINNA: I think the difference here was the memory hierarchy that we had on the 7600, kind of dictated the rest of the things. On the 6600 then it's just the one memory and on the 7600 we had the LCM and SCM.

NEIL: Why?

BHEND: I was involved in the LCM, I really don't know why they used the architecture except I think they kind of looked at it more like an external memory like ECS.

NEIL: ECS was pretty far along by the time that decision was made?

GRINNA: That's right.

NEIL: Was there any such concept on the 6800, or did this come after?

MOE: It came after.

NEIL: That was Seymour's idea then, to move to a memory hierarchy? Was it a cost thing do you think or electronic?

MOE: Just more memory, as required.

BHEND: Faster access.

NEIL: Why not 500,000 words of the small core memory?

MOE: Oh, that certainly was cost. Small core was a single module, 4 wire, classical memory.

NEIL: What's the difference in cost between the LCM and SCM?

PAGELKOPF: Chassis price is about the same and what did we have for banks there? 125K per chassis on LCM and 16K of small core memory at equivalent cost.

GRINNA: Almost ten to one - eight to one.

NEIL: Well, you could still have had the memory hierarchy and say for example, still had a cross-bar network with the PPUs and channels and stuff.

MOE: I think I mentioned last time we did on the 6800 project, we did do a 10 PPU version of the design with 125-7600 modules. I said last time he threw that all out and said I want to do separate PPUs.

NEIL: I just don't remember the reason for it.

MOE: Well, I think I really don't know the reason and we never got any and really never bothered with them. One of the things he was thinking about was direct driving peripherals. He said if I can make a PPU for $5,000 why pay $20,000 for hardware controller. It was a very valid comment at that time and he was trying to prove that. That's probably one of the biggest reasons.

GRINNA: A key point there is the flexibility. To able to throw out one PPU and change price and configuration.

MOE: It wasn't early when the decision was made (he had decided on the 7600) that Arden Hills would supply him with the peripheral subsystem. And Arden Hills really never did that so he went on doing some of the things of his own trying to use PPUs, the direct drive tapes, the display system, the.....I guess that was about it at that time.

NEIL: My recollection was that he was supposedly waiting for an I/O subsystem based on buffer controllers. ADL went through the evolution of the buffer controllers, changed the buffer controller design and then redid the whole thing and ended up not matching his schedule at all. I remember he came down to visit us and said he could no longer wait for them, he had to go on his own but he was waiting for buffer controller oriented peripheral system. The funny thing is that a lot of the ideas that we put into the second generation buffer controller and I/O system, spun off from things that we got out of our visits up there to Chippewa with his 6-pack station; so it came full circle. What other things like the stunt box do you figure were significant that you examined and decided to change for 7000?

BHEND: I was trying to think of all the things he had on the 6000, can't remember what they were. The reservation tag scheme.

MOE: Register reservation scheme, yes.

GRINNA: Well, the registers themselves were simplified.

BHEND: Another thing I think they were trying to do was find out just how many registers did you need to get simplified control.

NEIL: Internal registers or visible, that is program accessible registers.

BHEND: Program accessible registers.

NEIL: And you decided to stay with the same number?

BHEND: And also how long the loop back feature (the instruction stack)would have to be. What would be a good number for that. That was one thing they were looking for.

NEIL: In other words, the 6600 did really have some addhoc features which you now, after the fact, examined to determine whether or not they were really any good or whatever. What did you conclude about the registers? Obviously, you kept the same number for the 7600, but was there a reason?

TOTH: The instruction stack too.

MOE: That went from 8 to 12 words. I think that tended would be more a hardware driven decision, the implementation tended to be more hardware reason than anything else. The implementation of the instruction stack and a fantastic improvement in the jump speed within stack. He redesigned the 7600 instruction stack;pyramided right next to it was an address stack and that permitted 12 addresses plus 12 registers or instruction words to be transferred in 3 minor cycles for in-stack jump.

RESNICK: He effectively made an associative search on the thing, there was no arithmetic or anything. You just ask and thou shalt receive.

MOE: Whatever the address was,it was translated into a 60 bit word which was sent to the instruction register.

NEIL: The choice of 12 was dictated primarly by just logic rather than because somebody concluded that 12 was a nice number.

MOE: I think Seymour(when he layed it out) found that he could put segments of the stack into 3 rows of modules. It was a case of 4, 8 or 12 or 16, and 16 was more than 8 on the 6600 and probably would make a significant difference.

BHEND: He usually didn't tell us any reasons why he decided to do something. We fed information in, but he more or less did what he wanted to.

MOE: The net of the simplification of the 6600 as I remember being told was it cost 7% in degradation, it took out a significant amount of hardware and that 7% in degradation was strictly prohibiting further issues until the instruction could be completed.
If you did a shorter instruction you translated whether or not the results from all other instructions you had issued that weren't completed yet would in fact, complete at the same minor cyle. If they did, you didn't issue. That was at the end of the issue translation and the front part was if the registers weren't free for some other reason you didn't issue. Now that took a hell of a lot of hardware from the 6600 in the queues and all the registers that saved instructions and save instruction register contents, things like that. The number I remember was 7% degradation over just redoing a 6600

in 7000 logic at four times the speed. That was 4 times -7%. The PP scheme was certainly not explained to me although it should have been, but I didn't bother. It was done and in a conversation with Les Davis a year or so ago he admitted they wished they would not have changed the 7600 even minutely from the 6600 to the point where it prohibited running the same operating system. And talking about the future, that's what we're going to do. They will probably try to put CYBER 170 PPs on the 7600 in the future.

PAGELKOPF: It's called the model 175.

MOE: Whatever.

PAGELKOPF: Another thing you get with those PPUs, the 7600 PPUs, is independently dead-startable units which was a system problem of the 6000 PPs. There's more than one solution to that, however, but they didn't come up with one for our CYBER line as far as I could tell.

NEIL: There were several major problems with the 6600 structure from the systems point of view. One was that one, which was simply that a PPU could hang a channel and all the other PPs couldn't get it. The PPs had unlimited access to central memory and because of that as far as operating system developer was concerned it was a disaster. The way that we used the original Chippewa operating system any PPU could go in and accidentally clobber location 0 and automatically destroy the system and you could never find out who did it or why. I kind of thought that the reason he went to that scheme was to get some form of protection. But you're telling me that it was really driven from the electronic requirements or design requirements.

GRINNA: No, I'm sure that protection was involved in the decision on this. There's a whole combination of things here. It's hard to tell what went through Seymour's mind.

NEIL: Did it simplify the structure to eliminate the pyramid and to be able to just have the fixed addresses and memory and stuff like that? That simplifies it?

MOE: I believe the I/O structure of the 7600 was significantly more complex than the 6000. Like I said before my constraint was 125 modules for duplicating the 10 PPU I/O structure in the 6600. In reality he took, with 14 chassis, he took 512 modules, in addition to modules strung out through the memory chassis. He had a very big I/O fanout, what do we call that?

GRINNA: Multiplexer.

MOE: Oh yes, we had a very large I/O multiplexer. Another point in the I/O scheme (significant probably for the 7600) was if the 7600 in fact is 5 times or 4 times faster than the 6600 then hypothetically you've got to have more capability of getting data in and out. We weren't front ending much of anything at that time. So if we were selling 3 to 4 printers normally on a 6600 (there were 24 tape units on all the early ones we delivered) and, therefore, the 7600 would have 4 times I/O rates like that then 12 channels disappear rapidly. His ultimate I/O scheme there of PPUs with 12 channels each became I think, a reasonable decision. What has happened in reality is that's never been a stand alone 7600 computer and hence it would not need the multiple I/O channels.

NEIL: Was it acceptable from the outset to half-track discs? I mean we always stuck with that in the 6000 but I kind of thought that was an accident. Going back to 6600 days, the very first disc we put on we had to half-track because the PPUs could not transfer data fast enough. Was that an assumed beginning point for the 7600, you were going to half-track this with 2 PPUs?

MOE: I think even early 6600 he had duel channel access. For ping-ponging had been used prior to that already.

PAGELKOPF: I think the SCOPE people finally found that ping-ponging only got them 5% though because of other software overhead problems. That's true but we always said that if we didn't have to half-track it would be better than that because the software management of the half-track scheme could be a big pain.

MOE: On the 7600 today the disk subsystem works at very good speeds. I can't say what it is, but it's quite efficiently used. I think in retrospect the 4000 words of memory - 12 bit words of memory in PPUs has been a hell of a limitation for the corporation in philosophy anyway. That was expensive and each PPU already took 31 modules.

MOE: Cost the customer 50 grand.

NEIL: One of the things on the 7600 was originally space in the main chassis for what I'll call CPU PPUs very much like the 6600. Then you have the stand alone 6-pack. What was the original purpose of those PPUs that were in the mainframe? Were they to be used like in the 6000 PPUs?

GRINNA: One thing about this, it wasn't original, but on one of early serial numbers, I forget which one, 5 or 6, we designed the high speed channel for the mainframe PPUs to use for driving the disk so that the PPU could keep up with the disk file. That was possible in the main-frame because the PPUs were synchronous with the central channels and on the 6-packs they were asynchronous so they had to transfer slower. That came a little bit later. I don't think there was any intention to use them just like they were on the 6600.

MOE: No, he had no intention of using his 6000 operating system on the 7600. He fully redid the operating system and did that without using the 6000 cycle.

PAGELKOPF: Those first level PPs also served a role similar to the service station on the STAR. Nothing more than additional compute fanout.

MOE: The hard wire memory allocation for PPUs was implementing in hardware what he felt the circular buffers had done for the software in 6000. That's my impression of what happened. It seemed to be acceptable.

NEIL: Was it impossible to make that port go directly to LCM rather than having the SCM to LCM traffic?

MOE: Oh hell no, it would be very easy. The only thing you'd have to do is to get someone else to design the 7600. There was never any question in Seymour's mind that there would be no outside access to Large Core Memory. He would not allow it because it would disrupt what he thought was most important, 27 ½ nanosecond transfer rate per 60 bit words from large core memory to and from small core memory. So anybody who asked that question was told a categoric no, no way because I don't believe the machine's going to be used that way and I don't want it to be used in that way and hardware performance pricewise that's the most efficient way in his mind to use the machine.

NEIL: How did we manage to make such easy access on the 6600 to install ECS port considering how tight the design was?

PURCELL: Jim Thornton did it.

NEIL: Wasn't that pretty distruptive too, of the memory logic on 6000?

PAGELKOPF: Sure, the machine came to a stop. "Hold it folks, get ready, go ECS".

PURCELL: What do you think happened yesterday, Don when that statistician went over to talk to Seymour about redesigning the Cray-1 I/O system?

PAGELKOPF: He's probably still paddling down the Chippewa river.

GRINNA: One thing I'd forgotten about and Bob mentioned a little bit ago, at the time the 7600 was being designed we were still working at PPUs driving all the peripheral that we'd hung on the 6600 and that was part of the philosophy of those mainframe PPUs. They would in turn have networks of PPUs tied to them and to the extra PPU channels also. There would be this huge fanout of many PPUs out there that would feed in through this network.

NEIL: Was the 7000 designed with the additional port for additional ½ million words of memory, like we're putting on now?

MOE: Definitely not!

NEIL: It was not then. That means they've had to carve up that part of the design pretty badly.

MOE: You never know. I don't think the logic is terribly complex. There wasn't much reservation problem or logic problem so if you wanted to do that you could do things like that. What it affects is internal register sizes and things like that, and data transmission to and from there, but it certainly is not impossible to do as they've done it in 7000.

NEIL: Again, the reason I asked that is like my misapprehension about the amount of memory of the old 6600. I swear that when serial 0 was on the floor up at Chippewa, we were just about to tear it down, we talked about the memory size and if it wasn't Seymour, it was Les or somebody else said, "Oh yes, we could put on another 500,000 words, but we'd have to put them in a chassis." At that time the compressor was sitting at the back-side. So he said we'd put it just about here, it'll be about this size. At least that's what it turned out to be now made out of solid-state logic. So again, I'm surprised to find that there really wasn't any design for that.

MOE: Both on the 6600 and 7600 Seymour designed only one system, period. Full memory 6600 with 131 K words, he did no other and 7600 with 65K and 512K words, period.

GRINNA: We had quite a fight just trying to get external connectors on the large core stacks. That was a real battle. They didn't want to do that and it wasn't until we had trouble with one and it took me three days to unplug and replug in another one and they decided that maybe something should be done.

NEIL: Was it because of the potential unreliability of having a connector there or was it something else?

BHEND: Yes, they felt it wasn't going to be reliable with connectors. Several of us felt that if we didn't have connectors then we were not going to be able to get stacks to work so we had quite a problem there before they finally decided to try it even and then they did and it was successful.

NEIL: Is there anything else in the area of some things you did different on 7000 than 6000? You kept the same number of registers. You did do that study and decided that that was alright.

MOE: In addition the arithmetic was seqmented.

NEIL: Why?

MOE: So we didn't duplicate hardware for multiply for example and to increase instruction issue speed also. You couldn't issue a multiply every cycle , for example on the 6600 because the arithmetic units were not segmented, because it couldn't complete on operation in a minor cycle. Segmented pipelines on 7600 were very different.

NEIL: But you didn't use a common pipe for all your arithmetic?

MOE: Heaven no, an independent multiply unit which was segmented three times.

PAGELKOPF: We sent out modules to Bob Kowalsky in California at that time. He put a counter on the 6600 out there to see how often that multiply unit was really being used, actual practice at Livermore, (I don't really remember anymore). It was ridiculously low, in percentage of time; it was under 5%.Which meant that that hardware really was kind of a waste, in fact, I came across the records of the transaction yesterday. He still owes me those modules.

NEIL: Cleaning house, right? The segmenting idea, was that Seymour's again? And that was just to solve the problem of the issue?

GRINNA: Yes.

NEIL: Was there any thought of going to a common pipeline like STAR?

MOE: Not in Seymour's mind, I'm sure. Well, actually that's not true because Les Davis designed the 7400 in 1966 using the F series modules of 7600 logic and that was stopped cause we had to devote our resources to getting the 7600 done. There Les used a unified arithmetic like 6400.

NEIL: The 7400 design was actually initiated at Chippewa before the 7600 was done?

MOE: Yes, before the 7600 was on the floor, the logic design for the 7400 was done and logic design and logic modules existed, in fact.

NEIL: But in 6600 no effort was made, right until 6000 was shipped? The 6400 really was an afterthought?

MOE: No, Jim had a pretty good idea of what he wanted to do before he left Chippewa. He was pretty sure what he wanted to do then?

NEIL: At the outset then, we decided to build the 6400, at least engineering intent to build the 6400. Did Les ever pick that up again after this?

GRINNA: I built a 12-pack with one bank of large core memory over there. Les built an arithmetic unit that connected to one of the PPUs in the 12-pack.

NEIL: Was this Tom Parkins' telephone booth?

GRINNA: Tom Parkins got serial 2 and the last one. Serial 1 we used for our engineering work in Chippewa right up until the end. That was a very valuable asset to our 8600 development.

NEIL: Who did the programming?

GRINNA: Pete Dutton did most of it.

NEIL: Was it all independent memory or did you have any common memory between?

MOE: Just the common memory.

GRINNA: Each PPU had its own memory, of course, but there was one bank, 65K large or core memory, which was common to all of them.

PAGELKOPF: That's half a chassis.

GRINNA: Half a chassis of LCM. We had one chassis which had an MCU and one bank of LCM and then all the control for the system.

NEIL: You call it the telephone booth, did it have any other designation?

MOE: The 12-pack.

GRINNA: The 12-pack, MPP or multiple PPU station. It was designed initially with the thought that it might be a product but it turned out to be too expensive for what it did. Although it was a powerful machine, there was no market for it.

NEIL: Was that initially motivated by Parkins or did Seymour really have the original idea?

GRINNA: Oh no, it was done before Parkin knew about it. He came over and looked at ours and decided he wanted one.

NEIL: You sold him one. Got all your R & D dollars back?

GRINNA: The first one was built of scrap parts, it was an unfunded development. In fact, one of my design criteria was to build it out of existing modules and there was a lot of haywire in there. Once we got it working, we cleaned it up and designed four new module types.

NEIL: The intention was to only have one PPU directly control the arithmetic unit?

GRINNA: Yes. It was a peripheral arithmetic unit, you might say.

NEIL: Like a disc or a tape?

GRINNA: Yes.

NEIL: Did answers come back through PPU or did they go direct to core?

GRINNA: To the PPU.

NEIL: Was any more work done beyond that, thinking wise, paperwise?

MOE: Only that which Parkin might have done.

NEIL: We might as well carry this Chippewa story through that period. When did the 8600 work start? Any thinking about that?

GRINNA: Well, really the 8600 thinking was already going on in Seymour's mind at this time because the time frame now is ..... I was working on that 12-pack after the delivery of serial 1, in fact, that would be about a year after that. So Seymour was already looking at the 8600, he was looking at multiprocessors and that's another one of the reasons why he decided it would be nice to build a multiprocessor out of PPUs sharing a common memory. As sort of a pre-prototype for the 8600. Seymour was thinking about the 8600. I don't know what the first date was, but about the time that the 7600 design finished up after he wrote the documentation which took sometime.

NEIL: Was there any external motivation at this point, not for multiprocessors particularly, but for another machine following the 7600. Everyone expected, obviously, that Chippewa was going to build another one. But was there any external motivation from outside that led 8600?

MOE: I think Lynn and Seymour would definitely say no. It was just his mode of operation and that's what he wanted to do.

GRINNA: You never could say there was outside influence, but I think that although Seymour appeared to make all these decisions off the top of his head, he was listening to the customers all the time and he had a pretty good feel for what the market wanted. He didn't have to have a report from marketing saying this is the kind of machine we want, he just had sort of a feel for what was saleable and that's what he tried to build.

NEIL: Wasn't the original concept to be very similar to the 12-pack in that the floating point unit was not to be an integral part of each of the processors?

GRINNA: Maybe this would be a good time to read this report on the 8600 experiment. Some of you got copies of this. It was written by Dave Cahlender and it came out shortly after our last meeting here. When I read it, I realized that he was talking about a lot of the things we were talking about here. Some of the preliminary things he states in here are very good, and I think thought out, and it deals with our discussion here. When we get down to the last section especially on the problem areas, then we're going to have a lot of comments because this report was just one man's view and maybe there's a lesson here that there should have been a team effort to come out with a report. But that was not done. Not to criticize Dave, but this was really the first project of this type that he had been involved with and some of these problems that he calls out in here seem very large to him. Those of us who had been through it before weren't as discouraged by some of these things. I think there's another point, that Chippewa had a continuity and low turnover of engineers. Many of the same people who designed the 6600 or worked on the 6600 were still there on the 8600 and had been through this process a few times and saw things in a little different perspective. That's just to say that for these problem areas Bill and myself will have some comments when we get into them. I'll just start reading and if anyone wants to interrup to ask any questiona do so at any time.

"The 8600 project was undertaken by CDC as our next entry into the large computer market. Several problems were undertaken simultaneously. Some of these problems were solved satisfactorily and some of the problems were not solved. This report indicates the problems that were attacked, the solutions attempted, and the results of these attempts. By the nature of the project and the personnel involved and the method of termination of this project this report will have some incompleteness and some inaccuracy. An attempt is made to minimize the inaccuracy and to label those areas where there is some doubt about the facts".

"Problem: Making a commercially saleable computer that is significantly faster than the 7600. The government has indicated that it will now allow its agencies to buy a machine that is not ASCII. A method must be made to allow several processes to share code for the machine".

"Discussion of machine selection: The speed of the machine was of course the important problem. At the time the machine was started

No significant improvement had been made in components since the 7600. This meant that the machine must derive its speed from other than brute-force speed improvement of the electrical components. An attempt was made to increase the speed of the machine with artwork, packaging and parallel operation of several CPUs. The ASCII requirements meant that the memory should be 64 bits rather than the 60 bits of the 6600 and the 7600. This change made it so that the machine could not be compatible with the 6600 or the 7600 and influence the selection of instruction set and machine architecture. An attempt was made to retain compatability with the 6600 and 7600 in areas like floating point format so that FORTRAN programs would indeed be compabible between machines. The multiple CPU selection was done after it was determined that we could not build a machine that was four times faster than the 7600. The multiple CPUs allowed a further multiplication of speed over the basic speed obtained by the artwork. Originally, the 8600 was going to have several CPUs with shared floating point functional units. This was abandoned when we could not figure out how to issue to the functional units from the 4 CPUs. We selected less expensive and slower functional units for the individual CPUs".

GRINNA: That addresses your question, Neil, and as far as I know that's a pretty accurate description of what happened.

NEIL: I guess there arises a question when you say "less expensive and slower". It sure doesn't seem very slow or inexpensive to me. Compared to what?

GRINNA: I'm not sure what that was compar ed to because the more expensive one was never really designed except that it would have been multi-access, multi-purpose. But the total cost may not have been any different.

NEIL: The reason I'm saying this is because I got the impression that the arithmetic unit was to be similar to the unit that you had on your 12-pack structure. It was to be a uniform unit and from everything I've heard, I don't see how, if you had made it out of 8000 logic, what you have now is any faster or slower.

GRINNA: I'm not sure it would.

"Instruction Set: The register arrangement and instruction set came from a logical extension of the 6000, 7000 registers and instructions. Since the memory word was 64 bits, the instruction would logically be 16 bits in length. The 16 bit quantity would logically be displayed in hexadecimal."

GRINNA: I would like to insert here that it may have been logical but it wasn't quite that smooth. There was quite a battle before we settled on hexadecimal notation some of the people didn't think that was very easy to understand.

TOTH: I don't think I ever saw it represented in hexadecimal.

GRINNA: Seymour started out using "DI-BITS", the 2 bit digit and that was pretty simple and easy to understand and it followed the hardware pretty well. Then each programmer started his own version in hexadecimal and we finally did all get together in the end.

NEIL: What did you end up with?

GRINNA: Hexadecimal notation that was fairly standard.

NEIL: Not U, V, W, X, Y, Z?

GRINNA: No, that one was in there for awhile some o.k. using 0 through F. Like he says, it was logical but it wasn't easy.

NEIL: Am I to understand that we did a register file analysis and concluded that the register arrangement that we had in 6000 was pretty good because it was retained in the 7000? Now what motivates the change in the register structure? Just machine logic or the fact that the 16 bit instruction dictated it?

GRINNA: I think we come to that in the next paragraph here. "Therefore, the instruction would be 4 or 8 bits in length and the register designators would be 4 bits in length. A 4 bit instruction designator would give only 16 instructions so an escape mechanism was derived where there were 11 4-bit instructions, 4 6-bit instructions and 32 8-bit instructions. In the 6000 series machine the 8 operating registers and the other 15 address and index registers made the machine somewhat cumbersome to generate code for. The 8-A registers were almost never used and code was needed to move data between the B and X registers as well as to move data between the various X registers so that the data could be read from memory or stored into memory. The 8600 used a general purpose register that had none of the problems of the 6000 series machine. In addition, the increased number of X registers allowed the FORTRAN compiler to generate better code for the machine. The 8600, therefore, had 16 general purpose X registers".

NEIL: Had that consideration been made in the transition between 6000 and 7000? Why not build a uniform register rather than stick with the A and X registers for 7600?

GRINNA: I'm not sure if that question was raised.

NEIL: Or it had to be compatible and that's why you stayed that way?

PAGELKOPF: You have to guess that there wasn't enough experience on the 6600 when the 7600 was launched.

MOE: That's true.

RESNICK: For one thing he'd have to change his instruction format, cause you can't have three 16-bit references in a 15-bit word.

NEIL: But again, let's take this case. He could have gone with 8 registers and used 3-bit designators and had a bigger, more generalized of code for the 8600. But 8 registers were not enough, I guess.

BHEND: I don't know why 16 was picked, it's a good number, let's face it, if you have to change from 8, but I don't know if they made some kind of a check to see if that would be how many registers they needed if they did away with B box or B registers. Or if this was just decided that this would be the next best step to go to.

GRINNA: It turned out, we mentioned here, that the increased number of X registers allowed the FORTRAN compiler to generate better code and we did do a little study later on when we were considering adding a scratch-pad memory and we asked Garner how much could we add and what benefit would it give you. This would have functioned somewhat similar to the registers. It turned out, at least for his analysis that there wasn't that much to be gained by adding a scratch-pad.

NEIL: Because you had so many registers to start with.

GRINNA: Right, he was happy with what he had. He had his code written and I suppose that affected his opinion, but there was some analysis later on, at least.

HUTSON: Isn't it true that register X7 in his compiler for the 6600 wasn't used because of hardware problems with the exchange which was never fixed and this wasn't discovered until a long time, up around serial 20 or so and with the problem really fixed, he didn't even use X7 at all?

NEIL: In the first versions of the compiler I saw, register X7 was certainly used. He alternated between X6 and X7. That would have been in 1965.

GRINNA: It could have been, Garner did program around problems occasionally.

HUTSON: There was definitely a problem related to exchange and register X7.

GRINNA: I remember the problem now. It wasn't really a design problem, it was a timing problem.

NEIL: I think you're referring to the "store out of order" problem that we discovered at Palo Alto.

GRINNA: I didn't hear that story, but it's quite possible. Ok to go on.

"Some special instructions were necessary for the operating system to use for the tie-breaking between CPUs such as the exchange/read/write instruction. A flag register was added to allow inter-CPU communication when the CPUs were all working on the same problem. The I/O instructions were a combination of the instructions on the 7600 and some logical extensions. On the 7600 LCM was used for file data. The two level memory on the 7600 turned out to be very hard to use for the user. On the 8600 all of the memory was to be considered to be LCM so the I/O instructions were changed to allow linear buffers in memory rather than the hardwired buffers used on the 7600. This selection seemed to be workable and could be extended to the 6600 type of memory access with little problem. A problem that was never properly addressed was at what speed must the 8600 channels operate. The type of ready-resume logic of the 7600 was used in the machine but it was clear that this type of logic could not be used for a higher speed disk that we thought would be necessary. An 819 disk with 12 or 16 heads in parallel rather than the 4 heads was being considered."

GRINNA: He mentions that the problem was never properly addressed but we had a channel design for a general purpose channel and I think the reason that this problem wasn't "properly" addressed is we just hadn't come to that point yet. There was a concept developed already for these higher speed channels and given a little more time they would have been there.

NEIL: What was the concept?

GRINNA: I guess that there were a couple of concepts. One was a variable bandwidth from the channels, having a 32-bit channel or 16-bit channel and thus, building up the word before transferring to the disc. So you could adjust the bandwidth to get the speed that was required. I guess that was the one that was primarily considered.

NEIL: You would still run in a ready-resume mode though?

GRINNA: That was the other thing .... we considered running in a synchronous mode where we could get rid of the ready-resume, but that was a little bit more of a problem.

NEIL: I take it you decided to avoid that, if at all possible?

GRINNA: Right.
"On the 7600 some of the FORTRAN library was resident in LCM for faster loading speed. Since all the memory was considered LCM on 8600, the resident library was in main memory. It made no sense to make a copy of this program so a library mode was added to the machine and the data came from the RA/FL area."

GRINNA: That should read "from outside the RA/FL area."
"This mode seemed reasonable and took only a small amount of hardware to implement. It was not clear just how much this actually will buy us in an operating environment however."

GRINNA: That's the end of the discussion on the instruction set. From there it goes into problem areas. Any comments before we proceed?

NEIL: We all heartily endorse memory share-ability around this place.

GRINNA: Ok, problem areas.
"At the time the project was stopped it was clear that several areas were still problems and that the solution of these problems was not in the near future."

"Artwork: The method of generation of the artwork took the entire time of the logic designers as well as some support personnel. This made the design come slowly and very tediously, however. This may be a transient observation, rather than a real one."

MOE: Stop!

GRINNA: Right, I was going to stop. Ok, obviously if the generation of artwork took the entire time of the logic designers, we would never have had any design . But that's not true. The logic designer did lay out the overall layout of the board down to the placement of the components and the development of a set of equations for the module. From that point it was turned over to the layout technicians and draftsmen. There was an interface between, they would come to the engineers with questions but at that point it took a very low percentage of the designers' time, I think.

NEIL: When you say "a board",do you mean a module?

GRINNA: Yes, a module, 18 boards that were laid out as a unit. What did we use for a rule-of-thumb, Bill? Six months for a logic designer to block out a design and to design the module, write the set of equations and place the components on the board and that was to get it to the point where it was ready for layout for drafting. After that there was still some involvement but it was not 100% of his time.

NEIL: What was the elapsed time then from the drafting to the point where you could actually cut metal?

GRINNA: Do you remember? I'd be guessing, but probably a couple of months.

BHEND: I was trying to remember ..... it seemed to me that it was like 6 weeks from the time we got into the computer digitizing process to when we actually started making the boards. I can't remember how long it was actually in layout. I would say around 3 months, or 4 months.

GRINNA: I should insert, this is something that Dave doesn't address in his report. During this time that the module was in layout there were other things going on. We mentioned that we used that 12-pack quite a bit and primarily in the generation of the artwork. We did our plotting right at the console and there was Pete Jutten. He had written programs that took the file of equations that the designer had written for the module and did various checks on the artwork to make sure that the layout people had connected everything properly. Meanwhile, Cahlender had written a simulator which ran on a 6600. Eventually, we had a 6600 over at Chippewa for specifically for this, well for two purposes. One was for software development, but the other was for running the simulator on the hardware. We could do this on a module level or a machine level. When the logic designer had finished his design to the point where he had a workable set of equations he could take that set of equations into the 6600, load it into the simulator and actually program that module and that portion of the machine then and debug it for logic design errors and that was extremely valuable, especially as we built up and got several modules designed and got to the point where we had a CPU completely designed and we were actually running diagnostic routines on the simulated machine. There were a lot of design errors that were eliminated at that point. In many cases we were able to find logic errors and get them corrected while the module was still in layout. You know we kept working towards the point where we would get far enough ahead of the layout people that we'd get them before they got there but in some few cases that might happen, but not as a rule.

NEIL: Sounds like the STAR project, Maurice, send it over to layout and then give them a quick phone call every day.

BHEND: One thing that Dave never mentions in this report is the simulator which I feel was the most valuable tool that we had as far as 8600 was concerned. It was going to reduce the cost significantly of rework on these modules because you could actually take and run diagnostics on them and find out whether the multiply unit, say for example, is going to work and it checked for long path problems and it checked for short path problems. You could vary the clock. At least 90% of the problems that you normally faced were going to be found in this program.

GRINNA: You could make changes and get the results instantaneously. When we're still sitting at the console you could change an end-case, and you could run the test to see if you detected any other end-cases which had been a terrible problem in the past. That was probably the biggest step that I saw on the 8600 project. When we got down to the checkout, we came into checkout with a whole lot more confidence than we could have ever had without that. In fact, just the last day that we were working on the 8600 at Chippewa, we had gotten power on to our prototype. Well, we had put power on a couple of days before but we were checking it out and trying to do an exchange jump at the very last point before we were called to the lunchroom for that final meeting. We got down to the point where we could see why the exchange jump was not working and it turned out that the problem was an RC module. We had an obsolete one in then, which we knew. We already had found the problem on the simulator and had ordered a new module. The new module would have been there a week later. The problem had already been solved and here we found it in hardware finally, but by this time the module was already on its way, and with the turnaround time we had on modules it would have been six months later instead of one week so I just can't say enough about the significance of that.

MARIS: Were you able to use that simulator to eliminate bottlenecks in the hardware, or were you looking for that?

GRINNA: Yes, I'm not sure if I know exactly what you mean.

MARIS: Conflicts and buffer sizes and so on.

GRINNA: That's right, you could look at it and you could look at how registers were being utilized. You could look in the instruction stack and you could actually watch or you could display the whole instruction stack and you could see how it filled up and you had access to any element in the whole computer, not just what you were able to get out on test points or pins but every inverter in the computer could be displayed on the screen, limited at any given time, but you could ask for any inverter on the machine and see it.

BHEND: You could also step through to get to any inverter.

GRINNA: One inverter or one clock period at a time, if you wanted.

BHEND: You could start at a rank of registers and go one inverter at a time and see what information came out on each inverter as you went through.

GRINNA: Ok,we'll get back to that, but let's get on with the problems here.

"Packaging: Packaging was without a doubt too ambitious."

GRINNA: There's some doubt about that.

"The modules were laid out in 18 6 x 8 inch boards stacked together. This complex module required special board testers and module testers to be built. Even with these testers it was very difficult to find several types of problems. Although for logic speed having interconnect pins between adjacent boards every one of the modules was very good. The stacking and unstacking of the modules was something to be taken very seriously. After unstacking a module and making a repair, the restacking of the module usually bent an interconnect pin and required replacement of this head. During replacement other things could be done and the cycle repeated. A successful stack was declared after a board was stacked and unstacked three times without a bent pin. On the fourth stack it was assumed to be good."

GRINNA: Ok, I'm going to stop again. That may have been true on the first module, the first time we stacked, but Bill was talking to Rollie Moore who was in charge of our assembly production over there and the actual turnaround time on unstacking, repairing, and stacking the module was down to the point where Rollie said in one case, (this is just one case where they had a known problem) they turned the whole module around in two hours. That includes unstacking it, replacing the transistor and restacking it to get it back to the engineer. I would say that a more typical turnaround time was a day. You could send it into repair and have it back the following day if there was a specific problem called out. Now, if it had to go in and be completely torn down and you had to send a large number of boards through the board test operation followed by repair than that would take a little longer. But still you could have it back within another week. So from that respect we had gotten to feel that it was easier to repair than the 7600 module.

NEIL: Those we throw away.

GRINNA: We weren't going to throw away an 18 board 8600 module!

NEIL: I guess it comes to a question which is related to detailed design problems but considering the difficulty that the inner board wiring cost, was it really worth it? How much more would you have lost if you had gone to the board-edge and to an edge-board connector in terms of time involved?

BHEND: A lot.

MOE: There was no way to get circuit speed improvement anywhere near necessary to meet the objectives.

GRINNA: It would be a whole new philosophy. It's not a matter of how much time we'd lose, it's - does it work or not. We'd have to lay out everything with a different organization. The reason we were able to operate at these speeds was the whole concept of 3-dimensional packaging. We could have a clock on the center boards of the 18-board module and have that clock fan out to 16 bits of register on the "out" boards. That was the closest distance for running a clock, was vertically through the boards rather than distributing across the board and there was very little skew in the clock between the bits in the register and if you had imposed the requirement of going to the ends of the module that would have just shot down the whole scheme.

KRUGER: The 8600 was built out of wasn't it? There was a statement in Dave's report that said you could no longer get big increases in speeds through components. What time frame was that?

MOE: In 1969, when the 7600 design requirements were 2 ½ nanosecond outside and 1 ½ nanosecond inside and the 8600 switch time for what?

GRINNA: 1.4 nanosecond.

MOE: There were no integrated circuits available in quantity at that time at that speed.

NEIL: They're hardly any now. Let's see ISL was around 1 nanosecond, wasn't it?

KRUEGER: What you're saying is that "discretes" are faster; or were faster than integrated circuits?

GRINNA: While you're bringing that up (as to how we arrived at a circuit) first we used Motorola, that is we used the same transistor that had been used in the 7600. Then ISL came along and looked pretty good and it was close. And the transistor that we ended up using was actually the gate taken literally out of the ISL and packaged in the discrete package. It ran considerably faster in the discrete form as opposed to the speed that we could get out of ISL.

BHEND: 400 - 500 picoseconds.

GRINNA: We were speccing it at three quarters of a nanosecond, I think.

BHEND: That 1.4 number too, that included the wires, the estimated wire between switches. That was what your inverter time was supposed to be including the wires that went to the next inverter.

GRINNA: Right, we were actually counting on getting about 1 nanosecond switch time at the transistor level. With the eventual transistor, we actually got a little better than that. Which, luckily, made up for some of the ground rules we had stretched a little thin already.

HUTSON: What determined how many boards you stacked to make a module?

GRINNA: I think the 16 bit orientation. 64 bits per word, broken down into 4 modules, that dictates 16 bits per module and we needed 2 control boards so we had 16 + 2.....18.

HUTSON: Did that include a complete adder for 16 bits?

GRINNA: Yes.

MOE: How many boards were in a complete computer?

GRINNA: How many boards?

NEIL: Modules?

GRINNA: 126 or 132 slots in the original configuration, 12 x 11, and they weren't all used. There was slightly less than that, maybe 126 or 127. One module for comparison was equivalent to 27 - 7600 modules.

BHEND: Four modules did all the fixed point arithmetic and had the registers on them so they were broken down into 16 bits, 16 registers on each one of these 4 modules and did all fixed point.

MOE: I think to get everything in perspective, I think the 7600 and the STAR module are logically pretty close to equivalent. That represents 27 STAR modules.

GRINNA: Bill designed the multiply for the 8600 and it had 3 modules?

BHEND: Two identical types, only 2 types of modules, but it took 3 modules.

NEIL: How many pins was that?

GRINNA: 432 usable plus the ground and power. It was an array of 30 by 18 which was 540 total.

NEIL: In test points up front you had how many?

GRINNA: 12 per board which of course was never enough.

NEIL: And there wasn't any way really to get any other probe-ability?

GRINNA: No, this module has the solid plates on the side. During checkout we left those open so that you could probe the two outside boards and that really gave you a lot more because the signals that were fed out on feed pins to the outside, you could look at those. If you had a real sticky problem, you'd solder a wire to a circuit inside the module and run it out through one of the holes in the front plate.

BHEND: You could unstack several boards and if you had a problem you felt on C board you'd unstack A and B and put it in a jig and look at the C board. So design-wise, your design checkout was difficult but maintenance isn't. If multiply doesn't work, swap the whole shebang. Inventory costs would be great, but C.E. time would be minimal.

NEIL: What would be the inventory cost of a single module?

GRINNA: Numbers .... 30,000 transistors.

BHEND: I think, components.

GRINNA: What was that cost, around $15,000? About $15,000 cost per module? I'm not sure where that was headed.

BHEND: You remember how many types of modules there were?

GRINNA: No, I don't. I should.

BHEND: I can't remember. Seems like it's in the twenties.

NEIL: That's better than we can do on STAR. Have we got an idea of how many different types there are on STAR?

KORT: Close to 500.

HUTSON: It's under 500.

MOE: 7600 had about 250, probably very similar to the 6600. About the same number as 6600.

NEIL: On STAR we did try to hold types pretty tightly until we got into the second itteration where money was no longer an object, "just get the God damn thing built" and stop trying to be common. But the start-up time and cost for a STAR module is considerably less than on 8600 module?

MOE: Well yes, but if one of those is 27 STAR modules, that's not necessarily true.

BHEND: There's one thing there where Dave made the comment that we went to slower functional units. I designed the multiply unit and I really don't see any way it could have been designed any faster using the same module structure. You could only get so many components in a module. You were pin limited also. In order to make a functional unit that was faster, I came to the conclusion that you had to build it in one module which made it impossible to do with the ones we had. You had to have all the adder for all the bits in one place to do it. So I don't agree with that statement that we went to slower functional units. We did, as far as divide was concerned because we went from what we thought was going to be a reciprocal scheme to a 2 bit divide in effect.

NEIL: Rather than use the multiply algorithm? Why?

BHEND: You ask us that question all the time. When Seymour was there, there were very few meetings where we found out the whys and wherefores of what he was doing.

NEIL: He was gone after awhile though?

BHEND: Right, but this decision was made before he left.

PURCELL: Now he is doing the reciprocal on the CRAY-1.

KORT: That's probably why he gave it up on the 8600. (laughter) Strike that!

BHEND: Who was that that worked on it? Bob Allen?

GRINNA: No, that was Mark Willis. While I was implementing the hardware Mark was coming up with the software and we weren't getting very far with it. Mark ran into some problems and kind of came to a halt and then Cahlendar came on board. He picked up the work. Cahlendar did both functions, he rewrote the FORTRAN program to check out the algorithms and interpreted it into hardware. It just kind of grow to the point where it wasn't as great a thing as we had hoped. It wasn't going to be as fast and it was going to take too much hardware. The whole thing kind of died.

NEIL: Just cause you didn't micro-code it, that's all? (laughter)

GRINNA: Probably.

NEIL: I know it's kind of off the track, but how do you guys feel these days about the hardware divide versus the multiply algorithm?

MARIS: It's not BIT COMPATIBLE.

NEIL: How about that!

CURT: Other than the emotional feeling of never ever wanting to get burned that badly again as we did on STAR 65, it's no problem. (laughter) It's still believed that the divide can be done that way, which allows you to put more money in your multiply. Take the money you would have spent on divide to have a faster add/multiply unit.

NEIL: Explain what you meant by getting burned.

CURT: The STAR 65 implemented that algorithm and because of that, it wasn't bit compatible with any other STAR machine and we had really questions on that at least twice a day for several years.

PURCEL: It caused problems for diagnostics and customer engineering as well as the compilers.

CURT: Then we went to a process of designing into the unit a bunch of very dirty, very mysterious, rounding circuits, etc., to try to make it more and more bit compatible and so it would converge from the correct end and so on.

NEIL: The biggest problem that we ran into was that somebody found a FORTRAN problem on an integer result. I think the simplest case we came up with was to take 7 divided by various integers and come up with what would be the theoretical correct result rounded in the right direction. The first time around you didn't come close on 65.

MARIS: There was an error in the lowest order bit.

CURT: So you divide 2 into 7 and get 4 so you change the algorithm so it approaches from the small side. But then you divide -2 into 7 and you start playing around with signs and convergence end-cases. Finally, we were getting down to the point where we thought we had the problem fixed and then we'd run tests which would find one in a million errors.

GRINNA: This was similar to what was going on, on the 8600 with the divide. You'd keep adding bits to make the accuracy better, but that meant bigger adders and data paths.

CURT: All we thought that was necessary to solve our problem was to identify an integer divide instruction and then we would have implemented a "divide and multiply back" and correct the remainder.

KORT: You should have hung a "G" bit on it. If you wanted the correct result, it would run real slow.

NEIL: That's the STAR story - just hang a "G" bit on it.

HUTSON: Right, we would never have built the machine if it didn't have "G" bits.

NEIL: We would have never been able to palm off some of the junk we have had we not had "G" bits. Need a new option .... get another "G" bit. That's the nice thing about 64 bit instructions, Dennis. You guys worked too hard to try to keep your instruction sets too small. "A customer coming in would say I can't do this function that I need". "Hang on just a minute, we would say".

CURT: So how did you feel about that algorithm?

NEIL: Man, I've had it with divide units, high speed and low speed. Dan, there was our floating point designer. Dan, given your "druthers" now, would you pour your money into this multiply unit and use a divide algorithm or would you still build a high speed device?

DESMONDS: If I could do it with a multiply unit and stay out of the integer problems. (laughter)

NEIL: IBM does it. How do they get away with it?

CURT: We kept clinging to that argument. (laughter)

MARIS: They kept their integers and floating point separate so it may not have given them the problem. Schemes which are used for addressing that required division could come up conceivably with one in a million wrong addresses.

CURT: The solution was to identify an instruction as the integer divide. And then only use that instruction where you had to do an integer divide.

MARIS: And wasn't that also implemented then in the software in case you had to do an integer divide, you used a specific code sequence?

NEIL: We came up with an algorithm which said you always do your integer divides this way and at least then we could identify the cases that would fail. There were two kinds of failure. One was that the 65 didn't get the same answer as the STAR 100, the second was that you were not getting the theoretical FORTRAN integer answer. How did IBM avoid the difficulty?

TOTH: I'm not sure they did. IBM's had problems with 1 equals 2.

NEIL: We'll get to that shortly on STAR. 1 equals 2, 1 does not equal 1.

PURCELL: 1 doesn't even equal 0 sometimes.

NEIL: Sometimes. That was probably the biggest shock of my life on STAR when I wrote my first simple program and it didn't work. It took me three days to figure out why it wouldn't run and it was because 1 wasn't equal to 1.

GRINNA: We're still on packaging.
"In an attempt to cut down on logic space, registors were placed in transistor cans and were used as center-tapped resistors with one-half of the resistor used and two halves wired in parallel or in some cases the two halves wired in series. It was very hard to determine if one of the circuits with two halves wired in parallel was bad when

when one-half the resistor was open".

GRINNA: I know it sounds kind of confusing. Sometimes we would encounter problems that are hard to find, but it wasn't an overwhelming obstacle.

"Top level engineering talent was used to debug the component failures in modules and the debug time for a working module took from several days to over a year in some cases".

GRINNA: We did have one module that we kept recycling through and I think everybody agreed that the module was never going to work but we just kept pushing it through anyway because we were learning things about packaging and about assembling and disassembling, about transistors, resistors, and feed thrus. We'd look at it for a couple of days and check it out and send it back and it would be gone for a couple or three weeks and it would come back again and again; though that wasn't really a problem either. It's just that it was early in the program.

BHEND: The one thing I don't think Dave has really understood because he hadn't been associated with a project like this before was that top level engineering talent (if you want to call it that) is always required at this stage of the game because you don't really know just whether you have a circuit problem or a logic problem or if it is just a component failure.

NEIL: Isn't that true of every machine we built?

MOE: You could say it wasn't required but it certainly was the mode of operation at Chippewa. You could go back farther. 6600 and 7600 never had any design equations, never had any design automation of any kind. The circuits were laid out, the logic was laid out, the module mylar taping was done by the engineering staff, the modules were checked out by the engineering staff, most of the chassis were wired by the engineering staff and that's the way the organization was run. There wasn't a structure of workers and worker bees, there was a structure of a very small number of engineers. It gives you a hell of a good background and hell of a good understanding of the computer when you look at it through that level. It maybe is expensive, and everybody gets credit for redoing the artwork a dozen times. A lot of things like that happen, but that's the way it was done in Chippewa.

BHEND: The one thing he makes it sound like here was we had engineers in there sitting over the top of modules trying to figure out why a board didn't work. That wasn't true. We had technicians in there that were doing that.

NEIL: You were in better shape than we were.

BHEND: And it was only in extreme cases where we were called in to help them out (you know, if they had a problem that they couldn't find). But in general, using a board tester they were able to find all the problems on the boards.

GRINNA: Sometimes they'd come and say "we can't find it." "How much of this do you want us to shotgun"...that would be an engineering decision.

"We kept hoping this debug time would increase but it was not obvious that it would".

GRINNA: (Decrease) .... I think it already had decreased significantly.

"The job of pinpointing a problem when you were unable to put probes even in the general area of the problem was very difficult.

GRINNA: True. But I think the reason that these modules were turning around was not that it was repetitive. There's kind of a hint here that we kept sending the module back through and looking for the same problem. That didn't happen very often. Usually you found the problems that were logical, the engineering tested a module at a module level and they called out the problems and sent it into repair and very seldom did it come back with the same problem. It made many trips around but it was always new problems. That was basically a problem of reliability in the components and that was improving. But you know, once in awhile you had some problems that you had a real hard time pinpointing. And that's where we resorted to things like I mentioned before, adding some extra "test points".

MARIS: But you only had access to what you have here on the outside, right? So you would have to pinpoint the problems using that.

GRINNA: With things distributed the way they were one bit and its entire path were normally on one board so the C board would be bit 2 and it was failing on an add and yet it worked on a boolean operation. That says the register's ok, even though you couldn't look at the register since there was no test point. You said well, it must be in the add since boolean worked. Here you could call it out and say ok look at this particular circuit and the technician would look at that and if that wasn't the problem he'd look at stages on either side and 9 times out of 10, he'd find it.

NEIL: How many modules did you have that were primarily control?

GRINNA: Two for the central processor. The "IA" module and the "IW" module had the instruction address stack, the instruction word stack and all of the control for the central processor.

NEIL: That's roughly equivalent to 56 STAR modules.

GRINNA: That was the 12 word stacks and all the control. Now the memory control was two more modules.

NEIL: It appears to me that those would be the tough ones to debug. How many control modules did we finally end up with on STAR?

HUTSON: If I ever knew, I quickly put it out of my mind.

NEIL: It's got to be in the range of 200 or so 60 paks.

CURT: In trying to troubleshoot some of these more complicated problems, did you try going back and using your simulator to say change the equations? Thus you could simulate the failure and see if it showed the kinds of symptoms you were seeing.

GRINNA: We always talked about it, {that's what we could do}, but I don't know if we ever really had to resort to that.

BHEND: No, we never got to the point where somebody just finally came up and said we can't find the problem on this module. We had board testers and we had also a computerized module tester.

GRINNA: The board testers were computer-driven too.

BHEND: Right. And one of the programmers was designing the program which was going to take the equation for the modules and break it down so that you could test all the combinations for this module by using this program. That is where we felt it would be a real benefit as far as picking these things out.

GRINNA: The tester that we had for a module level was again driven by this 12-pack that we had there. You sat at the console and checked out the module, you would feed in the inputs and outputs. It was hand written, you'd key in what you wanted to feed into the input pins and you'd key in what you expected to get at the output pins. So this was where a lot of that "top level engineering talent" came in. It was in writing these tests and like Bill mentioned, it was an effort to replace the manual work with a computer program. But that was not operational yet. Maybe we would have had all the tests written by the time it was ready, even so, it would have been good because we would have had more confidence in that. You were never really sure that you checked every gate 100% through all possible modes. Ok, let's see.

"Components. A significant amount of problems came from the components used. It is not possible to ascertain if the problems associated with components were solved or not. The lead time on putting feedback into the components was significant, over a year. We had resistors that opened up, transistors that opened up, plated through holes that opened up and feed pins that opened or shorted to other boards. We tried to burn in our transistors as an experiment to determine if we could get rid of infant-mortality failures. We did not run this as a controlled experiment so we cannot say whether this improved the transistors or had an adverse affect.

GRINNA: Well, I didn't look on that burn-in of transistors as an experiment at all. To me it was a proven technique. It did improve our results. All these other problems that he mentioned, it's very true that they were all there.

BHEND: They're always there.

GRINNA: They're always there and I think there was so much improvement from where we started to where we had ended up that I was confident at least that they would all be solved.

BHEND: He makes it sound as if there wasn't any improvement there but we had started to keep records of component failures. In fact all failures on a module were recorded, exactly what was found, whether it was a short because of a solder bridge or whether it was a component that was turned in the wrong direction, things of that nature. Good records were being kept of this and I talked to Rollie Moore who was in charge of them, just before we came up here, and he said that at the time that the project ended most modules that came in only had like one or two transistor failures. Now this is out of 18,000 maybe, per module.

GRINNA: Approaching 18,000.

BHEND: Right. And about twelve total errors were found on the first tests of a module. This included all types of errors like feed thrus, solder bridges, turned components...all those things, so it had significantly improved at the tail end. That's why we were able to get modules working as a matter of fact because we had a significant improvement.

GRINNA: There was a time on the 7600 too, when we were convinced that we would never be able to get things to run. In fact statistically we had proven that it would never be possible to get that many transistors all working at the same time and if we'd continued with the same transistors I guess we never would have; but along came Motorola with some transistors that did the job.

MARIS: Were you plagued by intermittents at all on these modules? It's not mentioned in Dave's report.

GRINNA: No, it's not mentioned. There were,I wouldn't say we were "plagued". We had intermittents, particularly in the feed-thrus where they would open up. These were mechanical contacts between the boards and in some cases the bundle of the connector was squeezed down so it wasn't making contact with the female portion attached to the next board we would have some intermittents and we could get them to work by squeezing a module and different things like that. But those pretty much disappeared as we had improved the feed-thrus and component-wise there weren't too many intermittents.

NEIL: I don't think we really were ever bothered by a true circuit intermittent. Most of ours were really cold soldered stuff, right?

HUTSON: Not a whole lot of that either. Cracked foil was probably the hardest one to find.

NEIL: It seemed to me that when a circuit went sometimes it went to a 1/2 level and that sometimes gave the appearance of an intermittent situation but they didn't intermittently fail or not fail--except the God damn filters in the memory.

HUTSON: Well we had problems with differential lines, one side of the differential could be open and the thing would still work 99.9% of the time and they were hard to see.

GRINNA: David mentioned this on the resistors where we had parallel resistors. This happened sometimes there too. If you had one side of this parallel resistor was open it just changed the voltage level and sometimes it worked, sometimes it didn't. I think there were probably other cases but usually when we observed this it was a case where we had seen a problem on the module test, it had failed there and they got it in "board" test and they couldn't find it. The board tester wasn't showing it up; but as they got in deeper and started digging around they'd see some funny voltage levels and they'd dig it out and it finally turned out to be a resistor that was open. Maybe there were others that were open like that and they ran on the module level too. We hadn't gotten far enough to know if that was going to be a problem or not.

NEIL: Did the STAR-65 have similar problems?

CURT: Not intermittent problems caused by circuits, but 1/2 open differential lines were a problem, as has always been the base with TCS and we had some modes of failure in some of our " buffer" circuits that were intermittent in nature. One mode of failure of buffer circuits would be that data left in there too long would decay so that we had some buffer circuits that worked very well as long as things were going in and out of them at a high rate of speed; but when you get a certain backed-up memory conflict situation where the buffers would back up and the data would decay in the buffers.

NEIL: Were those the 16-bit buffers or were those the same register-file buffers?

CURT: They were not any chip you used on STAR, it was a 16-bit chip we got from the 3500 project.

NEIL: I've got one of those things. I stole a board from the 65. I couldn't believe the circuit that was on there. Where'd this come from, right in the middle of the board? This deviation you see is from an operation which was going to be absolutely strictly limited to TCS circuits and that class of circuits. Bob Wesslund says "we won't have any hybrids in STAR." How long did that last? First, second or third month of design?

CURT: There was <u>no</u> point in the design when we <u>weren't</u> going to use that circuit.

HUTSON: I always remember that Ron Hintz came around and asked us if we had any need for a TDX with a not output. Everybody says heck no. Once he put that in our mind you know it wasn't a week later that we found a hundred places to use it.

DESMONDS: That wasn't me that said no.

HUTSON: I was standing with a group of people at that time when Ron asked that and everybody said no to begin with and it came back to haunt us.

NEIL: The lack of a NOT output from either of our OR gates had to be one of our really pernicious blunders.

HUTSON: Getting back to the intermittent thing, actually on STAR-100 the open ground on the single ended wires I think was the hardest thing to find because it had a tendency to do funny things with the signal while it was in the wire. With tight enough timing it caused intermittent failures and the depth of the wiremat made it almost impossible to examine. You had to go and spend 20 minutes to digit all the way out to find out if you really had a problem there. It was kind of like your packaging problem. You had to spend a lot of time and effort just to find out if you were going in the right direction or not.

NEIL: I think in a way we shifted all of our problems to the wire mat. How many weeks did we spend on the 03, 04, trying to win that bottle of booze?

HUTSON: That's an interesting problem. It was R.F. noise being generated by the power supply radiating up into the backpanel. It had the effect that any circuit which had a slightly lower than normal logic output and went through the backpanel continued to be the weak sister you know, it picked up the R.F. noise and failed first. I think we spent about a month going through isolating intermittent problems and discovering that instead of the logic "1" being 1.8 milliseconds, it was 1.7 or 1.5, something like that. They'd change the chip and sure enough that would fix that problem but it still wouldn't have the test so we'd just go to the next one, and eventually found some bad diodes in the power supply.

NEIL: It wasn't bad diodes, it was a mismatch between two different brands of diodes.

HUTSON: That hasn't been determined. Evidentally there's a problem in inspecting the diodes. It was two different manufacturers and they were a mixture; there's 12 of these diodes and five of them were of one manufacturer and five the other and the mixture didn't work very good. It generated voltage spikes at the transition point. It was cutting off from one diode to the other which made the spikes 20 to 30 times normal.

NEIL: A circuit inventory then discovered that the optional wing of memory on serial 2 was about the same way, 50-50.

HUTSON: I think they found five of them on there.

NEIL: Just a different brand caused a problem. Well, why don't we go ahead with your discussion, Dennis.

GRINNA: Getting at the cooling next...of the problem areas this probably had the most time spent on it and had the least results. The basic cooling of the 8600 module was the conduction of heat through a thick ground and power plane out to the edge of the board. On the edge of the board a copper wedge was fastened with a thermally conducting epoxy. This wedge mates with another wedge which in turn makes contact with the cold bars. The cold bars are heavy aluminum, cooled with evaporating refrigerant that flows through copper tubing imbedded in the cold bars. The idea of this cooling method is to have the pressure in the copper tubing such that the refrigerant is boiling at a relatively high temperature, 60°F. It had thermal resistance to the heat sources low enough so the components are not too warm. Too warm is about 180°F junction temperature. The system was originally designed with relatively low margins on the thermal resistance. This low margin was eaten into by several factors. More current was used per board than was projected. The thermal drop at the interfaces was greater than projected and no tolerance for mechanical tolerance was made in the design. Larger holes were cut into the ground and power planes than the original design called for. These holes were in attempt to solve another problem. With the heavy ground and power planes a component could not be unsoldered without heating the entire board. A plated through hole to power or ground was changed to be a combination pole and tab. The tab was a small tab with some thermal resistance allowing the soldering of the component to one of the stronger power holes. Another factor, also affected the cooling of these modules. The cold bars were assumed to be rigid, however, with the wedges tightened and power turned on the thermal expansion forces were extreme and the cold bars moved or bowed making the already poor contact worse. The insertion of a cold module into a row of modules would move the cold bars away from the other modules as soon as the new module heated up. We went to individual wedges for individual boards in a module in an attempt to get better contact with these bowed heat sinks. This provided some help but was really not a solution to the problem.

GRINNA: I'd just like to comment on that. That was not really the reason we went to the individual wedges. The reason for going to the individual wedges was the mechanical tolerance he had mentioned earlier where if you could have had one large wedge mating with nine small wedges you can imagine that each wedge would have to meet just perfectly with that large one or it wouldn't make contact at all except for a few points. By having individual wedges to mate, you had more contact.

" The calculations showed that the 8600 module could be cooled by forced air cooling but no experiments were done to confirm these calculations."

GRINNA: Okay, those calculations were made, I might add, by Cahlender himself and none of the mechanical engineers would really take them seriously because they'd been through that route before. As I understand it the calculations showed how much air it would take to remove that much heat and on the surface it would seem to be feasible. However, I don't think he took into consideration the amount pressure it would take to move that amount of air through the restricted spaces that we had in that module.

PURCELL: Or the frequency of the whistle that would result.

GRINNA: There were a lot of things like that. For example, what kind of heat transfer do you gradually get from the transistor to the air stream.

NEIL: If he'd ever seen the ILLIAC IV he would never have suggested it.

HUTSON: It's a bad environment; because you get into problems of mechanical vibrations and cleaning the air. The air has got to be brought from somewhere and is usually loaded with oil and dust. It's a bad environment for any type of electronic gear.

NEIL: In addition to that when you consider the relocating that you're talking about to move that air creates all sort of interesting little wind tunnel effects. One transistor stacked behind another doesn't get any air. All sorts of problems arise with baffles and other mechanical stuff in there. You'd actually get fluid dynamic effects because of the relocating and pressure of the air stream. Just ask the ILLIAC IV people, the biggest mistake they made was not freon cooling that beast.

MARIS: What was the power dissipation of that module?

GRINNA: About 600 watts. Some of them were higher.

PURCELL: Was that one watt per transistor?

GRINNA: Not quite. {What was the design, , 150, 180 watts?} The original power supplies were designed to go to 180 amps at 3.2 volts, then we ended up going to 3 volts and I think we had modules that were as high as 240 amps so there was a lot of heat there to take care of.

CURT: Was that expansion problem really serious? Did you have to cool all the modules off before you added a new module?

GRINNA: No. As far as the expansion, I was never convinced that it was really thermal expansion that was the problem. We hadn't really run any tests on it. It's true that when we plugged in a module and tightened these wedges that you would move the bars but then if you'd go back and tighten all the modules (which was not an insignificant task) you'd finally arrive at some point of equalibrium where they were all being cooled. What we tried to do, of course, was use torque screwdrivers so that you would not tighten that new one too tight and cause a problem and we normally would plug in a module and not go through the whole ritual of going through all the modules. You might change the temperature a few degrees but it wasn't significant and there were mechanical engineers looking at making the bars more rigid and there would have been some changes in that area.

NEIL: Was there any thought of putting any kind of sensor in the hot spot so that if you failed to get proper cooling you'd get a warning?

GRINNA: Well, we did have that. There was a thermistor placed on the "hot spot" of each module and this was brought out to the distribution cabinet. If the module got too hot it would first sound an alarm then turn the machine off.

NEIL: Also you could make your power pins out of a metal that melts at 185°.

GRINNA: We had some solder that melted at that point for a while. Sometimes the components became a little loose on the board. That's not so funny. That was one of the reasons why we mounted the modules horizontally. We had always planned to mount modules horizontally but we made sure that the components were on top so that if they did get too hot the transistors wouldn't fall out. {Laughter}

O, let's continue. " Experiments were conducted with additional plates between the boards in thermal contact with some of the soldered blobs. These experiments indicate that this would be a satisfactory method of cooling a module.

Inter-connection between modules: The 8600 modules were connected together with coax cable. A pin was designed that could be soldered onto the coax cable that would be plugged into a connector block that the module would in turn plug into. The wire that we used had a very high propagation speed, 10.4 inches per nanosecond. However, the wire did not lay well into the back of the connector block. It was hard to plug in and it was hard to change a wire. Several inches of wire were taken up in going into the connector block and making a bend out of the block. If smaller, more flexible wire were available it should be used."

GRINNA: That's obvious, but as far as plugging wires in and out of the connector block the new connectors which were designed and ordered permitted removing a wire from the front side, you would remove the module and push the wire out from the front instead of digging through the wire mat to get at the wire. Also you'd send a little guide to aid in putting the wire back in which I think would have been a big improvement in this area.

NEIL: How many wires per connector block?

GRINNA: 540.

NEIL: Was there any reason for having so many per block?

KORT: That's not too many pins for that big module.

GRINNA: We ran out sometimes. Bill said there also could have been more pins; you could have done different things with multiply and other areas where you needed interconnections. There was just an awful lot of logic in there and depending on where you chose your major interface you could control the number of pins you use. We tried to minimize the communication with other modules but even at that I think some areas that it just built up. I'd say that most of the modules used most of the pins so there was a pretty heavy wire-mat there.

BHEND: I don't think it was any more difficult than diggint into chassis six on the 7600.

NEIL: I think I've got one better than that one. How about the JB panel on STAR 100?

RESNICK: Chassis six was worse.

GRINNA: The only thing with the 8600, you might have had to crawl inside a barrel to do it.

BHEND: Yeah, right.

GRINNA: All right, it wasn't that big a problem. Ok, we'll read a little paragraph here on software.

" A significant amount of software work was done on the machine. This effort included the FORTRAN compiler, assembler, utilities, simulators, and some operating system. The only problem that appeared in the software development was that it went too fast and got ahead of the hardware development. After the software work was done in order to check out the FORTRAN compiler and library several benchmarks were run on the simulated 8200 (which was a simple CPU 8600). These programs ran from 2 1/2 to 3 times the speed that they ran on the 7600.

Ok, in summary.

The 8600 experiment was an experience in the design and manufacture of a very large fast computer. The techniques used apply to different instruction set machines and the lessons learned should be valuable to CDC in the future."

GRINNA: That's the end of this report. Lynn had asked me to write and talk to some of the other fellows and prepare a majority report. Bill and I and some of the others have talked about this and have even hesitated to do what we did today. It seems like almost a matter of trying to save our own pride here, defending what we had or had not accomplished. I think there certainly are lessons to be learned. By talking about them in this way and bringing out theproblems we can all do better next time. Hopefully I guess that's what this whole series of meetings is all about. To talk about the problems we've had.

PURCELL: Was the software really in that good a shape?

GRINNA: Well yes it was. Garner got to the point where he just took a leave of absence. FORTRAN was ready to go and he had nothing else to do so he left. These other routines that Dave mentioned were checked out on the simulator. The operating system; Greg Mansfield was working on that and I'm not sure what state it was in but it was taking shape at least. I'm sure that by the time that we had a machine that would have been ready to some preliminary work with at least. Certainly the software was in a lot better shape than any machine that I had ever seen at that stage, I think it's safe to say, better than any machine that CDC ever built.

NEIL: With the major exception of STAR 100.

GRINNA: Maybe not. With some developments there were machines that came where software existed but for a new development I think it was pretty far out.

NEIL: What would you say were the lessons to be learned out of this. We've had Dave's observations.

GRINNA: Well, I think I probably commented on what my strongest feelings were as I read. But if I can go back and repeat, I think that the first lesson I would mention was the great difference of opinion that was shown here and that is a continuity of personnel on development projects where you can see a natural succession of the machine designs from like 6600, 7600, 8600. Each one was a growth from the one before and the personnel stayed somewhat the same. There was a lot of benefit from this experience. I think that too often a design team is successful in one program, you make managers out of them and they never design another machine. As a company philosophy it would be better to keep these guys designing machines, which is what happened at Chippewa. But beyond that I think the techniques that were developed on the 8600 were really significant. I think we had gotten to the point where we felt we could not design another computer without the aid of the compute power that we had at the lab (that 12-pak and the 6600 and the two other 6-paks that we had in the house were going full time and I think being utilized very well). As far as generating our art work and checking our equations, checking our art work, checking the logic itself, maintaining the records of failure data, coming from a point on the 6600 where if it was done at all it was done by hand or kept in somebody's log book. We just couldn't have built the 8600 with that kind of technique. There's a big evolution there.

BHEND: We tried to visually check these boards at first before we had a computer program to do it...to take each board and go through the tape up and find out if there were any mistakes on it. We found that it was almost impossible looking for it like that to find all the mistakes. You could go back through them 3 or 4 times and still come up with something that you'd missed. It was just so complicated.

NEIL: Bill, do you have any comments?

BHEND: I've been trying to think of a reason for change, you know; look back and say that if we had to do it all over again what we'd do differently. I really felt that on the 8600 we had put in a lot of changes for the good and we could look back on the 7600 and the 6600 and say things that we should have done that we didn't do, documentation probably on the 6600 was the main thing, things like this. In the case of the 8600 I felt that we were coming along real good and I felt it was going to be a success.

GRINNA: There was really a major emphasis on the8600 on reliability. You mentioned this on the question of "burning-in" the transistors; but it was more than just that, it was an overall philosophy that we really had to pay attention to what we could do to improve the reliability because there was a great fear that these modules would be unrepairable which turned out to be not true. There were a lot of other things that we did, burning in the transistors was one of them that cost quite a bit of money but it was worthwhile. Other things we did before any components were put into a board, a girl would sit and "ohm" each pad on the printed circuit board, checking for continuity to ground and power. The components insertion was somewhat mechanized. The machine didn't insert the components, it put a beam of light on the three holes that the transistor was supposed to fit into. It was by using these techniques that we got to a point where we had new modules coming off the line with 2 bad components and 10 assembly errors and this reduced our infant mortality and allowed us to get the modules checked out but I think it would have paid off even in the long run. There was an order of magnitude of difference in the effort that was made to improve reliability.

BHEND: We had our "shaker" too.

GRINNA: Oh yes. We would mount boards and modules on a shake table and try to break the solder joints and feed throughs. In fact we could mount a board on the shake table and vary the frequency and amplitude of vibration while the computer was running a test on the board. We were just looking for failure modes that we had overlooked before. I don't know that we found too much that way.

NEIL: Was there a projected set of numbers for the expected mean time to failure?

GRINNA: Some study had been done.

BHEND: I talked to Rollie about that today and really couldn't remember what they did but they were keeping track. They had come up with a mean time to failure;they were watching it very closely to see what it was going to be, what was projected. But he said that he had seen this stuff at home and forgot to bring it down. He did have it at home.

GRINNA: There were some theoretical studies based on the 7600 failures that showed that it wouldn't be too bad. {I can't even remember those numbers.} Basically the transistor was the same as what was on the 7600, and other things were similar. The study showed that because of the larger number of components we would have a pretty bad mean time to failure which is why we concentrated so heavily on reliability. I think we did improve our reliability.

NEIL: Do you think you can get that number for us next time? That's the thing that of course kills "big machines". ILLIAC IV is shooting for 4 hours. The TI machine is shooting someday for 30 and the ILLIAC is giving less than 2 and the TI machine is giving less than 6. We predict that STAR will be 100+ hours someday.

PURCELL: We're getting about 20 right now between parity errors.

ROWAN: How do you feel about discretes versus integrated circuits? Would you do it over again with the discretes?

GRINNA: Starting when we did, yes, there wasn't really any choice at that time. Now, of course, that picture has changed. I'm sure if we started it today that we would do it with integrated, but to get the speed back in 1969 there wasn't really much choice.

BHEND: I think the idea at the lab was that possibly the next thing we'd do after we got this machine running would be to integrate this machine.

ROWAN: Which was what Seymour started to do.

NEIL: Seymour did do a couple of things. He backed off on clock and changed, if you will, mode architectural innovations and changes.

PURCELL: He went back to the single accumulator.

KORT: He always started out trying to do the perfect thing and then he had to back off because of problems.

NEIL: Does that imply that these were things that Seymour felt needed changing after he got that far along with the 8600.

BHEND: You see now, the 8600 clock was faster than Seymour's clock. Even his projected clock. He started out; he figure on a 10 nanosecond clock, and we were running 8 and he backed off to 12 I believe.

NEIL: He was talking 14 for a while.

BHEND: So if you look at that now I don't know if that's a true statement or not but if you look at that that says that in order to get an 8 nanosecond clock you had to use discrete circuits because if Seymour could have got a hold of integrated circuits that would have gone that fast he would have used the 8 nanosecond clock.

NEIL: But on CRAY-1 also he has loosened up on his packaging inter-connection.

BHEND: He went out to the edge of his boards, he didn't use feed throughs.

NEIL: That allowed him to be a bit sloppier with his clock but then to get his speed he had to use some other techniques and so in the extreme he uses vectors to get the maximum performance rate out of that machine, thereby vindicating STAR to some degree. I'd like to back up and bring the STAR story up to date then we've got all the machines described.

TOTH: I'd like to ask a few more questions regarding the architecture of the machine, whether or not anybody was able to use the 4 separate processors effectively, even in concept rather than in practice.

GRINNA: No

TOTH: Secondly {I don't know if that's an adequate answer}, I'd like to also find out if it was felt that the memory size might be rather limited since you have 4 processors sharing it.

GRINNA: Okay, first question. No, nobody had come up with a means for utilizing this 4 processor for true multiprocessing. In fact our approach as we spoke to customers was nobody at CDC knew what to do with it but we just had a feeling somewhere, someday, somebody was going to come up with something and we said, we put in some hooks in the hardware that should facilitate what we think somebody might do someday and you're free to use them if you can find a way but we hadn't heard of any applications of those.

TOTH: Garner's FORTRAN used only one processor to do the compiling?

GRINNA: Right. The memory size, yes, we'd gotten feedback already of course as you always do, that it was too small and we had looked at the possibility of expanding the memory. Well, we looked at it in 2 stages here because we had decided that we could expand to a million words theoretically and it would involve making some hardware changes. With our present techniques and everything it would theoretically have worked but we weren't really sure we could do it. We were limited by wire lengths and just how far out we could reach with it but we said okay we're just designing for a quarter of a million words right now. That is the first machine and after then we can look at expanding. That pretty much is what was done with previous machines, the 6600 and ECS. But then also we had made a decision really to change from a core memory to a semiconductor memory and this would have given us a reason to hope that as the technology improved in the memory chips that eventually we would have been able to get larger chips and package a larger memory in the same amount of space so that we would not be limited by wire lengths. As far as expansion, that's where we were kind of pinning our hopes. We were thinking about it but nobody was ready to design it.

MARIS: I have a question on maintainability. Did you consider parity inside the machine?

GRINNA: I guess it was mentioned but no we didn't really consider that seriously.

TOTH: There was parity in memory though?

GRINNA: There was parity on memory and of course for the new memory we were designing we were putting in error correction but our experience with internal data transmission is there wasn't a high enough failure rate to justify it. That's not much of an answer, it's just a consensus.

MARIS: Currently we put a lot of emphasis on putting maintenance features inside the central processor. I was wondering if you gave any consideration for such things, such as failure isolation although you had the large module that could be pulled out and replaced so you didn't really have that problem as much. There wasn't really any consideration given overall anyplace for such features?

GRINNA: No, I can't say there was. I suppose where that idea came from was probably an input from customer engineering like why don't you consider this but we didn't do anything, there was no study done or anything like that.

NEIL: Let me ask Chuck to recite the liturgical chant of the inception of the STAR project. Think back now to 1965, 1965 and bring us forward. We now have gotten to the 8600, let's go back and pick up the STAR pieces.

PURCELL: Probably the first thing that happened was when Dolan Toth went out to Livermore and attended the meeting which Dr. Slotnick chaired where the requirement for........

TOTH: Let me correct that, he didn't chair that meeting.

PURCELL: He didn't chair it? Who chaired it?

TOTH: I think it was Sid Fernbach. In fact I don't remember seeing Slotnick there.

PURCELL: Well, the sense of the meeting at any rate was to describe to an eager group of possible manufacturers of computers the requirement that Dr. Fernbach had for a very large scale computer that was going to be a lot faster than the 6600 computer that hadn't even been delivered yet. This was in May 1964. They described the Solomon computer concept of Dr. Slotnik's. The Solomon computer had 256 PE's (processing elements) all supposedly joining together to compute on one problem. May of 1964 was just a month following the introduction of system 360; I believe that was April of 1964. It set the stage so that the system 360 is now introduced and Dr. Fernback was talking about his computer needs for a very large scale computer and we had the delivery of the 6600 in August of 1964. Among other things that were happening in that period was that Dr. Chou at University of Maryland was a consultant to Control Data at that time. Dr. Chou had the idea for an associative accumulator. (I can't say it quite the same way he said it.) This associative accumulator had again 256 registers which were filled in parallel and then he peeled out the data serial-by-bit and did tests for comparison of 256 elements in parallel. In addition to that background then later on in the fall and early in 1965, manufacturers began the use and abuse of the word time-sharing. IBM had what they called the great time-sharing machine, the 360 model 67, and Honeywell (it was GE then), had their Multix computer in the GE 645 and Control Data dabbled a bit in that area with some extensions to the 3600 so that everyone was concerned with time-sharing, trying to describe what they had for sale by using the term 'time-sharing'. That was one way of thinking about it. Really there was something behind time-sharing and that was to design a computer that would provide a resource to the maximum number of users with the minimum of contention between each user. So let's see, IBM had the 8-bit bytes on system 360, we had time-sharing considerations, we had data processing considerations with Dr. Chou as well as system 360 and we had Dr. Fernbach's requirements. We had then a request (not a proposal) for information from Dr. Fernbach. He requested information on what it would cost Control Data or what would CDC charge Dr. Fernbach to build such a machine. So Control Data responded to this first request for information from Dr. Fernbach with a machine that was referred to as the 7400 that Les Davis had been thinking about. It had 6400 sort of logic, with 7600 sort of hardware and we put down on paper a 4 processor 7400 sharing the memory and using a simplified arithmetic unit and using the 6600 instruction repertoire and we sent off this response to Dr. Fernbach with this information and we got the response back to his proposal, his proposed system, Dr. Fernbach said, thanks a lot, your concept is interesting but we're going to explore the 6600 instruction repertoire adequately and thoroughly with the 6600, and we don't need to explore that repertoire with this sort of an architecture. Interestingly enough the 4 processor idea showed in the 8600 sooner or later. At any rate the concept came back from that session with Dr. Fernbach and about that time, the spring of 1965, Los Alamos was sort of sitting there with another requirement for very large scale computers and IBM had won the competition for delivery to them of a system 360, model 91 or 95, (2 of them). Los Alamos was in the process of negotiating with IBM for the delivery of this system 360 model 95 to Los Alamos. I went down there to Los Alamos anyway even though they were negotiating with IBM and asked them what kind of computer would they like after they were done with the system 360's. Jack Whorlton and Ben Carlson of Los Alamos had some papers which described their ideas of how a computer should be built, how the architecture should be built. Ben Carlson had some ideas about the arithmetic and the floating point format and the indexing capability (the true index, etc.) and Jack Whorlton had some ideas of what sort of machine he would build if we were to exploit extended core storage and place the instructions in extended core storage, execute from extended core storage and try to exploit the potential bandwidth of extended core storage. So to put all of it together, we took the 8 bit byte, the time-sharing ideas, the requirement of Dr. Fernbach, the data processing ideas, the arithmetic ideas from Los Alamos, put it all together in a box and ended up with the 64-bit version of the STAR computer. We made a proposal to Livermore, submitted that to Livermore in January 1966. This was in response to another request for information from Fernbach and he said fine, you've got a great idea there, looks like it will meet our requirements but we didn't ask for that kind of machine. We asked for a parallel processor like ILLIAC IV therefore your proposal is thrown out. Dick Clover sort of rebelled at such a response and said we met your requirements. You're not really supposed to tell us how to build the machine. So they reopened negotiations and we submitted the proposal a year later to meet his specific requirements, specifically required was 40 million floating point adds per second in 64 bit modes. That was the specifics of the requirements. Everything else sort of washed out if you could make that speed. So we did start to meet this 40 million add per second requirement. In addition Dr. Fernbach said he would like 32 bit capability in the machine and he would like a hardware square root, Dan Desmonds. Those were his only two specific requests, and he almost crucified Dan Desmonds with those two ideas. At any rate that was Fernbach's specifics, the 32-bit and the floating point square root. So we rebid again, now finally in the form of a response to a <u>request for proposal</u> from Fernbach in January of 1967 and it turned out about June 30, 1967 we were informed of the intent of AEC to negotiate a contract with Control Data for delivery of the STAR computer. The project started in July of 1967, the contract was negotiated by May of 1970 (they weren't all too swift either) and the delivery was done in fall of 1974 of the machine. So that's some of the ancient history of the contractual relationships and sort of the intent of the machine was to fill these varieties of requiresments.

NEIL: We should indicate that Slotnik had pretty much convinced not only Livermore Laboratory but several other customers that the only way you could get this kind of power was with parallelism, considering the circuits of that time. As for the Solomon architecture he had really trotted around and carpetbagged that, partly because he had his hands on money from the Advanced Research Projects Agency (AREA) but he needed 2 things. He needed a home and he needed to demonstrate a viable end user. The upshot of this was that Fernbach set out to get an ILLIAC IV. The normal procurement process in the government is somewhat like this: A guy with an idea goes in and gets somebody in a government agency to buy it, the government agency then writes a request for proposal specifying exactly what this guy walked in the door with and generally after the proper time has passed and everybody has submitted proposals the guy who came in with the original idea gets the award because he obviously is the one that they modeled it after. This was rather an unusual circumstance in the case of STAR because that didn't occur, although it almost did. Texas Instrument Corporation (TI) was also involved in that procurement and they were also one of the organizations that violently protested the fact that we were all being asked to bid a Solomon type machine to LLL. Because even at that time TI was considering building an ASC type machine. However, I don't think it was what the ASC is now. I think it was another one of their seismic goodies that they had.

PURCELL: It was the vector processor but if you recall they were a little bit confused about how to minimize memory conflict and they had some prime number of banks like 29 banks or 31 banks.

NEIL: That came later...in 65, 66 and 67 they were really looking at a souped up version of what was really a seismic computer. They later came up with this other one which had a higher bandwidth memory, it had ECS {kind of} and had a small core memory. The memory was 31 banks and I'm trying to remember why.

PURCELL: Probably they figured it unlikely you'd have to take every 31st element, no program was designed to take every 31st element so you'd never really get into the bank busys. But to do that the hardware had to calculate the bank address and that was a big messy divide.

NEIL: That could be a big problem. Besides that they were originally going to go with the high speed memory that was going to be the same memory that flunked the ILLIAC IV and the large memory was going to be a very dense core memory which they proposed. It changed though, quite a bit. Now along about 1966.....when was the EM1B project developing?

PURCELL: 1967, it was finished in 1968.

NEIL: I mean what was the rationale there?

PURCELL: Well in the back of Jim's mind on the design of STAR he had received a recommendation from Ben Carlson that we should use 2's complement arithmetic. He also discovered when he was trying to minimize the hardware complexity of the 6400 that you could never go to serial by bit arithmetic with a 1's complement machine at least not neatly because you'd always have to go through an additional recomplement cycle. In order to get a real simple machine you should use 2's complement arithmetic so you could do serial-by-bit arithmetic. He had gone through the ideas of using 2's complement arithmetic as being a good thing, when we designed STAR we had this justification from physicists that we respected at Los Alamos that 2's complement arithmetic was really better from the physicist's standpoint. He decided we could do a serial-by-bit kind of machine just as an experiment to see how low he could get in the hardware design of a machine like the 6400. How much lower can you go with a 6400, so we sort of selected the instructions and data of STAR to also meet some needs that Jim thought, that exposed themselves when he did the 6400.

TOTH: It was also a conscious decision to build a machine that would have the STAR repertoire so we could get a leg-up on programming because we realized that we never yet built a computer where we had any decent software at the time the machine was delivered.

PURCELL: A simple machine like that with serial-by-bit arithmetic would have to run at such a high frequency as to really exercise some of the logical components too which was another concern (using a 50/25 megabit clock).

NEIL: The 3500 was already in existence. It was already built and running by 1967 wasn't it? With those same circuits?

PURCELL: But it wasn't going at that rate. The rise times were slower than the EM-1 would require.

NEIL: So in 1967 Ron Hintz was made project engineer.

PURCELL: On the STAR 100.

NEIL: And who project engineered the STAR 1B? Was that Pagelkopf?

PURCELL: I guess that was before Chuck Morris took over.

NEIL: Then whose idea was it to turn EM-1 into a product? I arrived on the scene when STAR 1B was being considered as the next MPL, that was MPL, or PLX or XPL. God knows what.

KORT: The PL something.....

NEIL: PL something. Now where'd that originate from?

PURCELL: You mean ZIP? The Zenith of Information Processing.

NEIL: Oh, is that what that means? The Zenith of Information Processing. Who dreamed that one up?

PURCELL: Jim Thornton

KORT: That's why I was confused here, because the EM-1 was originally called the ZIP.

PURCELL: Then sometimes we called it the NIP - Nemesis of Information Processing.

KORT: It wasn't a serial processor. The only reason I thought we went to serial was because of money problems.

NEIL: You mean it wasn't meant to be a parallel arithmetic machine?

PURCELL: 4 bit parallel or something like that.

KORT: It wasn't really specified. They just said you could use X amount of logic or money.

NEIL: You were told you could use X amount of hardware?

KORT: Well, like I say, from where I was sitting that's all I was told.

TOTH: Maybe you know whether or not there was a conscious decision made to use the STAR memory design right from the beginning. I know that that became one of the goals eventually to check out that memory design on the STAR 1B but I don't know if it was conscious or not.

KORT: Like I say the first machine wasn't specified as to design goals. I think it turned the way it did because of money problems. One day he came in and said it will be serial, and that was it.

NEIL: How long had we been "perking" on ZIP? Was it just started in summer of 1967?

TOTH: They spent approximately six months or a year maybe, taking the sketchy information available on each individual instruction trying to make some sense out of what happens, end-case conditions and that sort of thing before they ever began design.

PURCELL: It was no more than six months that Pagelkopf was working on the instruction repertoire, then by about January 1968 the instructions repertoire was fairly firm, in the summer of 1968 the EM1 was running with the microcode.

NEIL: But...the design work only went from summer of 1967 to summer of 1968.

PURCELL: I believe so, it went pretty good.

TOTH: Pretty fast.

PURCELL: They used the 3500 chassis so they had all that mechanical design already done, the blowers and all of that.

NEIL: And power supply.

PURCELL: And that can take a lot of time.

NEIL: Okay now, it was called ZIP. How did Ray Allard's group get in the act of suddenly trying to turn the ZIP thing into a product? What you're telling me was that it was not intended for this laboratory experiment to be a part of a product offering; it really was just a breadboard machine.

PURCELL: I never did really find out why the MPL got abandoned.

KRUEGER: You mean the PL.

PURCELL: It was MPL before PL.

NEIL: Going back, as I remember it there were task forces established as early as the 3300 day to define an MPL.

PURCELL: 1964 was the very first task force that I remember..Irv Dorf, Chuck Hawley and Dolan were in on that task force.

NEIL: And you were doing essentially a task force to define a medium product line machine?

PURCELL: A byte-oriented machine.

NEIL: A medium sized business data processor.

TOTH: I don't know if it was defined that well, but it was to be a new product line.

NEIL: Now the 3300 was already in business and this was to be the follow on to that?

PURCELL: Well that level of a product, that range of machine line for that level customer.

NEIL: So they had a task force...what did you come up with?

TOTH: I don't even remember.

PURCELL: Dolan, I saw the instruction repertoire once and the machine that reminds me that most of that instruction repertoire is the Texas Instrument ASC machine.

TOTH: Oh.

PURCELL: Except for the arrays, all the funny things they did with operands like shifting them, and complementing them and all that.

NEIL: Okay, there was a task force...but there was one after that, after the 64 task force, wasn't there another one after that?

TOTH: Yes, at least one more.

PURCELL: One ayear.

NEIL: Suddenly Ray Allard's group was doing programming. The Advanced Software Group has responsibility for doing the assembler and the simulator for the ZIP and the next thing I knew they were working on the productization of the EM1 as the new ZIP, as the next product line. This is after having gone through the 8600 rationalization process, like going from the sublime to the ridiculous as far as marketing input...this one had all the cooks stirring the pot. And Ray Allard's group worked on that. You guys weren't brought in at that point were you?

KORT: How many different PLs were there?

NEIL: Several, but some were the same thing with a changed designation.

CURT: There was an NPL in which I think represented over a period of time several different instruction sets, at least 2. There was something between NPL and the MPL. We were around at that time and we were aware of those instruction sets.

NEIL: When I arrived on the scene the EM1 {the ZIP machine} was the one they were looking at, they had some guys working on the FORTRAN for it, some guys working on the COBOL for EM1, and at that time I didn't even know STAR 100 existed. To my knowledge the only thing existing was the EM1 and I didn't even know there was a compatibility between the EM1 and STAR 100, and I worked on that operation for seven months before I discovered that. They were concerned about FORTRAN and COBOL. Allard's group supposedly was to become a basis of the software group for its new product line. That was why they were working on it, except that we had a guy by the name of John somebody or other who sat in his little cubicle working the FORTRAN and almost daily he would come in, sit at his desk, and say shit, there's too god-damn many registers in this machine to write a FORTRAN compiler for it. The next day he'd come in and say shit, there's not enough registers in here to be useful. This was in a bullpen with 7 foot high partitions and these sounds would emanate over the top and obviously affect everybody sitting in all the other cubicles. There was the operating system group (Martin Carter had the operating system group), Jim Vernon's group was working on the COBOL compiler in there. We had a consultant by the name of Richard Dorrance who had written a COBOL compiler for the IBM 7090, and 360. Mr. Dorrance was a consultant who came in and took a look at the instruction sets and said shit, you'll never write a COBOL compiler for that and this kind of began stirring the pot because he kept saying there are better instruction set alternatives. Mr. Dorrance proceeded to offer his which was why don't you make it look like the 360. Since it was microprogrammed why can't you make it look like the 360? Then he said on second thought there were some things that were wrong with the 360 so I would like to suggest (and he then proceeded to write up an instruction specification for an ersatz 360 machine). I still have copies of the instruction set. The instruction description runs about 10 pages. We later labeled this effort PA1 since he was located in Palo Alto. They called it Palo Alto #1. This unfortunately did stop all work on EM1 software, software-wise everything died. Everybody stopped working on it, they all started writing instruction sets for their own machine. This is at the very same time when Ray Allard and somebody else made a major executive council presentation. We had made a market survey, we went around and collected everybody's requirements, what they had to have, and of course, we had considerable " help" from marketing. There was a document almost a foot thick. We went through that and Allard went up and presented to the executive committee the fact that the ZIP should be the new product line. While he's doing that and while the executive committee is deciding on that, back at the ranch Mr. Dorrance and John Thompson are busy typing up new instruction sets for a machine which they think is better than the ZIP machine. The day the executive committee says Go is the day that Ray Allard was told by Mike Schumacker that his people decided that the ZIP machine will never fly. We made a re-presentation to the executive committee that says don't worry about the instruction set, it won't be the ZIP but we're going to build you a neat MPL. Thereupon ensued AH-1 {Arden Hills #1} which was the group at Arden Hills coming up with their instruction set. Each one of these were different flavors of an ersatz 360. Then there was PA2, Palo Alto #2, then there was Arden Hills #2, Arden Hills #3, and Palo Alto #3. Then Curt and I got in the act about that time. I said good heavens, you don't want to build another 360. And what was the name of the guy, that red-haired fellow?

CURT: Ken MacKenzie

NEIL: Ken McKenzie had an office and he worked for Sam Slais. They had built an organization which was an engineering organization, which was going to build this NPL and Curt worked for Sam.

CURT: No, Sam had a directorate then that had systems responsibility for MPL.

NEIL: The executive committee says go build it, we don't know what, just go build it. We formed the directorate and Ken McKenzie was working for Slais. A conversation ensued between us one afternoon where we simply stated that you know this is insane. We've been going for about three, four, five, six and eight weeks with people submitting instruction sets without any regard for engineering at all. I went into Ken's office and I said this is all wrong. Let's go back and build something that we know how to build. The idea was let's put on a blackboard those things that we know how to build and can do well, not what IBM can do. The first thing we put up there was memory and we wrote large cheap, reliable {question mark}, we put up there virtual memory even though we didn't have the STAR virtual memory system really running. We had virtual memory in the EM1. We also wrote #2, I think, high performance components, #3 cooling. We came to the conclusion that wasn't anything on the board there that gave us a lever IBM as far as trying to go head-on with IBM. But we thought we did know something about high speed processing. So we proposed to take what we knew from the blackboard and come up with a whole different architecture, separate PA1, PA2 and whatever else. We "genned" it up that afternoon, just "flat-out" and then spent the next week selling it to everybody. Did everyone piss and moan, {jiminy Christmas} cause they'd all been writing their own instruction sets. We descended on them with all four feet and said we think we've got a neat idea. It caught fire because it was a good idea, right. I'm still very proud of that even though I helped shoot it down. Anyway there was one major kink in it which had to do with instruction formats. It was the problem of how to express what we wanted to express in 32-bit instruction format, and fit all information in. You and I {Curt} and Ken and somebody else, sat in that little conference room, late on a Friday afternoon, and somebody had a brilliant stroke of genious, which solved the problem. Immediately then we had ourselves a machine and everybody bought it. Tuesday afternoon I went in and I said okay now that we know how to build a machine let's now stop and do it right. Let's go back and do the whole process over again and the answer came back no we can't, we haven't got time because we're losing sales daily. We've got to get out and build this thing. That's when I walked away from it, that's when they started generating instruction sets for this thing and that went on for another couple of months and these guys then started the design work.

PURCELL: Can I talk a little bit about what was going on in Canada at this time?

NEIL: Well this is before that.

PURCELL: Yes, but before they went to Canada there was some selling going on. There was a salesman in Ottawa that got quite infatuated with the instruction repertoire of the ZIP machine, and its capability for data processing, etc. He wanted to use that machine for seismic processing on ships for anti-submarine warfare.

So he went around to different offices in Canada trying to get money to use this little instruction repertoire in a military computer aboard submarines and he didn't really find any home. But he did find this Pate committee. The Pate committee had all kinds of money to build a computer for commercial purposes and so that was going on while you people were down here at Arden Hills. That money was floating up in a big bundle of " 50 cent" dollars available for engineering if the engineering would take place somewhere in Canada. So that's going on in the background while you're talking about engineering details.

NEIL: Well, they obviously formed a full division, I guess, to build a machine. Bob Duncan I guess was in charge of it, Ray Allard had the software, Sam Slais had the hardware and then you guys supposedly set about designing the machine.

CURT: We set about designing, writing the specs, and starting to design three different models of that machine but the models were based on what was then believed to be the requirement in the market place as far as the cost and rental. We went through a process I think after that of changing the machine quite a bit in terms of what the operating system was going to be and the features that we were trying to put in for the operating system. I think that probably the most significant thing that happened after that was {I believe} the first time they came and told us that there had been change in the market place. The first time they told us that we then redefined our three models to set at a different cost/performance level, and went back to designing and then not too long after that they came and said there'd been another change in the market place and that after they went back out to look at that again they cut off then the top model. But we went on designing. We had finally gotten about to the point where the smallest machine's logic design was probably 60% along and the microcode was all done at that time. We did the microcode first cause we had time. Then the decision was redecided that the next product line would not be MPL but would be the STAR architecture so we went and began to design for the STAR instruction set. At the same time we had generated a new package, a thing we called the 56-pak because they wanted a package that was air-cooled and that was through-the-board mounted. It was a planer 56-pak with an air-cooled, hybrid-cooling system and we had also been able to show that it was cheaper than freon cooling, a cheaper package. We were then working on the PL50. I can't really remember how far along the design of the processor was but the memory design was completed, logic design of the memory was completed, when we started going through a series of iterations about comparing the price of this hybrid cooling system to the STAR freon system. No matter how many iterations we made it still came out cheaper until finally we decided to change it anyway, to the STAR freon cooling system. It was changed and the memory was redesigned then and about that time was when (late 1969, 1970) the economic crunch came and we were running out of money. The thing I remember most about that was we knew there was a crunch and we had a meeting. We were called up to this meeting and on the way up Wayne Speckers said to me, " Well, a lot of things have happened but the last thing that's going to happen is that they will not take our money away from us because they need a computer they keep telling us, to sell." So we went in and sat down at the meeting and they said we're taking the money away from you. They then presented several alternatives of how we might go on and one of them was to go to Canada and get Canadian government money. We talked about that for a while. This is the one time that Wayne Speckers prediction mechanism broke down for about a year because at that time he said to me, " whatever happens we cannot ever go to Canada."

KRUEGER: When he reviewed the STAR 100 project two years ago he said " we won't ship earlier than January 1976--no way--impossible."

TOTH: Is that what he said?

KRUEGER: He said it was virtually impossible. Don't you remember that, Neil?

NEIL: Oh, I remember it very well.

CURT: I was involved in that too. He generally was very good at predicting what was going to happen next but as I say...the things that were happening to us at that time were so illogical that... {laughter}. The next point was that we would get up and be in Canada in July. We couldn't believe that the company could make a decision and implement it that fast and we were wrong about that also. We went to Canada and built what was finally called STAR 65.

I can't remember exactly what year it was that they decided that it was again not a product but a QSE.

MARIS: May, it must have been 1972.

CURT: May of '72, yeah, by that time the thing was fairly well designed. I guess we were probably in the process of building it. When the decision that it would not be a product, it would not be documented as a product, and it would not be sold. We were in the process of building 3 prototypes and did build 3 prototypes. The first prototype was shipped to the Arden Hills plant in November of 1973.

PURCELL: In May of '72, or more or less that period Nienburg came down here to Headquarters and made a presentation for the budget it would take to build the software for the PL50. He asked for, realistically I think, $45 million spread over a period of time. Everyone was shocked to death at such a terrible high price. It turns out before we're ever done with the SOS replacement we're going to spend many times that same $45 million, but his realism was not well received and that sort of helped turn it off in all directions.

CURT: That's about it. The eventual customer for that STAR 65 was supposed to be Langley. It wasn't running well enough for us to predict that it was going to be able to ship on time and finally in about December, I guess of 74, it was shut off. The prototypes in Canada are sitting there and will have to sit there for at least seven years because the Canadian government can come and take them if they want to. They can take the machines, they can take the design.

NEIL: Haveyou got some observations on the lessons you learned in that project?

CURT: Well, the first lesson that we should learn is that it's never too late to change anything. As we said in those days and the days when, a couple of weeks after that MPL architecture was beginning to be defined we were saying at that time that unless Control Data produced a product in a very short period of time we were going to be in very bad trouble. We're saying the same thing today and we're using the same reason to move along as fast as we can in implementing a new machine. I think that applies now, in the definition of architecture, it applies to problems that you find inside the machine. Inside of STAR 65 we came to understand a couple of mistakes that were made in the basic design about the time it was being built or before then. It was decided that it was too late to change that and that we would fix it with patches. That, I think probably looking back, was also a mistake; had we stopped at that time and redesigned that part we would have in the end gotten done sooner.

PURCELL: You would have saved many months.

NEIL: Well, I want to come back and revisit some of those detailed mistakes later but I wonder if you've got some great philosophical insights.

TOTH: Let me ask a question. How difficult was the communication and the getting of answers to your questions back there at Arden Hills working way off in Canada? I'm sure you must have run across a lot of questions, detailed questions.

CURT: Detailed questions concerning what was meant by the spec, I didn't have trouble getting those answers simply because I knew Chuck Morris and people like that I could call up and ask and I could get answers. In general, the kinds of questions that couldn't be answered and weren't answered, were questions involving the reasons for things or what kinds of things are important, more important than others, priorities, features of the machine for instance. That was always a problem in that we could really get along despite it. We learned to live with it so that we could back off and say that we don't need to know this and that's essentially what we did. Essentially what happened to us is that after a time we switched to the STAR architecture, we quit asking questions. We had the spec, at that time it was something we could work with for a long time without asking questions and that's just what we did.

PURCELL: The spec was pretty detailed by that time.

NEIL: That is something I do want to revisit again also in detail. That is the kind of 'topsy' instruction set; How it grew and also and how absolutely it cost a considerable amount of money to implement on the STAR 100. Had we had some cost ceilings that we had to meet we would never have built the machine and I think that's the same situation you guys were put in.

PURCELL: If you recall a lot of that growth was in an attempt to broaden the applicability of the machine. In response to John Thompson's FORTRAN investigations.

NEIL: There were major points in that instruction set definition. You had your original ZIP instruction set. When Ray Allard's group inherited it at that point it was the first time the question was opened as to additional changes in instruction set. Since you hadn't really gotten very far in STAR design there was some latitude given here. The next big jump was when the decision was to change over to STAR 65. The software people came down with a specific list of requirements they had to have, which would require changes in the machine.

PURCELL: That was the list of 13 instructions.

NEIL: That was the next big time.

CURT: I think another really big thing about the STAR 65 was that we were given at the beginning certain rules {cost performance rules} but there was also rules that said the STAR 65 was to be a COBOL machine firstly, to do register-to-register scalar FORTRAN in a specified performance relationship with the 6600 and to implement anything else any way you could. Vectors had no performance priority. It was therefore a kind of a non-sensical architecture with those kind of rules. After we were working on it a while they would come to me and say look, the management is very concerned in Minneapolis that this machine is going to be a better FORTRAN machine than it is COBOL machine. And I said well I can understand that concern but I think it's likely. In light of that it didn't appear to us to be an architecture that would be the best COBOL machine in the world in terms of cost/performance or the best scalar FORTRAN machine in the world in the terms of cost/performance. It wasn't. By the time we reached the point where we were wondering who was going to buy it the only interest that people had was in the vector performance. The only interest there, was absolutely no interest in COBOL performance and little interest in scalar. It was no surprise to me that people would find that disappointing.

NEIL: What was your vector performance.

CURT: Well, the floating point ADD would be the best vector performance, which was at the 100 ns rate and the performance was the same on 32 and 64-bit arithmetic. If anything got any more complicated the rates would slow down. In other words, if you were starting to talk about square root or divide or multiply the vector rates slowed down. So we didn't make much of an attempt to maintain that vector rate across a large number of instructions.

NEIL: Let's back up to EM1. We got the EM1 built in 1968, powered on in August.

ROWAN: Early 1969

NEIL: It was a one of a kind breadboard. We used it for software work. When was the decision made to build an EM1A?

KORT: EM1A? That was for Canada, for Slais' group.

NEIL: That was because they needed some EM1's.

KORT: They wanted a better response time and I/O capability.

NEIL: Okay, the 'A' machine then really was a slightly modified version of the original EM1.

TOTH: How did it differ, I don't recall this machine.

NEIL: I/O channels.

ROWAN: Wasn't there also the capability of going from 32K to 65K memory? Or was that the 1B?

NEIL: That was the 1B.

KORT: I think the only reason was the I/O and speeded up exchange.

NEIL: Oh, I remember why the 'A' was an "A'. It had nothing to do with that. It was that bloody ICL channel. It was the ICL channel that we had to put on, that's why we built the EM1A and we did it in a hurry. So we built what, 2 of those?

ROWAN: One orphan EM1A.

NEIL: Then we built the 'B'. Didn't you do the 'A', Ray?

KORT: Yes, but there wasn't much change to it.

NEIL: Ok, who did the 'B'?

KORT: There wasn't much change there either.

NEIL: Larry, what was the difference between the A and the B?

KRUEGER: The channels were different, there were more channels. 8 channels on the B. The 'long read' capability of the memory was in there. It was different enough from the 'A' so that we couldn't support it.

In fact about 3 years ago Hank Munneke told me to write a memo saying we could no longer support the EM1A; and we did. We got no reaction from anyone cause nobody even cared.

NEIL: So we ended up shipping what, 2 STAR 1B's, up to Canada?

KRUEGER: Three, I worked on 3 up there when I was there; and one to General Motors. One to Sunnyvale, the programming group out there.

NEIL: Yeah, so they've got 2 now.

KRUEGER: General Motors, California and there was 3 up in Canada and there was one sitting down here along with 11 ROS stacks which are now just junk.

NEIL: Junk?

KRUEGER: Junk.

NEIL: ROS was a technical exercise independent of STAR 100, wasn't it?

TOTH: That started out as a lab project from what is now Tony Vacca's group. As a matter of fact it's remarkable that the thing worked at all since there was a fantastically small amount of energy involved and high impedance levels besides.

KRUEGER: It's like looking at the great pyramids and saying it's amazing somebody could build that thing. It just takes a lot of people to haul rock and die.

NEIL: There are some people who continue to contend that it never has worked.

KRUEGER: Well, I was very critical for a while there when I first started and didn't know much about it. You know that if you moved the strobe timing plus or minus 1 nanosecond you'd start getting parity errors. That was intolerable.

KORT: That's really not too bad considering it was a first go-round for that memory.

KRUEGER: It was pretty bad compared to anything I'd ever seen.

NEIL: That ROS memory hardly incurred much more development after the first cotton-pickin' bench model was built.

KRUGER: What happened is they built 4 of them. Realize what that means for complexity. Every time you had to change a data sheet you had to take the old one out and metal filings would drop in there. So Bruce Becknell {a pretty sharp engineer} worked on the thing. His restriction was that he had to use the thing just as it was, just improve it. He looked at the design and found that the data sheets had metal lines running up and down them--etched metal lines. He said 'aha' we don't want metal touching metal, so what we'll do is coat the sheet with a thin sheet of MYLAR, while still retaining the correct capacitance to get one's and zeros out. That seemed to work quite a bit better. That was available on Serials 5 and up. He also improved the drive board but the margins were still less than a nanosecond. I think what Curt said is very important because you never felt that you could do something else cause they always said there was no money to do something else except build more of these pieces of junk. {Laughter} That's true though, but finally last year they said that we would no longer support the ROS stack and Phil Florine almost fainted from gratitude. To change a data sheet you had to have needles {two knitting needles} to align the sheets. And then if you had a short then you were in big trouble. You know you'd get all the sheets aligned and close up the stack and torque it down, and then it doesn't work--you've got parity errors. That usually meant that you had some metal filings in there. You could tell which sheet it was from the parity tree so you'd open up the stack, thereby dropping some filings down in there someplace else. Then you'd take a plastic sheet and run it down in the proper spot, all the while looking at an ohmeter to see when the short disappeared. Sure enough it would go away so you'd tighten the whole stack up and hope like crazy that another one wouldn't show up.

NEIL: Sound like your 18 board module on the 8600?

GRINNA: Well, not quite.

KRUEGER: There were other things. The girls that wired up the ROS stacks had to wear lights on top of their heads. There was this one girl down at 'core storage' who was the only one able to solder those stacks. You know she had to solder legs on chips that you couldn't even see, at least I couldn't see. She would get in there with her little light and solder away, but it was a hairy operation. It was fantastic to watch her.

NEIL: The sad part of it was that it was, when we discovered that we needed more microcode than we had, we were going to have two stacks, weren't we? Wasn't that the original scheme? The very first power supply was built to hold two stacks and then we decided to only have one. Then the idea was you take all the microcode that you're never going to change right? All that Kernel microcode that is never going to be changed at all, we're going to put it down in the stack and everything else will be in core. The spec is pretty frozen by now fellows, right? About 1968-69?

KRUEGER: So was the design of the STAR 100.

NEIL: I can remember the very first time Chuck Morris almost killed me. We went through the analyzing of the code; I put all the stuff down and decided what should go in the stack. We got the first set of sheets all made up, beautiful, and they hadn't even gotten them in the cotton-pickin' box when we came up with a change in the instruction set which had to go all the way back down into the basic R&I sequences. Talk about moving targets.

STAR INSTRUCTION set certainly was.

ROWAN: I made a little study on that last year. The original proposal for Livermore had 167 instructions--over the process of the design that grew to 231.

TOTH: Yeah, but let me hasten to point out that most of that probably was due to the fact that Fernbach wanted 32 bit floating point instructions.

NEIL: No, that 167 in the proposal...that proposal of '67 included 32 bit format. That was just me and them. Me and Schumacker's group adding 'goodies' to the STAR instruction set--like the D7.

HUTSON: I think the thing you've gotta learn out of this is that if you're going to design an instruction set, use up all the codes and that way not only do you have something invested in keeping it the same, somebody else does too because usually if you use them all up if you want to put a new one in there you've got to give something up. I think things became more stable that way.

TOTH: If you do that don't use the 16-bit instruction op-code. {Laughter}

RESNICK: Well they did use up all the instruction on the 7600, there wasn't any left and still they managed to plug in a whole new bunch.