Interviewee: E. L. Glaser
Participants: Fred Way III
Interviewer: Robina Mapstone
Date: November 8, 1973

MAPSTONE:

Okay. Today is November the eighth, and this is Bobbi Mapstone, and I'm talking with Professor Ted Glaser and Professor Fred Way the third, at Case Western Reserve in Cleveland, Ohio. And this is an interview for the Smithsonian Computer History Project. Professor Glaser, perhaps you'd start.

GLASER:

I got into the computing business in the following strange way. As a senior in college at Dartmouth, I had gotten somewhat interested in finite mathematics and actuarial work, and I had gone to a lot of insurance companies. They looked at my application, found that I was the best-looking candidate they'd had in a long time, I walked in the door with a seeing-eye dog, and that was the end of it. Their attitude was: "We are due probably a good mathematician, but there's no way you could do the IBM work." And a few other minor things like that. The one particular group that offered me a job, the people want to pay me a premium to come there, but their personnel man explained that they had a standard policy that handicapped people they knew couldn't do work as well as anybody else, they were going to hire me at 25% under the minimum. "And, besides," he said--and I quote--"you ain't got enough mathematics." Unquote! So, to finish the story, I was down visiting my girl, who became my wife, at Bryn Mawr, and they had what they called a Job [?] Weekend of different opportunities for women in science and industry. They had two people down from IBM talking about the SSEC. So, I went up there to see the SSEC, and I figured as long as I was here, I could ask some questions about this IBM stuff that I couldn't do, because I'd memorized a couple of boards in about half an hour and learned how to wire them. They talked to me for a while and figured out there was nothing I couldn't do, so I got hired. And I was the only one they got from Bryn Mawr that year!

ALL:

[laugh].

GLASER:

Yeah, how else would you get into the computer business? I told you, you didn't want to know!

MAPSTONE:

I want to know! I'm going to do a study on how people got into the computer business. You've just given me a beautiful story. [Laugh].

GLASER:

Look, it's a little like eating in a Chinese restaurant. There are certain things you're happier not knowing.

MAPSTONE:

[laugh].

WAY:

I spent weekends in Bryn Mawr and nobody ever hired me.

GLASER:

Because you ain't got enough mathematics, that's all.

MAPSTONE:

And which part of IBM did you get hired into?

GLASER:

I got hired into Applied Science, working for Cuthbert C. Hurd, and, in fact, I worked for a guy named John Sheldon, and a few other people, in what was called a Technical Computing Bureau, using the Mod 1 CPC. Oh, we had people around like Bill Heising, John Sheldon who ran it, and Elmer Kubie. Those are the main ones you might have heard of.

WAY:

Charlie Karl [?] must have been there.

GLASER:

No, he wasn't there. I hired Charlie three months later.

WAY:

Ah.

GLASER:

I interviewed him. And let's see. Bob Barton came in about three months later to learn what a CPC was all about. No--sort of fun games. One-time John mis-estimated a job; it looked like we were going to lose quite a bit on it, so I asked him if he'd let me do it my way. And he said, "yeah." So I went and got somebody to look over the prints of the 604, and I found that by pulling out two tubes and changing a couple of wires on the backboard and stretching one wire between the two plug-boards I could get it to do some things that it never intended to, and we came in under budget. So--[chuckle] he realized that I as a troublemaker and transferred me. So that's how I ended up on the 701 planning and had my hand in on the original mod 2 CPC, 701, 702, and from there on to other forms of obscenity.

MAPSTONE:

Yeah, well--okay, good. I'll pick that up. Would you like to now do your little introduction?

WAY:

Sure. I got into it the easy way. I had a friend who was working for a company, and they told him to start looking into computers, and he said, "You'd better come over here and help." Because I didn't know anything about it, and he thought I'd be a lot of help.

MAPSTONE & GLASER:

[laugh].

WAY:

So, I transferred. That was Babcock and Wolstencroft's [?] Research Laboratories.

GLASER:

Oh, what did you do before you were in the computer thing?

WAY:

My formal title was physicist, but I don't think I ever did anything like that. It was applied spectroscopy and things of that nature. And then that was also at B & W, but at Beaver Falls, Pennsylvania. So I transferred to the research lab, and then--Cam Pallo [?] was a friend. He and I had gone to school at Pitt together.

MAPSTONE:

Cam Pallo from IBM?

WAY:

Yeah, sure. Oh, we're bosom buddies.

MAPSTONE:

He was my boss.

WAY:

Oh, is that right? Well, isn't that incredible, because he and I--[laughter].

GLASER:

Well, okay, what were you doing at IBM?

MAPSTONE:

Well, I'll just turn this off here. [Recorder off]. Well, how about sort of going back now. I'm really very interested in what you were doing when you were on the 701-planning committee. I believe you took a look at the Wooden Wheel and 650 and various others.

GLASER:

Oh, no, no. That came later.

MAPSTONE:

Okay, you give it to me as it was.

GLASER:

[Whistle]. Okay. I was working for Hurd, and they moved me up to 701 planning. That committee was headed by Bill McClelland. Other people on it were, let's see, Ted Codd and Paul Matland. Now the 701 had been designed, and, well, in those days it was still called the Defense Calculator. And we were to do some of the programming for it, utility programming.

MAPSTONE:

Could I just get a date in here, please?

GLASER:

Okay. Early '52, about January, '52. By that time, I'd already gone through the mod 2 CPC and looked at some of the things and happened to make some modifications with it so that it was possible to program it. We did a lot of things. IBM, as I'm sure you understand, was then--and ... still is, in certain areas--a highly political company. At that particular time, the main battle was between decimal computers and binary computers. It had nothing to do with decimal or binary; it had to do with their backers. One of them became Product Planning--at that time it was called Future Demands--and on the other side was Cuthbert Hurd and Applied Science. There was an equivalent counterpart of both organizations within Engineering. The 701, however, was progressing. There was also a battle between Ted Codd, who believed in what are called regional assemblers, and Nat Rochester, who believed in symbolic assemblers. And it was towards the end of that--No. That's pretty much as it was. The machine was pretty well designed and put together. I had some effect on it by getting to re-layout the console. Ted came up with the standard binary card format, which was used all through the 7000 series and I think is even used today for all binary, including the checksum and the Padress [?] and the number of words, etc.

WAY:

... Binary card?

GLASER:

That's right.

WAY:

Was that when the mighty decision was made to fix it up so it would only read seventy-two columns?

GLASER:

No, that decision was made before that. I can even tell you how it came out.

WAY:

Well, two times thirty-six is seventy-two.

GLASER:

That's how it came out. And the reason that it was picked as seventy-two was that thirty-six bits seemed to be enough, thirty-five bits and sign, and the additional eight columns were found to be sufficient to put in everybody's name and card serial number, which, after all, didn't have to be read by the computer. Now it turned out a peculiarity of the 701 that everybody has damned ever since, all through the 7000 series, there were two extra bits to the left of the accumulator, called the P and Q bits, which were put in at the insistence of Nat Rochester, because he felt that by having two bits of overflow, which weren't cleared except under funny conditions, it was ideally suited so that you could shorten the inner loop of a double precision multiply by one instruction. Now how does that grab you?

ALL:

[laughter].

GLASER:

Oh, there are some funnier stories than that.

Well, it turned out that, probably my main contribution was the fact that there was no such thing as a bootstrap loader in those days, for peculiar reasons. There was a reset button that cleared all of memory and it cleared it all at once. Now this was nice, because when you're punching out something or printing something out, and you happen to get out of an area that you'd set up properly, the effect was, well, you noticed it, first, because of the deafening sound that came from either the punch or the printer, which at the same time was doing the interesting process of trying to tear itself out by its roots and walk across the floor. And this was a very exciting time.

WAY:

[Laugh].

GLASER:

All right. Oh, several things came out. I won a steak dinner from Ted Codd because he bet me I couldn't write a one-card program that would load itself into memory and then pull the cards behind it, using his nutty format. Now it turns out that the 701, the read instruction didn't mean "read." It meant "start the device." Now, for every word you want to bring in, you get a copy. If the copy came too early, it would wait, but if the copy came too late, well, that's it. You're dead. It'd stop. So, the idea was that you gave the copies and got the thing in. And the only boot-load it gave you was that one word has two instructions to it. So, if you pushed the load button, what it did was to start the card reader, copy the first word into the first location of memory, transfer to it, and from now on you're on your own. He lost his bet. So, we put some utility packages together, including ones that would--incidentally, the way that first got keypunched, which used to drive Ted out of his mind, was I'd sit there and keypunch on a keypunch in binary, which turns out to be amusing. So, the way those first things got done was by hand, and later on, our decimal load programs, which were very simple to copy, one wrote a card to do the decimal conversion while we were waiting for the next one to come in. It was so primitive it was positively savage.

Oh, also a few other things came out, but there's not much to add. I started working with a guy named Pat Beebee, who later went off to RCA and hasn't been heard from since. We got the first extract of the thing and a few other little niceties, but that's pretty much all you can say about the 701. At the end of that period, Hurd was making his major political move, so he decided to establish a research or planning group in Poughkeepsie. The head of that group was Dan Mason, who is now president of Computax. Other members of the group were Beryl Smith, who married a guy whose name I can't--Bill Bradshaw--Bob Barton, and myself. A fifth member of the group, Larry Sweeney, showed up, was a member of the group for three days, and got swallowed by the Sage project. He hasn't been heard from since, either.

Well, it turns out that we were supposed to work on the design of a new machine. Well, that sort of died. We worked on all kinds of things. Then they told us to go look at a magnetic drum machine which was being done up in Endicott, now that we'd all moved to Poughkeepsie and bought houses. So, we started commuting to Endicott for awhile. They moved Elmer Kubie up there. And that became the 650. Now it was a tossup whether we could do it, because originally it depended upon having re-circulating loops on the drum, which looked like it impinged on the UNIVAC patent. Therefore, IBM wouldn't touch it. However, that came up. In the meantime, Bill Woodbury--you remember Bill Woodbury from Betsy and Nancy--had been hired. He came in and got to looking at what could be done with computers, and a funny kind of cathode ray tube storage he built was called the Wooden Wheel.

Now the Wooden Wheel got its nickname because three people in the group were involved--Bill Woodbury, Greg Toben, and Truman Wheelock. The machine was interesting, so we looked at it to see what could be done with it, made some modifications to it that meant that you could turn it from a plug board programmed machine into a stored program machine where a plug-board became a microcode, although it was a very powerful microcode. And did some other things on it. Got Bill a little unhappy because I showed him how to do matrix conversion in five steps and square root in two and a few little gadgets like this. On the other hand, I got some very obscene algorithms. So that's how that happened, if you want to, you can go back and push on that. It was delivered to Northrop as an experimental machine. The group went out with it. They then decided to build a follow on, which was basically part of the original version that we wanted. It was not going as far as I had hoped to go. But it was a finished machine, and three more were built and sent to Northrop. Well, they were built, interestingly enough, by--oh, boy, I can't even think of the name of the company. It was in Artvald, [?] Pennsylvania, on a subcontract to IBM.

MAPSTONE:

Oh, I don't know.

GLASER:

Sundstrand. No, that's not right. I'll just have to think about that.

MAPSTONE:

It'll come.

GLASER:

Anyhow, other things that sort of got hatched during those days were a rather fancy tape-operated printer, that you plug-board-programmed, and it was supposed to become the--I forgot what we called it, it was a sorter-collator. [Chime rings three]. And I think they called it the 770. It was later delivered to Treasury. It used barrier-grid memory, had several tape units on. It could do a sort by collation or arbitrary collation; it could do a fair amount of arithmetic. I'd done some work on the 702, which in those early days was called a TPM, Tape Processing Machine. The original version was never built, for very good reason. The later version still had some problems we got put together. The 702 and 705, of course, in many ways were precursors to the 1401. As far as the sorter-collator, I've forgotten what they called it, but it got some people very upset because here was a machine that was plug-board-operated but outperformed the 702 for about one-third of the price. As much of these things were going on at that time, because it was quite clear that the 407 printer, which was the wheel printer, had come to the end of its days. In fact, after looking at the mechanism, one wonders why it was ever built.

WAY:

I think it was a tribute to the genius of the mechanical engineers that it ever worked.

GLASER:

It turns out that there were several printers around. There is one that was built by Buck

Beattie and Ted--I'll have to come back. It was a stick printer, and it only ever saw the light of day as the printer part of the 305 RAMAC system. It was a rather clever mechanism. It was a little stick with some sides that oved around and hammered against the page, and its position in the concept, again, was a precursor to the golf-ball typewriter. Interestingly enough, there were two other printers that were being looked at--pardon me, there was one other printer that was being looked at, and two others that we tried to get people to look at, but they kept getting shot down in flames. The one that was being looked at, later got brought out as the wire printer, which can best be described as a catastrophe. Remember this one, Bobbi? It was like the wire printer on a keypunch, except that the wires would sort of shift back and forth, you know, over a couple of column positions and things come up at the ends and hit them, and these wires at this point were about five feet long and I think once a day they had a minor repair they had to do on it. I think it was called replacing the gear train. It was a very bad scene.

WAY:

...one. Each head printed two columns and then the other one, each head printed four columns.

GLASER:

That's right. And they were both bad.

WAY:

There was a five-by-seven wire matrix that came down and expanded out at the ends, so it was thirty-five high, and then there was a rod printer with…

GLASER:

Holes in it.

WAY:

…holes in it, which moved actually, and then you'd bang it into the wires and exclude the ones you want, and then hit the paper and so forth.

GLASER:

It was a ghastly mess.

WAY:

There were problems in making it go.

MAPSTONE:

It sounds kind of Rube Goldbergy.

GLASER:

Well, it turns out there are two printers. One of them was actually in competition, initially back in '48, before I ever came into the company, with a printing mechanism with the 407. It was designed by Bob Paris. I do not to this day know why it was, but the Paris printer would never be considered. For some reasons he had stepped on somebody's toes politically and his work would not be used. He had--that printer was operating in '48. His second printer was operating in 1953. That sound familiar to you? The '48 printer was the bar printer, which was the cheap printer used on the 1620. And the one he had running in '53 was the 1403 chain printer.

MAPSTONE:

I'll be darned.

GLASER:

It turned out they finally went to it in desperation because they otherwise would have to use a UNIVAC patent. There was no other way. And they tried everything else first. Now this is washing dirty linen, but so be it. Because this is the one place it's got to be recorded, because if it's not here, it won't ever be. Now it turns out the problem there, I can tell you that at the time there was a group convened in early '54 to study this as to which printer sheet we should use. And the choices given were the stick printer or the wire printer. So, after six hours we came out and said, "Mr. McDowell"--this is Wally McDowell, W.W.--"we think the stick printer should do it." "Gentlemen, that's an interesting decision, but I don't think you've studied it properly." So, we went back for six hours again, came back, "Well, Wally, we've done the best we can. It's still the stick printer." "Gentlemen, I don't think you understand. I'm sure there are some facts you've overlooked." So, we came back in, and Gene Amdahl was with us by then. Incidentally, Gene was my roommate; he was fresh out of Wisconsin, with all kinds of bright ideas, just abounding with facts about programming. So, I said, "Let me try something." So I went back and said, "Wally, do you think the wire printer might do?" He said, "Gentlemen, that's a fine job! Committee is disbanded."

MAPSTONE:

Agh!

WAY:

Does that tell you something?

MAPSTONE:

I'll say.

GLASER:

Now you know how things were done.

WAY:

There's an interesting sidelight on that 407. As far as I know, IBM never published a manual anywhere
that showed you how to, without fail, print every character on a card.

GLASER:

And it turned out, it was—

WAY:

Nowhere in that manual is there any information on the subject, and the acid test is to take a card that had all pluses, all minuses, all zeroes, three cards, and if you can print those, it can be done. And there's not a clue in there anywhere as to how you would do it.

GLASER:

And it's a non-trivial problem.

WAY:

Yeah. In fact, the CE's were coming around here for a couple of years after I succeeded copying this diagram to see how it was you actually printed cards with the machine. [laughter] That's just crazy.

GLASER:

So, let's see. A whole bunch of machines came out. Now I've talked about the 650, 701, '2, '4, '5, the 7070. Oh, boy. There's a bunch of oddball things. 607.

MAPSTONE:

What was the 607?

GLASER:

The 607 was a 604 with a back door that lets you add some more memory all in vacuum tubes. Okay, so just more registers, more of the same, and badly done. But it was a quick way of doing things. Later on, there was a big battle between Poughkeepsie and Endicott and the system based on the wooden wheel was picked. There's one thing that has to be understood, and that is that--and it's true of most computer companies, but it's probably more true in IBM than anyplace else--computers are designed by committee. This is one of the reasons that computers--Had we not had the great progress in the electronic technology we had, computers would have had a catastrophe long ago. But we were always able to overcome our inefficiencies. But within IBM, the engineer did not have the choice of running the computer. Any salesman had enough political clout to have his feature put in, because the "Folding Castors Corp., this is important to them, and we can sell five more machines there." The fact that it couldn't be built didn't make any difference, or that it would totally wreck the design of the system. The classic case of that was the Stretch. Our last act was to try and shoot down the Stretch. I failed. I tried to shoot it down, not because I was a disloyal IBMer, but because I felt it would be a catastrophe because the people were talking about things, "Well, you cant' make it work with this technology, so we'll go do build in-place components, and that will make it work cheaper." And my feeling was they're going to come, but not by wishing. The attitude of the pioneering group as expressed by one individual was, "If you tell an engineer to do something, it doesn't make any difference whether he wants to or not. He has to do it. Don't bother me with whether it's possible."

MAPSTONE:

Isn't there some validity--or wasn't there at this time--to being in touch with the customer's needs and trying to fulfill that?

GLASER:

No. In the sense you say, the answer is "yes." But in another sense, "no." Because, you see, they were not in touch with the customer's needs. They were in touch with the customer's wants, and they're not necessarily the same. A customer who is there, who is trying to get a job done, all he can think about is, "What little thing can you add to our account now that will make it easier?" We knew better. We had no power to change it. Case in point: Stretch was built to meet a certain set of needs. It was originally to be called the LARC, but then the LARC was won out by the UNIVAC. And LARC, of course, was to be the Livermore computer. So, the next name they put out was the Datatron. Well, Ed McAllister was a noisy salesman at IBM. He got hired to go out to be head of marketing for what became ElectroData. It was just then being spun off as a separate subsidiary rather than just a computer project within Consolidated Engineering Corp. And he thought the name Datatron was a great idea. So, he had ElectroData trademark it. And some people at IBM were very unhappy, because they thought it was a neat name. So, then they decided, "well, Stretch is a great one because it will be reaching out, taking a giant step."

And I'll give you an example. Much of the problem Stretch had was that they did things in some very ridiculous ways, to be honest. An attitude that was expressed was "Make the instructions just as complex as possible, because you can straighten that out with software, and don't worry about making them regular or easy to handle, because programming can finish that." Of course, the people who were saying that did not really understand the principles, as rudimentary as they were at the time, about compilers or anything else. They just did wishful thinking. And that's one of the reasons Stretch was a catastrophe. Stretch was basically a good machine. But the instructions were so complicated, with so many idiosyncrasies, that there's no compiler in the world that could untangle them. [Pause]. A good rule of thumb--this has nothing to do with history: If you can look at a manual, and sort of measure how long does it take to describe each instructions and how long does it take to describe each exception, by the time you're done, if the exceptions are more than ten percent of the instruction description, you're in Trouble. In Stretch, the exceptions were by a factor of ten larger.

All right let me talk about another one of the products of your part-time employer, so to speak. There're instructions on the 360 that, my guess is, are never used, because there is no compiler that can use them. Too many strange things. They were put in to satisfy somebody's benchmark.

MAPSTONE:

Okay.

GLASER:

...And then again. The reason I am using 360 as an example is that you are more familiar with it. If, instead of 360, you want to talk about 1108s, I'll pick a different set...Okay. So.

MAPSTONE:

So that was really an early established philosophy?

GLASER:

That's right. Because, as it turns out, remember way back. The card equipment that the salesmen installed. The salesmen in those days were part salesman, and part sales engineer. He wired the boards to get the problem going. If he needed three more pilot selectors and a co-selector, he could ask they be put in. If he wanted a funny little widget to appear on the board that would make this job do better, it would be done that way. And the idea of carrying that into computing was a very simple thing. Remember that the 701 was thought of as a calculator.

MAPSTONE:

Yeah.

GLASER:

The 702 was the tape-processing machine as opposed to the Card Programmed Calculator, the card-processing machine. In fact, the card mentality went all through IBM for many years. The new virtual memory systems, the VS2 operating system, is the first operating system IBM ever brought out where vestigial effects of the card mentality are impossible to find. You can see it in OS, you can see it in DOS in spades. In fact, there are things in OS that you can still see, that I can show you are directly traceable to concepts that went back to limitations of and constructs in card-programmed--not even card-programmed, but card-operated--tabs and summary punches.

MAPSTONE:

And some things stayed around longer than we realized.

GLASER:

Yes. In fact, the Hollerith--the eighty-column card is not the Hollerith card. The forty-five-column card is the Hollerith card, which, incidentally--its size was picked because it was the same size as the dollar bill of the early 1900s.

MAPSTONE:

Yes, that's a good story.

WAY:

That's part of that exhibit.

MAPSTONE:

Right. [laugh] The classic story--something that wouldn't change. [laugh]

GLASER:

That's right. Incidentally, when IBM went from forty-five to eighty-column cards, they had three projects under investigation: a seventy-column, an eighty-column, and a ninety-column card. I don't know what the seventy-column card was. The ninety-column card was a double deck card more reminiscent of the Rem-Rand card, although a cleaner coat. The eighty-column card was picked for a very simple reason. The guy in that project was a better engineer and produced a better card-reader, period. It worked and the others jammed. That's how it got picked.

MAPSTONE:

Good reason. Who was that?

GLASER:

Daly.

MAPSTONE:

George Daly, yeah.

GLASER:

George Daly. And of course, Daly became the power. Lake was the one who was the real genius in Endicott. He was the one who produced the first punch ever to permit punching of cards that didn't stop, because in punches up through the late fifties, the cards stopped every row to be punched. It was called a Geneva movement. And Lake came up with the idea of what was called the swinging dial punch, where the punch actually moved with the card while it was being punched. He was a true mechanical genius.

WAY:

And they had the punch that Rem-Rand used for awhile that Bull produced. The card stopped once, and
every hole that was going to be punched was now on a four-by-eight punching mechanism.

GLASER:

It turns out that actually Bull produced that, but there's an earlier version that Rem Rand produced in their old wire equipment, for forty-five column cards, that would do that. And now and then, because it was mechanical, both of 'em would slip and the tray would drop them all through at once. And the thing would just stop.

WAY:

I'll tell you from sad experience what happened to the Bull punch when it did that.

GLASER:

I don't think I want to know.

WAY:

[Laugh]. It took twenty minutes to take it apart and get the jam out, and it took four hours to reassemble it.

GLASER:

Okay.

MAPSTONE & WAY:

[Laughter].

GLASER:

Sounds typical.

MAPSTONE:

You know, actually, I just thought an interesting thought to discuss is the dominance in the market of the Hollerith-IBM versus Powers and maybe UNIVAC. It always seemed that really Powers and Rem Rand had some darned good stuff.

GLASER:

They did.

MAPSTONE:

And somehow or other, they--even, right back in the 1910 census, they couldn't get that edge. Having been a UNIVAC user, I wonder if you have any sort of philosophical thoughts about it.

WAY:

I think it's summed up best by the statement that Lou Rader [?] made when he quit as president of UNIVAC. He said, "Just because you have the better mousetrap does not mean that people are going to beat the path to your door."

GLASER:

I'll phrase it in a different way.

MAPSTONE & WAY:

[Laugh].

GLASER:

Many 1108s have been bought in this country. I don't think one has been sold.

WAY:

Same comment went for the 1107.

GLASER:

And this was true of UNIVAC, but it turns out it's been true in other ways of most of IBM's competitors--gross incompetence. In UNIVAC it was marketing; although in the case of Remington Rand cards, they stuck with the mechanical sensing of cards and all mechanical systems. With a power failure, you could put a crank in the side and turn it, and it would go on working. There were no plug boards. You had what's called a system unit you plugged in, and that system unit was a bunch of cables with little wires running through them and levers and so on, that was put together by a mechanic. And that was how it was programmed, period.

MAPSTONE:

And this is as late as when?

GLASER:

1955.

WAY:

[Chuckle].

WAY:

All right. Some other types of things. And again, you can look at some of the other competitors. UNIVAC had the edge on IBM in everything except marketing. They did not know how to sell. They had a far better machine. Burroughs with the B5000--this is a machine, good heavens, Bobbi, I did my last work on that machine in the summer of 1960. It is still a competitive machine today, the 5500, which is just a slightly reworked version. But here they had a machine, and through a mixture of arrogance and totally not understanding what the power of it was, they blew it. They expected keyboard salesmen to handle a computer; everything that's electric.

This is an aside. We had an interesting design for a machine. It was a little later than '60, but before '63. That's when I left Burroughs. I was in Paoli by then. We developed the idea of a centralized machine. We had a bunch of key terminals out there, and each one looked like sort of an accounting machine that a person used and an accounting salesman would sell it. But, in reality, it had almost no logic in it. No logic was done in the central machine. It was subdivided by sector. So, each one had its own part. We did all of it, programming, collection, that was just with a piece of power equipment down in the basement. And it fell on its face.

Okay, another fine one. That's another time when I won a steak dinner. Duncan MacDonald. I'd built and designed a machine out of desperation, for reasons I'll get into, called the VRC. And the VRC--it turned out the proposal and design was put out with the name VRC on it, and everybody assumed that that was its name, and nobody ever asked what it meant. And so, of course, I didn't tell them. And Duncan [?] and I had come up with the name. It's got a little of three, "Vors [?], Rechen Computer," "calculator," as a matter of fact, or VRM, "Vors [?] and Rechenmaschine." And it turned out it was to be in competition with the Librascope machine. It was a small thing, built into a teacart. I'll tell you about it later. But, in any event, Duncan said, "Okay, Glaser. You're pretty smart. Let's see you do this one." And I said, "Okay, what?" He said, "I bet you can't design a desk calculator that can compete with the Friden." This was when the Sperry Friden was out. It was about late '57 or early '58. I said, "Well, I don't have a device that will give me output." Duncan thought for awhile and said, "Hey, you could use a belt with letters engraved on it or a drum and illuminate it with a strobe to it." So we whomped one together, actually, yes, took a strobe from the mechanics' lab--only a screen--put a simple photocell on the edge of a disc, you know, to synchronize it, threw some logic together and just painted some numbers on it. Actually, we took some--tied them up, put them down and had a draughtsman do them, on lettering and just glued them onto this little card-board disc, got the thing spinning, and two days later we had a display. It worked like a charm. Nobody could tell it was moving. You know, yeah, any place that can use a strobe knows that. And so, he said, "Okay, I got your output device. Now go build one." So, you know, I hadn't had a steak dinner for awhile, so we built a very interesting machine. Now by today's standards, no, it wasn't. It was a machine with ten registers, and he wanted it to be an accumulator that could add, subtract, multiply, divide, square root, and automatic decimal, never saw a complement number, built-in diagnostics. The shot cost approximately fifty bucks. We were aiming at the Friden, which, after all, in those days was a $1500 machine. And we got one job, showed it would work, and it died, because they were convinced--or at least data product planning was convinced--that no computer or no calculator did, in fact, have any future. Now a few couple of years later, George Hare and L. P. Robinson, well, all got teed off, left the company, and went to a place called Friden. Now the Friden machine isn't the same one, but there are many things that will look very similar if you squint hard. Okay?

MAPSTONE:

Okay.

GLASER:

So those are some of the ways decisions are made, for very strange reasons.

MAPSTONE:

You never know exactly.

GLASER:

All right, the 220. You know what the 220 was?

MAPSTONE:

The 220. That's the Burroughs machine.

GLASER:

You know what it is?

MAPSTONE:

No.

GLASER:

It turns out that what the 220 was supposed to be was the 205, with the loops removed, the recirculating loops, and a small core memory put in. That was my assignment. And it became quite clear that with the speed of core that was then available, this couldn't be done and it was a ridiculous balance. There was no way I could prove that to anyone.

So, I did one of the lowest-down, dirtiest tricks I've ever done. I designed five machines before I told anybody about any of them. And each one had glaring faults. And I was able to escalate Burroughs from one machine to the next. Well, it turned out we got stuck on four. We didn't build five. So, four--220 was the fourth in the series. It was not the machine I really wanted to build, but it was close to it.

MAPSTONE:

The fifth one was it?

GLASER:

No. The fifth one was the machine I wanted to build, yeah. Wait a minute. [Interruption in interview; recorder off]...the formalism of IBM. So I ended up working quite a bit with him for a couple of funny reasons, first, because he liked to speak German and I could still handle it fairly well; he also felt that my dog was more sensible than most of the people he talked to. I have a lovely memory of him sitting up in somebody's apartment in uptown New York, and I can't remember whose it was. And he's sitting in the middle of the floor trying to pour a Martini down my dog. I kept saying, "One of us has to drive." And he said, "No, Ted, no. Driver must converge."

ALL:

[Laughter].

GLASER:

And that was it. That was literally what he said. And he just could be an absolute, delightful nut, which was a side most people never saw of him, but he was just a joy, a very warm person.

MAPSTONE:

Well, actually one of the things I've been interested in is just general reminiscences of John von Neumann, since he's not around, and to build a history about him.

GLASER:

Well, of course, he's a fascinating guy. A typical story about John came from Max--[interruption: "Hi, Jim. Just a second"]--oh boy, he used to be, he was at UNIVAC for awhile giving level was level four. It sounds like a strange way of doing things, but it turns out for multiple line printing or the kinds of things you have to do with minor and intermediate major totals, and it was intended to be like an electronic account own instructions, and that was the way of doing things in those days. It turns out that oh, a lot of things were discarded. Indirect address was first powered on the 701. It was discarded because the access time to memory was so long. But it really was pushed into it because of the aerospace companies. It was in '53 that we were looking for something to come below the 701 to really replace the CPC, which was electromechanical, was hell to keep running and—

WAY:

[Chuckle] The whole idea of tape that you'd wind around wheels and that 941 storage unit!

GLASER:

That's what it took.

That's what it took was that little box; it had relays in it. It really was a ... Did you work on that box?

GLASER:

Yes.

Did you put it on a 604 to make it a CPC?

GLASER:

Um-hum.

About that big square, with relays. That's all. That was the big bulk storage.

GLASER:

Now it turns out that—

MAPSTONE:

I think you could get ten of them, couldn't you?

GLASER:

No, five. Three were normal and you could get two more in a RPQ.

Five were abnormal. [Laughter]

WAY:

The whole concept. [Laughter]

GLASER:

It turns out that, I think, the reasons the 650 got picked, Hurd had a lot to do with it, but it was the safe way of doing it.

This was over the Wooden Wheel.

GLASER:

That's right.

WAY:

How did it come out with half the memory the 201 already had in it?

You mean this...

WAY:

Sure. That thing was a 4000-word machine, and the first 650 was a two with an option of one.

No, the first 650 was one. The Model A was a thousand words.

WAY:

I've seen one of them.

GLASER:

That's right. It turns out to be because of the problems of making the drum. There are two reasons. One, it was self-checked, and two, it was a simpler concept for people to understand superficially. Do you know anything about the structure of the Wooden Wheel, Bobbi?

MAPSTONE:

Not much.

GLASER:

Is it worth taking up time with a description?

MAPSTONE:

Very definitely.

GLASER:

Look, why don't you get into something else and let me think about it for a couple of minutes to see how to get it across in a few words, okay?

MAPSTONE:

Okay.

GLASER:

So, go with these guys and I'll probably be listening.

By the way, one--it doesn't have to do with the structure of the thing--but one even that is sort of interesting: I often wondered about the politics and I'm not sure what happened. About mid-1954, summer of 1954, the--let's see, how did this go? Hughes became the head of the Endicott Laboratory,

GLASER:

That's right. And there was a movement...

GLASER:

Of a whole lot of people out to Poughkeepsie.

Well, not only that, but there was a guy by the name of Walgren who was made Head of Planning for the Endicott laboratories in 1954. He got some instructions that came directly from--who was the head of engineering in Madison Avenue at this time?

GLASER:

Wally McDowell.

WAY:

Oh, that's right! McDowell. I remember he had orders for Walgren's group that I was in, got orders to reinvestigate the whole design and process of the Wooden Wheel. This was about a year and a half after--At this time, the 650 was almost a commercial item. It actually—

GLASER:

I can tell you what caused that. So it was about a two-year interval after the thing had apparently been snowed in favor of the 650, there was this attempt to revive it that lasted about three months.

GLASER:

Yes. There was an attempt to revive it. It was to be the 750, and that's when the whole thing got scrubbed, and I finally left.

MAPSTONE:

What brought it up again and what scrubbed it, do you know?

GLASER:

As it turns out, some people wouldn't let it die. They were able to show how much more powerful it was than anything around. But the problem was that the group--I had requested to go talk to the group in Endicott that was making the evaluation, and I was denied that request because I might prejudice them, because they had a guy give the information--I know what documents he got—

WAY:

They weren't even documents.

GLASER:

And it turns out most of the reasonable stuff had been cut out. A couple of things--let me describe the machine and let me tell you what happened. Oh, one of the arguments was that because it was a plug board machine, it couldn't be self-checked.

MAPSTONE:

Was that true?

GLASER:

It turns out all of the arithmetic and transfers could be checked. In fact, it's one of the few cases that I know of that a--now serial-serial was no problem, and parallel-parallel was no problem. And the intermediate set was parallel-series and would transmit all the bits of a character or a digit at the same time. And because of the funny nature of the machine, I didn't worry about it. We'd transmit all the ones and all the twos and so on together, and it made for a very fast arithmetic unit and was fully checked. The trouble was, it was being sold as a plug board machine rather than as a micro-program machine. All right, real quick description: the machine basically had a double-length accumulator. You can think of it almost as the old-fashioned desk calculator. Remember the carriage that moved back and forth? That you didn't shift in the accumulator but you could shift an arbitrary amount and it would go in and come back out again. The concept was what they called the program step. It was an event. An event that you wired from the top plus the various actions you wanted to take place. One of the top bits, however, was a special one that said a sequence, you know, that went into the input of the next one, so the sequence of steps was just totally wired. There were electronically controlled switches. They could alter the sequence or alter functions. The addressing of a small ono-memory was done by--if you wanted to know would zero this little bank, plug zero into the word one, or something like that.

However, there was a fairly separate arithmetic unit with its own channel. It had number registers on it. We could say, "Give me the contents of the memory whose address is in this register." And the registers would be used to address memory in this way as a totally general, add and subtract arithmetic unit. The real power of the machine came because you could do some highly compound things on it. We had what was called the, in the vernacular, the spotter function. It would pull a word out of memory and you would then wire it for an arbitrary break, so that various portions would go to different address registers. We had a couple of funny special registers in there where you could put in a digit and it would decode it, make a decision based on it. It was a decoding structure. And, in fact, one of the things we did was, it was a decimal machine, and we wired it up one day and the full machine at that point had 180 program steps to work. It was 100 program steps to the decimal equivalent of the 701. I think that was one of the things that killed it, because as long as we had some left over, we threw in square root, matrix conversions, for the fun of it. Well, that was not considered a good move. So that's a quick description, not a very cogent one, perhaps, but it gives you a feel.

Now what happened was they tried to take this architecture and style it into a stored program system. This happened about the time I was leaving. And it couldn't be done. As a matter of fact, I remember showing mathematically why they couldn't do it, and that they could change the architecture and do some things with it, but don't try to take architecture and bend it like that. It took them about a year and a half to figure it out.

MAPSTONE:

So that was the final death-knell of the whole thing?

GLASER:

Yes, that's right.

MAPSTONE:

Well, the other machines under consideration then were the 650 and the—

GLASER:

No, 650 only.

MAPSTONE:

Oh.

GLASER:

And the 650 got picked in mid-'53. So then later on they were looking for some other kinds of things that could be done. The 650 was still a sort of a computer kind of a thing, and they wanted something for business, so they looked at the wooden wheel again and some other things. There was a lot of gyrations, but very little was getting done.

A year later, '64, they looked at something that had a name which was the 608.

GLASER:

Yes. The 608 was going to be a 604 with magnetic core.

GLASER:

With magnetic core and serial arithmetic. There were a lot of little things like that going on both at Poughkeepsie and Endicott. I never, you know...

GLASER:

We built a funny drum machine at one point with a giant, funny card walloping gadget and printer on it. That was a stored program machine in a funny way, and we got that.

MAPSTONE:

Which was this, Ted?

GLASER:

It was called the Card Processing Machine. Oh, yeah.

MAPSTONE:

Oh, yes!

WAY:

I remember that.

GLASER:

I did the logic design on it.

MAPSTONE:

I always just presumed it was designed right after the CPC.

WAY:

No, oh, no. This was a 1953ish type operation.

GLASER:

'54.

WAY:

'54.

GLASER:

Yes. In fact, that ended up with a very funny one, because what happened was we continued this interlace game, except what we did was use, the word-length was six or twelve digits, and we'd grab an alpha in there. You couldn't operate on the alpha, but you could compare it and pass it through. It hadn't-- Often, for example, if you wanted, it had different motive--I can't think of it, it was a transmitting and receiving mode, so in each sector, I think there were twenty sectors around the drum, you had seven registered and the transmitting mode was, let's say, level three, and the rece…

Actually, they were called the B-lines, because they merged with the lines in the Williams tube. And the A-lines were the accumulators, and they originally wanted, the original reason being that he wanted to be able to increment addresses because he didn't have much storage left to work, and he could test out his memory. And that was Harry's main contribution to that machine, about his only work. Let's see, what else came out of there? I never did know the gentleman--he was a Norwegian--who really designed a bunch of that original machinery.

MAPSTONE:

I was just going to ask you that. Ed Selmer.

GLASER:

I never met him. The 204 existed when I came, and the decision had already been made to build the Cardatron, 205, when I arrived, and John Alrich was to be building the point [?]. I took a look at the Cardatron, and my rationale was stated to the group as, "If you insist on doing it this way, this is the code you're going to use." Because they had an idea to use some strange code, and I said, "No. The code you're going to use, I guarantee it'll end up being in the same collating sequence as the collator, and don't ask questions. Just do it." [Chime strikes four].

MAPSTONE:

You laughed at that. The Cardatron?

WAY:

No matter what was the matter with the computer, you looked there. It didn't matter if the floating device was failing, the Cardatron was probably the source of the trouble. [Laughter]

GLASER:

Unless it was the tape control unit. That had something to do with it. [Laughter]

WAY:

That matured!

GLASER:

That was on the 220.

MAPSTONE:

You were weaned on the 205, weren't you?

WAY:

Yes, right.

GLASER:

It was fun, but it was a good machine.

MAPSTONE:

Tell me something about John Lenz's machine. And if you know, one of the things I can't understand is why did he go to ElectroData to build an IBM machine?

GLASER:

No, he didn't go to ElectroData to build an IBM machine. He was sent to ElectroData to build an IBM machine because John had been playing with this machine since '51. At the time he first built it, it was an interesting idea. But it was getting longer and longer and older and older. It turns out, to paraphrase MacArthur's saying, "At IBM old projects never die. Only the project heads are transferred." So, John Lenz was a friend of Tom, Senior, and therefore he was one of the untouchables, and they really couldn't just arbitrarily knock him off, because Tom, Senior, was still alive then. What most people don't know was that Tom, Senior, was a very senile man in his last days. He had this great dream of building a single building taller than the Empire State in the middle of Manhattan, and it was to be the new corporate center. And he'd periodically ask, "Well, what are we doing about that thing?" And they'd have to distract him and get him going and he'd get out the plans and make a few more changes in them, then he'd forget about it. But John Lenz could always get in to see him. So, they had to do something with him. They didn't want him building in the place, so they sent him off to ElectroData because they didn't have the engineering capacity to do it. And John was a crack engineer, but everything he used was different. For example, there was not a flip-flop in the machine. Everything was done with Schmidt triggers and Fenestron [?] circuits right out of the old radar, and the addition was done with a well and bucket integrated circuit.

WAY:

... circuits.

GLASER:

What? You remember those.

WAY:

Except that when you see them...

GLASER:

And you remember what a well and bucket integration circuit was, too, and that's how the adder worked. And, for example, to show you how it was, do you know what the code that was used on the drum? Hollerith code. And when I say Hollerith code, I mean it went nine, eight, seven, six, five, four, three, two, one, zero, X, R. X meant minus and R meant--it turns out R was used for other things to keep track of carries and some of the strange things like this. And that is literally how it worked. Okay? Now it turns out they didn't know how to kill it, so they finally got the thing built, and John at that point was getting violent. "If you're not going to build, I'm going to go off--if you're not going to market, I'm going to go off and do it!" And he'd halfway conned some people at ElectroData to push for it, because he was a very forceful man, a brilliant engineer, so the most painless thing they could do was to offer it as--in the division they knew they were going to sell off, save some time and equipment.

MAPSTONE:

That was a vacuum tube machine and also used relays, right?

GLASER:

Yes. It turns out IBM's first major transistor machine was the 7070. And IBM initially designed a core 650. Actually, it was to be the 660. The 660 was built, they advertised in the literature someplace. It turns out there were some people that, when I left IBM, kept trying to pump me on what was going on. So, I suddenly realized what was happening, because I wasn't trying to leak information to them. I kept telling them the truth, just not all of it. They didn't know what I was doing, so I kept telling them about my problems in trying to put core arms, yeah, 205, to try and do a faster machine with a core and the loops only. Every time they asked me a question, I'd give them a true answer, but not all of it. I had several people afterwards tell me that the 660 was a direct result of that. Now the 7070 was a machine that was very similar to one I'd designed in '53 and early '54, but they went back and rebuilt it with transistors. In fact, I had a compliment paid to me. What was the name of that Southern California computer group? Jim, you were a member, and so was I for a long time.

JIM:

DCA.

GLASER:

That was it! Digital Computer Association. At the time, there was the first presentation of the 7070, and I wanted to hear it, and I suddenly recognized what it was. I didn't say anything. I didn't want to embarrass anyone. It was late '57. I've forgotten who was presenting it. It was one of the guys from IBM. So, it was all over and the questions, and finally somebody asked him was that a plan? This guy from Lockheed. He said, "Gee, this is a fairly smooth machine. Who designed it?" And bless him--I can't remember his last name, his first name was Jerry--he said, "Well, let's be truthful. He's here tonight. It's Ted Glaser. He just doesn't happen to work for us anymore." Which I thought was very sweet, because I hadn't even seen him since I'd left the place.

MAPSTONE:

That's nice.

GLASER:

Yes, I think it is. So, let's see. That's about all I can tell you.

MAPSTONE:

One of the drawbacks of Lenz's computer, just for a minute, one of the things I understood it had was the ability to do automatic square rootings. Was this so?

GLASER:

Yes. This was added later. The original machine couldn't.

WAY:

2.23 seconds average square root.

GLASER:

Mhm.

WAY:

And the auto point mode and fixed point was 1.99.

GLASER:

That's right. It turns out remember that you programmed this by running through a set of steps you wanted to take while it punched a tape. An then you used that tape to run back through again, and that's how you programmed it, although it also did have a plug board.

MAPSTONE:

Were other machines at this point doing automatic square rooting? Had it become sort of fairly standard
device?

GLASER:

No. No.

MAPSTONE:

Was it required?

GLASER:

No, it was something they just saw how to put in, so they added it.

WAY:

Well, somebody already put it in a desk calculator, right?

GLASER:

No. It wasn't in the desk calculator yet. The first square root Friden came out in about '58, '57, right around in there. The desk calculator was used to find square root by subtracting out numbers.

WAY:

Yeah, oh, yeah, but I mean—

MAPSTONE:

But it was just never built into anything until, what?

WAY:

Until the Friden came out. But I didn't realize what year it was.

GLASER:

About '57. And it turned out, however—

WAY:

Well, Rea's machine had a--the READIX had a square root in it, didn't it?

GLASER:

I can't remember whether it had that or not.

WAY:

I think it did.

GLASER:

Now the Wooden Wheel had a square root built into it in the second version. The first one didn't. But all they did was we'd put my algorithm in behind the board, which had a number, you had a wheel with and played some games with it, so you divided by two, see. As it turns out, we end up using the odd number, it's the McClaren expansion. You know, the square root is the quotient, and your trial divisor is twice the square root. Well, by playing games, you know, dividing by two ahead of time and doing a couple of other things, it comes out smoother, so you didn't have...

WAY:

That's just the way you learned how to do it in school. I guess they don't teach it anymore. Bells and Whistles.

GLASER:

You end up getting rid of one register. Sort of like the old days. You'd run your cut. How can you do integration with the fewest number memory locations, because they used to be critical.

WAY:

That's how Gill got his name on it.

GLASER:

What?

WAY:

That's how Gill got his name on the cut, picking the coefficients right. Everything.

GLASER:

That was Stan Gill, wasn't it?

WAY:

Yes.

GLASER:

Yes. He was a delight.

WAY:

You had to precede every written instruction with an instruction to turn on the typewriter. [Laughter]

GLASER:

That's right.

MAPSTONE:

Which machine was this?

GLASER:

This was the Auto point 610, and it turned because that turned on the drive to the typewriter. Oh, it had some pretty strange things in it. In fact, the basic built-in operation did the function of A over B time C plus D in one fell swoop. And by plugging ones in the right places, you could get it to do something you wanted it to, so they decided to make that a feature. It turns out that the reason it did that is because that was an algorithm from the old mechanical ones. It was a little-known version of the 601-mechanical multiplier. It was sort of between the 601 and never really came out very much. They played with it for a while and then came out with the 602, which was actually a much better machine than the 602a, except if you've ever tried to program it, you ended up in the booby hatch, because it did it by left and right components and you had to remember which was which and how to add them together yourself later. But you were happier not knowing. That was a mechanical monster also.

WAY:

[reading?] "___a step at a time, execution or typing out the program are particularly easy on the 160. To step through a program, the programmer must hold down the IMT key with one finger while he depresses the RSM key for each digit read from tape or plug board. Of the forty tape codes, nine are double column characters whose execution requires two RSM key depressions." [laughter from audience] "If the programmer wishes to type out his state program, he must: A) turn Type Program switch on and depress the RSM key; B) when he sees the typewriter nearing the end of the line, he must stop the operation by depressing the INT key and manually return the typewriter carriage; C) then he must depress the RSM key to type out the next line. After repeating this process until the program has been printed, the programmer will find that his instructions appear in typewriter code single character symbols, only sixteen of which are common to the keyboard symbols, of which there are forty-one that he has used in preparing this program initially." (laughter from audience) "Normally, one would expect that the cathode ray tube display would be helpful in debugging. But a glance at the 160 CIT display shows that it is not only limited in what it displays, only one internal register at a time, no instructions, but also very difficult to read."

GLASER:

What are you reading this from?

WAY:

It's a sales primer I wrote in October 17, 1957.

GLASER:

Oh, good saints!

WAY:

It's one of the best things I ever wrote.

GLASER:

Keep going, I like it.

MAPSTONE:

?

WAY:

"...but also very difficult to read. Position coding on a 30x10 matrix of spots, negatives, and component form, etc. Here again, the IBM advertisers have played up this feature as a big asset. Some book should have referred them to the IBM manual of operation, which states: 'The cathode ray tube is an indication of the existence of data and the selective register, and is not recommended for the reading of the data in the selective register. If a person wishes to examine in detail the contents of the selective register, he need only read it out on the typewriter.'" That's all. [Laugh]

Talk about mad machine interaction!

GLASER:

Well, this was something else again.

MAPSTONE:

I'd like to get a copy of it.

GLASER:

I'd be glad to.

GLASER:

By the way, Jim, do you have any of the old documentation? I'd mentioned it to Bobbi. The valse [?] Rechenmaschine?

REA:

No.

GLASER:

I don't think I do, either. That was one that never saw the light of day.

MAPSTONE:

That's too bad. That would be nice documentation.

GLASER:

You shouldn't have. Well again, remember that the READIX--that was Jim Rea's machine--that was a ten-digit machine with sign. Now it turned out instructions were four digits for address and one digit for operating. Now since he had a sign left over, if it was a plus it didn't do anything, if it was a minus the left-hand word had the index register added to it but the right-hand word you couldn't do anything about.

REA:

Similar to the structure in the A1. If you did a transfer, a jump of any sort, it was only a right-hand instruction and address was used for that. It didn't matter what the left said.

GLASER:

Now it turns out that in that machine, if you jumped you always jumped to the first word, the first instruction in a word. It didn't make any difference where you wanted to go. Although of all the machines that were ever sold in quantity, the G15 was probably the most insane.

MAPSTONE:

Tell me why.

GLASER:

Are you sure you want to know?

MAPSTONE:

I'm positive I want to know.

GLASER:

All right. In the first place, every word in that machine was on the drum, and all the words were recirculating registers. Now it was a binary machine; the registers went up from zero to twenty-three. That was the lines, and there were 108 words per line. Are you with me so far?

REA:

Talk about weird numerology, this is—

GLASER:

Now it turned out you could only execute instructions in lines zero or one. Now it turns out--I'll tell you the sensible part first. Now there are four additional lines, each one of which had four words in it for immediate access, plus two lines of one word each, two words each. Now depending upon whether you were in the odd or even word of the sector told you whether you were opposite the odd or even word of the two-word line, which meant that you'd either add or subtract unless the invert bit was on. [everybody laughs except Glaser] Now that was the sensible part of the machine. It went downhill from there. Just like that! Just never mind, it gets worse!

MAPSTONE:

I had been told that it was difficult to handle. I was curious as to how it was.

GLASER:

Are you beginning to get the feel?

MAPSTONE:

You know, the thing, just listening to all this, it's incredible that you were able to use machines. I mean,
that just says so much for the ingenuity of the people that were using them.

GLASER:

That's right. That's right. Bobbi, did you ever get a chance to use any of the old IBM punched card
equipment?

MAPSTONE:

Um-hmm.

GLASER:

You know about table punches and summary punches?

MAPSTONE:

Um-hmm.

GLASER:

Did you ever hear of a thing called progressive digiting?

MAPSTONE:

I think not.

GLASER:

Shall I tell her? Maybe you don't know about progressive digiting.

REA:

Secret.

GLASER:

What?

REA:

Sure, tell us.

GLASER:

Progressive digiting was the way to multiply and get sums of products using nothing but your bare hands, a tabulator, a summary punch, and a sorter. [Laughter] Okay? Now it turned out what you did was--because this was used for all kinds of correlation. Let's say you had a bunch of numbers by a lot of different ways into it, you'd end up saying, "I have a bunch of numbers on these cards and I want to multiply all these numbers by this common set of factors and get sums of products." We used to have the multiplier appear on every card. Okay? You'd then sort the cards in order on a lower digit. Okay?
And what you'd do is put the nines in first, then the eights and the sevens and so on. The only thing that was multiplied by a nine appeared nine times. Anything that appeared by an eight, it appeared eight times, and so on. Every time you got a break, you got a summary punch. And you'd do it in the tens and take all the summary punch cards out and take the zeros and the ones that were in the units' position would be marked with--would be not marked, and the ones for the tens would be marked with an X, so next time you put them through, with a switch thrown, the X gets offset, and you add the

whole bunch together and you get the sums of products and all the stuff and it would come blatting out. That's called progressive digiting.

REA:

No wonder the parts business was so good! [Laughter]

?:

Be quiet, I wouldn't be surprised if you get another problem.

GLASER:

Or how does that grab you?

?:

Another sort of a gem.

GLASER:

Yes, that's right.

REA:

And they didn't stop, either, they just chewed the stuff right out.

?:

They just kept right on going until you pulled the plug out of the wall.

GLASER:

That's right.

MAPSTONE:

I really think it's a tremendous comment on how we ever got here when you consider where we've come from.

GLASER:

Yes, that's right.

MAPSTONE:

And the obstinacy of—

REA:

In some respects, we can see where we are, and it's a little harder to foresee to progress! [Laughter] Look at J.C. Elms.

?:

Ooh!

GLASER:

But there are all kinds of interesting machines that it turns out that every once in awhile a whole slew of machine companies burst forth. This is a general phenomenon. Each one is directly identifiable with the emergence of a technology. Now the whole group of machine companies sprang out really in 1951, '52, 53, and they all pretty well by '57 had died that were going to die. Now that was a great sweep-out. You had ERA, NIVAC--by the way, do you know the origin of ERA, Engineering Research Associates? You do?

MAPSTONE:

Yes.

GLASER:

Where did it come from?

MAPSTONE:

I talked to a couple of people, Al Fenaughty and Emmett Quady, who were with it.

GLASER:

From when?

MAPSTONE:

1946? Or '47.

GLASER:

Did they join it in Minneapolis?

MAPSTONE:

Yes.

GLASER:

Okay. That's not when it was born.

MAPSTONE:

It was in the Navy, right?

GLASER:

It was born in the National Naval Security Service, which was the Navy branch of NSA or what became NSA. That's where Bill Norris and Frank Miney and a whole bunch of them worked. And they actually tried to get IBM to build this drum machine they wanted, and IBM didn't want to do it. It had been designed there, so they said, "Well, we'll form the company," because they all wanted to go back to civilian life. So, they all agreed to it. So, they actually signed the contract on a machine that was designed, actually on a Navy post, and they signed it the next day after they were civilians to make it legal. The original machine, the 1101, was Binary Thirteen. This was built and designed in Building Thirteen at Arlington Hall Station.

[break in interview]

MAPSTONE:

Let's talk about CEC.

GLASER:

Yes. Howard you know about; Lloyd Cody is back with the Russian.[?] In fact, originally they built the arithmetic unit and the memory separately. One day they decided, "Gee, let's for the fun of it try to throw them together." So, they did. For a long time that was around just as a test thing, and there were things put on two by fours to keep the cables together. One was known as Foster's Fall and the other was known as Cle's Causeway. And Gerry Foster is with IBM San Jose and Cle is Burroughs, Howitch you know, Gloria Bullock, one of the earlier programmers, she is working with Allstate on the West Coast. Kreuder was not one of the early ones, but he's still around with

Xerox. I don't know what happened to Allan Beek. He did the early design on the tape controller for the 205. He went with Logistics Research and then dropped out of sight.

MAPSTONE:

He's _________.

GLASER:

Good heavens. Okay. Most of the others--McDonald is really in retirement, Duncan McDonald.

MAPSTONE:

Yes. Robinson?

GLASER:

Robinson is free consultant with someone up in the Bay area. Bradburn you know about; Rice is retired. Those were really the main players. Most of the others are totally out the computer business.

MAPSTONE:

Now, was the 220 actually your architecture?

GLASER:

Yes.

MAPSTONE:

Would you talk about that machine for a bit as you saw it and conceived it?

GLASER:

Let me tell you as it was. I was constrained to build it to be more or less the same architecture as the 205. Therefore, it had the A-register, the A-register which is this tension on the right, and the B-box. In terms of the architecture of the arithmetic unit, there's not much to say other than the A-register got turned into a working register. It was not blind any longer. The shifts got generalized. We—

REA:

It had that partial word business.

GLASER:

Went to a partial word so that you can manipulate them. I wanted to put them on all the instructions but got overruled. And it would have been just as easy. I also wanted to turn the air register into a full-fledged accumulator. All the paths were there. But that also got overruled. And the original 205: it had what was called a D register or drum register, because they wanted to bring the whole word out so you could see the high order sign before you knew what to do with it. They then added it from there into the accumulator, and then went directly from the accumulator back into the memory because they could do that in serial fashion. Because of this, it was insisted we do it the same way. I at least got them to bypass the D register so you could hook direct with the accumulator so we didn't have to transfer twice. Actually there was no way we couldn't have sent the thing over in parallel.

REA:

In fact, we did.

GLASER:

Well, it turns out that it was designed to be able to do that but that was one of the things I couldn't get in,
so that people here put the modification in because it was obviously easy to do.

REA:

It was, too.

GLASER:

Now it turns out that the core memory in that machine was a parallel memory, therefore most of the time it was loafing. The reason for this was that ultimately I wanted to cut out the Cardatron and cut out the kind of tape we were using and go directly to and from memory, which meant it could have been doing simultaneous read, write and compute. But again, I got overruled by some people in product planning, because they said it was much more important to be able to modify instructions as they came into memory by the B register, a feature nobody ever used. Oh, there were some other funny things that were done. I wanted to be able to interrogate the little keyboard directly from instruction. When a light came on, you could do it directly. Well, so nobody was afraid they might hit something else. When the light came on, he had to hit two buttons and flip the switch down before he could do it. Again, it was an unneeded safeguard. But if you really looked at it, it turned out that could be defeated by bridging one contact. I don't think anybody ever noticed that.

[Chime]

MAPSTONE:

Do you want to just hold it while I quickly change tapes?

GLASER:

Sure.

[End of Reel 1, Side 2]

[Begin Reel 2, Side 1]

MAPSTONE:

This is Side Three of my interview with Ted Glaser et al.--if you don't mind being et al. at this point! [Laughter]--at Case Western Reserve. Okay.

GLASER:

Now let's see. So that's it. The Cardatron and the tape--the Cardatron had to be kept intact, so they just put an adapter box on it. The tape also wanted to be kept intact, so I tried to restrain that a little bit from what it was on the 205. It turns out there was a separate control, believe it or not. The tape unit was an interesting one, but there is not much I can add there. The guy who designed that was a guy named Paul Gelson. I had very little I could do about it. Let's see, what else can I tell you about the philosophy? Yes, there was some sort of interesting kinds of things. We did some funny things with the sign, again to make it easy to identify what was in memory as an alphabetic—

REA:

Oh, yes, a two.

GLASER:

A two. There were a couple of signs in there that nobody quite knew what to do with. One was a minus infinity and the other was a plus infinity. Anything you put on there would compare--minus infinity would compare lower than anything else you could create any other way. The plus infinity did the same thing. It turned out that it had very little use in scientific work, but if you're doing data processing, setting up your things right and putting minus infinity in the right place, or plus infinity-- depending on what you were doing--you wouldn't have to have anything special in terms of programming when you were updating files, because this turned out you started out with those as dummy records that came in, which you set up on an initializing, then all your programs would work right.

MAPSTONE:

That was an unexpected bonus.

GLASER:

No, that was done on purpose.

MAPSTONE:

Oh, that was done on purpose.

GLASER:

Yes. Let's see? What else in the architecture? It turned out when I couldn't move on to the simultaneous thing, due to all this argument of how a buffer stopped to and from a machine. Well, the way the machine was designed at that point, we went ahead and fixed it and did something strange, that as long as you couldn't buffer you might as well operate in the most efficient way so that, for example, when you were reading tape or writing tape, the entire machine was working off the tape clock. The same was true of the Cardatron. It became a totally synchronized machine, because at that point you didn't have to do any buffering. It was the easiest way to handle it as long as you couldn't do it the other way. I guess the best description in that sense of the architecture, we tried to keep it as clean as possible with as few exceptions as were humanly available at that time. The other thing is that nobody really knew how to design a control and people had all kinds of strange ideas of separate time ________, so I analyzed it and it appeared the simplest thing to do was come up with a funny kind of central control. I came up with a funny kind of control that was really a microprogram, sort of a two-level thing because it turned out you had two kinds of pulses, sequence pulses, which basically set up basic logic. Since this was a serial machine, you handled it a digit at a time. You had digit pulses. On a sequence pulse, you could set up a counter to measure out so many digit pulses. And on those you'd set up certain logic and then it would step to the next thing. Now there was nothing I could read onto the memory, but you just took simple diode gates and put them together. It turns out that John then grabbed this design, because I'd laid out all the instructions this way and had a bunch of kids there were fresh out of college--literally. They'd graduated in June. And they put together the 220 that summer.

MAPSTONE:

Oh, great.

GLASER:

Oh, boy. Ken Crossa is the one I can think of; there were others, but the one I can come up with a name for right now. But there were about two or three kids, and that's how we threw the crazy thing together.

MAPSTONE:

Fred, as a user of the machine, how did you feel about it?

WAY:

Excellent. One of the best machines we had. Undoubtedly it had the best console and ____ we'd ever seen. It was swell.

GLASER:

The console, by the way, again, the only thing I wasn't happy about was a couple of registers' positions I would have changed slightly, but that was a minor thing.

WAY:

But everything in the machine was visible from the console. They had some extra doors that would open up so you could see all the internal flip-flops and what not. So, it made it nice for the maintenance man. See, we maintained the thing ourselves.

GLASER:

What we did: again, there was a big argument that you should never show the customer too much. So, I got them to compromise that there were doors on each side. When they were flipped back, you could flip them back out of the way or even take them off, and everything was there. If the customer didn't want to be confused, you could close those, and all the working registries that he normally used were there. He could set anything or clear anything. Anyway, he had certain quite simple controls available to him. And that was about it.

WAY:

Oh, it was an excellent piece of machinery. It was great.

MAPSTONE:

A good machine to program?

WAY:

Oh, yes, it was beautiful. Really was.

MAPSTONE:

And who developed programs for this?

GLASER:

The thing was shipped with almost no software initially. It was a register-forward machine. Assemblers came in later. Several people did that. The Burroughs Algo Compiler--got to be known as the Balcom—

WAY:

Yes, that was good.

GLASER:

That was done by Jack Meyer and Joe Lurgman. I don't know where Jack is now. He's drifted around to a lot of different companies. 0ne was with Honeywell for awhile, then RCA. I don't know who he's with at this point. Up in the Bay area. Joe Lurgman was only with IBM for a very short time, and then went to--not IBM. What am I saying? Burroughs. He then went to Computer Science as soon as his project was done.

WAY:

Oh, sure. He even had his hands in the Exec II.

GLASER:

Exec II, that's right. Bob Barton pulled those guys in. Bob had left IBM before I went with Shell, the ______Shell development, and I conned him into coming back into the computer business and coming to Burroughs. He put that one together and then somewhat later I left the West Coast and went to _______. Things worked out the way he wanted them to, so he apparently came to ______ with two of us plus a guy named Keither Spiremand and Jim Anderson. Did a lot of the original conceptual work on what became the 5000.

MAPSTONE:

Which was a highly successful machine. That then became the 5500.

GLASER:

That's right.

MAPSTONE:

Were you really ahead of the state of the art or were you kind of just working at the maximum of the state of the art with this machine?

GLASER:

Which?

MAPSTONE:

The 220.

GLASER:

Bobbi, I don't know. The circuits were old. Electronically it was no great shakes. In terms of design, it was not the best we knew how to do but it was the best we could get people to accept. Main decisions were highly political. At that time--and indeed, even now--I'm not sure what state of the art really means in architecture. The 5000 was about as good as we could do at the time, although again there were mistakes at the time it went together that I thought were in it. But it was the best compromise we could get because even then we had to get people, too many people, to agree and sign off that yes, it had their individual ideas in it. So, there were some mistakes in it. The 5500 carried some of them. The floating channel, for instance, was something that we'd wanted to do earlier. We finally got it into the 5500. This is a thing Fred worked. You don't have a tape on a channel. You just have diode devices sitting out there and you'd have one to four channels. You'd ask for an iota device and if the channel was available, you'd do it. But it turns out that both Duncan and McDonald came from a telephone background and I'd studied them and got to the point where I was qualified to design diode networks, because I wanted to see how it worked. And a lot of that thinking went into that machine. And in fact, well, you talked about stacking a program table, and it turns out Bob had never known about Polish notation, so he went back one time and says, "Geez, why don't you do a stack?" And I say, "Do Polish." And that's how that thing came up. There was an earlier machine called the 2111, but that killed. It came up with the idea of a syllable, you know, which was the sort of arable length instruction type of thing that you have on the 5000.

MAPSTONE:

Yes, yes.

GLASER:

The program reference table and a lot of that descriptive concept came out of a machine that was not too well-known in this country called the Parnmasch, built in Germany, built at the University of Parnm in Germany, P-a-r-n.

MAPSTONE:

And who did that one?

GLASER:

Don't know. We just saw a paper on it and it looked like an interesting idea, and we started to draw in some things from there.

MAPSTONE:

Oh, that's interesting. I've never even heard of such a thing.

GLASER:

First rule of thumb: the only kind of pleasure is an excusable steal and a bad idea. It turns out our
attitude was that if somebody had a good idea, look at it, read everything you can. ____?

MAPSTONE:

You know, earlier we'd kind of been talking about some of the problems with using machines and all those rather negative aspects. What, in both of your minds, were some of the good machines and the machines that really moved the art on to help the technology grow?

WAY:

Well, from a user's standpoint, I don't know whether it really has to be the machine or the software available.

GLASER:

The combination of the set.

MAPSTONE:

Both, either, or, and both.

WAY:

Because we both here created the first symbolic assembler that ever ran on a UNIVAC I. This was late '57, which is incredible. We also had the first, yes, we had the first algebraic compiler that ran on the 1107. The 220 is about the only one we ever had that we didn't scissor up the software on. We used the Burroughs compiler mainly. We also wrote a symbolic assembler for it. What was it, Star?

GLASER:

Yes, which was a catastrophe.

WAY:

Oh, it was a mess.

GLASER:

It turns out there was another assembler that got built and wouldn't be released. It was a very nice macro-assembler. But they built it primarily for playing with and nobody ever used it. I don't know what ever happened to it.

WAY:

Connelly and Stelp put one together that we had here. It was called Save. And then we did add a rudimentary operating system on the top of it. We added daytime clock to it so it could log stuff on it. And we didn't really get into the software up here until the assembler on the UNIVAC I and the compiler on the 1107. In fact, the first compiler we had, the 1107 ran compatible with the 220 so it could use the same data. We had both machines in at the same time, and we thought we'd ease the transition for the campus log. So, it was a Burroughs compiler on an 1107.

GLASER:

For awhile.

WAY:

For awhile, and then we went to Algol 60.

MAPSTONE:

Oh, ________, Algol 60. I never talked to a user of Algol.

WAY:

Oh, boy!

GLASER:

That's a major language, as I understand.

WAY:

That's the major language for using here.

MAPSTONE:

Is it?

GLASER:

Yes.

WAY:

Yes and it started on the 1107.

GLASER:

And I'd say--now I wasn't here then--by my guess is that the main reason that the campus did not go to a
5000 or 5500 was not because of Algol but because of Burroughs' arrogant attitude about not letting the
user ever inside to do what he needed in the software.

MAPSTONE:

Well, Burroughs had that attitude, too.

WAY:

Oh, they did. They said right out, in fact, as soon as they announced the machine, they did not want anybody--in other words, if you weren't going to use Algol, you weren't going to use the machine, in essence was what it was all about.

GLASER:

And it was the arrogance.

WAY:

Yes, that met with great resistance in this quarter. I don't know about any builders.

GLASER:

Had, I'm sure, had Burroughs not announced that attitude, which turned out not to be their real attitude, my guess is that this would still be a Burroughs shop, because the 5000 and the 5500 would have done a lot of things that Bill wanted to do at the time.

WAY:

Sure.

GLASER:

And you and George and others.

WAY:

Yes. You'll find Knuth's name, and Winch, for instance, both involved--well, in fact, we've created some compilers for the 650.

GLASER:

Knuth, remember, did all his computing here.

MAPSTONE:

That's Donald.

GLASER:

Yes.

MAPSTONE:

I didn't know that.

WAY:

And he also had his hands up to the elbows in the MCP on a 5000.

GLASER:

Yes.

WAY:

So, he's been at it for awhile.

MAPSTONE:

So, as a user, that Algol really was... The Algol that came out in '60, which was the second go-round...

WAY:

Yes, right. Well, that was what we implemented on the 1107.

MAPSTONE:

And that's the one that took off.

WAY:

And that's still around today on the 1108. It's virtually the same compiler.

GLASER:

And that's what all of our people here use.

WAY:

At least, we start all the students on it. There's a tremendous amount of Fortran being done, but we start with Algol.

GLASER:

Well, it turns out that a lot of people who have to on contracts deliver a Fortran program use a restricted set of Algol and develop all of their stuff in Algol and then transliterate to Fortran because Algol diagnostics are better. It's a more forgiving, it's a friendly compiler. That was what we tried to do with the MCP on the 5000, is make it friendly so it like people.

MAPSTONE:

That's a nice touch. So, any other thoughts on machines that you feel really were significant?

WAY:

Well, the 650 certainly was because of the sheer numbers of them.

GLASER:

The numbers of them. I'll pick a couple of others.

WAY:

The _____ would have been better, I think, but I don't know why they didn't sell anymore.

GLASER:

Well, in the first place, it was never announced; it was leaked. I'm being quite serious. I thought that was the way they saw it. Some other things--let's see. Well, it wasn't on the West Coast, but I think it was a significant machine, the E101, because it was the first machine that was conceived of as a personal computer.

MAPSTONE:

By personal you mean that you could use it?

GLASER:

A person sat down and programmed at the machine. You didn't come up with a card deck. It was sort of meant to be interactive somehow in a very crude way.

MAPSTONE:

Is that the first interactive machine?

GLASER:

Well...

MAPSTONE:

Or, I mean, the first—

GLASER:

Commercial one? Yes, I would say so. Now there were others that were used this way but none that were really sort of designed in that way from the beginning. I think the 5000 was a very significant machine. It should have proved more to people and it didn't, namely, really what you could do by designing the architecture well. Now it was a machine that was slow by anybody's standards, and yet it's still a competitive machine. But in some funny ways not even Burroughs learned from it. Sort of sad.

MAPSTONE:

Was this again political?

GLASER:

No. You see, the people that really understood innards of why things were done, the 5000 and the 5500, weren't all around at the time of the 6500. And anyway, that was a step backwards. They optimized certain things but often it was local optimization. I had a terrible time getting it done. They were going to improve it just a little bit without really understanding what they were improving. Now what else? I think that other machines I'd pick as sort of interesting milestones... Certainly, well, I'd say the Burroughs 5000 was important for one reason: it was the first machine by five years that was marketed within the ______ operating system. It was doing in '61 when the rest didn't come up until '67, Exec, anything. Now the things it does have become quite standard now.

MAPSTONE:

Like this.

GLASER:

Yes. I think 1107 is important, because it was the first machine that showed a multiple register architecture.

WAY:

You had ______ation on that one. We got the thing in when this went wrong with the 220, and I think it executed instructions out of the registers in it. So here comes this thin-film memory machine and we thought, "Hot dog, look at this!: Of course, that was not one of the options. [Laughter] It's funny how a small change in the design of the thing affects the way the user sees it at the machine ________? level.

MAPSTONE:

Oh, sure.

WAY:

Yes, incredible.

GLASER:

I was trying to think of the other sort of interesting machines that were real milestones.

MAPSTONE:

Another thought while you're thinking about it. I think this would pertain to you, would be the language milestones, too. We talked about Algol.

WAY:

Well, Fortran was the one—

GLASER:

Fortran was the one that kicked it off. There's just no question.

MAPSTONE:

And basically, everything from there is just building up on the original principle.

GLASER:

Well, some of the concepts that got laid out there, although, okay, I'll give you another milestone, one that most people don't know about--you probably didn't tangle with it. It was GP, which was much before its time and could never work on the UNIVAC.

MAPSTONE:

GP?

GLASER:

Yes, it was an early kind of an operating system that was done by two guys named Holt and Teransky. The problem was that, had Teransky lived, it would have made all the difference in the world. Holt was the wild-eyed theoretician and philosopher, and Teransky was the hard-nosed guy that made it practical.

WAY:

Those guys hammered that whole thing in through the console.

GLASER:

That's right. And made it work.

WAY:

They couldn't find a card-to-tape converter within UNIVAC, and none of the UNITYPERS were working, and they handled it all in through the console.

GLASER:

Just typed in through the console.

WAY:

Wrote a block on the tape every time they got a block's worth of it done. [Laughter]

MAPSTONE:

That's interesting. What—

WAY:

An incredible piece of work!

GLASER:

Bill Teransky was hit by a car. It was a very tragic situation.

MAPSTONE:

Oh, dear. What time period was this?

GLASER:

Mid-1950's.

WAY:

'57, '58, somewhere around in there, because it was after we had our machine.

GLASER:

Right in through then.

WAY:

I think we got our E1 in '57.

MAPSTONE:

How about... Do you feel the West Coast, the developments that have come out of the West Coast, and I think we talked about machines and software, maybe even people, have had influence on the industry as a whole?

GLASER:

Floyd Steele was probably the first, because although the generalized DDA never took root, the concept is still around and used in control systems. Maybe it's only a couple of registers here and there, but it's the basic concept. Bob Barton, as an irritant and as a professional objector, was a guy that made things move. And certainly, much of the 5000 is his. Knutt did his computing here, but he received his recognition after going to the West Coast and certainly is in many ways <u>the</u> computer scholar of our generation.

MAPSTONE:

Where is Knutt now?

GLASER:

At Stanford. Any others? No. Those would be it.

MAPSTONE:

And Bob Barton is where now?

GLASER:

He now only works for Burroughs at _________ and he lives on Catalina.

MAPSTONE:

Oh, how nice.

GLASER:

Well, some people that don't like him point out that this is so he can walk to work. Other people? No, those are the main ones, I would say.

MAPSTONE:

Do you have any thoughts on it?

WAY:

From the West Coast. This sort of narrows it. Of course, Grace Hopper ought to get included in any list of the major _________.

(laughter)

MAPSTONE:

Of course, yes. She does.

GLASER:

Fletcher Jones, although he might not be, although you might not like him, would have to be included because Computer Sciences was a very unusual venture when they put it together. That would be it. Machines from the West Coast: the 5000, and the LGP30 just because it was an interesting small machine that was the precursor of the Mace. But the thing that was interesting was that when the transistor came, you had a whole additional _____of machine people that suddenly appeared. But they weren't West Coasters this time. The West Coast had its Golden Age and really interesting, alive, and vibrant things going on in the early '50s. The next big chunk occurred with the transistor and really integrated circuits was the thing that pushed it over the edge. It was the mixture, because the one came right after the other, and printed circuit boards so it was cheap to put machines out. But no really machines got spawned there with one exception, and that was Hewlitt Packard. The rest all really...

You know, he'd work by the cut-overs from existing mechanical builders, but the decks, the computer control corporations, and Harvey Wells which fell by the wayside, most of these were spawned around--and again, it was the TX2 people, the people that learned it there, or spinoffs from the Cape in Florida, and a couple here and there elsewhere, but really nothing on the West Coast. What is it, the Automation Something-or-other, it's out here, but it's not a major factor in the field anymore. Your major push hold in the Bay area, Deck, Data General, Hewlitt Packard, SEL, Integrated. And the ones that are left in California are, with the exception of Hewlitt Packard, jaked. Some of them just... I don't know why, but there was not this--maybe it was because of the aerospace. Maybe it was because it was too soon after it happened once before. But I went out there, and the very alive "Anybody can do it, let's try something new" didn't exist.

MAPSTONE:

When was this?

GLASER:

This was in the early '60's, late '50's and early '60s. I left California in '60. You see, I'm not a West Coaster. I consider it home because I love it and my family lives there. But I moved to California in '55 and left in '60, so it was between IBM and setting up the research group in Burroughs. Really, the majority of my life has been East.

MAPSTONE:

But there was this marvelous spirit, wasn't there, in that first thrust?

GLASER:

Yes. And DCA was one of the most alive, exciting places. Anybody would challenge anything. But it's gotten old and very tired and very jaded; and that happened by '60 and '61, where no matter what you brought out, "We've heard that before." And it was a tragedy, because people went from the exciting times to the above-it-all--I don't know

how to put it. The smart-aleck putdown is the only way I know how to put it in simple terms. It's not that simple, though. Interestingly enough, I saw the same thing happen again to a community. Cambridge... Ah, it's almost symbolic for me. I arrived from Cambridge to teach at MIT the night before Kennedy was shot, my last trip there with any kind of--you know, I'd already resigned from MIT. In fact, I was out here actually on campus when I had to go back for one last signoff. And I left there the night that Bobby Kennedy--not Bobby Kennedy, but Martin Luther King--was shot. And the thing that was interesting was that I recognized it in retrospect--you never see it at the time--but that was really the Golden Age in Cambridge. Then Cambridge started to come apart.
And the interesting kinds of people that were around were again by the time I left starting to get old and jaded, and they knew all the answers and "Don't tell me about that, I've heard it before." It's the type of thing you don't see when it's going on, but in retrospect it's very sad.

MAPSTONE:

I like the comment. They call it the Golden Days of the computer business, I suspect.

GLASER:

But you see, the Golden Days aren't over.

MAPSTONE:

They shouldn't be.

WAY:

They're not, either.

GLASER:

The most exciting things are yet to be done.

MAPSTONE:

But there probably had to be a slump, didn't there, sort of a rise and fall maybe?

GLASER:

No. They didn't have--All right. Perhaps there had to be because we're dealing with the exigencies of real people. No, the vision was never gone; but they arrived because of complacency or because they didn't want to make something and say, "We made a mistake."

WAY:

I'm going to zoom on. I'll look through my files down there and I'll give you anything I have—

[pause in interview]

MAPSTONE:

Earlier, you had talked about you had made changes on Mod 2CPC. I thought it might be interesting to just talk about what those changes were, what capacities it had?

GLASER:

Well, this is detailed kinds of changes. They had some very rigid structures on the model 1CPC called channels. You still had to wire a fairly heavy board over on the tabulator to get the thing to work. But the channels were really sort of built-in _________ structures. And on the Mod 2CPC I said, "Let's just give everybody all the selectors they want for ample relays." Well, it turned out, you look at the board and it's so heavy it's just not feasible to do anything. So, we put in control structures for busses so it would do some harmless decoding for you off the card and make life a little easier. But that was the main thing I did. And it made it possible to get the control to do the--there was a kind of rewinding memory to do, but by putting it behind the board you saved a lot of board space.   And that really gave us some... You can think of just some simple math. So, we did that. I got them to do some funny kinds of change in wiring. Again, are you familiar with the type of wire that was on these boards?

MAPSTONE:

Yes.

GLASER:

The usual collectors, you know, have common, normal, and transfer, in that sequence. But it turned out the thing we wanted to be able to do was change things together so that one _______ could be built. Well, this made for very messy wiring. So, you changed the normal transfers, counter transfer and normal. Well, then you stack the relays one above the other, so it's common transfer, normal common transfer, and all, so you just use the _____ which was pretty normal at the time of adjacent relays. We took off the bridge points, or you know, remove the points of a tree off the transfers. It really wasn't a tree, it was a sort of a chain. And then by using some slightly longer wires and bridging sideways, you could build a minor tree very easily. And that's how we did it.

MAPSTONE:

How do you feel about the CPC as a significant breakthrough in computers?

GLASER:

It was a significant breakthrough in the common sense as the first thing remotely resembling some kind of a computer where the program was not in it and part of the wiring. In that sense it was extremely important, because although it was a miserable machine, evil-tempered, electromechanical monster, the fact remained that it got computing out the door at a time that nobody thought it could do it. People forgot that it was on that computer calculator that such minor things as the first packing factions on a fusion shock, that would be really gigantic shot of the age ________, original conversions of potential control fusion when the Matterhorn Project was built, the first did the final analysis of the true error of the ____ helicopter, returned the feasibility of the photo analysis by sections on a hull which later became the DC 7. It did a lot of useful stuff.

MAPSTONE:

It sure did.

GLASER:

And, now, as a monster, it worked. You see, I really came in as a programmer, a junior mathematician. And nobody ever told me to do computer design, but I seemed to want to do it and be able to. I did it. So, I went from programmer to hardware designer, then I went back. And I built a couple of compilers and some long-range systems and so on. I started out as a user and I guess in a funny way I still am. I'm not sure of really what I am at this point, except I like computers. So that there again it's a very personal thing. I don't know how else to talk about it, because it's... It's the only life for me. I got out of college in 1951, took a master’s degree, and went to IBM. And I've been building machines ever since.

MAPSTONE:

So, you just came at the right time.

GLASER:

Yes, with a very fierce and intense desire to do certain kinds of things.

MAPSTONE:

That's, you know, to me this is the part of the history that's exciting, just this feeling.

GLASER:

Yes. I feel very fortunate. I was at the right place at the right time. By the time I left IBM I'd built nine machines that went out the door, even though they didn't know where to sell them. At this point I've built over thirty machines.

MAPSTONE:

Good heavens! Which were the nine?

GLASER:

Several--when I say built, I had at least an influence where I can say, "Look, this is a piece of hardware that I did, that's in the order code, or it's in the architecture, or it's in the software." So now one, two, four, and five, the same thing that became a sort of collator--I don't know the final number. It was 770 or 774, one or the other. And that was a tape-printer which was a rather sophisticated machine in its own right. Six and seven I had something to do with. Mod 2CPC 650. And that's all. I don't even count the two wooden wheels, although on the second one the changes that went into the machine became normal. And some military stuff. I remember one was the 220, the modern 5000. And much more military stuff with the 205s, special grid delegator. And I think on the 645 and _________.

MAPSTONE:

You've been busy.

GLASER:

Yes. And I still enjoy it.

MAPSTONE:

That's really great. That's really nice.

GLASER:

Except that it's time for me to change.

MAPSTONE:

Ah ha. Out of computers or into a new avenue?

GLASER:

No, neither. It's time for me to change in computers, and which is perhaps the next revolution

MAPSTONE:

That's... I would like to hear you talk about that but I'm going to not take you back, unfortunately.

GLASER:

No, let's put it this way--I won't tell you very much about it, other than just hold on a few years and I'll
have a few goodies to show you.

MAPSTONE:

Okay.

GLASER:

In fact, I won't be able to this trip. Suffice it to say that I have a machine which can be programmed in a very strange way, in English for my teaching. I built it with a cobalt compiler with a kind that didn't know anything about cobalt. In the parking lot with pliers ________, we produced compilers better than the equivalent IBM wanted to build in two weeks. So, we think we've got the next one going. I'm going to keep it under wraps because we think it's a pretty good patent.

MAPSTONE:

Oh, it sounds exciting.

GLASER:

So that history I'm trying to document for you, so come around one of these days. I think it will be fun.

MAPSTONE:

I will. It really does sound fun.

GLASER:

Ask the next one.

MAPSTONE:

Okay, why not find a copy of some ideas? What about the sort of great West Coast-East Coast battle on Boolean algebra as opposed to block diagrams? Did you ever get into that or have any feelings about either way being significant?

GLASER:

I'll do it either way, because you only have two idiots or four idiots, because by the time you're worrying about that lever of the thing, the thing's already frozen. You've got to worry about the bare number of relays to do the whole architectural structure of the machine. All they're worrying about is the details. So, it really doesn't make that much difference to do either thing; you can just interchange them. Oh, there's one thing that might be of interest. Probably the first interactive missile machine was the wooden wheel. That was the very first one with what they called a parameter board on it, which was ten words of ten ________ storage. The board was set up so it could request you to do something. Turn on the light--come on! (person enters) Well, hi, come on in, Bob. It turned out it would ask you about just turn the light on so it could spot itself, and you could change the parameter to lead them on and hit a button which says, "I've changed it." Well, you can see that what we were doing here was actually the design of circuits to show it could be done. I had been close to the Sage operation and knew about the cathode ray tube--the light concept, as it was then called. We didn't conceive you could draw with it nor track it. We came up with a tracker just one day--"Hey!" This isn't usually the way to work with a machine. Now the machine could come up with a bunch of questions, you know, a bunch of things and multiple choices. Now it turned out that by answering these questions, in a sense you were asking the machine questions. And I tried to get them to do this in the wooden wheel when we came up with our add interesting structure and these funny kind of tree structure of questions where we didn't think it was an interacting computer, you would ask it a question, it would ask you a question, and by playing back and forth this way, you were really having a dialogue. You thought of it as you wrote on one side and the ______ wrote on the other. Now we didn't know how to write with it. All we knew how to do was to know which one we pointed at and press a button, but that was enough and that, we thought, would be sort of fun. We never got very far with it because again it was just a little bit too rich for, well... It's funny.

I've often wondered--I've not seen this document, but there are fads in science as surely as there are in clothing or anything else. And God help you if you are doing something that is not a fad! It turns out if you're lucky you'll be a renegade, but most of the time you'll be a martyr. But if you worry about such things, they will bite you. If you don't, you just go ahead and do them. But this was not in the fad then, so it went away at that time. But, oh, there were some nifty times we had on that one! It was sort of fun. But the East Coast-West Coast battle again at that time was the preoccupation with "how do I save
diodes?" But it turns out that at the time I had the tubes to save the diodes, so it was already quite clear that that was no longer the issue. It turns out that legends or the unwritten laws die very hard. The first multiple-register machine really only came in in the early mid-sixties. This was because flip-flops in the early days cost you four tubes, or really two twin triodes, two tubes, two envelopes. That's expensive. But those days were gone by the late fifties. The people kept worrying, though.

And I'll give you one other funny story. Core memory is slow and watch, it gets very fast. Therefore, you could do all the logic in one step. Well, when people suddenly realized that was the case, that in fact things like multiply and divide, even the trick of multiply and divide was fairly fast, so you wanted to go directly to the main memory. That was a great idea. So now all of a sudden, our logic speeds have exceeded our memory speeds again. We have our earlier idea and we're still time-sharing into memory and you get all sorts of memory conflicts. When we get over it, why, the bank switch will be good. It's a new idea that goes back to the old days of the air raid horn. Here's the quote many times, I've always oved to hear it said ______"That nation that does not read history is doomed to repeat it." Well, it's true of industries, too. And we keep rediscovering it in different forms.

MAPSTONE:

This is something I've actually been trying to trade, is how often have these rediscoveries taken place in
the computer business and what are they?

GLASER:

Well, it turns out there are a lot of them. In fact, one night I went and just sat around shooting the breeze to see if we could track them. At about two hundred we stopped counting.

MAPSTONE:

Good Lord!

GLASER:

I'm being dead serious.

MAPSTONE:

Give me a few of them.

GLASER:

I've given you two just now, with the change of thing. Another one: the big deal in drum or delay line machines was minimum legacy programming, sometimes called optimum programming, so that by the time your execution is done the next instruction comes up in place. It turns out if you cross your eyes and go back and look how they did it, the games that people play with pipeline machines are identical. They just don't recognize it. And they're rediscovering the same set of alga rhythms. Okay, that's an example.

MAPSTONE:

It's a good one.

GLASER:

Some overflow rules and the fact that the DDA's are unstable because to get serial you have to alternate one and zero. The big thing that happened with the D12 Bendix was they said, "Let's have plus one, minus one, and zero, so we have a true zero." And all of a sudden, the whole thing no longer oscillated, it no longer had a bias in one direction, it was smooth. People in some control systems using DDA's which are now coming out on ships and some traffic systems are just rediscovering it. That's odd.

MAPSTONE:

Yes. I think—

GLASER:

And to my way of talking, it's one of the reasons I decided to teach, because I suddenly found this terrible thing, that the only way I could get some people to learn what I knew was by the apprentice system. And damn it, that's a ________, but it's not a computer science. And that's one of the reasons that at all we built this funny project here called Logos, which is a formal design system so we could move the arrangement a little and do it by more than an alteration, which is much of computer technology right now. I know most people don't like to admit it, not the things in the textbook but the things that really make a system work. Again, there are three machines: there's the machine in the programmer's manual; there's the machine in the engineer's, you know, customer-engineer's, guidebook; there's the machine that's on the floor. They're similar but they're not identical. Again, I'm not trying to be funny; I wish I were. You know yourself if you want to get something done on a Less, you don't need to do that, you can let Less use his own machine. That's because we've had this great burgeoning of technology because of this lead. You've had a tradition, then the companies have prevented us from following sound engineering precepts to get something done because it "wouldn't sell". Well, you also had a very difficult problem, something that was on the order of magnitude tougher than ________ mechanic trying to get enough nautical definition to know where to turn, because he could go look at the universe. And I hypothesized an experiment to test his view: that we weren't wriggling, that you don't want to watch for the universe you could go to find another one. And the rules that are consistent are barely understood. Remember, people talk about [?], which is a bunch of hogwash. Any decent physicist by 1950 could understand his general theory. But a number of people that really understood physics and mathematics as recently as, in fact, hadn't even heard of it back in 1965, much less understood it--very few. And there you are.

MAPSTONE:

Okay, we are nearly out of tape, so perhaps this is a good time to stop this.

GLASER:

Well, I think as it is I've given you a highly biased, one man's view of I'm not sure what, except I've had fun. [Laughter]

MAPSTONE:

Well, I've had fun, too.

[END OF INTERVIEW]

JWM:

Yes. One of the things that I had glossed over was that in l930, I guess, l929 or l930, I had earned some money during the summer working at the Bureau of Standards, and the first year I worked at the mechanical laboratories calibrating water current years and testing fire extinguishers. We'd set a fire in an old building we had for the purpose, and then see whether this soda acid extinguisher would last for two minutes or whatever the standard time was supposed to be. If it did, it was okay, and if it didn't, it was no good, and things of that sort.

We also tested numbering machines such as the Post Office uses. Some of them will stamp the number changing each time, and some of them essentially have a binary counter in them, and stamps the same number twice and then changes, and then stamps the next number twice and then changes. They had tests of those going on for the Post Office Department.

I did mechanical miscellaneous type testing one year. The next year I worked with people in the wind tunnels. My boss was a fellow named Hugh Dryden, who just recently died. They had some of the most advanced equipment there for aerodynamics testing. They had a 3-foot wind tunnel, I think, that would get up to perhaps l00 mph. There was one of them they could run up to l80 mph, something of that sort. They had a little jet some place which worked from a compressed air tank, where just for a few seconds, you could momentarily get very small jets of air which were practically the speed of sound. I never used that, but all of these wind tunnel things were very interesting. The work we were doing mainly was with, what was called a hot wire anemometer, which attempted to measure the rapid fluctuations in air velocity close to the surface, boundary layer effect, and you get periodic oscillations in this.

We measured these with a hot wire which would be cooled more when the air velocity passing through it was high. Then you can measure the resistance of that wire which had changed with its temperature, and
you would infer then what the oscillations in air velocity were near the surface.

I forget whether I spent one summer or two summers in that wind tunnel, but at any rate I was what at that time called a junior physicist or something or that sort, and I earned like $2400 a year. When I got my PhD, I went to take the Civil Service exam again, because I didn't know what the chances might be as to what jobs I'd get, and there were no offers at all for senior physicist or whatever I was called. Supposedly, you were able to earn $3200 a year or more as a senior physicist with a PhD.

After some years at Ursinus, with the depression only easing gradually, I finally got a query from the Civil Service roles, asking if I would be interested in applying for a job if offered, all the ifs, at Ft. Monmouth or someplace like that, Signal Corps in New Jersey. It turned out that the times were such, you see, that they could get PhD applicants who were eligible as senior physicists to take jobs as junior physicists at $2800 a year or something like that. It didn't seem very profitable to change from getting $2800 a year teaching 9 months a year, to getting $2800 a year in Civil Service, where I guess you got 30 days' vacation, working 11 months a year. So I turned that one down.

UCM:

This was about the mid thirties?

JWM:

That was somewhere, I don't know exactly when, but it was well after the time I had gotten my degree, the time when I was beginning to think about leaving Ursinus because the money wasn't too good, you know.

Finally, in l940, somewhere in that era, I began to think very seriously about the possibilities of building electronic computers. I was also thinking very seriously about how to make more money because of the question you asked a little while ago about the college supporting any of this, and the demands they were making on me. As I say, the only way they supported me was to pay me salary. I had no budget for any research, and no means for tools at all. I bought this $75 calculator out of my own pocket.

UCM:

So there were two of you?

JWM:

There were two of us who were distinguished from the others by the fact that we had Ph.D.'s. I think some of the others had master's degrees in some subject, but there were two of us who were Ph.D.'s, which meant that we were further along in age and training and were accustomed to a higher starting salary maybe. In particular, I had a family. Dr. Burks, who was the other Ph.D., in mathematical logic, from the University of Michigan, was as yet unmarried, I believe, so maybe he didn't have as big a problem as I did. Wherever I would go for employment, I wanted to be sure I was making enough to take care of the family. I couldn't take any job just for the love of it. Obviously, it would be a little bit more difficult to take something in another city.

So I was very pleased to find that Dr. Chambers selected two Ph.D.'s to invite to be members of the Moore School staff, for teaching purposes primarily, on a temporary basis, that is, for the duration of the War, if nothing else, because at that time they were just then losing, I guess, two members of the staff to war efforts. One of the persons who either had left or was leaving, was Knox McIlwayne, who left to take charge of some manufacturing for the Napleton [?] Company in war work.

The other one, to my consternation, was Dr. Travis, who was that member of the faculty who was supposed to conduct the course in the design of computing devices which was offered in the catalogue. I had already learned, of course, that even though courses are in the catalogue, it was not necessarily given, and I guess I had tried to enter that and found out it wasn't offered at one time. At any rate it turned out that Dr. Travis was leaving for active duty with the Navy, and so the Moore School was temporarily short of teaching staff, and they looked to the two Ph.D.'s here as suitable candidates for replacement during the war period.

Both Burks and I accepted the appointments as instructors on the Moore School staff, with the assurance by Dr. Chambers, that even though the pay was not great, it was always possible in an engineering school to supplement this by various other contracts on jobs which needed to be done. This indeed turned out to be the case, especially with the war developing. They began getting some contracts from the military agencies, such as the Signal Corps, and secondary contracts from the Radiation Laboratory at MIT, which was working on war contracts. It looked like there would be a build up in their contract work in association with Ballistics Research Laboratory in Aberdeen. So, all in all, there would be opportunities to collect a small salary for the staff teaching work, but also some additional salary for working on contracts at the same time.

As I say, that turned out to be the case. Burks and I got employed on some contracts thereafter, so the salary that started out lol analyzer. That was MADDIDA. And those patents went over to Bendix.

MAPSTONE:

That's right, yes.

GLASER:

And they built the D11 and D12, then they built the G15, which—

MAPSTONE:

Harry Huskey's.

GLASER:

Yes, and it turns out that Harry Huskey got the original design for the G15 from a machine I think they called the ACE.

MAPSTONE:

And that also now ties in with the Datatron, because Huskey was a consultant in the Datatron, and that used the B-box.

GLASER:

Yes. But the B-box did not come from Harry Huskey.

MAPSTONE:

It came from England.

GLASER:

It came from England and Ferranti.

MAPSTONE:

Yes, the Manchester machine.

GLASER:

The Manchester machine, and actually it was Williams that designed it. Do you know why it was called the B-box?

?? entirely caught up in navigation teaching. Jacchia left for the Center of Analysis but used his spare hours during the war to do research about variable stars and to worry about