

An Interview with
DOUGLAS T. ROSS
OH 65

Conducted by William Aspray

on

21 Febraury 1984

Waltham, MA

Charles Babbage Institute
Center for the History of Information Processing
University of Minnesota, Minneapolis

Copyright, Charles Babbage Institute

Douglas T. Ross Interview
21 February 1984

Abstract

Ross, the founder of SofTech Corporation, recounts some of his early experiences working on MIT's Whirlwind computer in the 1950s. He explains how a summer job at MIT's Servomechanisms Laboratory operating a Marchant calculator led him to use the Whirlwind for greater computing power--and to seventeen years in the MIT computer labs. Ross reports on his first use of Whirlwind for airborne fire control problems. Soon after that the Whirlwind was used for the Cape Cod early warning system, a precursor to the SAGE Air Defense System. Ross describes improvements made to Whirlwind, including the introduction of the first light pen and the replacement of the paper tape reader with a photoelectric tape reader (PETR). Ross also discusses some of the programs he wrote or used on Whirlwind, such as the Initial Data Processing Program (IDPP), the Servo Lab Utility Program (SLURP), and the Mistake Diagnosis Routine (MDR). He describes the IDPP as particularly interesting, because it involved pattern

recognition and was thus an early example of artificial intelligence research.

DOUGLAS T. ROSS INTERVIEW

DATE: 21 February 1984

INTERVIEWER: William Aspray

LOCATION: Waltham, MA

ASPRAY: Could you tell about your upbringing, your parents' careers, and your early education?

ROSS: My parents -- both mother and father -- were medical doctors, medical missionaries and psychiatrists. I was born in Canton, China. I stayed there for all of six months because that was the last of their missionary tour as medical missionaries. I actually grew up in New York State.

ASPRAY: I see. You must have been born in the late 1920s. Is that right?

ROSS: December 21, 1929, the longest night of the year I would say. At first my parents were stationed at Rockland State Hospital in Nyack, New York. Then, when I was about 5 years old, we moved to Brigham Hall Hospital, a private mental hospital. We had drugs, alcohol, and mental people there in Canandaigua, New York, which is about 20-30 miles from Rochester. That's where I grew up with my two sisters--one older, one younger, both born in the U.S. The hospital had 100 acres, a big farm and all sorts of estates, with trees all over, and 120 beds. There were both men's and women's wings. We lived in the middle, in part of the building that was built in the middle 1800s.

ASPRAY: Did you go to school on the premises?

ROSS: Well, no. Canandaigua was a town of about 8000 then, and is probably still about the same today. I'm not sure. I doubt if it has grown. It had public schools and I went right through them. The high school was called Canandaigua Academy, but it is just a public high school. I was interested mostly in mathematics, science, and music.

ASPRAY: Did you play musical instruments?

ROSS: Yes, all of us were quite musical. I played the piano, clarinet, and saxophone. I composed and arranged and improvised, and got into boogie-woogie, jazz, and that sort of thing. That took up a lot of...interesting times. At one point I was in thirteen musical organizations at the same time: marching bands, symphony bands, jazz bands, various choirs and singing organizations. That sort of thing.

ASPRAY: How would you characterize the school? Was it a reasonably good place to get an education?

ROSS: For that size town I suspect so. I have very little on which to judge. I actually have very few memories of my grade school and high school education -- key teachers and so forth. Not very many stand out.

ASPRAY: Did you outstrip the offerings of the school, say in the sciences and mathematics?

ROSS: Yes, and I was sort of a hellraiser with a bunch of friends, most of whom were squeaking by with C's and D's while I was getting an A-average and should have been doing better. I did a lot of things on my own. I liked to make things with my hands. There was a lot of woodworking, model building, and model railroading. I collected stamps, and did various things in chemistry. I used to make things that exploded, and all that sort of thing. So I evidently got quite a good background in science. I always had a knack for that sort of thing. The hospital had a subscription to Scientific American and Science. So things came every week and I consumed them. I evidently did obtain a fairly decent background because I remember being off as a summer camp Junior Counselor in 1945 when the report came of the atom bomb explosion. I rushed back and explained to the cook and kitchen help all about how atomic bombs must work after having absorbed enough things about Einstein, and $E=mc^2$ and so forth from my own readings. It certainly didn't come out of high school. So I evidently did absorb quite a bit on my own.

ASPRAY: What about science fairs and projects and things like that? Were there such things when you were going to school?

ROSS: No, there really weren't. Not at all from what I recall. We were reasonably well off. My father had died when I was eleven, so all during the war my mother ran the whole hospital, including extra people because we had some Army WACs (the Women's Army Corps people) which they couldn't handle at the Veteran's Hospital in town. So that was a very busy time. Also, you see, all three children (there were more older half children who had grown up in China because my father had been married before) had quite a bit of independence because our mother was so occupied professionally. She did a good job with us too. We all ended up very independent and able to get around.

ASPRAY: Among your hobbies, did you have an interest in radio or ham?

ROSS: Yes, except I had no electronics or biology either. One of my older half-brothers was an ichthyologist. When he came back from the war, he completed his doctorate in fishes at Cornell. I used to go on field trips with him collecting algae and plants. I learned a tremendous amount from him. So even though I never had a biology course in my life, I have always had a pretty good foundation in principles from that. But the science fair type activity just wasn't organized. One high school physics course during the war had either six or eight teachers in one year. That's why I say I don't think there was too much that actually came out of the formal side in the way of science. I never took any chemistry in high school. I did it all on my own.

ASPRAY: Did you work while you were growing up?

ROSS: Yes, I used to rake leaves. My aunt and uncle had a large farm in Avon, New York with horses, cattle, sheep, chickens and the whole bit. I used to go over and help there for a few weeks at a time and got quite a bit of farm

working experience which I put to use just a little bit at the hospital. But I never had a paper route or anything like that.

ASPRAY: I understand you went off to Oberlin College. Is that right?

ROSS: Right.

ASPRAY: Did that happen immediately after high school?

ROSS: Yes.

ASPRAY: In what year did you go off to college?

ROSS: 1947.

ASPRAY: Why did you choose Oberlin?

ROSS: Actually, we knew about it because a half-brother and -sister had already gone there. Then my older sister, Peg, had started at Vassar but then decided to transfer to Oberlin the same year as I went. (My mother went to Vassar, and my younger sister, Nancy, went to Vassar, too.) I applied to Harvard, probably, and maybe to Cornell, I can't remember. My brother and brother-in-law were at Cornell, the latter in a Graduate Faculty position. In addition to liking the music -- you know Oberlin has a conservatory of music -- I also had gotten, during my later high school years, very interested in religious things. I went around helping my friends. One friend had a very bad lisping problem. I drew him out and that sort of thing. So the combination at Oberlin of music, a school of theology, plus a very high reputation, very good math/science, and liberalism attracted me. Oberlin had been one of the first schools to admit women and blacks. So it was quite a natural thing to go to Oberlin.

ASPRAY: What was your major there?

ROSS: Mathematics.

ASPRAY: What did that consist of at the time? What kinds of courses did you take?

ROSS: I remember there was a beginning mathematics class. You didn't get calculus until the sophomore year, I believe. The first year I took a class with the astronomy professor which covered trigonometry, algebra, and that sort of thing. It was a continuation of what one had in high school. I remember that we were using a new trigonometry book. The very first meeting of the class, having thumbed through this paperbound book, I found myself with my hand in the air. When the professor called on me I said, "I've just been thumbing through here and there is an error in the table of logarithms. The logarithm of 2 to the base e is .2010, not .2101 (or whatever it was)." How I even knew that I never could figure out. But he was suitably impressed with my innate abilities so I was the only freshman who had privileges to use the upper-class and graduate (they ran a Masters program when they had students for it) reading room in the mathematics library. In the main library they had a special seminar room with oak paneling and that sort of thing. I was the only freshman to have privileges there. I did a lot of reading on fourth dimension, geometry, relativity, and all sorts of things in my freshman year to keep myself entertained.

ASPRAY: What other courses were offered as part of the program?

ROSS: Well, I really don't remember. I took chemistry and French and so forth and an excellent course in English writing which I think was very informative for me. In fact, in the past couple of years my wife and I have made a modest gift to Oberlin to promote technical writing and writing for science majors because I think that stood me in very good stead. It was quite a good liberal arts background. Chemistry was one of the reasons I never made Phi

Beta. I would balance all the equations and the instructor would merely say, "that's not the way nature does it". For no reason. It was like learning a foreign language.

ASPRAY: Did you have any courses in numerical methods at all?

ROSS: No. I had just a calculus course. I guess it was in my sophomore year. I had this calculus course which was an hour a day five days a week. We never really got in all that deep -- though we got to some differential equations at the end of it. I'd been at loose ends, in a sophomore slump. In fact, at one point I remember I read an article that said you get the most productive sleep in the first hour-and-a-half. So for several weeks I had myself on a schedule of sleeping for an hour and a half and then getting up and doing stuff. It completely fouled up every schedule possible, but I was terribly keyed up and productive with everything I was doing. It really was just symptomatic of my lack of purpose. I had all sorts of things going on. The professor (Chester Yeaton) who taught the calculus class was a geometer. At the end of the term he asked if I had ever thought of doing an honors program in mathematics. This was the first time anyone had ever paid any attention to me at all in a semi- professional way. I was completely stunned because my average was down on account of chemistry and French, things I didn't do particularly naturally. Therefore I wasn't doing very well, but everything else was going fine I guess. With that grade average I hadn't had any aspirations at all; but when he suggested this and described the program which involved taking extra courses and doing extra, special work...Well, that sounded really great! So I did! By the time I was in the senior year I took every single course that they offered. Some day I should try to find out if I still have that class schedule. I was sitting in on so many courses and seminars that I had no time to do the homework. I'd take exams and was also doing special reading in geometry with this same professor, topology and tensor analysis and so forth. Things that were way over my depth. In any case I managed to absorb everything they had to offer. But in the process I didn't learn anything terribly deeply I suspect. It was just a lot of very interesting, hard working times.

ASPRAY: A wide introduction, though?

ROSS: Oh, yes. In the middle of my senior year I got married, completely against the wishes of this professor who got me into the program in the first place. In fact, he thought I would ruin my career. My wife and I had met the year earlier. She graduated the year before, went to Europe with her parents, and had come back to work in Cleveland and live at Oberlin. She got an apartment there and I got special privileges from the Dean, with my mother's approval, to live off campus. I guess I had a car, though we didn't use it much. To have one was very unusual. Instead of eating in the coeducational dormitories as was normal, I had a job washing dishes so that I could eat meals there and then eat with Pat once she got done with work. We ended up getting married in January, between semesters of my senior year. The honeymoon was spent visiting graduate schools. The first honeymoon night was spent in the Statler Hotel in Boston on the way to visit MIT. In any case, to wrap up this educational honors program topic, the same professor informed me that the honors program required an oral exam with the people one had worked with (and I had worked with everybody there). I had never had an oral exam in my life and I was all keyed up. I had been married less than four or five months and living in an apartment we found, feeding ourselves on a \$1.55 a day. We have a little book that we kept that mentions that. In all this tension and pressure, I entered the room for this oral examination and found the whole department sitting there. The chairman of the department was a very good friend, (E.P. "Fuzzy" Vance), a very nice guy. He said he thought he'd give the first question to Professor Yeaton who was this old professor who had gotten me into the program in the first place, who was so against my marriage, and this sort of thing. I was so keyed up that I burst into tears. They gave me ten or fifteen minutes to collect my thoughts and I completed the exam afterwards. But as a result of all that I got a *Cum Laude* instead of *Magna*. I wasn't close to getting a *Summa*.

ASPRAY: While you were at Oberlin did you take courses in physics?

ROSS: Yes. I took all the physics that was offered. However, the first year of physics was the only time I had laboratory. I ended up being a lab assistant for a little bit in that course. But because of all the other things I was doing I didn't take physics courses that required laboratory work. I didn't have the time for it. I had enough physics

courses to earn a major in physics except for the lab requirement. I actually had enough hours, it's just that they weren't of the right kind.

ASPRAY: Were there any engineering courses offered and did you take them?

ROSS: No on both counts. I realized after I'd been at MIT for a while that I had never even known the semantics of the word "engineering". You see, all my relatives and contacts were medical doctors or biology and chemistry professors. In fact, I'm almost the "black sheep" in the family for not being an MD or Ph.D. because everybody was doing that sort of thing. There was no contact at all with engineering. I didn't even know what the word meant. I remember I had a book that should have shown me what civil engineering was. It was all about how to make railroads and aqueducts and dams and so forth by damming streams. I actually did follow the book somewhat on a little brook we had there at the Hospital farm at one point. But even though this book, I'm sure, used the word "engineering", and was all about civil engineering, engineering as a concept didn't mean a thing to me. I knew math and science and that was it.

ASPRAY: Very strange.

ROSS: Very strange. A very myopic view of things during all that period. There are many instances where you just can't imagine that you were so naive. Yet it's true. You see things very, very specifically -- in certain ways. Even with all this science and physics and so forth that I absorbed. I did have one early experience with engineering, however. When we got the *Book of Knowledge*, I found on one page a diagram for a short-wave radio. I thought it would be neat to try to make a shortwave radio, so I arranged with all the radio repairmen in the town that whenever they were going to junk a radio, they should set it aside and I would pick it up. I got all these old radios -- really classics now -- that I stripped. I had huge transformers and loudspeakers and huge condensers -- the whole works. Boxes full of this stuff. I didn't understand it. I didn't know a thing about it. I just liked to take things apart and learn how to solder. I discovered out of my collection of parts -- with the tuning condensers (with movable plates), the knobs, and all that

stuff, that I had what seemed to be needed in this one page diagram of a shortwave receiver. All I needed was a big B battery, (I had only had A batteries up to then). I got a B battery. Those were huge things. I soldered up this thing, just taking a board and pounding nails in to hold things down, and soldering them all together. I strung up an antenna. I had to buy the antenna wire. It was the only thing I had to buy besides the battery. I already had a pair of earphones which I had earlier sent away for. In those days on the comic books, the A. W. Smith Company, or something like that, of Chicago, Illinois, had this fantastic catalogue where you could send for almost anything. Among their items you could send for a little crystal set receiver. It had a little cat's whisker. You just put a pair of earphones on it and sure enough you could get AM radio. I had done this and had gotten a cheap set of earphones with that set. Also I had gone over one weekend to visit my brother-in-law, (Jose "Pepe" Gomez-Ibanez), who was a Professor of Chemistry at Cornell at that point. To make things interesting he showed me around the chemistry lab and what he did. We took time out to make in the laboratory a crystal for a radio receiver. I think we used a crucible to fuse zinc and sulphur, making zinc-sulphide I suppose. If I knew enough electronics, I could explain it. It probably makes some sort of diode. Sure enough, just putting powdered zinc and powdered sulphur in a crucible, heating it up, and letting it fuse together we made a crystal. I went home and had myself a crystal I hadn't bought. I had made it and, sure enough, it worked! So I took this shortwave radio that I had put together, put the battery in, put the earphones on, and swept the spectrum. Sure enough, "beep-beep-beep", some sort of code and noise came out of the thing and that was all. I had a faulty component. I didn't have any idea of the proper voltage. Something had burned out. I didn't have any instruments. Except for that one time when I heard a little bit I heard nothing. I never could figure out what made it go wrong and soon got interested in other things. So that was a close as I came to engineering. I assume that I still have some of those old, big magnets out of the speakers and so forth. But I never understood engineering or knew anything about it until I got to MIT.

ASPRAY: Did you have any experience with any kind of mechanical calculating device?

ROSS: We might have had a Frieden in the physics lab, but I only took physics lab in my sophomore year. I don't recall using one there. I do remember that I used to read science fiction magazines, and probably in senior year there

was an article in *Astounding Science Fiction*, I believe, about the Selectron. I remember being fascinated by that. I understood how it worked from the article; but that was the extent of my contact with computing.

ASPRAY: Were there personal friendships that you formed at Oberlin that continued in you later career?

ROSS: Only two really, and that was because both these guys ended up coming to MIT. Skip Mattson, (H. Frazier Mattson), was in the math honors program the same year I was. He got a *Magna Cum Laude*, and was very suave. He came to MIT as a graduate student at the same time I did. Then in the Physics Dept. there was Bill Lange, who I didn't know all that well at Oberlin. But by the time we came to MIT he had also gotten married, and so we lived in the same housing compound there. Neither of those friendships lasted more than a couple of years.

ASPRAY: Let's turn to MIT and to your choice of graduate schools. What had you wanted to do when you went to graduate school and where did you consider going?

ROSS: Well, I can't remember the full set of schools where I applied to graduate school. I could look it up I suppose. I remember Chicago was one, and also Cornell. Oh maybe not Chicago, but instead Michigan. Because I know we went to Michigan on that "honeymoon". I remember having a very nice and friendly lunch with Barkley Rosser, who was then head of the department at Cornell. Also Carnegie. It wasn't Carnegie-Mellon at that point, but was Carnegie Institute of Technology. Then Michigan and MIT. I don't think I applied to Harvard. We visited these places during that honeymoon. I had never even applied for any kind of scholarship at any time because our family view was that scholarships should go to people that really needed them, and so we shouldn't compete. The family gave no credence to the supposed "honors" that went with achieving a scholarship.

TAPE 1/SIDE 2

ROSS: Being married, with a working wife who had a \$40/week job when we lived at Oberlin (although she spent \$5/week on commuting costs) is how we managed to pay all the rent and feed ourselves for \$1.55, and were saving money. I wanted to be self-supporting. My mother was planning to pay tuition. That was fine. I guess she also gave some room and board, but I wanted to have some scholarship, or preferable a teaching assistantship. Out of the applications, Carnegie-Mellon offered me a full teaching assistantship, a good stipend, and all sorts of things. MIT was able to come up with only a scholarship because Skip Mattson was already coming from Oberlin with a teaching assistantship and his record looked considerably better than mine. But there *was* the offer for some scholarship help. I thought it over and even though I didn't know a great deal about either of these schools --in terms of what I now know -- I decided that MIT was MIT and I'd take my chances there. Besides that we had enjoyed visiting there. We met with, I believe, both Hildebrand and George Thomas, who had just written his book (on calculus). They were still using the first preliminary printing of it at that point. So we decided to go to MIT and as soon as I got there it turned out that Prescott D. Crout, of Crout's Method for solving matrix equations (a lot of applied mathematicians know it), was scheduled to teach one of the twenty seven sections of freshman calculus, because they always cycled through with their big names doing that type of assignment. But something came up and he couldn't do it. So, lo and behold, here was a teaching assistantship for me as soon as I got there. Sure enough, I had two sections of freshman calculus out of the twenty-seven. While leaving Oberlin I had somehow run across Polya's book, "How To Solve It". Inside the covers, both front and back, they had duplicated a checklist of "how to solve" problems -- just titles and methods. I thought that was so neat that I approached the department and actually convinced them to allow me to get permission from the publisher to duplicate that spread and distribute it to all the freshman calculus classes. Sure enough, we did that. I had done some tutoring type of teaching at Oberlin, spoke at the Math Club once in a while, had the gift of gab a bit, so the teaching aspect didn't phase me. I was also very interested in "Gee, what can I cram into the heads of these youngsters?" You get all of these young (in the first place they were at most three or four years younger than I was, but it was pretty striking to me), bright minds. So I remember from the schedule of teaching calculus, the first class was sort of light because they were just getting their books and so forth. So I snuck in as an extra assignment (because there was no regular assignment) to my classes the following problem: "Suppose I have a Cartesian coordinate system and I want to describe things in it. What happens if the x and the y axes have

different calibration?" I just wanted to see what the response was. Well, of course that's now well understood (affine transformations, etc.) I don't know why I picked that one, I just thought it was sort of interesting. I wanted to see how their minds worked and how they would get a hand in on it. I don't remember at all what came out of that; but I remember learning many, many years later that this problem, which in this case is one of the simplest formulations of it, is right at the very core of modern physics, relatively, gauge theories and whatever. So I was pleased that I stumbled on that one as a way to test my students. Later on all the grades from these twenty-seven sections were pooled and I got called on the carpet. I think George Thomas was the one who was in charge of the course. We had uniform regular quizzes and they looked at my grade sheets and my kids were getting too many high grades. That wasn't right. Things had to match the curve. What was going on? Was I being too soft, easy? I said, "Well, I don't think so. Let me show you what I do." So I showed them what I did every time we came out with one of these quizzes. I would solve the problems and would break each problem into sections of what were the important ideas and what were the right answers. I would break down a fifteen point question into two points here, three points there, one point there, and so forth. I had it all laid out that way. I showed them how I kept separate grade sheets with the student's name and all the sections that I had broken the problems into, how I graded each one individually and lumped them all together, and then only wrote the final grade on the blue book that the student handed in. Several times I had students complain that I was grading them too harshly, too low. So I would just show them my solution to the quiz and how many points went for each thing. I would give them their quiz book beside it and I'd say "You go grade yourself. I haven't made a single mark on your book except check for right and x for wrong and then the final number. So you don't know what my view of it really was. You go and grade it yourself according to my breakdown, add up the points, and come back with your number." Then they'd say, "Well, that's very rational" and go off and do that, thinking that they would be very nice to themselves on the pieces that I wasn't. Every single time they came up with a lower grade than the one I had given them. The students themselves were grading their own stuff. By the time I showed the professors all of these charts and excesses of analysis and processing that ended up with those grades, and they went around and checked with a few of my students, they said "The grades aren't out of line. You do enjoy teaching. You do rather well with it. How would you like to be a Freshman Advisor?" So in the second semester of my first graduate year (Spring 1952) I was the only freshman advisor of that low a rank. I guess

there was one advisor who was a lecturer, but everyone else was an assistant professor or above. So here I was as a Freshman Advisor responsible for fifty-four students, or some number like that. I really enjoyed teaching and had a great time with it. But in the summer of 1952 I needed a summer job. I like to say I've never gone looking for a job properly. My wife, Pat, had been working at Lincoln Laboratory. She was the first "computer" at Lincoln Laboratory before they officially took over the Whirlwind computer and moved all the people around in Barta Building. She punched a Marchant calculator. They were working out of Building 20 which is still there at MIT (an old wooden building left over from Rad Lab). One of the jobs she had, since they weren't set up yet was to work with a mechanical correlation computer in the Servomechanisms Lab, Building 32, next door. It was an analog correlation computer, built by Norbert Wiener, with two ball and disk integrators in it and little handles with which you traced strip chart curves. Tracing of radar noise data is what they were analyzing. Two girls would sit there, each one visually tracing this curve at a fixed separation from the other, as it was pulled through, and the ball-and-disk integrators would compute the cross-correlation function or the auto-correlation function, one point at a time, depending upon how far apart the two pointers were that they were using. Pat was describing this thing to me and I said "Well, that sounds neat. Can you bring me home something that describes it?" So she brought back a seven page purple ditto description of the principles of the machine -- how to run it. Came summer I called them up. I have no idea where I got the gall to propose to doing this. But I just looked in the phone book, and called up the Executive Officer of the Servomechanisms Laboratory (Al Sise) and said, "I'm a math graduate student and would like a summer job. If you could find an electrical engineering student, I'm sure by the end of the summer we could make you an electronic calculator that would beat the pants off that little mechanical thing that Wiener has put together. Are you interested?" "Well, I don't know. Let me take your number and I'll call you back." In a few days he called back and he said they couldn't swing that, but would I like to come and push a Marchant calculator for the summer just like my wife did? I said, "Well, sure, a job's a job." So that's how I got into the whole sweep from then on. Because one of the first jobs that we had...No, there were two things. I was associated with a project that was evaluating airborne fire control systems. You know where you put on a radar and track a target plane and try to servo a gun to it and shoot it down.

ASPRAY: What year was this?

ROSS: This would be the summer of 1952. I ended up working for John Ward, who still is a very good friend and still is at the Lab at MIT. He was the Project Engineer who I went to work for. I forget exactly what the first jobs were that I had, but I had to calculate and plot graphs and so forth. Here I was banging away on the Marchant and then they came up with a...No, I've got that mixed up. You see, what I want to get to is how the first programming language...No, that's right, it must have been earlier because the first programming language I ever designed was a programming language in which the computer was a collection of people. That was before I actually encountered my first real computer, Whirlwind. I may be wrong on that, now that I think about it, that's the way I told it for years. Because one of the tasks -- now that I'm thinking about it it must have been the second summer, I'm not really sure. No, it must have been the end of that same summer. Anyhow, here was the problem. We had this very complicated set of equations that described this fire control system and it had to figure out gun aiming errors and that sort of thing, much more than I could handle, myself. So we hired a bunch of part-time students. They were supposed to help carry out these calculations. The way that you set up calculations for a mechanical calculator (in those days) was that you make a column of x's and a column of x squared and a column of difference of x squared and take the square root of that. It was all organized by columns. Well, if I had used that normal way of laying things out across the sheets of paper that were supplied to us -- they were very, very complicated set of equations to go through -- we would have had columns going out the door. So instead, I said, "I'll make a form so that each of these people can just do what I say on the form." On the Marchant calculator you had little windows where you could read the accumulator and what the numbers were, see the numbers punched in, and read these various things. I gave those various things names and places and I wrote out on purple ditto ("Classified Confidential" because the formulas were classified) about a six or seven page set of steps to go through saying: punch this number in here, multiply by that number, copy it down here, and fill in from line one to line seven -- like your income tax form I suppose. It was in fact a complete little programming language so I could set up any calculation I needed to have other people do.

ASPRAY: This was actually put into use?

ROSS: Oh, yes. We had six or eight people there grinding away at these things. It was interesting. The end numbers were the elevation and deflection of the gun-aiming error, or whatever. Well, back to that mechanical correlator. A problem had come when, just for the heck of it, they had set the two pointers as close together as they could, about a millimeter apart, and had run that. They got this very strange number out of the mechanical correlation computer. They tried it again, and again, and again. As long as they had very short spacings they were getting these points way off the curve -- surprising the dickens out of them. And so I looked into it. The previous year, while I had been teaching freshmen and so forth, I was also taking complex variables and that sort of thing. Several times Pat had come back bubbling over that she had gotten some really tough equations to solve; that this would be a neat thing, that it would take her at least two months to get through it; and that she enjoyed the person she was working with. Describing it to me, I said "Gee, that sounds interesting; show me the problem." And two or three times it turned out to almost exactly match what we had just gotten through covering in Theory of Functions of a Complex Variable.

ASPRAY: Solving real integrals using complex techniques?

ROSS: Yes, and so I would send her back with a closed-form solution. It happened two or three times you see. So by the time I got to the Lab I had this reputation of now and then coming up with things that were useful to these engineers. So when they had this problem with this correlation function anomaly, where there is a point way off in the boondocks with a small time separation in the computer correlation, they asked me to look at it. Well, I didn't know anything about correlations. I never heard of it except for reading the one little manual for the correlator. Remember, I had offered to make them a better correlator. Meanwhile, I discovered part of the reason why that wasn't necessary. There were already about six or seven electronic correlators in one form or another around in different labs. So why didn't I check the anomalous result with one of those? We could transform the signal into whatever the paper traces one of these would need and then check it out and see if something was wrong. It turned out that nobody there in the lab, you see, really understood beans about power density or correlation functions. They were just doing it.

ASPRAY: You said there were half a dozen other electronic correlators on campus. Who had them? What were they used for?

ROSS: That I couldn't really tell you now. But there was one (I'm sure there must have been something like that) over in Dynamics Analysis and Control Lab (DACL), which also was one of the projects that spun off from the Servomechanisms Laboratory, which is where I was doing all this. I really don't remember where the others were because where I was they were all either broken or busy. Somebody said, "Why don't you go over and use Whirlwind? Try Whirlwind." And I said, "Whirlwind, what's that?" "Whirlwind is a computer." "Oh, it is. I've never heard of it." I remembered a little about the RCA Selectron memory tube, but I didn't know anything about Whirlwind. Well, it turns out that the Whirlwind computer and Digital Computer project had spun off, like the Dynamic Analysis and Control Lab, which covered the analog computing area. Both had spun off from the Servo Lab, which is where I was. Servo Lab was the lab that was set up during the war by Gordon Brown. It was closely associated with the Radiation Lab, but I think it may have had its own characteristics fairly early in the game. Anyway, after the war they formed the Research Lab for Electronics (RLE), which carried on in the Radiation Lab, proper. Servomechanisms Laboratory, with Gordon still running it, was this separate entity of the Engineering Department, itself. I believe RLE was split between EE and Physics and maybe some math was in there. I'm not sure. I've always been pleased that the Servomechanisms Laboratory (even though the name was hard to say) spawned so many of the critical things in our field. DAAC for analog computing; the Whirlwind Digital Computer Lab for digital computing, and also the thing I got tied in with later -- numerical control -- all grew out of the same place. Even though Servomechanisms, proper, got so well understood after a few years that the things that (when I was there) we were still giving masters and doctorate degrees for now the average mechanical engineer has to know in early undergraduate days or he isn't considered a mechanical engineer. In the old days this hotbed of activity, doing things, making things work, spun off these different labs -- groups of people -- that have gone on and been quite influential in various areas. Well, at the time the people working on the fire-control systems didn't really understand correlation much at all. So they suggested I go see about Whirlwind. I found where the Barta building was and I went over. Somebody had mentioned that Jack Arnow had written a correlation program for Jules Charney

of the Meteorology Department (I think). And so I found out where the Barta Building was and looked up Mr. Arnow, whom I had never met. Sure enough, he was very nice. He sat down with me and showed me the correlation program which he had written and described to me how the instruction codes worked. All this was brand new to me. All in just a couple of pages or so. It was not very elaborate programming but, I was hooked. That was really neat stuff. This must have been the end of July or August, summer of 1952. He told me I could sign up for some time and get a programming manual. It wasn't a manual. Just what there was about programming. On the way out I stopped by and got one from, I think it was, Donna Neeb. I don't know how I remember that name. She was nice, one of the programmers there, a staff member and really sharp. So I took the stuff home, taught myself to program, and started out by writing a correlation computer program to compare with this other one. Well, it ended up getting to be quite elaborate. I decided I should do it with Simpson's Rule; thinking that it would make a big difference. It turns out that that it didn't make a bit of difference. But I went to all the trouble of working out the logic for computing a correlation function using it. You see Whirlwind at that time had exactly one "k" (1024 words) of sixteen bit memory. By today's standards that doesn't seem like much. I can't remember whether it had a working drum. It may have had a working drum. The tape units were barely functional. In fact it wasn't until one of our skilled instrument makers in the Servo Lab got a hand on working with them that the Raytheon tape units actually got to be useful. But it was mostly just paper tape that you punched on the Flexowriter to run the program. So I had a computer with this very limited memory and I wanted to be able to process long ranges of data: functions of time. I decided on a program that would compute one-hundred-and-one values of the correlation function, each one with a separation of one more delta (delta equals zero to one-hundred) with one pass of the arbitrarily long data going through. That meant that I had to lay-out my memory so everything was shifted. Then applying the alternating coefficients of Simpson's Rule was pretty tricky. But sure enough it was a program that you could feed an arbitrary length of data in and it would always have in memory a hundred values of the correlation function, but handle any amount of more data. I wasn't completed debugging it when they started to tear down Whirlwind for changing over to the new display system for running the Cape Cod System (precursor to the SAGE Air Defense System). They completely changed the order code that had to do with input/output, introduced the "SI" (Select Input function), and restructured all the input/output system of Whirlwind. That shut the whole thing down for weeks right at the end of the summer when I needed to do

this debugging. So I had a lot of time on my hands and I proceeded to write a program to compute the Fourier Transform of the auto-correlation function in order to see the power density spectrum. At the end of the summer I had these two partly developed programs. John Ward said to me, "Did I really want to go back to teaching freshmen? Wouldn't I like to stay around to finish this up? Wasn't it sort of fun, anyhow?" I decided yes, it was! So that Fall of 1952 I taught my two sections of calculus (having given them notice that that would be the end of it), took a full graduate load, and also worked full-time.

ASPRAY: You were busy.

ROSS: Having fun, yes. So anyhow...I can't quite pick up exactly where I was with this saga.

ASPRAY: It was your first term that you were working in the laboratory, still teaching.

ROSS: Yes, right. I've gotten a copy finally of the book that was written on the history of Whirlwind, but I haven't read it yet to see whether I think they got the story right. A lot of people are not familiar with the fact that when Lincoln Lab grew out of the Project Whirlwind (which were just names to me at that point) for the purpose of looking into radar air defense -- processing radar signals with the computer and then using radio controls to defend with fighters and so forth -- Whirlwind had been originally designed as an aircraft simulator for individual airplanes. (Of course, nowadays, has not only that function, but also all the satellite tracking and the FAA making the airlines safe -- all of which came out of Whirlwind.) They always had the scope digital display output associated with Whirlwind. But in the summer of 1952 it was still, I believe, just a single scope. For this application of tracking multiple aircraft they needed to have many different displays being driven by this thing. Whirlwind only had thirty-two instructions. They managed to keep all the useful instructions the same while changing just a few. I can't remember how many there were before, but what they did was change the computer so that it had this select input (SI) instruction which had a parameter that would select one of several display lines or one of several sets of buttons for input, and that sort of thing. From that they were able to make these display station consoles for the air defense problem. All I knew

at that point was that they were shutting the thing down to change the instruction set and I had to change a little bit of my programs to correspond. That was not too bad. But again, what that did was give me time to press on and do the Fourier Transform program. By then I had gotten far enough into what they actually used this correlation information for in the fire control system evaluation to determine that they were also interested in power density.

TAPE 2/SIDE 1

ROSS: Whirlwind shut down to make this change and as a result I had not only the correlation function that would handle arbitrary lengths of data, but I had this companion program that would compute Fourier sine and cosine transforms (though I only actually used the cosine transform because of working with the power density spectrum. The correlation function is an even function so you don't need the sine.) This pair of programs was what I was going to use to check out that anomalous point that they had gotten, because I could compute the whole correlation function at the small deltas that they could only get a few of by the hand method. So by the time Whirlwind came back up in the fall I got back to debugging and I couldn't find out what was wrong in my correlation program. The transform program was working fairly soon I think. But the correlation function program with all the intricacy of Simpson's Rule completely baffled me. I decided I would have to make what I called a "mistake diagnosis routine", a program to analyze my program and find out what was wrong with it by putting in break points and printing intermediate out things. I remember that the summer of 1952 was when we had a going-away party for Jack Gilmore, later Key Data president, at the computer lab as he left to go do a stint in the Navy. This was right after I had first gotten involved with Whirlwind. What he left behind as he went off to the Navy was the first, what would nowadays be called an Absolute Assembler, an assembly program which made it much easier for all of us. Instead of writing binary code or 5-5-6 (is what it was called, sixteen bits were put down on tape in 5, 5, and 6 groups), in Jack's program you could put "CA" for clear and add, "SP" for subprogram which was the jump instruction, and "CP" for conditional subprogram. You still had to put down the absolute octal address, but it was a real boon to mankind to have Jack's program working. So that's actually what I had been writing most of my stuff in. But that was the extent of our tool support really, that plus the Octal-Dump. You could get a memory dump in octal form because they had a

character generator that made a figure 8 -- you know like you have nowadays on the little hand calculators. They had that with a vector sweep on the display scope. They had written this scope dump program to where you could dump memory on the upper oscilloscope. They had two there. They displayed them together, one that you could look at and one that had a 35mm camera mounted on it. Everyday, maybe twice a day, but certainly everyday they would develop this role of 35mm film and make great big 8 X 10 prints. The cost must have been outlandish. But that was the way that we did things. Lots of things were done with movie film in those days including analog instrumentation for those fire-control systems we were working with. They would make a box that had all the aircraft instruments, dials and so forth, all crammed into a dark box and take a picture with a camera. Then they would have these gals sit there and manually read all those dials and write down what they said. That was how we got hours and hours of very intricate data to process. They also took movie films of the fighter aircraft with a camera mounted on the gun turret. The girls would put a crosshair cursor on each frame and measure each coordinate by hand. They would be punched in by hand. Really something. In any case, we had Jack's input program. We had the dump program. You could also do the "post-mortem dumps", as they called them. Do the post-mortems on the Flexowriter at ten characters a second output. That was why I was having a little trouble debugging my program. I decided that I needed to put some break points in, but I decided not to do it straight. I would do it in a systematic way. Later on it ended up being quite elaborate, the MDR (Mistake Diagnosis Routine) system that would take any program. Remember, Whirlwind only had one "k" of memory at this point, so if you had an elaborate program, all of the memory was very, very valuable. So sprinkling through lots of break points and print instructions which we had in -- everybody called on a print sub-routine -- really chewed up space. Since you already were using probably all the memory for both program and data and you really didn't have any slack at any time, squeezing in a program to help with debugging was not a straightforward task. Not only did you have to recalculate all the addresses whenever you moved things around and put them in by hand again with all the attendant errors, you just didn't have room. This is right in that fall of 1952 or maybe sprinkling all the way to 1953, because I'm not sure when they put the drum in. They may have had a drum that was able to store six k by then. The concept of the Mistake Diagnosis Routine was to copy out a portion of the program and replace it by the MDR with some working space, put in some patches, copying instructions from each place to the working area, and chain everything back together logically as though it were undisturbed. Well, the

precursor to the full MDR didn't have all that elaboration, but was on the same principle. It took me a long time, of course, to debug this before I could get to checking out my correlation program. I can't remember if it was in the preliminary version, first version, or the final MDR; but the very first time the MDR finally worked (got its bugs out), I discovered the problem with my correlation program. The problem was precisely one bit. In other words the program consisted of two main parts -- basically input calculation and output, I guess -- and there was one instruction to link these two phases of the program together. In that one instruction I discovered there was one bit wrong. So merely complementing that one bit made my program work like it was supposed to! It had taken me three or four months to get there, and I don't know of any more classic example of the brittleness of digital programs. In fact I made that point a few years ago when I was doing some consulting in Philips with the high-level managers. They didn't understand anything at all about computers. I said, "You hear about software, especially coming from an engineering place, that you can always fix it up by changing the software. That's a lot easier than the hardware. Well, actually, there's nothing more hard than software -- and I don't mean hard to write. It's actually brittle." I gave them this example of one bit making the whole thing shatter and be no good at all. There is nothing else that is that brittle. I think of it like a crystal structure splitting. That was my classic example. Anyhow, that was the start of my computer exposure: this pair of programs plus the Mistake Diagnosis Program.

ASPRAY: You continued to work in the laboratory from that time on?

ROSS: Right. Never went anywhere else until I founded SofTech, seventeen years later.

ASPRAY: Why don't we go back then and talk about your education? When you went to MIT, was this to go into a Master's or PhD program in mathematics?

ROSS: A Ph.D. program in pure mathematics.

ASPRAY: What sorts of courses did you take?

ROSS: Well, you still had requirements so you had to take theory of functions of a complex variable. I did take some algebra -- basic algebra: isomorphisms, homeomorphisms, and that sort of thing. I remember I took a topology course with Warren Ambrose, getting off into fiber bundles and manifolds and this that and the other thing. And the theory of real variables, measure theory. I had a fascinating course with a then young CLEM Instructor, at MIT in the Math Dept. They named these instructorships after C. L. E. Moore. Sigurdur Helgason, who is now back at MIT as a Professor, was then a CLEM Instructor, and I had a fascinating course from him in measure theory or real variables -- I forget exactly. I have the notes still. What was neat about it was we ended up with a series of theorems -- theories -- essentially about systems with continuous flow. In other words, things could swoosh around all over the place, but they never could form a bubble or a break or anything like that -- just continuous. And very deep theorems and proofs. At the end of the semester he ended up with one or two lecture times free. He'd gone through everything that he really had to cover. So, just for the heck of it, he threw in a couple of closing lectures in which we derived Quantum Theory from the theory we had just established: Heisenberg's Uncertainty Principle, the Wave Equation, and everything else. They were just an application of the theories we were proving. That was really a blast. So that was the sort of course that I was taking. Then also, to establish a minor, I decided to do take courses in Electrical Engineering because I had never had any of that sort of thing. I took switching circuits with Sam Caldwell and Dave Huffman and something on Control Theory with Yuk Wing Lee. Y. W. Lee was professor there. I don't know if he's still around or not. He must have retired by now. But he used to be very close with Norbert Wiener. Wiener spent a lot of time in China. I did my Master's Thesis under Lee. Because I was working all this time, and by this time had several people working for me, I couldn't take time off to do much with these courses. In fact, I was continually missing classes because I had to go off on various trips and so forth. In particular, in order to get a degree you had to be a full-time student while you do your thesis. So what I did was I'd gotten interested in the Fourier Transform aspect of my programs. I'd run across a paper (or someone had referred me to it) of some work that John Tukey of Princeton had done on minimizing Gibbs' Phenomenon. Gibbs' Phenomenon is the $\sin x/x$ ringing, jangling around. It was very familiar to the Servo engineers. They knew when you tried to have a servo follow a step function, if you didn't have a feedback control loop, it went up and jangled around like crazy before it damped out.

Well, I remember being baffled by the first description that I heard of it. I didn't understand it at all...Tukey had multiplied the $f(t)$ (the time function that you want to transform) by the "arch of a cosine". "Arch of a cosine" somehow that just didn't set in my mind at all. What it means is half a cycle of a cosine function -- smoothly up and then down. Anyhow, what this did was greatly minimize these various fluctuations. There was all sorts of confusion at the time because with the Gibbs' Phenomenon in there, when you just plain computed the transform of an autocorrelation function, it would come out with negative power density in places. "Oh, we can't have that! Terrible!" So Tukey showed that by pre-multiplying by the arch of a cosine, (strange thing) that you couldn't get negative power density spectrum estimates. An autocorrelation function is an even function. If you do a Fourier analysis of it to obtain the power density spectrum, all the sines and cosines turn out to be just cosines because the frequency components all have zero phase. So you don't need to worry about the sine transform for power density. You only have a limited length of autocorrelation function $f(t)$ up to some maximum. The full $f(t)$ goes out to infinity, so this truncated function is as though you have pre-multiplied it by a rectangle that is unity up to T and zero after T . In Fourier analysis, if you multiply two functions together and do a Fourier transform, you get the same result when you independently transform the two functions and convolute them in the frequency domain. So multiplication in the time domain corresponds to convolution in the frequency domain. Tukey saw that Gibbs' Phenomenon was due to convolving with $\sin x/x$, the Fourier transform of the step function. You're computing the weighted average of the product of these two functions at that individual point. If the true spectrum has a pair of peaks, two bumps that are fairly close together, when you center this convolving "window" function over the left-hand bump, then maybe one of the bumps in the $\sin x/x$ window function will pick-up and multiply times the big value of the second bump; whereas, if you were off a little bit, maybe there's a negative number bump like that times that other peak. In any case the result was the bumpiness of the power density spectrum estimate. It has all these extra wiggles because you've been picking up things that are far away from the center frequency when you do the sliding window. You can't get rid of the window unless you go to infinity, and in the time dimension we have to truncate. To improve matters you want to reshape that spectral window so it has a sharp peak around the middle and goes down to zero rather quickly, so you don't pick up these far-away frequencies. Then you can discriminate frequencies closer together. Tukey chose the cosine function. By the time I got through understanding all this stuff, I asked why worry

about all this complex stuff? I'd like to be able to squeeze my spectral window just as high and narrow as I could. I shouldn't describe the whole thesis, but by taking an engineering approach I was able to compute better and better averages which corresponded to doing the same kind of window function processing. But I had a whole family of them for $N=0,1,2,3$, on up. You could squeeze the tails on your spectral window arbitrarily so that they went down as t^{-N} instead of $t^{-1.4}$ like $\sin t/t$. So that was my thesis. I had it all worked out and the programs and so forth before I approached Professor Lee saying I'd like to do a thesis. That way I was sure that during the thesis term I could make the final runs of my programs, write it up, and be sure I would be done with the thesis. I had an error analysis that I did, too, which I had naively based upon absolute values. Prof. Lee's one suggestion to me on the thesis was that it would be better if I used a square function because then I could do an analytic treatment. (Absolute value isn't an analytic function.) So, I sort of shuffled around and said, "Well, I've already got it done this way." But he probably was right. I probably should have rethought that part. I also had a closing section in which I continued with my engineering way of looking at things and claimed that I could beat Nyquist's Sampling Theorem limitation, where, theoretically, you're supposed only to be able to obtain frequencies separated by half the period -- two points per cycle. By using these same techniques and playing with the window function I demonstrated crudely --and have never known anyone else to do it since -- that by making a specially formed spectral window for a given power density estimate you could actually discriminate finer than the Sampling Theorem would allow. I believe it's somewhat related to what they do with various picture enhancement methods nowadays. I'm not sure. I never studied in this field since I did the thesis in 1954.

ASPRAY: Where did you get your training in mathematics, Fourier transforms and such?

ROSS: Oh, at MIT. It was used in a course that I took with Professor Lee. I'd just gotten exposed to it from the fire-control project and it's not very complicated. At the time, because I knew I didn't know much about it, (again, like the Poyla book thing) I organized a seminar series (I don't remember exactly what it was called) open to all at MIT in which the problem was "integrals of products of two functions", which includes the correlation functions, the various Fourier-like transforms, and anything that is an averaging type of thing. They're very heavily used in all sorts

of engineering areas. In this seminar series I got various people who I had run across to come and give talks. Norbert Wiener gave one or two; I gave one or two; and I can't remember who else. I hope I have someplace in my records the notices for that series because that was sort of neat. There were various people at Lincoln Laboratory and RLE that worked in these same areas; so it went on for a semester or so.

ASPRAY: These are the sorts of things that are sometimes taught today in applied mathematics courses. Did you study that, too?

ROSS: No, the only applied mathematics course I ever took was one in probability, I think. I just took it as part of the master's program. You see the master's program was supposed to be minor for my Ph.D. in pure math. But I also found that it would be to my financial benefit to have a master's degree, moving up a notch in the pay scale. I didn't want to be, and I wasn't, an electrical engineer. I had no idea about anything in electrical engineering except what I found there in lab. So, I applied for the master's "without course specification", which is what I actually got -- which you can do at MIT if you're spread around. I was then supposed to go on and complete the doctorate. I did do all the course work for the doctorate and more or less on schedule -- doing well with the courses, the grades, and so forth even though I traveled a lot. By 1956, when I finished up my scheduled course -- taking I was just starting with the APT Project and already had 8-10 programmers in 2 or 3 places in the country working on more of the fire-control data reduction stuff. Later I ended up with 55 programmers at 19 company locations at one point in the APT project, I think it was. I couldn't possibly crank myself back to courses I hadn't studied since 1951 in pure math and pass all those very tough exams. Furthermore, I figured I had done the equivalent of about three Ph.D. theses by then, including my master's thesis. I still say the master's thesis is as strong as many doctorates that have come out. It would be good if it were beefed up a little. But I'm an amateur scholar. That's the trouble. I never really did what a doctoral program is supposed to do -- competing with scholarship.

ASPRAY: Training to become a professional scholar?

ROSS: Yes, I used to say I was a pure mathematician by training but an engineer by heart because I discovered what engineering was: get in there and make it work and understand as much as you can; do things systematically, but make it work. Another one of my phrases for years has been "Don't foist your viewpoint on the problem." That's how I got the Master's project going, for example. Just by working from the bottom up. Analyzing what was happening in the computation and smoothing things out. Then when I got to looking at what I did, sure enough it was marvelous, very neat, elegant, much easier to compute than cosine arches. I had complete control over the thing.

ASPRAY: Along the lines of your master's thesis for just another minute. At this time Tukey was splitting his time between Princeton and Bell Labs. Do you know what kind of problem he was working on that interested him in this technique?

ROSS: No, I really don't. I never even met the man. I always wanted to. When I submitted the thesis to IEEE (it was then IRE then, I suppose) for publication in the transactions, they took a long, long time to respond and then wanted to have the whole thing completely restructured in ways that seemed to me would decimate the important ideas. And it was a lot of work; so I never completed that. It never got published, even a summary of it. No, I did do a summary of it in 1956 --summer of 1955 was the first year, I guess. Al Susskind organized a Summer Session course at MIT spinning out from our work in the fire-control project on Analog/Digital Conversion Techniques. He asked me to do a section (he had various lecturers from the Lab) on Sampling and Quantizing Theory. So I did do a summary of my thesis in chapter two of the book that we published through MIT Press. It still is around. But that was the only publication of the thesis ideas. When he asked me to do this, I didn't know anything about sampling other than this stuff that I had done. It turned out at the time that Bernie Widrow had just finished his doctorate thesis under Bill Linville. Bernie's thesis had come up with what he called the Quantizing Theorem, which is the same as the Nyquist Sampling Theorem, but applied to amplitude.

TAPE 2/SIDE 2

ROSS: His thesis included, among other things, this Quantizing Theorem. It turned out that back in the 1860s or so, there was a statistician named Shepherd who had worked out what are called "Shepherd's Corrections" for getting correct moments of a statistical distribution from sample data. For example, I remember that the exact variance for a continuous function can be computed from its samples except for a factor of $Q^2/12$, where Q is the quantization width. Well, Bernie had run across this and he was interested in working with sampled and quantized data, breaking the Y axis of a function up into discrete values. You take the continuous Y axis and put a bunch of tick marks on it like a ruler. Then any value that falls in one of those regions, between tick marks, is attributed to be exactly the value at the middle. That quantizes the signal. That's the same sort of thing as taking a continuous time function and sampling it like the Nyquist criterion is for discrete time series. If you're only interested in the statistical properties of the function $f(t)$ as you quantize it, make it discrete both in time and amplitude, if you're only interested in statistical values, which is what correlation functions do (they average), then it turns out you can work with the distribution function of the amplitude. The way I showed this in Chapter Two of the Analog Digital Conversion Techniques book for the summer course, I said, "Imagine you have a thing like an abacus, a whole bunch of horizontally strung wires with beads on them, and you position the beads on these horizontal wires so that it has the shape of the function $y(t)$. T goes to the right. You have these horizontal value lines of wire going to the right and you just make the shape of the function. Now take the frame that has those wires and the beads strung and tip it up vertically. All the beads then slide by gravity to the end. In other words, you slide all the function values back and jam them up like a histogram on the Y axis. Now, let's do the tip so that the histogram is vertical. The piles are the way we like to look at them. Imagine that to be a function of time and you've sampled it in time, just as we were used to in the Nyquist stuff." So that's the connection that Bernie found. Quantizing and computing a statistical moment (some kind of average) from the distribution of amplitudes can be handled like sampling a time function. All the information needed is in that histogram we've just made, which is the quantized distribution function, for the values of the Y range of the function. So that's what I combined with my own stuff about the Sampling Theorem, weighting functions, and minimizing Gibb's Phenomenom -- getting better spectral estimates --and a little bit of reading about the Z Transform that Zadeh had just come up with. That was the treatment of sampling and quantizing in this course. It's all spelled

out in the book. That's the one place where the thesis work was published. That training almost completed that kind of mathematics that I was doing. Actually, earlier -- about that same time, 1955-56 sometime -- there was a member from Westinghouse (I believe it was) visiting Lincoln Laboratory, studying magnetic amplifiers and magnetic materials, and so forth. Bob Ramey, was his name. Prof. Frank Reintjes, who was then the head of the Servo Lab had some contact with Ramey. Ramey had a problem. They had been working with donuts of magnetic material, putting various pulses of current through them, and measuring things with oscilloscopes. If you take a sample of magnetic material, apply a steady current through a winding, and measure the voltage change on the oscilloscope as a function of time, you get a bell-shaped curve. It starts from zero, goes through a little hook, goes up to a maximum, then tapers down to zero. What you're measuring there is the change of the direction of the magnetization of the little elements -- tiny magnets that make up a big chunk of material. You're successfully overcoming the local tendency of them and making them line up with the field you're imposing. Every time some of them change, that makes another bunch easier, easier and easier until you get them all switched. So the area under this curve measures the total flux change. Well, they put in a square wave function of current, driving a piece of material. In other words, zero current, then step up to some fixed current, then step down to zero, then up, then down, on and on measuring the voltage. What they found were little spikes of voltage over time at different heights. When they looked closely at them, the top shapes of the spikes had the same shape as the corresponding portion of this curve that they got when they just drove the thing in one sweep to saturation. In fact what you've done is taken this curve and chopped it up into pieces and stretched them out, so that if you take a continuous shape like a bell-distribution curve that starts from zero and goes up to something and comes back to zero in a bell shape, think of chopping it into a bunch of vertical strips, and then put half an inch of space between each strip, that's just what they saw.

ASPRAY: With things going up and down?

ROSS: Yes, there was zero voltage change when the current was zero. But if you took away the spaces between each piece and jammed them together (just like we did with the histogram), you would get back exactly the curve of the continuous case. They noted this and didn't understand it. So, Frank asked me to meet with Ramey and see if I could

give him some ideas about how to go about analyzing this phenomenon. Sure enough, I went on and did a whole theory of magnetism based on this set of experiments. If you do it for different levels of current, the curve keeps the same area under it. If you drive it with a large current, it switches fast; then if you do it with a low current, it switches slow. Anyhow, you'd have a bunch of things you could measure in the laboratory and then you plugged them into my formula and you could determine both the voltage and the current just as if you had a non-linear resistor. I guess nobody had ever had that sort of thing before. Later on it turned out you could take one of my functions and factor it into two separate functions, if you approached the same subject matter going all the way out through domain propagation with quantum theory and everything else. I never published my theory, but I was happy to have it vindicated in that way. Anyhow, it was an engineering kind of solution again you see. That combined with the step that I had done with the Fourier transforms, my thesis, and with the work for the summer course (the same thing in quantizing) got me off into a spurt. That reminds me of the one set of notes I think I've lost, that was the finish to that line of work. I was working out a theory of non-linear systems that combined all these approaches together using impulse responses and doublet responses and so forth, then combining them together using non-linear integrative mathematics. Several years later when I stopped doing this and I was describing it to somebody they said it sounded very similar to what they understood to be Wiener's last work. What he was doing with non-linear theories sounded very similar. I think I've lost the notes. I've still got the magnetism theory notes, but I think the yellow pages are lost. (We used yellow pads of paper so you wouldn't make copies of our classified work on the Ozalid blue-printing machine. I never knew the reason, but yellow is black as far as the Ozalid processing -- which is how copies were made in the fifties. Anyhow, we used to say, "find it fast in the yellow pages" because we had all these notes. That's one set I think I've lost. If you're interested in my mathematical background, that completes the step that had to do with classical real variable analysis. All the rest of it, the mathematics, has been much more influenced by algebra, topology, and formal systems, although I never studied "formal systems" as "formal systems" -- never took a logic course of that sort. I took a simple symbolic logic course at Oberlin where we studied Carnap and Reichenbach. I did my term paper for that philosophy course in logic on "A Translation and Analysis of Godel's Theorem into Reichenbach's Symbolic Logic Notation", showing that by so doing you got a much deeper insight into what Godel's Theorem was all about, which was not bad for a junior in the sticks.

ASPRAY: Godel's Theorems are quite complicated.

ROSS: Yes. Well it was to me too. And I was really smitten by symbolic logic and by the fact that it worked. I can remember, it may have been about when I was reading that Selectron article, walking down the square in Oberlin with one of my buddies and exuberating about how some day you'd be able to take this symbolic language (this formal stuff) and make something that was actually thinking -- really solving problems and so forth.

ASPRAY: Little did you know. I assume from what I've read of your materials this is going to play a rather major role later on --in terms of formal language theory.

ROSS: Yes, that's right. Again, its sort of a funny mix because I'm not sure I really know, today, what a proof is. I am supposed to be a trained, pure mathematician. I have done my share of proofs going through the courses. I've set up lots and lots of formal systems, and so forth; but I'm not really that kind of a mind somehow. Algebraic systems I can't think with. I can do it, sort of; but I really don't think that way. If right brain/left brain means anything, I'm strictly a picture man and a modeler. I make it tangible. Matrix algebra, for example, always eluded me. I mean I could do it, sort of. But I couldn't live it. Things from it just don't stick with me. That's why I say I'm really an engineer or philosopher much more so. The only way I understood all that quantization stuff was by looking at what was actually happening in the fine structure of the calculation of those things. It wasn't until I got my picture of the little abacus tipping up and making the distribution function that I really came to grips with what that was all about, and seeing the calculation of statistical moments as a question of balance and so forth. I have to see it, feel it. Anyhow that does complete that branch. Now to the other branch of my math background, besides this working acquaintance with algebraic systems and formal systems. The other key thing in my mathematical background and the way I think, the way I create, is geometric vectors, space, seeing. Remember, I did four dimensional geometry when I was a freshman in college. I really had a ball with it. I got to where I could do Euclidean-type proofs in four dimensions, understand in four dimensions how you rotate around a plane. I was actually able to see that. On the

geometry side, that's the one thing I remember from high school, too. I did take a course in geometry in high school. I had a very nice old teacher. But she kept marking all my proofs wrong. I would always proceed to show her my proof was correct. So she was magnanimous enough --enough a real teacher to say, "By George, you're right. It's not the way the book does it. I've never seen it done that way before." Every time it would come out different, screwball. Some of the times my proofs were much shorter than the ones she'd seen. But she did understand both Euclidean geometry, and the value of what a proof was. That was the only real proof work I'd ever done until I got to MIT. There was terrible pressure there for me that first term.

ASPRAY: Did you take a course in abstract algebra?

ROSS: Yes, and all those whizbang guys from New York super high schools could solve things in their minds and prattle on for hours at a time on Banach spaces, etc. I was just completely swamped! I did start to understand what a proof really was after that. But the geometry and spatial things always have been very important in my envisioning things. As part of that Oberlin honors program I whipped through projective geometry, homogenous coordinates, covariant and contravariant tensors, and that sort of thing. My main work at Servo Lab on fire control was actually spun out from the other work where we did all the complicated calculations and I made that programming language for the people to do the actual evaluation of the gun-aiming errors and so forth. It was a thing that I very quickly go into. They had just come up with a new form of ballistic table called "Air Mass Ballistics", which I won't discuss in detail. The net effect of it was that they wanted me to look into the possibility that these tables could be used to do a more efficient job of calculating and reducing the data that was measured by photographing all those dials and using computers. So, sure enough, working with Ranny Gras, from the Instrumentation Laboratory -- who, by the way, even in those days 1954-55 built himself a solar heated house, an interesting guy. I worked out a whole new set of calculations for doing this evaluation. A very heavy use of geometry, vectors in three dimensions, and that sort of thing. Also, getting back to the early days of Whirlwind, when they shut down to put in the new input-output system, that piqued my interest. "What's all this top secret stuff? Got to know about that. What do you mean secret? I have secret clearance. I can know about secret things." So sure enough, I talked to John Ward. He talked

to, I can't remember whether Reintijes was head of the Lab then. Maybe its still Bill Pease. But he said, "Sure, we can get a tour and see who's over there. See who would be useful on our project." I can still remember walking into this back room, behind the green door type thing, and there was this huge room. I believe it was already either shrouded or painted black but it was dark all around the ceiling. It had something like twenty-four 19" oscilloscopes in consoles all sitting there glowing. Sure enough, these were all various work stations, from the commander's station on down, for tracking all these airplanes in the Cape Cod air defense system. Pushbuttons all over the place. Nothing but pushbuttons and a light gun, which was a photomultiplier device shaped like a pistol with a wire coming out of the handle. You pulled the trigger and the computer would then note the spot that had just been put on the screen. That would be the way you initiate tracks of airplanes. Also, over in the corner was the thing called the "Light Canon" rather than the "Light Gun." It was a permanently mounted photo-multiplier tube on a tripod looking down on a horizontally surfaced, 19" screen. The yoke was pointed at the floor. They laid a circle of yellow plastic (yellow lucite) about 18" in diameter on the middle of this tube so that the Light Canon above it would only see the annular ring around the edge. That's what they slaved the raw radar data to so that whenever new data came into the field of view the computer would automatically see it only in that little ring. It blanked out all the radar returns in the middle that were already assigned. The computer would pick up the new track and put it on somebody's console. They would pick it up with their light gun and pass it on to somebody to track. So in that room there were just jillions of buttons. Of course, the new SI instruction could tell you to read this bank of buttons or that bank of buttons, to activate this register bunch of switches at the different consoles around the room. I said to John that we could use this stuff very well for our data reduction problem. Sure enough, we did over the next seven years, roughly. In fact, we were the only project besides Lincoln Lab using all that equipment for man-machine interaction for all that time period.

ASPRAY: Were you actually using their equipment?

ROSS: Yes. You see I don't think we ever paid for a computer. That was all just run by Lincoln Lab or the Office of Naval Research. I remember our project being charged. We kept track of time; but I don't think it cost money. I

never bothered with money matters. With this new set of airmast ballistic equations I was able to add display programs. By then they had gotten the core memory into Whirlwind, which gave us much more reliable memory and four times as much. Also, the drum was then working and the tapes were almost working. So we could do some pretty big things. I had several programmers working for me and later added more programmers at Emerson Electric in St. Louis, which was making the firecontrol system itself, and some Air Force and Kelleck people down at Eglin Air Force Base in Florida. So they were all programmers on this thing that I was directing. We did man-machine interaction for this on-line data reduction for quite a number of years. Out of that came quite a number of things that were firsts. Do you want to discuss some of that too?

ASPRAY: Yes.

ROSS: O.K. I have no idea whether I'll get the sequence correct, but I'll hit some of the interesting points. I mentioned that Jack Gilmore had done what was the first of the assembly programs. Also, though in the next summer (I believe it was the summer of 1953 but it might have been 1954), the Digital Computer Lab started having Summer Session programs where other people would come in and give a one or two week course to tell you what they were doing. To do that they started giving the Summer Session computer system or software system. They didn't call it software then, but Summer Session System. They started putting a lot of neat things in there. Charlie Adams was running the group and was my main contact. (I never worked closely with Jack Arnow after that beginning. I knew him. He later ended up going to Lincoln Laboratory and then spun off from there to start Interactive Data Corporation, which is right next door to SofTech. Before SofTech's first office space at 391 Totten Pond Road, across the street from where we are now became available, we worked out of my study at home and a borrowed office from Jack Arnow's IDC operation. It's amazing how you get these long distance ties.) The main people I worked with during the 1953 period were John Frankovich, (who is still at Lincoln Laboratories. He was then at the Computer Lab). Frank Helwig. I've lost track of where he has now gone to. He was at the Computer Lab for a long time. He lived in Lexington. I knew him and his wife slightly that way. Then later, Arnie Siegal and Sheldon Best, when they were doing other portions. But mostly I worked with Frankovich and Helwig at the technical level, and Jack Porter,

who was at Mitre, but I don't know if he still is. He was the group leader after Charlie Adams left to form Adams Associates. I remember John Proctor was also the Lincoln Laboratory Administrative Officer-Manager. He was making various deals and so forth. During this period I had my own programming staff that I brought along -- you know, that I trained and got doing things. But we always had much more than we could handle. So what I would do is get interesting things that I needed, I would describe them to John and Frank, and we would find a way to have them go ahead and make something for my project to use with my project paying or not paying (I don't know how that worked out). Anyhow we'd make this deal between my project and the Computer Lab, with them actually doing something interesting to expand what Whirlwind was able to do for general use. Out of that came a number of interesting things. The first one...let me go back to that Summer Session system.

TAPE 3/SIDE 1

ROSS: From Gilmore's first system they did the first Summer Session, I think. I can't remember. But out of that, in turn, came the CS (Comprehensive Service) System, which was a full symbolic assembler. Why their terms for important ideas never stuck I don't know. They were much better than those introduced later on. We had "flads" for example. Have you ever heard of a flad?

ASPRAY: No, never.

ROSS: A flad is a "floating address". We had flads all over the place. They started with a letter (a signal letter) and one or two digits. Their system would keep track of what they were. You didn't have to figure all these absolute addresses and patches as you did before. You could start writing bigger programs. I forget if we had comments. I don't think we could afford comments. You tend not to put those in when both space and time of machines and people is at a premium. To input all this stuff through a Flexowriter punching paper tape at ten characters a second is a lot of work. The girls that used to do that were marvelous. You'd hand in your manuscript, they'd be banging away at it and call you up on the phone and say, "Did you really mean this?" because they would go by the patterns of

things they saw. They frequently found errors in one's programming. Typographical -- you know writing -- errors they would correct on the fly for you. It was really neat. In any case, you could make these big programs and we proceeded to do so. In fact the whole system of programs got to be rather outlandish. The Whirlwind console was a bunch of toggle switches and lights. It started out with just the Flexowriter with its mechanical tape reader that you read the tapes in on, the mechanical tape punch that punch taped out, the scope, and its camera. That was about it. Well, shortly after I started, they got the first PETR (Photo Electric Tape Reader). It was a marvelous device that could read 200 lines per second instead the ten or twelve as before. They later upgraded it to 400 lines per second. That was a real boon! They hooked it in, through the new SI input system. You'd give one SI command and select the photoelectric tape reader or another SI command would select the old Flexowriter mechanical tape reader, but that was unusual. To read programs in you set the toggle switches to pre-load the registers of the machine. In particular one of them was read when you push the re-start button. Whatever you had set as address it would jump to. Normally that starting address was an Octal 40. The reason was the first thirty-two (i.e., Octal 40) decimal registers of the machine were actually toggle switches. One switch per bit. When you pushed the boot-strap button or start button on this machine (not the restart button, but the start button) it would start at register one, I believe maybe it was register zero, and it would execute the program that you had manually toggled in. So that if your program started at the next available memory location, you ran right through the first thirty-one words of storage into thirty- two, which is Octal 40. That would be the start of your program. So, normally, your program started at Octal 40. The other neat thing about it was the exotica of writing thirty-two line programs to start that system up while initializing everything properly and using wraparound (when you went off the end of the memory space you were back at zero, and there was some wraparound thing that they used there). Anyhow, you did have to toggle in where you wanted it to start the program. You normally also then initialized the accumulator with something and maybe the B Register with something, too. They each had their own switch sets. Then you would push the go button. For starting out very complicated programs normally you would read in a tape, set some switches, read in a tape, set some switches, read in a tape, set some switches, read in a tape, and so forth. After you had finally gotten all those things in, some of the switch settings would be things to dump memory off onto tape or disk and make room for the next one to be called in, like my Mistake Diagnosis Routine moved pieces around. Anyhow, it got very complicated and very error

prone even though we had excellent operators. They really loved their work and had great times doing it. Still it was getting a bit out of hand. So, I said to John and Frank, "Why don't you make me a program for which I can put all of the switch settings on a tape to be read by the mechanical tape reader. Then all I have to do is splice together with scotch tape all my other tapes in one big role." Then, with one start up, I could get the whole thing done and not have those errors. "Oh, that sounds like a great idea, yes! We'd like to do that." So they went off and they did it. My claim is, to my knowledge that's the first Operating System and what was going into the switches that went on to that Flexowriter tape was the first JCL (Job Control Language). It was forced by necessity because my programs were just too complicated. In the Fire-Control Evaluation Project, under John Wond, John Brean was in charge of the engineering side of the project. He was making a set of digital instrumentation to replace reading those film pictures of dials by hand. They were using analogue-to-digital conversion devices, selsyns, and that sort of thing with coded discs. Gray coded discs with optical readers to make a system that would measure about forty different variables in the airplane and record the numbers on magnetic tape. At that time, telemetering doing it in real time to the ground. So they were making this Digital Flight Test Instrumentation, the "DFTI" Package, which was also written up in the Analog-Digital Conversion Techniques book. It was the source of all this. What it did was directly make digital readings of numbers (frequently Gray coded) out of all the different things that were measured, and would record them on a magnetic tape, a big Ampex one inch wide magnetic tape, with a clunky device. Well, at least it was flyable. Included in this, by the way, was a digital camera. They mounted a flashing xenon flashlamp on the fighter aircraft that's attacking the bomber whose defense system that we were measuring. Then, mounted on the bomber's gun, looking along the barrel, was this little box that had two cylindrical lenses -- lucite lenses -- mounted at right angles. Take a cylinder and cut it lengthwise so it has a semi-circular profile. That makes a line image out of a point source. The angle that you're off with respect to the axis of the cylinder is maintained by this line image so that with the cylinder lens mounted vertically, the line image would sweep horizontally, measuring X, and the similarly mounted horizontal one would sweep up and down measuring Y. This line image then swept across a coded glass plate with black and white, dark and clear, regions with photo diodes behind them. That would then directly convert the angular offset of the target from the gun aiming into a line sweeping out that Gray-coded digital signal, which would go directly onto the electronic recorder of the magnetic tape. So, instead of having people later look at pictures of

airplanes with cross-hairs, we popped the numbers right straight on the tape. That combined with the other things measured by code wheels, measuring up to thirteen bits of accuracy was being slammed out to this tape. The intent was to read the tape into the ERA 1103 computer. Eglin Air Force Base in Florida had one of the first 1103 computers installed down there. My programs (the programs that my group would write) would analyze all this data and give the answers. We were writing the first programs on Whirlwind; but the fixed programs were supposed to run on the 1103. So again I subcontracted to Frankovich and Helwig to make a system that was like the Comprehensive System, but for the 1103. They worked with me on what features it would have and so forth. It would allow us to do the same kind of symbolic programming for the 1103. At the time, you could only program the 1103 with what was called Bi-Octal (two sets of octal digits, two digits per word). They had this little relay box with physical plugs about the size of an apple that were just dummy wired inside but pronged on the end. You plugged two or three of those in this box that had a mechanical tape reader on one side and a mechanical tape punch on the other. That was how you converted the octal code, all numbers that you typed on the Flexowriter, into the bi-octal tape that came out the other side. They didn't even use the computer for the conversion. So we couldn't imagine writing this huge program, very sophisticated stuff, to run that way. I got the Whirlwind people to write this symbolic assembler system for us to use. We would use Whirlwind to write the programs, make bi-octal tapes, and then fly them down in an airplane to Florida to run them. Early on, I also hand-wrote in 1103 language a little companion system that essentially did what Jack Gilmore's first system did but ran on the 1103. That way we could at least make some patches to help with debugging, without having to fly all the way back to Cambridge. We did all sorts of fancy debugging systems there at Eglin. One was called "Save the Baby" (STB), don't throw the baby out with the bathwater. They had a drum memory as well as core memory on the 1103. So we arranged to pair a previous drum image with the current core memory and just do a changed word post-mortem. Combining that with break points (which were like my Mistake Diagnosis Routine) you could put timed break points in various places and say what memory cells you wanted to compare with STB. They also had one of the first high speed printers hooked up down there. You would set up these break points, use the changed word post-mortem for just those things that you were interested in on the fly, and dump them out on the line printer. That was Save The Baby. So you could get debugging information dynamically. Stuff like that. So we had better tools, in fact, with the line printer down there than we were used to at

Whirlwind. The Whirlwind Scope Dump was faster than the Eglin live printer, but we couldn't get the turnaround time to get the film back. Anyhow, we did various kinds of tool-building activities, and then later we just completely subcontracted tools to the Computer Lab people. It was Arnie Seigel who had been working with the Milling Machine Project before I began working with them. Al Susskind was in that project. I knew all of them but hadn't actually done much work with them. I'd been involved in all these other areas. One of the last things that Arnie Seigel did before he left MIT (he was an electrical engineer, in fact), was a job for me. It had gotten to where we needed to have more capability with respect to those fancy Cape Cod System Sage- type consoles in the secret room at Whirlwind. We needed to have full character (full keyboard) input, and they never had had keyboard input to Whirlwind. I guess probably you could, but I never even knew anybody that tried to use the keyboard on the on-line Flexowriter. That was part of the computer itself. We used the mechanical punch and reader. I don't remember anything ever going through the keyboard because you had the switches. In any case, there was nothing that corresponded to a keyboard input on the computer, and certainly not in the back room. So again I subcontracted with Arnie to design and install a Flexowriter that we could wheel up and plug in in the secret room out there beside the master console, which was the main thing we used. To my knowledge that was the first interactive keyboard use at MIT. I don't think that anybody except my project ever used it that much, because we were phasing out of Whirlwind into the IBM machines (704) at that point. But that precise set of equipment (the design of the relays at least, but I believe the equipment as well) was the first keyboard into the IBM 704 at MIT when Frank Verzuh was there and "Corbie" Corbato was working for him. So when they decided to do this same thing later it became the time-sharing project (the precursor to the CTSS Compatible Time Sharing System of Project MAC) and all that sort of stuff. They knew that I had this one, and I said, "Well, why don't you just have Arnie do that and see how that works for the 704 and improve on it for your next round." So the starting of that whole time-sharing development had its roots back there with my project's needs. The same way the operating system idea came up because our tapes were too big, this came up because the Master Console had only pushbuttons. Earlier, in 1954 or so, I had planned and thought through an algebraic compiler, but after I had done about six weeks work on that, I discovered that Lanning and Zierler already were just about to check their program out. So I said, "Oh, hell with it, I'll go out and do..." I think that's when I decided to do my Mistake Diagnosis Routine properly as a general tool. So I also was

going to do a Fortran-like something or other and had it started, but decided "Well, phooey, those guys will do that. I'll do something else," thinking they would finish it up and there wouldn't be anything left.

ASPRAY: Do you still have your plans for that?

ROSS: Yes, probably.

ASPRAY: How much did it look like some of the later languages that were written?

ROSS: I really don't remember. You see, in terms of background, at that point all we would have is the Gilmore Assembler-like thing that we were regularly using. We might have been using an early version of the Comprehensive System/Summer Session System, but I never knew very much about that. (Frankovich and Helwig were rather secretive with me about the early development of that.) The main thing was to be able to do algebraic equations the way they looked. Then I wanted to be able to do vector equations too, I think. I had all that coming out of my fire control, and the airmass ballistics, with all the three-dimensional vectors. Also at that point, through a visit or an acquaintance of some sort, the Computer Lab people became familiar with work at the University of Manchester, where they had the B-box Index Registers. So, even though Whirlwind didn't have Index Registers or a B-box, they did introduce that concept in the floating point interpretive library system that we had on Whirlwind. Back in those days everybody had to do their own number stuff. In fact, all my work for a long time was done in fixed point, where you had to control where the mantissa was and you had to learn how to do "Sign Agreement" before you printed things out. All these neat things that are now buried in chips and youngsters don't even know they exist, you had to do. But we did have for floating point operations this library system that had indexing and did the abstract. Whirlwind had a special instruction (SF, for "scale factor" I believe it was), which shifted a binary number left until it found the first non-zero digit and kept track of what the amount of the shift was. That was the heart of doing the floating point. I can't really remember what my Fortran-like project was going to be. Again, I only spent at the most six weeks part-time on it. It was just an interest to try something else while everything else was going on. That was

as far as I got before discovering that Lanning and Zierler were already so far along, that I said, "Heck I'll do something else." If they were going to do that one, there wasn't room for two of us.

ASPRAY: It's interesting that several different groups were trying to do that.

ROSS: Yes, these things were in the air. So you find a natural... You asked me to say a little more about the day to day operations of Servo Lab and my group. When I started I was just doing things that came directly out of what the project needed doing. It wasn't until the end of the summer when I got going with Whirlwind that I had my own set of problems generating from what I was doing. In fact, I still can remember the first day that summer that I realized that I had spent the whole day and I hadn't done anything except think. And I was paid for it! With all my other jobs, during college and so forth, summers had been spent working as a laborer. I used a jackhammer to break up streets for the gas company or did factory work at Eastman Kodak, and things like that. So it was a real eyeopener to me to realize that you could actually be paid for doing nothing. It wasn't long before the various things came out of the follow-through of both the radar noise analysis and fire-control system evaluation, which had been my summer job. Since I was going to be there full-time, getting off into the air-mass ballistics, it got more like pacing myself. Some of the people who had been there started to work as my assistants, you see. They got work out of what I was doing. I can't really remember much about the way that things were done, except that they were pretty much dictated by the way that the Whirlwind operation was set-up, with the keypunchers and schedules. We'd sign up for our own time, which frequently would be at 2:00 or 3:00 in the morning. We also kept quite a number of records of the different runs, designed forms for keeping track of things. We came up with forms to keep track of what happened on a run at the Computer Lab that the operators would fill in for you. So there are interesting things for someone to look into at some point of just how the operations of what would later become a computer center ran -- what that evolved from. I guess I really don't think of any significant comments about the day- to-day running. Very early I became convinced that the methodology that I was developing, and this is fully described in the Pre-B58 Reports for the B58 Airplane Ultrasonic Bomber (I think it was ultrasonic) that was to replace the B47 and B52. But the B52s are still around! They may still be using the same tail turret that we were testing then, for all I know.) That was why it was

called the Pre-B58 project because they had the Digital Flight Test Instrumentation to test those earlier planes, too. In order to solve that problem we developed a whole new method of analyzing and formulating problems that I knew was very significant for computers, right from the beginning. The idea was that you took very complicated problems and broke them down into a mixture of very simple calculating steps interlocked very intimately with very little pieces of carefully-chosen logical expressions -- choices of which one to do next. This was instead of solving a big problem by one big blotch, which is what they had done in their analysis before. For example they had great, huge calculations and all sorts of sines and cosines and arctangents all over the place, which didn't work out too badly for analog computer stuff. The same sort of thing was being done at the Instrumentation Lab, the analog computer hotbed, for fire control systems. They had all sorts of notations for naming variables with respect to coordinate systems, which later on was how they kept track of things for NASA. They had all sorts of ways of getting symbolic names associated with the many, many variables and coordinate systems they worked with. The calculations were big, horrendous things. And we only had one to four K of memory to do things -- and still wanted to do more things. The new air mass ballistics formulation was a new formulation for calculating what happens to a projectile when its fired in a moving air mass. In other words, when an airplane is moving through the air its as though there is a big wind blowing and, if you think about it, a spinning bullet coming out of a gun with a very strong wind blowing sideways on it acts strangely. You've got this spinning mass with very strong gyroscopic effects. When you push with the wind on this thing, in addition to the direction it's going, the gravity that's pulling at it, the changing of air density as you go up in height, and all that sort of thing, it gets to be a fairly complicated set of partial differential equations to keep track of this thing tumbling, spinning, precessing, etc. It's a mess. Well, they came up with this technique where they just measured the "drag coefficient", so called, of the projectile -- which is a function of it's shape. You come out with a bunch of functions depending only upon the speed of the bullet through the air and its shape, called the Siacci functions. It was all written up. The Siacci functions all depend ultimately on integrals of various simple algebraic operations: divide by the velocity or multiply by the velocity squared, that sort of thing. They tabulated these functions in tables. And then with enough elaborate computations around it that would give a solution to this complicated problem. It was quite different from the style they'd used before. That was what they asked me to look at. Well, the natural thing to me was to solve the whole thing in vectors, in three dimensions. For

the new DFTI calculations, we wanted to solve this thing every fortieth of a second. In other words, you imagined that you fired the bullet every fortieth of a second. If you calculated the closest distance between that moving bullet and moving target, then we could apply a dispersion pattern and figure the probable number of hits. Also you could reflect that back to gun-aiming errors and determine how to do a better job. In any case that was the nature of the thing. But we had to solve this complete problem for data measured every fortieth of a second. Well, I noted that the conditions of the problem don't change very much in a fortieth of a second. They're smooth. It's a natural problem. And, also, what you had as the fighter approaches the bomber closer and closer was that the time of flight between initial conditions, there your fire (imagine firing) and terminal conditions (the final end point where you're closest to the moving target) gets shorter and shorter as the fighter catches up to the bomber.

TAPE 3/SIDE 2

ROSS: Again, the time-of-flight charge is not very fast relative to the forty times per second which you're working with. So, both the initial conditions and the terminal conditions of a repeated problem are changing slowly. Therefore the essence of this method of approaching problems is by hook or by crook to crank your way through to one starting solution and then apply very small changes to both beginning and ending conditions, only keeping track of the differences so that you can carry forward the bulk of that starting solution that you have always maintained, to get the whole set of solutions. It turned out that by choosing the proper funny coordinate system and imagining the origin was on the bullet (so that you just saw the target airplane whizzing past you), this whole complicated set of equations boiled down to equations that were no more complex than, maybe, a single multiplication and two additions, or something like that. But then several of these things were all added together through the vectors. It gave a much more complete solution than they had ever gotten before. Later on we put gunbals and inertial coordinate systems into the Digital Flight Instrumentation System so we could measure own-ship motion, with roll, pitch, yaw, and accelerations. We then could reconstruct the whole bomber path. This was almost infinitely more complete than any set of analyses that they'd gotten before. By the time we got through programming this stuff for the 1103 computer at Eglin, we also got these much better answers fifty times faster than

they'd done them before. In fact, just the computation speed, a multiplicative factor of fifty, is really outlandish with respect to most problems. They were much harder problems that we were solving. I saw these same techniques as being applicable to any smoothly changing system, any physical problem. We published reports on them but don't think we ever prepared any refereed papers or conference papers on that. It was sort of a closed community anyhow, I guess. That background is what led to the approach that I later took to the numerical control (machine tool programming) problem of three dimensional geometry. I came to call my approach the "Systematized Solution" that you could adapt to specific conditions. Also, that same Pre-B58 System was the first application that I know of a learning program being used in a serious, useful way. After you got through setting up all those weird sets of vector coordinate systems -- various angles and this sort of thing -- it turned out that even though the problem conditions were changing slowly you did have to know when they were getting bad and recalibrate the software, recompute the parameters. It turned out that to recalibrate the system amounted to doing an extra four more complete evaluations of the miss distance of a projectile. That's relatively expensive. So, using this idea of measuring things that you could measure with simple computations, mixing them together with simple logic, but having a whole structured approach, we put in what I called the "Servo Check Loop", because it logically feeds back a correction as in a servomechanism. The idea was that one of the outputs from the evaluation calculation is the actual gun aiming error, in other words how much in azimuth and elevation should the gun have been corrected by. Well, I could take that error and feed it back in as a correction, as though I had that as the initial conditions, evaluate that projectile flight, and see if I came out with a zero error, as theoretically I should have. So I was able to have an independent calculation by feeding it back in. That then made an error signal that I could use to correct the calibration parameters used in the evaluation program. So we had the Servo Check Loop for keeping the parameters always up to date and getting good answers all the way through. And we were confirming that they were good answers by actually checking every so often.

ASPRAY: You wouldn't check it every time though?

ROSS: No, only every so often. But the real question is how often is "every so often", especially where recalibrating is so much more expensive than just keeping going. Well, that's where the Learning Program came in. First of all, the

data that came out of this tape, a crude, inch-wide tape recorded in the airplane, not the high performance computer tapes we're used to now. All those conditions, plus crudely being read into a crude computer on the ground, and so forth made for lots of room for bugs and errors to creep in. Data drop-outs and so forth. So, our data was frequently noisy. I'll get back to that. The idea was you didn't want to be too cocky or assume that you were getting good answers, especially when you had the ability to actually check whether they were good or not by closing this loop. So, we made this learning program which started out by checking the first frame, then every other frame, before long every eighth, every sixteenth of these forty frames a second. Whenever it ran into errors and had to go back and recalibrate, it would get more conservative and cut back to every eighth, every fourth, and so forth. We could even back the whole calculation up and just redo things. The idea was that it actually kept track of its own performance. It adapted to the conditions of the data. So if the data was noisy or had lots of dirty stuff around, then it would slog through it at a slow rate. If it was a nice clean run, it would just zap right through and give good confidence. It worked pretty well. In fact, it was unlikely that we could have completed this whole program without that program and another one that I should mention, which was the IDPP, the Initial Data Processing Program. This is a program written by Ben Scheff, who was a classmate of mine at Oberlin. Ben went off from Oberlin to work at NSA and he was in the Navy, stationed at NSA. Then he came to work for me at Servo Lab and later went off to RCA. Now I think he's still at Raytheon. Well, anyhow, Ben and I designed this thing which was intended to preprocess the raw data that came off this Ampex tape. One interesting thing about the tape was there was no tape transport for it that would start and stop; none of this thing about Servos that take up loop slack and all that sort of thing. In this equipment either the motor would turn and the wheels were going or it wasn't going. If it wasn't going, it was either coasting or building up speed. That was it. So instead of making all sorts of control logic to make a better tape transport... It's always easier to solve it with the software than the hardware. Well, I volunteered in this case, there that we could make a thing that could do pattern recognition of the tape. So whenever they shut the thing off, we'd watch the data from the tape as it slowed down. If they could give us a mechanical backup and replay of at least a minimum amount of time so we could be sure we were back past the places where we had had good data, then we would reread the tape going forward, recognize the good data, see where the data changed, and that's when we would really start reading it again. We'd accommodate all this starting and stopping of the tape with our software. And we did! That's the only

way that the whole thing worked -- because the IDPP had enough pattern recognition in it, again getting into Artificial Intelligence work -- enough pattern recognition on it if you did just allow the tape to coast to a stop and back up some amount. As long as it was in something that we had read before, we would find the break and read more. Meanwhile, we were building stuff up in the core memory and dumping it on the drum. When we dumped it on the drum was when we had to ask for the tape to stop and back up again. Furthermore, that only got the tape, the raw data, into the computer. The data was so full of drop-outs and problems (this and that and the other) that we also had to have interpolation schemes to begin the pattern recognition. We had to be able to recognize bad data and interpolate good data which we would flag and carry through. But we would at least be able to get the whole calculation dependent upon smooth data, which was essential with my method of calculation. This IDPP program did all that stuff in the most weird and wonderful way and would actually make a good run out of pure noise I think. But the combination of those two is what allowed that whole program to run. Even back in the middle-late 1950s, if I recall, it cost something like \$10,000/hr to run those tests. In today's numbers who knows what \$10,000 is? A lot of money! And that was just for the stuff itself, not counting the whole flock of other things about airplanes. We were just paying for people. Before our DFTI instrumentation and the Pre-B58 programs they didn't used to get the results back for three or four months. By that time, they'd shut everything down. So that was why this real time (or almost real time, next day turnaround) of Digital Flight Instrumentation and On-line Interactive Data Reduction was so important. We also copied much of the design of the Sage System or the Cape Cod System of Whirlwind man work intervention hardwired into the first use of the Charactron display outside of the Sage System, hooking it up on the Eglin Field 1103. We built them a console that had pushbuttons and keyboard and light cannon at Eglin. John Ward and I, and others, worked out the design of that thing. And the Lab, John's engineering group at Servo Lab, built the system. We took it down and installed it. That was critical to making the whole program work. While I'm at it, I may as well say, the other thing about getting that input data, probably 1954 or 1955 (I think around 1954. I can't remember), was when I wrote the first hand-drawn graphics input program to a computer. I was flying back from Eglin. On the airplane I got to thinking about this IDPP that had bad data points. I remembered the light cannon sitting in the corner. Remember, that's the one with the tripod. Well, if I took away the yellow disk in the middle so it could see the whole screen, then I decided I could track shadow. In other words, we had the light gun, but it just

pointed at whatever fixed spot you were aiming at. We didn't have light pens yet, you see; there wasn't any such thing. I decided that I could make the light cannon see whether my finger was in the way of a dot being displaced. This is one of the few programs that I ever wrote that worked the first time -- about two hundred instructions. I told about this over at the Computer Museum one time. I decided that if I wrote a program to put out a 5 X 5 array of dots (left to right, top to bottom) and use the light cannon after each dot was put out to ask, "Did you see that?" If it did see the dot, then I put the next dot out. As soon as it got to a dot that it didn't see, then I would recenter the 5 X 5 area around that and start over again. If you think about that, what that does is it makes a Maximum-seeker-from-the-left. Any shadow that's in the way, it'll climb up to the top of it and won't go down the other side because it's always going left to right. So then I moved my finger around, the little pattern would supposedly stick on the end of my finger. Well, how would I do it? I wrote the program starting out with a great big 5 X 5 array; so it would pick up the shadow of my arm and then shrink down the pattern size. The only trouble was that, in the secret room, the Master Control Console was in the middle of the room, and the light cannon (because nobody ever used it for anything because it was fully automatic) sat way over in the corner about twelve feet away. Now, the way I used to run was that I would take over complete control of Whirlwind from the operators, who would stay out in the main control room and smoke cigarettes, drink coffee, and gab while I ran my session entirely under control from the Master Console which duplicated everything in the other room. We also always ran Whirlwind with the audio on. I think they had a loudspeaker hooked onto the digit 13 in the accumulator, if I remember right. Something like that. They also spliced the various display lines together on the scope that you could see in the Main Control Room. That was the way the group normally ran Whirlwind. Then they had an intercom connection to the back room, at the Master station. So I'd just be in there and run, and when I was through I'd call them up and say, "OK, now you can take over". I'd wrap my stuff up and go back to the Main Control Room. I would have left all my tapes sitting in a big-roll tape holder there, with the master tape (or whatever we called it, the Director Tape program for that operating system), in the Flexowriter reader, so I could run the reading from the other room. Anyhow, I ran this graphic input program the first time. In order to get my finger transported from running the computer to twelve feet away to run this thing in the corner I had to figure that out. Well, realizing that it's a shadow-seeker-from-the-left, I just laid down (on the horizontal tube in the corner) a piece of paper at roughly a 45 degree angle. I

went over to the Master Console, fired up the program, pushed the go button, and walked back over to the corner of the room. Sure enough, when I got over there, there was a little patch (about a quarter inch square) of bright, very bright, dots sitting perched on the corner of my paper that I had laid on there. I snuck my finger up behind, in the shadow, and gradually pulled the paper out from underneath so that the little light gob was stuck on my finger. Still the program was working. Then I started to write my name, and instantly the loudspeaker of the intercom blasted out. The guys from the other room had heard this funny noise because it was a different kind of random tracking sound. So they automatically looked at the lights on the console and looked up at the scope to see what was going on. Here they saw my name coming out on the scope. And they jumped on the intercom and said, "What the hell kind of program do you have there?" They just couldn't believe what was going on. So, anyhow, then we did demonstrate that in various shows about the Data Reduction System that we had for people coming through. After that various people suggested that I even go and get a patent on it and that I write it up in an article for Aviation Week. I never got around to it. There was always too much to do. But we did use it. We fixed it up. I forget whether we actually ever got it put in or how much it got working at Eglin. But the reason I did it in the first place was to be able to hand draw the things that the IDPP (the Initial Data Processing Program) was having trouble interpolating. I wanted to be able to just have a display and fill in bad spots by hand. Then of course, it was clear that it was also related to the milling machine, and I did get into that a year or two later. We always had the graphic input as a part of the thinking. So, anyhow, a lot of interesting things came out of that early military work. The other thing that also shows in the literature (the reports), and I did use this a couple of years ago...There was a meeting at the National Bureau of Standards. I can't remember what it was about now, but because we were active in Ada (the Ada Language System that SofTech was building for the Army). I was asked to cover the topic of software development environments in ten of fifteen minutes. I couldn't figure out how could I possibly say anything in a short time like that. So what I did was look back and take material out of a report that was actually written in 1957 or 1958 and published in 1959. I had a handout for the meeting which is a chart that I had made then of our Experimental Software Development Environment that I had going at Whirlwind at that point. Because of out of all the Pre-B58 work we had branched out into other areas, again, because I wanted to have my methodology (chopping problems up into little pieces, etc.), be used other places. It was also sort of tricky to do all this stuff. We started working on tools

right from the beginning. I had that Mistake Diagnosis Routine. John Walsh helped finish that. At one point, there was a report on it. If you could spare twenty-three words of memory on Whirlwind, that was enough to put in a thing that would dump out enough of your program to move in the Mistake Diagnosis program and do the whole thing from there on. We could interactively debug any program that had twenty- three spare words. Furthermore, I did some experimentation at the time on using that for refining the calculations of these computing programs. Combining the Mistake Diagnosis Routine with the Scope Output Routines and another program that I had done for Polynomial Fitting (automatic fitting of polynomials -- I wasn't smart enough to use Chebyshev Polynomials, or anything like that, just the straight polynomial). But we had a Polynomial Fit program with error calculation and fitting higher and higher degrees of polynomials to ballistics tables. That's how we got them put in the evaluation programs. Combining all those things together I had this concept of being able to take a very complicated program; put in break points at different places; and plot those points on the X and Y axes of the scope. Now if there is a functional relationship (or almost functional relationship) between those two points of the calculation, you'll get a scatter diagram that will make a very high correlation on the scope. You'll be able to see that these two things are very closely correlated. Again, for certain kinds of problems you don't need the fine detail. So I was going to have it that you take a completed, checked out, but much-too-slow-and- complicated system; put in break points; experimentally determine functional dependencies; experimentally accommodate those dependencies by curve fitting; and then replace (in an engineering sense they're really) modules of actual software; putting in these shorter software pieces even though you couldn't figure out analytically how to do it. Then you would actually be able to whittle away at very complicated real problems and apply my methodology. We never got to the completion of all of that, of that aspect of it. But we did get a whole flock of other tools built along the way for writing very big, complicated programs. I forget what's on that chart. It would be interesting some time to just take the chart and go through it. The system was called "SLURP". It sounded good and stood for Servo Lab Utility Routine Program. That was the basis for our experimental program development environment. So in addition to the MDR (Mistake Diagnosis Routine), Scope Plot Routine, Scope Input Routine, and the various assemblers and debuggers and so forth. It was also intended to handle the problem of making and labelling various kinds of output: displays on scopes for real time use, displays on scopes for photographing and documentation, print-outs on the Flexowriter (we still didn't have a line printer). The idea was to

be able to handle all the formatting, labelling, page numbering (putting on confidential labels when they were needed) and so forth. But the trick was to do all this with only one set-up so that you could use the same set of stuff, either dynamically (interactively) in real-time control of the data reduction problem. (This is what we did with all that big room of stuff and duplicated on the Eglin 1103 in Florida.) You'd sit there and have a display of where the actual gun aiming errors were coming out and by button you could call up any set of variables any place, look at things from different points of view, and so forth, and interact with and control this Data Reduction System as it did its job, including getting all the plots out at the end. One of the driving forces for SLURP was to extend both the Director Tape Concept (that Operating System, ICL type thing) so you could control what programs were running when, but also to be able to control all this formatting and labelling, of both displays and final output. So we designed the Simulated SLURP Computer. It ran, in simulated form (emulation, if you will), a special machine that was done by Whirlwind programming. It was a machine that had as its normal instruction set the Flexowriter codes (in other words, characters), as well as some other things that had to do with controls, jumping, testing, and so forth. The main thing was that between character streams you sprinkled control codes that would have to do with patching them together on any selected output device or feeding those character streams to programs. To do that I designed what I called the "Multi-mode Control Element" for this simulator computer, in which you had two bits that would control the mode. It had three different modes of operating. The idea was that you could extract in one mode just what was appropriate for on-line abbreviated display; in another mode you got a somewhat condensed version that would be able to fit on the scope for documentation; and the third mode gave the full version on the printer. It was all done by just selecting out of the same character stream. Using the SI instruction, (that input/output control which came with the computer) it would go on any one of the different display devices. Essentially, we'd fixed Whirlwind up with what nowadays would be called a BIOS (Basic Input/Output Systems). The idea was to have this flexibility. I remember that the report about SLURP shows that to check out the Multi-mode Control Element we used the following kid's phrase, with each word appearing only once in the character stream structure, and the entire expression being formed just by jumping around the Multi-mode Element: "The skunk sat on the stump. The skunk think the stump stunk, but the stump think the skunk stunk." So sure enough it worked. We also did a Logging System for keeping track of all the button pushing that was done through any of the interactive elements (including

the keyboard) and a Log Playback System, so that anything that you found useful you could play back and automatically re-run the system on that portion. There was also a thing for editing logs. Every part was written, partially. Many of them were hooked together. But the whole thing was never really completed. But we were really serious about having tool support for building this big software. SLURP was also driven by this philosophy of computation that said "Do complicated things by collections of little things." I guess the only other thing out of that period of interest is what was called the "MIV Box." This also was written up in the report. The MIV Box was for Manual Intervention. Again, with all this complexity of systems and systems for building systems, and systems for analyzing the building of systems, and so forth, thinking of it in terms of language design (which I wasn't yet, if I put that view in here) the only way you might possibly keep track of all this stuff -- remember it's for on-line interactive use -- is to make the thing very highly structured and use hierarchic control. First of all, as I said in my APT History paper, there was no way we could do anything in those days without being very systematic and very structured. There just wasn't room on the machine and we were doing really smashing huge problems. So you just had to do that. You took for granted that you did that. But another way of these things was that for the very complicated controls you had to have this hierarchy or you couldn't understand it. You didn't know where you were. You couldn't keep that much in your mind. Therefore, I combined that with a lack of physical equipment (compared to today) and the MIV Box was the software solution to getting arbitrary control over arbitrary programs by means of one toggle switch (I believe it was one toggle switch) and one pushbutton. Maybe it was two pushbuttons and one toggle switch, I can't remember. It's written up. The idea was that you had a Wait Loop where, by the switch, you could control whether you were to stay in the Wait Loop or not. And while you were in the Wait Loop you could push one "activate button" which is hardware built to fire only once, and go out to do one execution of something; then you come back to the Wait Loop. That was a Minor Exit. If you wanted to go to another place where there was another wait loop (there could be a hierarchy of wait loops instances, you see), then it was a Major Exit. So with the major and minor exit buttons and the wait switch you could step yourself through any hierarchy of control, either doing stuff at that level or going to a different level. That's I think similar to what later turns out to be the small numbers (from 1-3) of buttons on the present mice that they have for controlling window displays and so forth today. The simplest mouse is like the most recent Apple MacIntosh, where they have only one button, and they make a big

thing out of it. Well, that's very simple. It just says that whatever you point at you say, "Do that," and then the display changes to show the next allowed choices. That's how you get away with one button instead of two or three. But it's nice. We used to call those things "light buttons" that you were pushing. Instead of pushbuttons, we'd call them light buttons. The trouble was that it was both expensive to put in the control -- the software --to put a light button on a display in space and time in the computer. And also we already had the physical pushbuttons and the wired lights - activate lights that came in registers at the same console. So we did many things with physical pushbuttons and lights that nowadays (and we wanted to do but could seldom could) you do with light buttons.

END OF INTERVIEW