

Over 50 Years in Computing: Memoirs of Raymond E. Miller

Raymond E. Miller

Abstract

These memoirs discuss my career in computing from 1950 when I started as a design engineer trainee with IBM through 2002 when I retired as a professor of computer science at The University of Maryland. I include many of my experiences and also some personal opinions. Readers should be aware that these remarks are my own and may differ from opinions of others.

Chapter 1: The Early Years

I was born on October 9, 1928 in Bay City, Michigan, but since my parents moved to Sheboygan, Wisconsin soon afterwards, my memories start as a young child growing up in this small Midwestern city on the western shores of Lake Michigan. My father, Martin T. Miller, was a kind and hard working man. He was a salesman for Hand Knit Hosiery Company that later changed its name to Wigwam Mills. He traveled extensively throughout five Midwestern states, getting orders for the sporting goods of Hand Knit Hosiery. My mother, Elizabeth C. (Zierath) Miller had been a secretary at Hand Knit Hosiery prior to marrying. She had her hands full raising my older brother Donald and me, as well as two younger children later, while Dad was out traveling.

The depression years were difficult, as most merchants were not buying much, so Dad spent many weeks away, often sleeping in his car to conserve money. I remember times when our dinner consisted of rice pudding with raisins, and not much more. Yet we managed to get along and actually were quite happy as kids. For many of my early years we lived in a rented flat (the first floor of a two-story house). Before I finished grade school at U. S. Grant School we moved to a house my parents bought. Their joy of owning their own home was evident even to me as a child. This was their first house, and, as it turned out, the only house they ever owned. It was a great house to grow up in: two floors, a full basement and an attic, and only two blocks from the Lake Michigan shoreline. Before we moved in Dad had it completely renovated inside and added a two car garage at the back of the yard where there was an alley. Dad had only one car, but later he bought a trailer in which we spent many summers with him as he traveled. Also,

the house was close enough to U. S. Grant School so I could still walk there for my schooling without having to change schools. Thus, I went there from Kindergarten through eighth grade before going on to High School. In those days there was no middle school between grade school and high school. Don was already three years into High School at “Central High” before I entered as a freshman. Right after he graduated he was drafted into the army, as World War II was going on in 1943. He got into the Signal Corps and was stationed in the Philippines and later in Japan. Before I finished High School in 1946 the war ended, and Don returned, so we started college together, both going to the University of Wisconsin – in Madison, of course, since that was the only University of Wisconsin at that time. There was a budding extension in Milwaukee then, and other teachers colleges in the state, but this was well before the explosion of state university systems.

Don and I, even though both majoring in mechanical engineering, rarely, if ever, were in the same classes. The university was bulging at the time with a massive influx of returning service men, so I remember even having some classes in Quonset huts that sprung up on campus to handle the need for more classrooms. One of the university student cafeterias that we often ate in was housed in a large Quonset hut. The mechanical engineering curriculum at Wisconsin had many “hands-on” courses as well as the theoretical courses such as thermodynamics and differential equations. In courses such as welding, foundry and machine shop, I got experience with the various operations done on metals. In metallurgy we etched the surfaces of various metals and studied their crystalline structures, in combustion engine laboratory we charted engine performance, and in mechanical drawing I spent many hours at the drafting table. A slide rule attached to my belt was the norm for an engineering student. A required speech course was a real challenge to me as I had never had to prepare and present material to an audience before. Although I feel that this background of material helped me significantly in my life in many mechanical endeavors, it is difficult to say that it helped me in my computing career. Don and I both graduated in June 1950 with our fresh bachelors degrees, ready (more or less) to face the world. For my parents this was a new experience. Mother had grown up and graduated from High School in Sheboygan, but Dad, being the thirteenth child of a Lutheran

minister, never had the chance to even finish eighth grade. Money was short and he had to help support himself by working in a clothing store. This was the seed to his later becoming a salesman. Don went off to Pittsburgh with Westinghouse and I went to IBM in Endicott, New York. Neither of us followed our Dad's footsteps by taking over his territory, although Dad would have very much liked that. Don became an expert engineer in vibration analysis, which later became quite important in nuclear power plant design at Westinghouse. Endicott, was known as the "Plant No. 1" location for IBM at that time. I rented a room several blocks from IBM for seven dollars per week, and started as a design engineer trainee in a class of about 35 of us learning about the IBM machines for data processing: sorters, punches, printers, collators, etc. for business data processing. No, there were no computers built by IBM at that time. However, as a mechanical engineer I was very interested in complicated machinery that did interesting things. This interest came mostly from field trips that I took from Madison to factories around Chicago that made breakfast cereal, soap and other products, and I became fascinated by the machines that produced these products, starting with vats of fluid mixes that came out the other end of these large and long machines with the products all bagged and boxed and ready for shipment, without hardly any human intervention except for checking that they were operating correctly. I thought, "wouldn't it be interesting to design such machines." IBM was a company that built complex machines, so it seemed like a perfect fit for my interests, and the salary offer of \$3,600 per year was very attractive. Although computers were already known, IBM was still in the business of making, selling, leasing and maintaining punched card machines. It also manufactured punched cards, and had quite a lucrative business selling cards. Thomas J. Watson senior, the founder of IBM, was still president of IBM at that time. I remember meeting him at Endicott. His visit was quite an event. It is said that he really didn't trust electronics, but wanted to "see" what his machines were doing. Thus, even though there was some work going on in secret corners of IBM development labs on electron tube machines, they were not thought of as projects for new products in the near future. While taking this design engineer's course I did see some more advanced machines that IBM had: the CPC, Card Programmed Calculator, and a storage unit

about the size of a large hassock that contained numerous relays for storage. Obviously its storage capacity was in the hundreds of bytes, not megabytes. The IBM Engineering Laboratory at Endicott had reportedly built the Harvard Marc I machine for Howard Aiken's laboratory at Harvard. It was a large computer type machine that used a loop of wide punched paper tape to instruct it in its operation. If you looked closely at the back of the Engineering Laboratory you could see the outline of removed and replaced bricks that showed how this large machine had to be removed from the building to be shipped off to Harvard. Neither the CPC nor the Marc I machines could be called general-purpose computers, however, as they really were not truly programmable. The first computers that the public became aware of were those used in national elections aimed at keeping track of the voting, and used for predicting the election outcomes. They were called UNIVAC's, not computers, after the first really commercial venture in building computers. The closest thing that IBM had at the time that one could denote as programmable were the machines that had program plug boards that one would place plug wires into to sequence certain machine operations for short loops. However, this was to change quickly. It was thus on August 7, 1950 that I started taking the design engineer class at IBM to which I attribute the start of my computing career. This training course was scheduled to last about one year to prepare us for our IBM careers in the development laboratories or in the field for maintenance and repair. IBM had an Education Building close to where our class was held, and one evening John Holland, who was in this class with me, talked me into attending a lecture that was being given there by a professor from Harvard. The professor lectured about an esoteric kind of algebra that struck me as having very strange properties and of no interest whatever. Having thought about that later I'm sure that he was lecturing on Boolean Algebra. It's interesting to see how one's perspective changes! Before 1950 was over IBM decided that the class needed to be terminated. There was too great a need for engineers in their laboratories, so we were dispatched to various IBM locations. Along with some of my classmates I was sent to Poughkeepsie, the "Plant No. 2" location, to work in the development labs there. Thus, my initiation as an actual design engineer in data processing (computing) was commencing.

Chapter 2: Initial Tastes of Computing

Upon arriving at IBM Poughkeepsie it seemed like they really didn't have a process in place to assign new inexperienced design engineers to projects. Although I may have had interviews with various group managers, I don't remember this, but rather remember just being assigned to a small project housed in the Carriage House of the Kenyon Estate. This project was investigating how to speed up the movement of paper and cards through various machine stages. Paper movement was a bottleneck in the speed of data processing. Possibly it was thought that I, being a mechanical engineer, was suitable for understanding the mechanical processes involved in paper movement. In any case, I worked with this group briefly, seeing how rolling wheels and belts could be designed, how brushes might help remove static electricity, and how airflow might help to speed the paper movements.

Soon it was discovered that I had another skill that could be useful; mechanical drawing. That should have been no surprise since mechanical drawing was a common requirement in mechanical engineering curricula. So I was transferred to work on a large new project with a firm deadline that IBM had contracted for with the United States Government, called the "Defense Calculator". This assignment really kindled my interest in computing of a nature that I was hardly familiar with at that point in time, and was much closer to general purpose computing as we know it today. The project occupied much of the Kenyon Estate offices as well as another IBM facility in Poughkeepsie called the "High Street Laboratory". This laboratory was on the second story of a factory building on High Street. The first floor was a tie factory, and often during the lunch break IBMers in the Laboratory would frequent the tie factory outlet that sold some of their ties – probably seconds. I was asked to design the "main frame" of this defense calculator. It was an electronic machine with vacuum tube modules fit into the main frame for the control and arithmetic functions as well as other tube modules for the memory functions. I designed the frame to hold these modules, ensuring that they could be accessed for maintenance, and had sufficient room for airflow to keep the modules cool. The frame included a swing-out door for module access with a swivel wheel that was spring loaded to roll over an uneven

flooring. Cost was not a factor. The frame itself was of cast aluminum for good electrical grounding properties. Thus, I designed this frame and drew the mechanical design drawings. Don't try to find my name on any such drawings if any still exist, however, as I was not permitted to sign them. Signing was done by the project manager, either for authentication reasons or, more likely, to be able to claim credit for the work. Nevertheless, some time later I saw pictures of the IBM commercial version of this machine, the IBM 701, in an IBM newsletter, with the mainframe of my design, of the spring-loaded swinging gate for maintenance access, and the covers directing airflow. That was satisfaction enough for my first design experience. Connection to this project made me realize that there was a new kind of complex machine coming onto the horizon that I had to learn more about so that I could really get involved with this new way of processing information.

Chapter 3: An Interlude: The Korean Conflict

My work in Poughkeepsie was cut short in May 1951 when I was called to report to active duty in the Air Force because of the Korean Conflict. I had taken Air Force ROTC at the University of Wisconsin. ROTC was required as part of the curriculum for two years for all male undergraduates that had not had military service, and I had continued this for the final two years to earn a little money to help pay for college room and board as well as the “exorbitant” university tuition of \$180 per semester. Thus, I ended up with a Second Lieutenant reserve commission in the Air Force. Even though I had not joined any reserve unit after graduation, this was the reason I was being called up to active duty. IBM, being a strongly patriotic company, had no interest in requesting a waiver for this call to duty, even though they had a shortage of engineers. So I left New York to report to Lackland Air Force Base, Texas on military leave from IBM. Since I had been with IBM for less than one year, this meant that I no longer would receive any pay from IBM, but military leave meant that I was guaranteed a job with IBM when I was discharged.

Upon reporting to Lackland Air Force Base I found that I was one of several hundred newly reporting reserve second lieutenants, all of whom had to be screened and assigned to various duties. This would require some time. I’m sure you’ve heard the military saying, “Hurry up and wait.” Well, that’s exactly what the situation was. I, along with many others, had to be housed “off base”, and in my case in a small motel close to San Antonio. We were all required to report into base every morning, but had no assigned duties. This went on for a month before I received assignment orders. The month was not wasted, however. There was a base golf course and an officers’ swimming pool, so along with one of my new friends I tried out the golf course, learning on my own, and have turned out to be a lousy golfer ever since. The pool was OK as well, but with that hot Texas sun I had to be very careful not to burn too badly. Eventually I received my assignment. I was to report to the Air Force Research and Development Command at Wright-Patterson Air Force Base near Dayton, Ohio. The “Wright” stands for the Wright brothers of Dayton who designed, built and flew their plane at Kitty Hawk, North

Carolina.

At Wright-Patt I was assigned as a project officer on a bombing computer project in the Armament Laboratory to oversee the transformation of an analog bombing computer into a digital computer. The project involved getting this digital version installed in an old B-29 bomber so that it could be flight-tested. After installation, which finally was completed at the end of 1952, I was assigned to fly along to Eglin Air Force Base in Florida to observe the test runs there over a test bombing range. At least the assignment to this project enabled me to keep in touch with computing in some way, even though it was quite a different type of computer. The flight to Florida was not uneventful! Over the Carolinas the pilot got on the intercom telling us that he had just lost an engine due to an oil leak, and he had to feather the prop. He said this should cause no major problem, but just to be safe we should all put on our parachutes. Yet another new experience, never having had a parachute on before. Fortunately we didn't have any more trouble and we landed safely at Eglin. I was assigned to temporary officers quarters there, and now we needed to get a new engine installed in the B-29 before bombing tests could be started. Another month of waiting for a new engine ensued. Again, I kept checking every day, but here the base golf course was really attractive. My golf game didn't improve, but I had to be doing something. It's said that practice makes perfect, but I assure you that that does not apply to golf at least not for me. Yet spending time in Florida in February is not all that bad. I never experienced any test bombing runs, as in February 1953, with the Korean Conflict over, I was discharged from active duty. With these experiences on active duty you might understand why I refer to this period of time as my service in the "Air Farce".

There were other interesting aspects to this period of my life, however. While at Wright-Patt I lived some of the time in a rooming house in Dayton. I met my future wife at the Lutheran Church in Oakwood, (a suburb of Dayton) that I attended. I like to say that she fell for the uniform, but I rarely wore it when off base. Also, the bombing computer project conversion to a digital machine was under contract with IBM, so I paid some visits to the IBM facility in Vestal, NY where the work was being done, and this was adjacent to Endicott where I first started with IBM. My role was quite

different now, however, as a member of the Air Force.

Now, what to do upon getting off of active duty? Rather than return directly to IBM, I decided to try to pursue my desire to learn more about computers and computing. I decided that this could best be done by enrolling as a graduate student at some University that had an activity in computing. I wrote to IBM and asked for an educational leave for this purpose. Although IBM apparently did not have a formal educational leave policy at that time, I was granted a leave in the form of a letter from an IBM Vice President (McDowell, I believe). The stipulations of the leave were that I was to stay in good standing at the university of my choice and that I would return to IBM shortly after finishing my education. In this case my time away would be counted as time at IBM toward all IBM employee benefits, except for one – the IBM Quarter Century Club. The question still remained as to where I could enroll to get more computer education. There were only a few universities that had ongoing computer projects. This was long before there were Computer Science Departments or even the term “Computer Science” or “Computer Engineering”. I was accepted to enroll as a graduate student at the University of Illinois in Urbana-Champaign. They had a recognized Digital Computer Laboratory directly associated with their Graduate School with Professors in Physics, Mathematics and Electrical Engineering associated with this laboratory. The laboratory had recently started to run the Illiac I computer, a machine that they had designed and built. Illiac I was modeled after the Johniac computer at Princeton, named after John von Neumann, the leader of this landmark programmable computer project. Prior to building Illiac I the laboratory had constructed a similar computer for the Army Ordinance Corps called the Ordvac, and I believe the laboratory really got its start through this funded project.

Chapter 4: Graduate Student Days at Illinois

I enrolled in February 1953 as a Mathematics major at the University of Illinois. The mathematics courses that I took during my undergraduate Mechanical Engineering program at Wisconsin, including several electives that I had chosen to take, were sufficient for my graduate mathematics standing. However, I didn't qualify for graduate standing in Electrical Engineering as I had not taken a number of required undergraduate electrical engineering courses. So I started working on a Masters in Mathematics, taking as many numerical analysis, Boolean algebra and logic courses as possible that were taught by professors in the digital computer laboratory.

I still had some issues to work out. How was I to fund my way through this education? I no longer had my military pay, nor any from IBM, and only modest savings. I also wanted to become more closely associated with the Digital Computer Laboratory. I applied for and received a graduate research assistantship in the laboratory and also applied for GI Bill monies for education. I really didn't feel as though I had earned the right to get GI Bill assistance, but it was there and would help me enormously. With these sources of income I could make my way. My research assistantship was for 20 hours per week to work on laboratory development projects. The funding was not for PhD thesis research as is common today. Yet, this provided tuition waivers in addition to satisfying my desires to get as much experience as possible in computing. I also wanted to get further education in electrical engineering since many of the graduate courses in computing were graduate electrical engineering courses. After all, these were called "electronic digital computers" at that time. So, I went to speak to the engineering dean, Dean Elliott, to see what courses I would need to get a bachelors degree in electrical engineering. Fortunately the requirements were not overwhelming, so I started to take these classes along with my graduate classes in mathematics. This meant taking 20 or 21 credits in a normal semester along with my 20 hours work as a research assistant. Needless to say, I didn't have much time for leisure. I rented a room in a home close to campus whose owner was working on his PhD in Library Science at the University. Not wanting many distractions from my studies, the TV that I had purchased used while

in the Air Force in Dayton, and modified by installing a larger TV tube and associated circuitry, I convinced my parents to let me install in their home in Sheboygan. This was the first TV in their home. They weren't really sure that they wanted a TV, but I didn't want the distraction as a graduate student. My sister, Ann, got much enjoyment from it and my parents soon became interested in some programs, so I felt that this had served a good purpose. The closest TV stations were about 60 miles south in Milwaukee, so an antenna was needed to get reasonable reception. I made the antenna out of lead-in wire that I nailed to the attic rafters along with copper wires for directors and reflectors, all directed south toward Milwaukee. The antenna design I found in a book at the Sheboygan library that provided me with the correct spacing and sizes of the antenna pieces, and I connected this via lead-in wire from the attic, outside, and then into the living room. The reception was quite good, since most of the direct line of sight to Milwaukee was over the waters of Lake Michigan. In addition, reception from some Green Bay stations was also possible. Cable TV was still many years away from being conceived. As time progressed this antenna was replaced by a commercial antenna in the attic, and later still the TV was replaced by a color TV.

But this is a digression; back to Illinois and my graduate days: I went through a series of room rentals, sometimes jointly with another student. For a year or two Gernot (Gary) Metze and I roomed together in a small room complex. He was an electrical engineering graduate student from Austria who also worked in the digital computer laboratory. He was quite musically inclined and used this talent to design a music program for Illiac I as a fun project. No high fidelity here, but the Illiac had a speaker attached to the sign bit of the accumulator register that allowed the machine operator to tell if the machine was running, and even could be used with a trained ear to tell if a program was stuck in a non-ending loop. Gary used this speaker to produce tones of the scale by programming the right frequencies of sign bit changes. I remember a lab Christmas party when we all stood around the machine floor listening to Christmas carols that he programmed. I thought they sounded great, but Gary agonized over the tones being somewhat off key, claiming that the machine was operating slower than normal. Since Illiac I was asynchronous, and it's speed not controlled by clocks, the speed would

vary due to voltage variations and other factors.

By 1954 I earned my B.S. in Electrical Engineering, and on June 18, 1955 I received my Masters in Mathematics. That date sticks in my mind because that is also the day that I married my wife Marilyn in Dayton Ohio. I like to say: "That's the day I got my third degree." Shortly thereafter we moved into an apartment in the University owned "Student-Staff Apartments" on West Green Street, a short distance from the campus that I had rented shortly before. Rent was affordable, \$70 per month for an efficiency apartment that included a kitchen area, a living room with furnished sofa bed, and a bathroom. I added a desk and an end table that I had designed and built. We quickly added other necessities.

Originally, on entering graduate school my intentions were to get a masters degree and return to IBM, but with urging from several of my professors in the digital computer laboratory I agreed to stay on and work toward getting a PhD. By keeping my research assistantship and having a bachelors in electrical engineering I could continue on toward a PhD in electrical engineering, which at that time provided more computing options than continuing in mathematics. After a summer session at the University we took the month of August off for a camping trip in Canada above Minnesota. This served as a substitute for a honeymoon, sleeping in a tent with cots and fishing almost every day. Then back to Illinois for more classes and worrying about PhD requirements. One of these requirements was to pass two foreign language-reading exams. Since my foreign language skills were nonexistent, the only way I could do this was to take special language courses designed for this purpose. I chose French and German, and took the French course the next summer along with another student friend, Bob Collier, from across the hall in the apartment building. This course was taught by a graduate assistant in the French department. I found him to be a terribly arrogant person with little regard to our needs, and decided to sign up for the French exam as soon as possible during the summer. By studying hard, and practicing past exams in the engineering library, I remarkably passed the exam and immediately dropped the course. I felt ecstatic by getting out of that class, but really never learned much French, and Bob Collier never forgave himself for not doing the same. German was harder, with all those compound words

and different sentence structures, so I took the full course in German, studied a book designed around learning seven standard sentence structures, read sample passages from previous exams, and finally, with the help of a German-English dictionary at my side, passed the German exam. It's good I was in engineering studying computing, as I never would have gotten anywhere in Foreign Languages! After finishing sufficient coursework Dr. Ralph Meagher, the director of the Digital Computer Laboratory and a Professor of Physics, became my official thesis advisor. However, Dr. Franz Hohn, a mathematics professor with a courtesy E.E. appointment was my actual advisor. I had taken several classes with him and he had written a short book on Switching Circuits and Boolean Algebra. I received an RCA Fellowship through the help of the Electrical Engineering Department for the 1956-57 school year to work on my thesis. During that year I did my thesis research and writing, sometimes long into the night. Fortunately, the Digital Computer Laboratory had acquired a house on the edge of campus just one block from the Student-Staff Apartments as an annex for some of their graduate research assistants, and they let me occupy a desk there for my thesis work. I finished my thesis that year, with the help of much coffee, and was awarded my PhD in June 1957. I'm not much for ceremonies, so I didn't attend the graduation, but just heard my name announced over the radio while sitting in an A&P Supermarket parking lot before going in to shop. My thesis was on bilateral switching circuit design, from which I published my first journal paper in the IRE Transactions on Electronic Computers in September 1958. This was slightly before the Institute of Radio Engineers (IRE) merged with the American Institute of Electrical Engineers (AIEE) to form the Institute of Electrical and Electronic Engineers (IEEE).

My graduate research assistantship work from 1953 through 1957 provided me with many opportunities and experiences. I'm still convinced that this provided me with a much deeper understanding of computing than I could have received through course lectures and course laboratories alone. It complemented the coursework perfectly. As a mechanical engineer my first assignment at DCL, the Digital Computer Laboratory, was to modify the type bars of a teletype printer so it could be used as the main output device of Illiac I. I still have copies of the documents I produced describing these

modifications. Later I worked on the Illiac I primary input device, a DCL designed optical tape reader which read the programs and input data from teletype punched paper tape into the machine. This reader was considered to be very fast for the time, much faster than commercial electro-mechanical readers. Once, while debugging one of our readers, I got a little too close looking into it, but quickly pulled away due to a shock on the tip of my nose. Illiac I was also considered to be a very capable and reliable machine. The addition unit was built around a digital adder design which was different than previous analog adders. An addition of two 40-bit words took around 200 microseconds, and there were 1,024 words of storage in its Williams Tube memory. The Williams Tube memory consisted of 40 cathode ray tubes each with a copper wire screening glued to its face for capacitive pickup. Each tube stored 32 by 32 bits in a raster on the face of the tube, one tube for each bit of the 40-bit words. The 0 or 1 was recorded and read off the tube by exciting the bit location and moving the electron beam slightly to record and read either a 1 or a 0. Programs were developed in machine language and then laboriously punched up on teletype paper tape for input to the Illiac. Some programs were run many times and a large subroutine library was developed for many common numerical function computations. This led to the need for multiple copies of tapes and the integration of these subroutine tapes into the total master programs.

One of the people that I got to know, who used the Illiac in running programs for his own research, was Bill Atchison. Bill later went to Georgia Tech as Director of their Computer Center, and then later to Maryland in their Computer Science Center, and he was one of the original faculty members of the Computer Science Department at Maryland. Even though Bill left Georgia Tech before I went there, Bill and I met each other at ACM functions and later at Maryland when I moved there in 1989.

After learning some things about logical circuit design and Illiac logical circuits from Jim Robertson, David Muller, Franz Hohn and other faculty at Illinois, I was asked to design, supervise the construction, and test the operation of two special purpose machines to reduce the labor of producing and duplicating program tapes. These machines were called a tape perforator and a tape comparer. The first was the tape perforator, which would prepare a

duplicate tape. Since a program tape that was used many times would deteriorate, the tape perforator could be used to quickly and conveniently produce a copy of the tape prior to it being damaged, thereby eliminating the need for reproducing the tape by hand punching it character by character. The second machine, the tape comparer, could then be used to verify that the new tape was identical to the original. These machines had additional features controlled by switches on switch panels that would allow the tape perforator to combine subroutine tapes with one another and the specialized program tapes into a single master tape, and allow the tape comparer to be sequenced character by character and in other ways, again saving much re-punching by hand. As I went about designing these machines I used the Illiac I logical circuit design components for the design, the Illiac I designed optical tape readers for inputting, and an electrically operated tape punch for producing the duplicate tape. Once designed and built to my specifications, the tape perforator would not work correctly. Thus, I spent many hours trying to debug the design. I could not find any design error, and was convinced there was something else causing the problem. Indeed, the basic logic design was correct. However, there was a timing problem, which I found after checking the signals using an oscilloscope. This was readily fixed by inserting two inverter circuits in series into one of the lines, thus producing sufficient delay in this line to provide the correct timing. Since Illiac I used asynchronous logic such a trick would not normally be necessary, but the tape reader and punch were synchronous devices and their connection to the asynchronous circuitry required correct timing; a lesson well learned. The tape perforator and comparer circuitry and controls were constructed to fit into standard circuit racks with attached shelves built to hold the readers and punches. These two machines were located in a tape preparation room right next to the machine room and used for many years thereafter. As a first real logical circuit design experience, the process from design to successful operation of these two machines was very satisfying for me.

Other developments for Illiac I were also going on at that time in the laboratory. A group from the University of Sydney in Australia came to spend some time at the lab, study the Illiac I, make copies and slight modifications to its design documents, and take them back to Sydney to make their own

machine. In honor of this Illiac I heritage, and keeping the tradition of naming machines, they named their machine "Silliac". Illiac I was being upgraded; A cathode ray tube display was added as an output device to provide a basic form of graphics display. As an extension to the 1024 word Williams tube memory a magnetic drum memory unit was also designed for Illiac I. The logical circuitry was designed in the lab, but the drum was commercially built by a company in Minneapolis, Minnesota. I was asked to travel to Minneapolis to oversee the final testing of the drum at the factory prior to its shipment to the laboratory. During this testing some crosstalk between tracks was detected in the tests. This worried me. After checking back with Dr. Robertson, however, it was decided that the level of crosstalk was tolerable, so we accepted the drum for delivery. When it arrived at the lab it was my task to record timing tracks on the drum to indicate origin, word and bit boundaries. I spent weeks recording and rerecording the bit track, using a tunable oscillator, until a good match of the signal recorded was obtained to join the start and end of the track with the correct number of bit oscillations. The origin and word tracks were easily recorded following the bit track being recorded. After we got the drum and its circuitry all installed and operational we had the phenomenal addition of 32,000 more words of memory!

The original Williams tube memory for Illiac I was quite reliable, but there was one weekend during which we all participated in trying to locate an intermittent failure that was occurring in the memory. This involved an intense weekend of memory module replacement, one at a time, until finally the errant module, that is, the one out of 40, was located. Had the failure not been intermittent it would have been much easier to locate. One other evening Jack Nash, a mathematics professor associated with the lab who at some point became lab director, held a birthday party at his house for Gene Golub, who was a mathematics graduate student and a laboratory research assistant. This was a very special occasion since Gene's birthday was February 29th so he had had only a few real birthdays in his life. Some of us banded together to buy Gene a car; his first car, as he was from Chicago and never needed a car there. We purchased a very old used car for fifty dollars and placed a strip of aluminum foil over the roof to make it look similar to the Ford Crown Victoria of that time that had a chrome strip on the roof.

Gene was impressed by a new Ford Crown Victoria that Ted Poppelbaum, one of the lab professors, owned, so that was the reason for the aluminum foil. We considered this to be quite a good joke gift, however Gene was somewhat overwhelmed and to our concern used the car thereafter for many trips to Chicago. To our amazement the car held up well and he never had a serious accident or problem with it. Somewhere around this time I became aware of the book “Automata Studies” edited by C. E. Shannon and J. McCarthy, of the Annals of Mathematics Studies, Number 34, published by Princeton University Press in 1956. This book had many very interesting papers on automata. I was particularly interested in Ed Moore’s paper on Gedanken-Experiments on Sequential Machines and J. von Neumann’s probabilistic logics paper, but there were other famous authors who published landmark papers in this volume as well. This book opened my eyes to many other things that were beginning, on the research front at that time.

In addition to the Illiac I upgrades that were taking place, another much faster and powerful machine called Illiac II was being designed at the lab. This machine was based on a much more asynchronous concept developed by David E. Muller that he called “speed independence”. A fat laboratory report, published in 1958, describes this machine design. I had taken courses from Professor Muller and was quite intrigued by his formal theory of speed independence. Thus, when I had finished my PhD I was appointed as a Research Associate from June 1957 to September 1957 and worked on the logical design of some speed independent circuits. Somewhat after I left DCL at Illinois they got Illiac II operational. I remember receiving an empty envelope from them. The only indication of what it was about was the official University of Illinois machine postmark, which stated that some large number was a prime, a Mersenne prime. This was the largest known prime at the time and had been computed by Illiac II using a prime number detecting program as a background job during otherwise idle time for Illiac II. This was indeed a novel way of publishing! They designed two more machines subsequently, Illiac III, a graphics and picture processing machine design project led by Bruce McCormick, and Illiac IV, a high-speed parallel machine. The logic design of Illiac IV was done at the lab, but with integrated circuits for implementation this needed an industrial partner for construction, since

integrated circuit fabrication facilities were beyond the scope of a university laboratory then.

As I was completing my PhD I also started to look around for research job opportunities. I interviewed with RCA, Sperry, Bell Labs at Murray Hill, N.J., IBM Research and several other industrial labs. I wasn't interested in a faculty position at any university at the time as I didn't see myself as a professor or teacher. I was bent on industrial research and development, and job opportunities were quite strong. I got numerous offers, both in development and research, but since the IBM offer was quite good, and they were aggressively building their research operation then, modeled much after the AT&T Bell Labs structure, and separate from their development labs, I decided to rejoin IBM. I joined J. P. Roth's group, having had a very interesting interview with him on a shuttle ride from Poughkeepsie to Westchester County, N.Y. where the new IBM research laboratory was to be built. Paul Roth's group was working on switching circuits and logical design, so it fit my interests perfectly.

Chapter 5: Return to IBM

In September 1957 Marilyn, I and our young daughter Patricia, left Illinois traveling to New York so that I could join Paul Roth's group called "Switching Research". Computing theory to me at that time meant switching circuit theory and logical design. Even though the Automata Studies book was familiar to me, I had not had any classes that covered automata theory; so Turing machines, undecidability and the like were still unfamiliar concepts. Paul Roth's group included Eric Wagner, Jim Griesmer, and somewhat after I joined, we hired R. M. (Dick) Karp who had just finished his PhD at Harvard. Also, Michael Rabin, of Scott-Rabin fame, spent time consulting with the group. I continued to do research in both combinatorial and sequential circuits and participated in writing many reports (which we called SR-Reports) with members of the group. We were housed in the main estate building of an old estate known as the Lamb Estate on a hill overlooking the Hudson River just north of Ossining, N.Y. in Westchester County. IBM had purchased property in Westchester County for a new research center being designed by Eero Saarinen. Thus, we could be near this new laboratory and be able to move in when it was completed without having to go through a housing relocation. Other IBM research groups occupied the main house on the estate, as well as ancillary buildings. Also, some other research groups were in Ossining itself in a place called the Spring Street Laboratory. Nathaniel (Nat) Rochester was department director for the Information Research Department in which Switching Research was one of the groups. Nat had been one of the principal managers of the Defense Calculator project that I mentioned earlier. Later Herman Goldstine, who had worked with John von Neumann on the Princeton machine, took over these and additional lab activities and this department became the Mathematical Sciences Department. Also, Emanuel Piore became Director of all of IBM Research. These were exciting times. IBM was investing heavily in research and trying to build its research structure much like AT&T Bell Laboratories; the most well-known and renowned industrial research organization. Dr. Piore, who was a physicist, reorganized the laboratory along traditional discipline lines; Mathematical Sciences, Physical Sciences, and the like, but also slanted it to-

wards IBM activities with Computer Sciences and other applied departments. He also emphasized interdisciplinary activities and stressed hiring only the highest quality candidates. For several years I served as an IBM Research recruiter for the University of Illinois. This enabled me to get back to the campus and the Digital Computer Laboratory to talk to my past professors and their best PhD students, who were completing their PhD's, to see if I could interest them in applying to IBM Research. I remember Dr. Piore addressing the IBM Research recruiters at a recruiter's luncheon held before we fanned out on our various recruiting visits one year. One thing that he said really struck me; a near quote, "Go out there and find people better than you are." How well I remember that! After all, having just gotten my PhD recently I thought I was pretty good, so I nearly took this as an insult. However, as time went on I realized how appropriate his comment was. Certainly hiring only the best people is the correct formula for building an excellent research organization. Keeping this in mind helped me enormously later as I recruited people for IBM, and later Georgia Tech.

Working at the Lamb Estate was a real delight. We had offices in this sturdy stone constructed building, which was built as an elegant residence for a prominent doctor, Dr. Lamb, and his practice of curing rich patients, primarily from New York City, from their alcohol addiction. Thus, the various suites of several rooms each had a separate bathroom and thick walls that provided a quiet atmosphere, with windows looking out onto lawns and nearby woods. The ancillary buildings apparently had been constructed by some of his wealthy patients who frequently returned for yet another cure. After Dr. Lamb's practice ended, and before IBM leased these facilities, they had been used by the Maryknoll Sisters from a nearby Catholic convent, so this led to an interesting history for the estate. Following our IBM occupancy, the Hudson Institute moved in as the next occupants of this facility. During our lunch breaks we often took walks in the adjoining woods, or threw frisbees on the lawn. Shmuel (Sam) Winograd started there as a summer student employee, and Michael Rabin proudly showed off his new VW Beetle that he purchased for \$595. On one occasion a group of us drove up to Poughkeepsie to attend a presentation about the new IBM Thomas J. Watson Research Center given by architect Eero Saarinen in the large au-

ditorium there. This was held in Poughkeepsie since the research physicists were still in Poughkeepsie then, along with other research groups, and this was the largest auditorium that IBM had in the area for such a presentation. The auditorium was packed, and Saarinen went through a detailed presentation including some demonstration models of how the offices and laboratories would look. One of our colleagues from the Mathematical Science Department asked him why the offices were not going to have any windows. His reply was that he had visited many research laboratories to understand their architectural needs and noted that invariably when he found labs that had windows he found that the researchers had drawn the blinds, and thus he believed that windows were actually not desirable. He then went on to say that since there were windows all along the front and back of the building you could easily get a view of outside by simply stepping into the hallway to see outside. All of us were very disappointed as we really appreciated the windows in our offices at the Lamb Estate, and realized that this ideal environment was destined to be only temporary. I felt that the reason for no office windows was actually quite different, however, since without office windows a much broader building could be designed, thus reducing the overall cost.

Of course our main activities were those of research. One of the projects that we spent several years developing under Paul's leadership was the logical design for cryogenic switching circuit devices called inhibitors. IBM Research had started a large project developing superconducting circuitry in an attempt to use these devices to build an ultra-fast computer. Thus, we were looking into how these devices might be used for the logical design of combinatorial and sequential circuits. We wrote a number of reports on various design concepts using inhibitor logic, and received several patents on our design techniques. We also considered the problem of how such cryogenic machines might be built and maintained even under the possibility of circuit failures. Circuit failures would be very difficult to overcome since these circuits would be immersed in liquid helium during normal operation. So it would be impossible to just go in and replace a failed device with a new module. Also, if repair meant removing the circuitry from the liquid helium to repair the failure, and then return the circuitry to the liquid helium for operation again, this process itself could cause additional failures due to the

heating and re-cooling stresses. We presented our results on reliable designs for these conditions in a conference paper, “The Design of Digital Circuits to Eliminate Catastrophic Failures” by J. H. Griesmer, R. E. Miller and J. P. Roth, and the conference papers were later published in the book, “Redundancy Techniques for Computing Systems” published by Spartan Books in 1962. Unfortunately it was later discovered that this cryogenic circuitry experienced unexpected degradation in speed when interconnected through numerous circuits, so after spending considerable effort and time, as well as money by IBM, these activities were unsuccessful and the project was terminated. It was many years later that IBM Research again undertook another large superconducting computer design project, using different types of cryogenic circuitry, only to fail again at producing commercially useful results. But by this time I was involved in other kinds of research.

My main research interests remained in switching circuit theory and logical design through the early 1960’s. I worked on sequential machine state minimization and asynchronous circuit design techniques, and occasionally worked on algorithms for some combinatorial and graph problems that arose in these studies. While at Illinois, Gary Metze and I had done some work on a matrix representation for sequential machines, and we wrote a paper on this with Professor Sundarum Seshu, which we published in the IRE Transactions on Circuit Theory in 1958. I also looked at a circuit layout problem that arose from the cryogenic circuit configuration of that first IBM cryogenic circuit project. It was a neat combinatorial problem of trying to see how such circuits could be layed-out on a rectangular board to maximize the number of useful circuits that could fit onto such a circuit board. This resulted in a paper in the IBM Journal of Research and Development with John Selfridge in 1960.

Having kept my contacts with Illinois through my recruiting visits there, and because of my continued research on switching circuits, I was invited back to Illinois for the fall semester of 1960 to teach an advanced undergraduate/graduate class on switching circuit theory and logical design. Fortunately IBM Research had quite a liberal “sabbatical” policy then, so this was easily arranged. I refined my notes that I had already started, taught the course, added more to these notes, and this later became an IBM Research Report

(RC-473) in June 1961. This later led to me being invited for another IBM sabbatical to Cal Tech Electrical Engineering Department in Pasadena, California for the 1962-63 school year where my title was Visiting Senior Research Fellow. Although this was a research title, I was expected to teach a three quarter sequence undergraduate course on switching circuits there, and this gave me the opportunity to write more on the subject. Before going, I decided to write notes while there to publish a book, and the publisher John Wiley and Sons gave me a contract for the book. The department assigned me a secretary for preparing the course notes so I decided to write course notes for handing out at each class period on Monday, Wednesday and Friday. This secretary was kept busy for the full school year as the process of preparing notes for the students required her to prepare ditto masters and then run off the 30 to 35 copies for the students and have them ready for each class. We didn't have computer word processing or copy machines at that point in time, so the process of document preparation was much more cumbersome than what we are accustomed to now. Indeed, the computer owned by the department was a Librascope LGP-30, an early magnetic drum machine. A graduate student at Cal Tech from Mathematics sat in on much of my course. Even though he wasn't officially taking the course, he was clearly the best student in the class. This student was Don Knuth, who became quite famous in our discipline later through his excellent book series called "The Art of Computer Programming", as well as his Tex word processing program. The note writing kept me very busy also as I was now including many new topics and learning about them from technical papers as I went along. Nevertheless, these notes became the basis for my two volume book "Switching Theory" that was published in 1965. Volume I was on Combinational Circuits and Volume II was on Sequential Circuits. Although I did the bulk of the work while at Cal Tech, there was much work remaining for me to complete at nights once I returned to IBM. This included the addition of exercises, references and reference notes at the end of each chapter, several additional chapters on asynchronous circuits, and a labor-intensive index preparation task using 3×5 cards. These two books were adopted at various universities for their courses in switching theory and logical design, and I used them for several years in my teaching as an adjunct professor at several universities

within easy driving distance from IBM Research.

About the same time my two books were published, Mike Harrison, of Berkeley, published his book intended for similar courses. Even though our books competed with one another for course adoptions, the books were quite different in style and coverage, and thus actually complemented each other in some ways. My writing of the two switching theory books more or less culminated my research efforts in switching circuit theory and logical design; however, some interesting things occurred subsequently due to the appearance of the books. One day, “completely out of the blue” I received a package from Russia containing the Russian translation of the books, with an inscription to me from the person who had done the translation. Some years later, on a trip to a conference in Russia, I asked if there might be some royalties for me for this translation. This was questionable since Russia did not participate in international copyright agreements at that time. Nevertheless, my Russian host said, “Yes, indeed, there should be some rubles somewhere in a Russian bank.” However, I never did find out where. On another trip, this time to give an invited address at the Fifth Yugoslav International Symposium on Information Processing, in Bled, Yugoslavia in October 1969 I was approached by a professor from Ljubljana University who said that he was using my books in his course and that he would like to be my host while I was there. He was a very gracious host, taking me to see many of the area’s sites. On one of these trips he mentioned that he had a sister who lived in the United States. I asked him where, and his answer was – a small town that I had probably never heard of. After some coaxing he said it was Sheboygan, Wisconsin! When I told him that was my hometown he almost drove off the road in surprise. After that, he took me to meet his family so that I could take some pictures to send to his sister when I got back, along with a letter that he wrote to her. They had tried to send pictures to her from Yugoslavia previously, but they never got to her, so he was delighted that I would be able to do that for him. There were many other times that people mentioned my book to me, either as having used it to teach from, or having it in one of their classes. Even as late as 1997 a professor from Cal Tech remarked that he had recently used it to find out details of speed independent circuits that I had included in one of the last chapters of the book on sequential machines.

This, of course, was many years after the book was out of print, but the notion of asynchrony was being revived for very high speed circuits in which a centralized synchronizing clock proved infeasible.

Upon returning to IBM Research from Cal Tech in the summer of 1963 I was ready to look into some new areas of research. Thus, it was quite appealing when Herman Goldstine, the Director of the Mathematical Sciences Department, suggested to Dick Karp, Sam Winograd, Larry Horwitz and me that looking into parallel computation might be of some interest. Even though this was long before parallel computation was on the minds of many, there was already a clear indication that parallelism could be used to speed up some special purpose computational tasks. IBM had a product developed for the oil industry called the “convolver box”. This was a special purpose device that could be attached to the main bus of an IBM computer to do convolutions very rapidly. Convolution was used extensively in oil exploration for the analysis of soundings taken over land and sea to show the deep structure of the earth layers and expose potential locations where oil might be found. With this suggestion we started to look at parallel computation. We designed approaches, algorithms and designs, for many different special purpose computations: parenthesis checking, macro-instruction execution, the Cooley-Tukey convolution algorithm, and others. Our design for the Cooley-Tukey algorithm even rated a footnote in their original paper “An Algorithm for the Machine Computation of Complex Fourier Series” in *Mathematics of Computation*, April 1965. We published a paper in *JACM* on uniform recurrence equations that described how parallelism could be used to speed up the numerical computation of systems of differential equations. After a while, the designing of parallel algorithms for these various tasks seemed to be getting quite repetitive to Dick Karp and me, and this led us to realize that there was a hidden approach that we seemed to be using over and over. We formulated this underlying structure, which we called “computation graphs” and published the paper, “Properties of a Model of Parallel Computations: Determinacy, Termination, Queueing” in the *SIAM Journal* of November 1966. In 1966 this was the only *SIAM Journal*, but now they publish a number of journals in specialized areas. Computation graphs proved to be quite effective for designing inner loops of computations, but were limited in not

having more general computation structures, such as conditional branching, that were needed for more general computations. This led us to a more general model that we termed, “Parallel Program Schemata”. We wrote a long paper about these schemata which was finally published in JCSS in May 1969. I continued to work on parallelism even though Dick Karp decided to leave IBM to become a faculty member at Berkeley with a joint appointment in computer science and operations research. I thought it was quite surprising that Dick left IBM at that time because within only a few more months he would have had 10 years service with IBM and thus been vested for retirement benefits. I wrote another paper on parallel program schemata that discussed some undecidability results based on a schema not possessing the property that we called “repetition freeness”, and this paper was published in the first issue of the SIAM Journal of Computing in March 1972. Yes, by that time I had learned about Turing machines and undecidability, mostly from teaching courses on formal languages and automata, at night, at the NYU Bronx Campus from 1969-72 as an adjunct faculty member. Before that, from 1965-69 I taught at the Stamford Branch of the University of Connecticut that had an evening masters program in electrical engineering. Taylor Booth, who was an electrical engineering and computer science professor at the University of Connecticut at Storrs, was the person who persuaded me to teach as an adjunct, and this led to my continued avocation in adjunct positions at NYU, Yale and Brooklyn Poly. Never did I realize that this would eventually lead me into being a professor as a vocation, since I was quite happy with my research work and all my excellent colleagues at IBM Research. From 1957 through 1980, some others that I had the good fortune of working with at IBM were Arnie Rosenberg, Larry Stockmeyer, Nick Pip-penger, Joe Rutledge, Jim Thatcher and Cal Elgot all in the Mathematical Sciences Department, as well as John Cocke from the Computer Sciences Department. One semester Sam Winograd, Willie Miranker and I were invited to teach a graduate course on parallel computation at Yale, and we went up there once a week for that, each teaching things in our own specialty. Later, I used some of my saved 13 weeks vacation to teach again at Yale four days a week; again a graduate course on parallel computation. This gave me some time to work with Dick Lipton and Larry Snyder who were Assistant Pro-

fessors of computer science at Yale. Later both Dick and Larry spent some time with us at IBM during a summer, so we could continue our research. Also Dave Muller from Illinois spent a summer consulting with us, and Rich DeMillo also spent some time there. This IBM Research summer program provided us with refreshing opportunities to work with people from various universities.

I served as manager of two different groups in theoretical aspects of computing from 1968 to 1979, and from June 1979 until June 1980 I was Assistant Director of the Mathematical Sciences Department, serving as Acting Director from July 1979 to September 1979 while our Director, Dick Toupin, was away. As Acting Director I led the Department's move into parts of a newly constructed extension of the Research Center, which had been built to handle the expansion of IBM research activities. We had been housed in a section of the original building that was designed for combined office and laboratory space, but the extension was mostly office space, so our move gave us more appropriate adjacent offices and released space for activities that required laboratories.

John Cocke and I worked on two interesting topics together. In the mid and late 1960's there was considerable research and development on techniques for optimizing compiled computer code. Fran Allen of IBM research, who had worked in John Backus's original Fortran group, was one of the leaders in these optimization efforts and she worked with John Cocke in this area. My work with John involved manipulations on the flow graph structure of programs where loops in the flow graph had more than one entry point. Usual optimization techniques, at that time, did not allow instructions to be moved out of such structures, but we devised a splitting and coalescing approach that modified the flow graph into single entry loops, thus enabling optimization to be carried out. A rather comprehensive book on optimization techniques of that period was written by John Cocke and Jack Schwartz of NYU. Even today people work on "compiler optimization" techniques as new computer organizations, such as parallel computers, provide new opportunities to optimize code and reduce running times of programs. The second opportunity to collaborate with John Cocke arose from some work that Joe Rutledge and I had done in 1965-66. Joe and I developed an approach to automatically con-

vert a sequential program into a parallel program, and we demonstrated the applicability of our approach by having a program written using our algorithm for conversion of Fortran programs. The technique produced a block diagram for parallel operation that depended on the data flow rather than instruction sequencing. An interesting sidelight to this work revolved around the status of patenting research results then. It was common patent practice to get patents on novel machine designs, but getting patents on computer programs was much less acceptable. In fact, protection on the rights for programs was usually relegated to getting a copyright, which provided much less protection. We saw that this application of the patent laws made no logical sense, since both novel machine designs and novel programs were simply different ways of expressing a computational process. To challenge this patent practice we submitted our approach to the IBM patent attorneys for patenting. This gave rise to the obvious dilemma. Who was the inventor? Was it the person who developed a new algorithm; was it the person who wrote a computer program to implement the algorithm; or was it us who had developed the algorithm that transformed the program into a special purpose machine design? As one might expect, this issue was deftly sidestepped by the attorneys who decided that our approach should simply be published in an IBM Technical Disclosure Bulletin (which appeared in April 1966) rather than trying to patent it. This provided protection for IBM, but left the basic issue open of what is, and is not, suitable for patent coverage of novel techniques. Since then some programs and algorithms have received patent coverage. However, novel working devices rather than programs still dominate the scene as patentable material. Anyway, John Cocke became interested in the “data-flow model transformation” of ours from quite a different point of view. As an expert computer architect he saw the potential of designing a computer that would use this data-flow as a method for configuring such a computer to directly execute the data-flow. We conceived several different computer architectures for doing this and published our paper, “Configurable Computers: A New Class of General Purpose Machines” in the Springer-Verlag series “Lecture Notes in Computer Science, Vol. 5, 1974”. This paper was actually presented at a conference in Novosibirsk, USSR in August 1972, and the Springer-Verlag volume consists of papers from this conference. When

we did this work we also published two IBM Invention Disclosures on the various ways such computers might be implemented. Work on “data-flow” computers was also springing up elsewhere around the same time, and there were numerous attempts to construct data-flow machines, but there were a number of difficulties in making such machines. As far as I know, no such machines that displayed efficient operation were ever made.

After several years of working on parallel computation I was offered the opportunity to spend the 1969-70 school year on another “sabbatical” from IBM Research as a MacKay Lecturer in the Electrical Engineering and Computer Science Department at Berkeley. So again we packed up the family for another cross-country station wagon trip, this time with all four of our daughters. I had flown out earlier to give a talk in the department and arrange to rent a house in Lafayette, CA, just on the Eastern side of the mountains East of Berkeley, from Professor David Sakrison, a faculty member in the department who was going on sabbatical for that year. I stayed in the Durant Hotel for that stay, which was right on the edge of the campus, and woke up very early on the morning of my talk with the noise from a circling helicopter overhead. I decided to get up and take an early morning walk. To my surprise this helicopter was not the air taxi that went from the San Francisco Airport to Berkeley and back, but was actually a police helicopter. This became apparent to me as I approached “Peoples Park” in Berkeley and walked into the big demonstration going on protesting the closing and fencing-off of the park with 10 foot high chain link fencing. I decided to quickly return to the hotel, not being interested in getting involved in this famous uprising. Later that day, as I was giving my talk on the engineering side of the campus, the demonstrations spread to one side of the campus with police using tear gas to disperse the demonstrators, but all was quiet on the engineering side of campus. UC Berkeley was on a quarter system, and since I was asked to teach special graduate courses on parallelism for only two of the three quarters, this gave me a reduction in teaching load over my usual adjunct teaching schedule of one course per semester. In contrast to my Cal Tech sabbatical, I did not prepare class notes for each class meeting, but did prepare a few notes and used numerous technical papers and references for class discussion.

Shortly after Dick Karp and I had finished our parallel program schemata paper I learned of another quite different style of representing parallelism called Petri nets. Dr. Carl A. Petri had devised this model and written his PhD thesis on it at Bonn University in 1962. Anatol W. Holt and Jack Dennis, along with others, had picked up on this approach at MIT and elsewhere and were using it to represent parallel computation processes as well as many other kinds of processes. Also, C. A. Petri headed a group at GMD in Bonn that was developing new theoretical results about Petri nets and their generalizations. Upon learning about Petri nets, I found an uncanny connection between our work on parallel program schemata and Petri nets. To prove some decidability results for our schemata we had developed a mathematical formulation called vector addition systems and shown that certain reachability questions about vector addition systems were decidable via a tree construction. This provided the basic result that led to our decidability results for parallel program schemata. I noticed that the vector addition system model also could be applied to Petri nets, and that a reachability problem for vector addition systems was equivalent to the liveness question for Petri nets. I found this relationship between parallel program schemata and Petri nets to be quite intriguing so I spent quite a bit of time looking into these interconnections. This led to me giving talks on these models, their results and their interrelationships, at MIT, GMD in Bonn, Venice, Japan, Bressanone Italy, Newcastle upon Tyne England, and elsewhere, during the 1970's. A short survey paper with many references on this appeared in the August 1973 issue of the IEEE Transactions on Computers.

Although I continued to work on parallelism through the 1970's I also worked with my colleagues on many other interesting problems during this time as well. Rosenberg, Snyder, Pippenger and I wrote a paper on optimal 2-3 trees; Lipton and I wrote on planar graph coloring; DeMillo and I wrote on synchronization primitives; Elgot and I wrote on coordinated sequential processes and Karp, Rosenberg and I devised some matching algorithms for strings, trees and arrays which was a very early contribution in pattern matching.

In late 1971 Mike Rabin, Shmuel Winograd, Jim Thatcher and I started organizing a conference to be held at the IBM Research Center featuring

invited speakers who were doing various studies in the complexity of computer computations. The conference dates were set for March 20-22, 1972, an impressive group of invited speakers accepted our invitation to present papers, and we sent out announcements of the conference with invitations to attend. Now, the IBM Thomas J. Watson Research Center is located in upper Westchester County, N.Y. near Yorktown Heights, N.Y.. However, at that time there were no suitable hotels close to the Research Center, the closest being in White Plains, N.Y. about 15 miles South. Jim and I arranged for a block of rooms to be reserved at the White Plains Hotel for attendees and chartered buses for them to be brought to and from the hotel and the Research Center. As an aside, I see from the invitation we sent out that the hotel rates then were \$23 per night for a single, and \$27 for a double. What a contrast to rates at conference hotels these days! We were astonished with the large response. Over 225 people registered for the conference, filling the research auditorium to capacity. Papers discussed various upper and lower bounds for different computational problems, but probably the most influential paper was Dick Karp's "Reducibility among Combinatorial Problems" which greatly raised interest in the P vs. NP problem. After the conference, Jim Thatcher and I edited the conference proceedings into a book entitled "Complexity of Computer Computations" which was published by Plenum Press later in 1972. The book includes an extensive bibliography for that time as well as a record of the panel discussion held at the end of the conference. Much of the panel discussion centered around what this area should be named, since it had not yet solidified to the point of being given an agreed upon name. I have long believed that this conference served as a major catalyst for the subsequent explosion of interest in what we now call computational complexity theory.

Earlier, I mentioned the visits of Larry Snyder, Dick Lipton and Rich DeMillo to the Research Center for collaborative research. The Mathematical Sciences Department actually carried out quite an extensive visiting researcher program, which stimulated new ideas and helped the vitality of our research program. We also hosted graduate students and post docs. C. K. Yap was a graduate student from Yale who did some work with me on parallelism and formal specification. I also had the pleasure of hosting David Harel

for 1978-79 and Takumi Kasai from Kyoto Japan for 1979 under our Post Doc Program. Although David Harel and I never wrote any papers together, Takumi Kasai and I wrote three papers together on parallel model issues, one published in JCSS in 1982 and the other two presented at conferences in 1979.

In 1980 I left IBM Research to go to Georgia Tech. My Georgia Tech experiences are discussed in a following chapter, but before that I'd like to recount several other episodes from the 1970's and 1980's that I found quite interesting. I was invited to the August 1972 "Symposium on Theoretical Programming" in Novosibirsk, USSR where I presented my paper with John Cocke on configurable computers. This was an invitation only conference with few foreign, i.e. non-USSR, invitees. Andrei Ershov arranged and hosted this conference, and some of these foreign visitors were J. B. (Jack) Dennis from MIT, Erwin Engeler from ETH Zurich. C. A. R. (Tony) Hoare from Belfast, M. Nivat from Paris, Jack Schwartz from NYU, Mike Paterson and David Park from England, Fran Allen from IBM Research, Robin Milner from Edinburgh and S. Igarashi and T. Ito from Japan. I arrived in Moscow from New York around 6pm Moscow time and was met by a graduate student who had been assigned to meet me at the airport, take me to a hotel so I could get some rest, and then take me to a different Moscow airport to catch my flight to Novosibirsk that was scheduled to leave around 12:30am the next morning. I didn't get much rest, however, since the hotel insisted on holding my passport at their front desk, but since the desk was closing at 9pm I would have to retrieve my passport then so I would have it to continue on to Novosibirsk. I was picked up in time to make my flight. Upon arriving at this airport and checking in at the Intourist desk I noticed Robin Milner sitting in a chair in their waiting room. He was delighted to see me, as we knew each other from a previous occasion. The main reason for his delight, however, was that he had arrived in Moscow 24 hours earlier and they did not have copies of his Intourist papers so he was not allowed to continue on to Novosibirsk. My arrival gave him renewed confidence that he would actually make it to the conference and not have to return directly to England. In our discussion, we found out that we were both scheduled to leave around 12:30am so we looked forward to sitting together and talking

during the ensuing 5 hour flight. When it came time to leave we both were loaded onto a bus to take us out to the tarmac to board the airplane. Only upon getting off the bus were we aware that we were actually on separate planes, leaving at the same time, and would thus not be flying together. That was very strange, and I've often wondered if that had been planned by them ahead of time. Upon reaching Novosibirsk I was again met at the airport and driven directly to the hotel, the "Golden Valley Hotel" in the science city right on the outskirts of Novosibirsk. "Golden Valley" certainly sounded fancy, but the hotel was rudimentary at best. The linoleum floor of my room had an open split down the center, the bed was narrow and hard, and the bath, even though private for my room, was far from luxurious, having an unpainted flakeboard toilet seat and a small square tubbed shower. Even so, the conference was very interesting and well hosted. Talks were given in either English or Russian, with pauses between sentences for translation. Dr. Ershov caused quite a laugh while serving as translator for a paper given in English, paused a moment after the speaker to compose his translation and then reconstituted what the speaker had said, but said it in English again rather than Russian. I had expected that this approach for translation, even if done correctly, would have made it difficult to present one's work, but it actually worked out very well, with speakers using the pauses to hone their next sentences.

Another occasion for Robin Milner and me to meet up was at an IBM Japan symposium on Mathematical Foundations of Computer Science in October 1976 at the IBM Amagi Homestead. This conference was hosted by IBM Japan primarily for young Japanese Computer Science professors. Robin and I were the only two foreign participants. I talked on parallel computation (naturally), and Robin on the calculus of computation. The IBM homestead was situated on a hill with a marvelous view of Mount Fuji that we both enjoyed. One early morning we were enjoying this view when the Homestead head chef came up to us and asked if we wanted to have the Japanese breakfast she was preparing for the attendees, or if we would prefer an American breakfast. Neither of us knew how to answer. Fortunately, however, she interpreted this correctly and served us American breakfasts of eggs, bacon and toast. I'm afraid that I would have had difficulty consuming the Japanese

breakfast served the others so early in the day, which included a small cooked fish staring up from a small dish, steamed rice, seaweed and green tea. Probably the most entertaining anecdote regarding this meeting, however, was their attempt to get into print in the symposium proceedings the panel discussion that concluded the meeting. Both Robin and I participated in the panel, which was tape recorded and then transcribed by some IBM Japanese secretaries. You can imagine the difficulties they had in distinguishing between speaker R. Miller and R. Milner from the speaker announcements that were given on the tape. Even though we both had the opportunity to proof-read the initial draft as transcribed, the final version still didn't come out completely right. Our names just sounded too identical as spoken by the Japanese.

Having started adjunct teaching in 1965 at the Stamford Branch of the University of Connecticut, I continued this evening teaching essentially every semester through 1973. In 1969 I switched to teaching at the Bronx Campus of NYU in the Department of Electrical Engineering and Computer Science that was headed by Dr. Herbert Freeman. Often I would go out to dinner there, before teaching, with a young assistant professor in the computer graphics area, Alvy Ray Smith III. Alvy was also involved with the theory community and was the person that drew the cover design for the SWAT Conference proceedings that appeared first in 1973 and remained the cover design for many years thereafter. At NYU I taught switching circuit theory but also automata and formal languages using an early book by Minsky, and later the well known text by Hopcroft and Ullman. One semester some of the graduate students approached me to ask if I would be willing to teach a special graduate course on supercomputers. I knew some things about supercomputers from my work on parallelism and knowing about the Model 91 IBM 360 machine design, so this sounded appealing to me. I approached Dr. Freeman about teaching this course. He was quite reluctant, saying that we first had to cover our regular courses. Since these students were petitioning to have me offer this course, and I believe that one of Dr. Freeman's concerns was that not enough students would actually enroll, I suggested to him that I would be willing to teach the course by an arrangement of NYU paying me only a certain amount per student enrolled, thus ensuring that NYU would

still make money on the offering even though I might be making somewhat less than the meager adjunct pay for a normal course. Dr. Freeman seemed shocked at the suggestion, and reacted immediately by saying, “We are not running a popularity contest here!” As it turned out, almost 20 students enrolled for the course, and I taught it. I covered the IBM 360 Model 91 as well as some CDC designs, and learned much more about supercomputers of that time in this way. As time went along, NYU sold its Bronx campus, terminating its engineering programs. Alvy did not receive tenure, but went on to work at a small graphics company on Long Island. He later moved to California starting his own company. He was quite successful and later became one of the principals in the “Toy Story” computer generated movie. I ran into him only twice since our NYU days; once at a conference in Atlanta when I was at Georgia Tech, and again in the hotel lobby of the 1997 ACM Conference in San Jose. There he came up to me saying; “I’m Alvy Ray Smith. I know you but I’m not sure from where.” I had seen him also and felt the same way. We had a good chat, catching up on the years that had transpired. He said he no longer considers himself to be a computer scientist, but rather an artist. I’m sure that’s appropriate. At that time he was a senior research fellow at Microsoft Research. That’s a long journey from being an assistant professor that didn’t get tenure, but it’s clear that he found his calling.

The hiring of Nick Pippenger for IBM Research in the mid 1970’s, right after he completed his PhD at MIT, is another experience that I remember well. Nick applied to IBM Research and had been invited for a visit with the Computer Sciences Department. I also saw his application and talk announcement, was interested, and got on his interview schedule. I was greatly impressed. He gave an excellent talk and was very well prepared mathematically, especially in combinatorics. I immediately recognized that we should offer him a position in the Mathematical Sciences Department. However, there was a barrier that had to be overcome. I was the only one in the department that had interviewed him, or even gone to his talk. Sam Winograd was our department Director, but he was in Israel at the time, and our hiring practice was to have a group of department members discuss a candidate and come to a consensus before urging that an offer be made. Without a

consensus what could be done? Well, I was so convinced that Nick would be an excellent addition to our department that I put a call through to Sam in Israel, told him about Nick, and asked him to call Peter Elias and others that he knew at MIT to corroborate my opinion. Fortunately Sam went ahead. We hired Nick and he clearly was a great addition to our efforts in theoretical computer science.

One final story before moving on to the next chapter: This occurred while I was editing papers for JACM. I received a paper that Gregory Chaitin was submitting to JACM on the complexity of computing random sequences. He was an employee of IBM Argentina in Buenos Aires; however, since this paper had nothing to do with his work at IBM Argentina he was asking that his association with IBM not appear in the publication of his paper. I found this unusual, but he apparently felt that IBM Argentina did not want to be associated with this highly theoretical work. Nevertheless, we got this paper, in fact a sequence of his papers, published in JACM, and I later had the opportunity of meeting Greg as he was attempting to return to the U.S.. It turns out that he had received his undergraduate education at the City University of New York and then moved to Argentina with his parents. With help from others we got Greg a visiting position at IBM Research, even though he had no graduate degrees, but had learned a tremendous amount on his own. This later led to him receiving a permanent position at IBM Research. His works, which are related to those of Kolmogorov, regarding the amount of information in random sequences, have become quite well known.

As should be clear from these anecdotes, and my comments regarding my professional colleagues, I have found much of the appeal of my computer science career in working with so many excellent scientists and individuals, as well as considerable satisfaction in being able to help some of them in the early stages of their careers. Pleasures have come from these many acquaintances I have made over the years, as much as in my contributions to various research discoveries.

Chapter 6: Professional Society and other Service Activities

My first association with professional activities started in 1961 when I presented a paper at the Second Annual Symposium on Switching Circuit Theory and Logical Design in Detroit, Michigan. This symposium was sponsored by the AIEE (i.e., The American Institute of Electrical Engineers) Computing Devices Committee. The first such symposium was held in Chicago, Illinois in 1960, and the proceedings of both of these symposia were printed as a single volume. My paper was an introduction to speed independent circuit theory in a session organized by David E. Muller who had formulated the speed independent circuit concept and developed the theory. These two symposia brought together many of the researchers in switching theory and logical design. After only a few years, the sponsorship of the symposium changed to the Computing Devices Committee of the IEEE, the merged organization of the IRE and AIEE, then later to the Switching Circuit Theory and Logical Design Committee of the IEEE Computer Group, the precursor of the IEEE Computer Society. As the IEEE organization was crystallizing after the IRE/AIEE merger, the Computer Group became the Computer Society and the Committee became the “Technical Committee on Switching Circuit Theory and Logical Design”, which later changed its name to “Technical Committee on Switching and Automata Theory”. I had joined the IEEE and ACM in 1957-58 so that I could receive the two major technical journals of that time; the IEEE Transactions on Digital Computers and the Journal of the ACM. Little did I realize how involved I would become in these two organizations.

Nearly from the beginning, through my switching theory interests, I participated in the IEEE/CS Technical Committee on Switching Circuit Theory and Logical Design. By 1969 this committee had changed its name to Switching and Automata Theory (SWAT). Their annual conferences, of the same name, had become a premier place to present research papers in theoretical computer science. I assumed chairmanship of this committee for 1969-1972. Somewhat before then I had become concerned that the paper acceptance rate for this conference, based on submitted extended abstracts, was getting so selective that quite a few excellent papers were being rejected from the

conference simply due to the limited number of slots available for paper presentation. From experience, I had developed the opinion that approximately a 1/3 acceptance rate was OK, but the acceptance rate had dropped well below 1/3. The issue was whether we should expand to two conferences per year. There was considerable objection to going to parallel sessions, so that did not seem to be a viable option. Patrick Fischer and I had discussed this some and he decided, and I encouraged him, to approach ACM about starting an ACM Special Interest Group for theoretical computer science. He did this, got ACM approval, became the first chairman of this group and called it the ACM Special Interest Group on Automata and Computability Theory (SIGACT). Thus, SIGACT held its first ACM Theory of Computing Conference in Spring 1968. The group of people involved with SWAT and SIGACT were essentially the same, so this enabled us to have two “theory” conferences per year, one in the Spring and one in the Fall. This relieved the pressure, and we had two different professional societies backing the theory community which gave us a very desirable stability. Although these two conferences remain as leading theoretical computer science conferences to this day, with the IEEE one, since 1974, named “Foundations of Computer Science”, many more theory conferences as well as journals have come into being as the field grew. After 1972 Phil Lewis became chairman of the SWAT committee. While I was chair there was growing interest in changing the name to a broadened scope. I had successfully averted this issue, feeling that it might cause a political turmoil amongst the governing body in IEEE/CS. Thus, it was only appropriate for me to move for a name change at the first business meeting of the committee after Phil had taken over. This caused some laughs from the assembled members followed by a heated debate before the group decided to suggest that the committee be renamed the Technical Committee on Foundations of Computer Science. Phil approved and thought this would be easily accomplished. To his surprise, there was considerable opposition. Many different groups felt that they were contributing to “foundations” in their respective specialties, but even with this Phil did get acceptance for the committee name to become, “IEEE Technical Committee on Mathematical Foundations of Computer Science”. This seemed like a slight restriction, but since we had some flexibility in what we would name our conference, we sim-

ply called it “Foundations of Computer Science” (FOCS). One might say, “We FOXED them out.”

1969 was becoming a busy year. Not only was I taking over the SWAT Committee chair, and that Fall I was starting my sabbatical year at Berkeley, but shortly after I arrived at Berkeley, Gerry Salton the EIC of JACM called me and asked me if I would consider becoming the JACM Area Editor for Theory of Computation. I accepted, was area editor until 1972 when I became EIC from 1972 into 1976, and then was an Area Editor again, this time for “Computational Structures” for 1976-79. Through these JACM editorial activities I became much more involved with ACM. I served on the ACM Publications Board for 14 years from 1973-87. During this period ACM greatly expanded its publications from its first three; JACM, Communications of the ACM and Computing Surveys, to a growing number of special area ACM Transactions as well as some magazines. Although this dramatically changed the character of the ACM Communications from a publication that published numerous original research papers into a publications containing articles for a much broader ACM member audience, the total number of original research papers published by ACM also increased substantially through the introduction of the transactions. As these new publications were being proposed we on the Pub Board had to determine how we were going to name them, and still keep some consistency in names. I remember going to the library to look up potential names. Transactions was a natural, but the IEEE already used this name so it might seem undesirable for ACM. Even then there was a fierce, but healthy, competition between ACM and the IEEE Computer Society. There were other choices; Annals, Record, Letters, etc., but after some discussion we decided that “transactions” really captured the flavor best of all, so that is how this generic name was chosen. I also served for six years, 1976-82, as a Member-at-Large on the ACM Council, and from 1978-82 while Peter Denning was President of ACM, I served as chairman of the ACM Management Board. Also, for 1979-83 I was an ACM representative on the AFIPS Board of Directors and Executive Committee. These were the times when AFIPS held their National Computer Conferences where attendance at one point reached almost 100,000.

I felt that it was quite important for me to serve my computer science

community by volunteering to do things through our professional societies to help the field mature. Certainly publications are an important part of the research community, and the expansion of the ACM publications into the various transactions has proved quite useful. These activities also provided me the opportunity to work with, and get to know, a much broader range of people in the computer field, thereby broadening my understanding of computing and its impact on society.

There were many interesting things that went on in ACM during 1976-82 while I was on the ACM Council. I already mentioned the start of the ACM Transactions. While Jean Sammet was ACM President one proposal that was brought to the ACM Council was for the introduction of a new class of membership to be called "ACM Fellow". I was strongly in favor of this proposal, as I felt that having Fellows would bring added luster to ACM as well as to those members who became Fellows. Also, having Fellows was a common practice in most scientific and engineering societies. Thus, I was quite surprised when strong opposition to this proposal arose, spearheaded by Herb Grosch, who stated that we should not be setting-up an elitist class of ACM membership. The proposal failed, and it was only many years later that ACM finally established the ACM Fellow membership class. I found it quite interesting, when after some years of ACM Fellows being elected, Herb Grosch became an ACM Fellow. He certainly did not turn down the honor then! Also, while Dave Brandon was ACM President, ACM sent a delegation to China in November, 1982. I was invited to be a member of this delegation and my wife, as well as other spouses, joined the trip. We visited Beijing, Xian and Shanghai, giving technical talks, visiting universities and research laboratories, and seeing many sites. The delegation was an attempt by ACM to build a scientific exchange between ACM and the Chinese computing community. The Chinese Institute for Electronics was our official sponsor for the trip, which gave us special delegation privileges. I believe that many of our contacts and experiences of this trip were later reported in the ACM Communications by Tom D'Auria, one of our delegation members. I also wrote a brief personal report on the trip, which still remains in my personal files. I was surprised by the massive interest in our technical talks. There were over 100 attendees in each of the talks, even when we presented

talks in parallel at some locations. Also, the attendees usually had been required to be granted attendance tickets to attend. Many of the attendees were well prepared for the lectures, having studied the speaker's papers ahead of time. In the question periods after each lecture many questions were asked. I know that I was quite unprepared to answer some of the questions put to me when they involved details about papers that I had written years earlier.

Another activity, in which I became deeply involved, also started about the same time in the early 1980's. I was asked by the chairman of the ACM Education Board to become a member of a combined task force of IEEE/CS and ACM members that was looking into the question of whether there should be some sort of certification or accreditation process set up for computer science undergraduate programs. ACM and IEEE/CS had long been involved in curriculum recommendations for computer science and engineering programs, and had cooperated on some of these activities. Also, ABET, the Accreditation Board for Engineering and Technology, had been doing accreditation for undergraduate engineering programs for many years, with the IEEE/CS participating in the accreditation activities of ABET for computer engineering programs. I agreed to become a member of this task force, but was quite skeptical of the prospect of accreditation activities for computer science programs. This skepticism was certainly amplified when the first meeting of this group that I was to attend was scheduled to be in Chicago in midwinter, and I arrived at Chicago O'Hare Airport to 12 degree windblown weather from the much more reasonable temperatures of Atlanta, Georgia. During this meeting I expressed my skepticism, but also kept an open mind. Many universities were just initiating computer science undergraduate programs then, due to the rapid increase in student interest in computing. Often these new programs were being started through setting up a program with unqualified faculty, borrowed from other departments, who had some experience with computers through their programming experiences, but no real understanding of what was involved in computer science curricula. Through this, there were places giving computer science degrees to students after they had little more computer science than a smattering of programming courses. Thus, even though there were many well-established undergraduate computer science programs that existed, which had no need for accreditation,

along with excellent graduate programs that had high reputations, there appeared to be no effective quality control for the undergraduate programs. It was quite likely that graduates from these “programming only” computer science programs would send a message to employers that computer science degrees were useless, and thus degrade the reputation of our computer science profession. In fact, there was already some indication of this occurring. On the other hand, the establishment of an accreditation program by ACM and IEEE/CS would send a message to the employer community that both ACM and IEEE/CS were interested in ensuring that at least some minimum expectations for granting a computer science degree were in place. These considerations convinced me that it would be desirable to establish some criteria for undergraduate computer science programs along with some sort of certification that a program met these minimum criteria. Thus, the task force continued to meet with the mission of attempting to develop criteria. The ACM and IEEE/CS curriculum recommendations, even though helpful, were not sufficient or in the right direction for program criteria. It seemed inappropriate to require specific courses, as done in the curriculum recommendations, rather than a coverage of various topics no matter how they were woven into courses. Also, there were those considerations of the qualifications of faculty, the laboratory facilities, and the supporting mathematics, science and liberal arts materials in the undergraduate education. After much study and discussion the task force, in a meeting on the Oregon coastline, came up with a set of criteria that it felt was appropriate to present to both ACM and IEEE/CS for approval for the establishment of an accreditation body for computer science programs. This involved the establishment of a new organization, the Computing Sciences Accreditation Board (CSAB), with a special Computer Science Accreditation Commission (CSAC) to handle the computer science accreditation process. As the task force was coming to this conclusion I was scribbling a potential logo for CSAB on a napkin. This design, after slight manipulations by a commercial artist, subsequently became the official logo of CSAB. Both the ACM Council and the IEEE/CS Governing Board passed resolutions to establish CSAB, along with an initial commitment to fund the startup, and authorized representatives to be appointed to CSAB. Taylor Booth, an IEEE/CS representative, became the first president of CSAB

for 1984-85, and I, having become an ACM representative, became the vice-president. I succeeded Taylor as President for 1985-87. We then had the job of initiating the accreditation process. Some of these initial activities were summarized in the article, "Computer Science Program Accreditation: The First Year Activities of the Computing Sciences Accreditation Board" by Booth and Miller, which was published in both the Communications of the ACM and IEEE Computer in May 1987. After Taylor and I had written the initial draft of this article I finished it after Taylor's untimely death caused by a heart attack. Much has transpired since these original days of computer science accreditation, in organization, in accreditation criteria, and in the accreditation process. On the whole, however, I believe that accreditation has proved to be beneficial by setting minimum standards for undergraduate computer science education. The criteria helped many young departments in establishing their programs and getting their university's support for facilities. Even for those programs that have never actually seen the need to request accreditation, the existence of the CSAB criteria has had a broad effect on these programs as well. I remained active in the accreditation process through 1996.

From 1988 through 1992 I served as a member of the IEEE/CS Board of Governors and as Vice President for Education for 1991 and 1992. I found it interesting to observe the differences in governance between ACM and its ACM Council, and the IEEE/CS and its Board of Governors. IEEE/CS being a part of IEEE as a whole, and having about 1/3rd of the membership of all of IEEE, still had to fit within the overall structure of IEEE, whereas ACM had no such constraints of a parent body. I felt a certain tension. Some of the IEEE/CS folks seemed to view me at times as an outsider from ACM, and I'm sure that some of the ACM folks viewed me as a traitor for becoming more active in IEEE/CS. Yet, I felt that my loyalty belonged to the profession more than any particular society, so service through both ACM and IEEE/CS seemed appropriate.

As an IEEE/CS Board of Governors member I was later asked to serve on their ad hoc committee to consider the question of whether IEEE/CS should join with ACM and the IEEE Communication Society in the joint publication of a new transactions on networking. Since my research interests had shifted

to communication protocols and networking in the early 1980's I was glad to participate in this activity. Also, this was somewhat of a groundbreaking activity to pursue. Although the Computer Society jointly sponsored some publications with other IEEE entities, there certainly were no joint publications with ACM other than the computer science curriculum recommendations. The idea of having a Transactions on Networking was both appealing and generally supported by the Governing Board members as a great new area for a transactions; however, there was a strong faction that felt that it should be launched by IEEE/CS alone, rather than jointly with ACM and the IEEE Communications Society. Since this was originally an idea from the IEEE Communications Society as a spin-off from their IEEE Transactions on Communications, and ACM SIGCOMM was a strong player in networking research, it made good sense to have a joint transactions. After considerable negotiation it was finally agreed to by the IEEE/CS Board of Governors. An agreement document was signed by both ACM and IEEE, and an intersociety Steering Committee was established to manage the publication. The first issue came out in February 1993 and this transactions quickly became the premier publication for technical papers in networking. I served as an IEEE/CS member of this steering committee since its inception until 2004, and feel that the joint sponsorship has been one of the primary reasons for its great success.

Another activity that was recently celebrated is the 10th anniversary of the IEEE International Conference on Network Protocols (ICNP). This conference has also had considerable success as a single track networking conference with high quality papers. The best way that I can describe the starting of this conference is by printing here the text of my opening session address at the 10th ICNP on November 13, 2002 in Paris. This follows:

From Whence ICNP?

The first ICNP was held in San Francisco in October, 1993. However, the initial concept for ICNP arose 2 1/2 years earlier during INFOCOM '91 held in Miami, FL in April 1991. I was at INFOCOM that year since the conference theme was on "Networking in the '90's" and my PhD student Sanjoy Paul was presenting a paper of ours on conformance testing of protocols in a

session on protocols. Krishan Sabnani was also there as a session chair, and Sanjoy had spent some time with Krishan at Bell Labs. Even then INFOCOM was an enormous conference, so I was quite surprised to find that our session on protocols had such a small attendance. Later, during the conference, Krishan and I were sitting on a couch in the hotel lobby talking and the topic of this small attendance came up. It was clear that there was a lack of interest by the conference attendees in these somewhat formal approaches to networks and protocols. Protocols were designed and tested in a rather ad hoc fashion without any need for formal design methods. Even so, there were a number of other successful conferences on formal techniques for specifying, analyzing, designing and testing protocols that drew this specialized group of researchers working on these formal approaches. What seemed to be lacking, however, was a fruitful interaction between designers of real protocols and the group of researchers developing more disciplined design techniques. This realization spurred us to thinking about “yet another conference” specifically aimed at filling this gap.

We decided to investigate this possibility: A conference aimed at bringing together people looking at formal techniques for protocols with people actually designing and implementing new protocols. Would this be feasible? Would there be sufficient interest? Who might sponsor such a conference? And what might such a conference be titled? We didn't yet have a name picked out, nor any of these questions suitably answered. To go on further we decided to recruit several others, namely Mike Liu and Simon Lam, to join us as a small task force. Both Mike Liu and Simon Lam were well known and respected in their communities and we knew them both. In fact, Mike Liu and I had an interesting interplay going on as to whether we should refer to this field as “protocol engineering” or “protocol science”. Of course he used the term protocol engineering, which was a term that others were also using at the time, but I, wanting to emphasize that more needed to be developed on the understanding and techniques side, espoused the “protocol science” moniker. Of course it was neither - or both - whichever way you look at it. Happily both Mike and Simon readily agreed to participate, and I credit Simon for giving us the final idea for the conference title. We knew we wanted to call it an “International Conference”, since there clearly were people across

the globe working in this area, and we also knew that “Protocols” should be in the title. Simon suggested “Network”, the obvious word we were missing. In fact, this was so obvious once he proposed it that I even forgot what other titles we were considering. As things have evolved, however, maybe we should be calling the conference the “International Conference on Networks and Protocols”, but this is not the time nor place to discuss this.

So now we had a suitable title, but other questions still needed answering. Mike Liu had been involved in starting many conferences and suggested that the IEEE Computer Society Technical Committee on Distributed Processing, of which he was a leader, would be a good group to sponsor the conference, so he looked into this. Mike also knew the details of running a conference through the Computer Society, so that was very helpful. Also, to assess interest we arranged to hold an ad hoc meeting at the April 1992 IEEE International Phoenix Conference on Computers and Communications, which was sponsored by this committee. There were 25 to 30 people from various parts of the world that came to the meeting. They expressed overwhelming support for the idea, so while the meeting was still progressing I started to wonder what might be a possible logo for the conference, and eventually those initial sketches I made of the large I and C with the smaller NP being inside the C, became the logo we used. It is interesting to note that we could still use this even if we inserted the word “and” between Networks and Protocols.

Since I chaired the ad hoc meeting, and also somewhat the initial task force, the other members drafted me to be the initial chair of the conference Steering Committee, and I continued as chair for around 5 years. This initial Steering Committee included Krishan, Simon, Mike and Bill Buckles - the chairman of the Technical Committee. Mike Liu graciously agreed to be the General Chair for the first ICNP - as long as it could be held in San Francisco! And Mohamed Gouda agreed to serve as Program Chair. With much help from others, we put the first ICNP together and felt that it was very successful. Many others have helped since, for the following 8 ICNP’s, and the conference continues to get better. Clearly the interest and reputation of the conference has grown during these first ten years. As far as the Steering Committee goes, other people have served on the committee. Krishan Sabnani became the second Steering Committee chair with David Lee, the

current chair, following him. As far as the “International” in the title goes, holding it this year in Paris solidifies our use of this word since we have held the conference previously in the U.S., Japan two times and once in Canada.

In closing, let me quote some closing sentences I wrote in the foreword of the first ICNP proceedings: It reads:

“We hope that the conference will stimulate research and development of new and better network protocols and the ways to design and analyze these protocols. If this occurs, as well as attracting good computer science and engineering researchers into this area of study, we will feel that our original intentions will be satisfied.”

Well, I think we have made good progress toward these goals. There is much more to be done, however, but this only bodes well for future ICNP’s.

There are other groups beyond IEEE/CS and ACM that have interests in computer science. One of these groups that I became associated with in 1983 was the “Computer Science Board”. This was a group of people that Marshall Yovits brought together some years earlier to start an annual conference dealing with computer science education. See William Asprey’s article on the origins of the Computing Research Association in the March 2003 issue of the Computing Research News to see more on how this Computer Science Board evolved into the Computing Research Association (CRA). I recount here my own remembrances, which are somewhat different than Asprey’s account. Marshall, who was chairman of the Ohio State University Computer and Information Sciences Department, saw the need for a conference where department representatives could meet and interview prospective new faculty members, as well as have these new PhD’s present talks on their PhD thesis research. These activities were later taken over by ACM as their computer science education conference. However, the Computer Science Board continued on as a self-appointed group interested in departmental and computing research matters. The annual Taulbee survey of faculty salaries, numbers, etc. is probably its best-known activity. However, it was from this group that CRA evolved, with an elected Board and many activities including the Taulbee survey and the Snowbird Conferences for computer science and computer engineering chairmen. I served as a member of this group from 1983

through 1991 as it transitioned into the CRA. A report of the 1984 Snowbird meeting appeared in the May 1985 issue of CACM. A particular issue that concerned some of us was the wide gap in computer science PhD production in the 1980's with the foreseen faculty needs in academic departments and researchers in industrial laboratories, as well as the lack of sufficient National Science Foundation funding for computer science research. With the blessing of the Board, Paul Young, David Gries, Bob Ritchie and I undertook a study of these two issues, with an attempt to quantify these needs and bring them to light. We published a short version of our findings in the September 1986 issues of CACM and IEEE Computer entitled, "Imbalance Between Growth and Funding in Academic Computer Science: Two Trends Colliding". I was pleased to see that this study, along with additional efforts of CRA and others in the computing community, appeared to raise awareness sufficiently to help in the transformation of NSF computer funding sources from its Computer Research Group organization into its current Computer, Information Science and Engineering Directorate. It is clear that the CRA has become quite an effective mechanism to provide a unified voice to the government for computer science and engineering research. This was difficult to accomplish earlier due to having two different professional societies, ACM and IEEE/CS, in the computing field; as well as industry interests and the IEEE/USA that speaks for all electrical engineering disciplines.

There are many differing views of professional service. The professional societies like ACM and IEEE/CS, as well as other organizations such as CRA and CSAB, highly depend on volunteer service to carry out their missions. Yet, many researchers spend little effort in professional service. They may do some technical paper refereeing and serve on conference technical program committees for conferences in their own special research areas, but never get involved as journal editors or in the governance of the organizations or their committees and special interest groups. Such activities take considerable time and energy, and rarely does one get rewarded for doing these things with ensuing promotions or salary increases. Universities evaluate faculty on their research, teaching and service for promotion, tenure and salary, but service ranks well below research and teaching in these evaluations. In fact, "too much service" might be viewed as taking efforts away from the primary

activities of research and teaching. In industrial laboratories service is also viewed somewhat positively since it may bring some prestige to the organization and provide external contacts for the company. However, the primary goals are to do research and development that leads to products for the company. Thus, inventions, prototype systems and technology transfer are the keys to individual salary increases and promotion, rather than service. So, much like in academia, “too much service” may be viewed negatively. If this is the case, why do more than a minimum amount of service to satisfy the evaluation requirements? I believe there are reasons to do more that provide rewards of a different kind. Let me explain. First, computing is a very young discipline. I like to shock my colleagues by saying, “Computer Science really isn’t a Science yet”. When they ask me why, I respond by saying that it isn’t a science yet because it isn’t 200 years old. The notion behind this comment is that to be a science a field must have some well understood principles and laws, and in computer science we have very few. I believe this will change with time. More fundamental principles and laws will emerge that will give a much firmer foundation to our discipline, and then give us a stronger argument to call it a science. Research is certainly needed for these to evolve; however, service also plays a vital role in developing the field and guiding its directions. Another reason to be involved in professional service are the personal benefits that ensue, beyond salary and promotion. I found that I got to know many more people through my service activities than I would have had the opportunity to meet simply through research. This helped me broaden my understanding of computing as well as to make many friends and contacts. I found their views often differed from mine as a researcher. Yet they were interesting and stimulating. This has helped me broaden my views as well as strengthen my own opinions. Also, I found that through my various service activities many people that I had never met knew me. For example, my service as Editor-in-Chief of JACM was invariably mentioned when I was introduced as a speaker in the 1970’s, especially when I presented talks in Europe during those years. It was clear that the researchers outside of the U.S., in particular, held JACM in esteem since this was a primary source for them to keep up with the research activities of that time in theoretical computer science.

Service to the computing community has been very rewarding and satisfying to me, and I highly recommend it to others. It is easily entered into by becoming involved with the professional society activities, be they conferences, journal area editorships, or even chapter activities. This can lead to membership on technical committees and special interest groups, and build from there into more. No, I don't believe that these things increased my salary, nor provided direct opportunities to promotion. However, I found the other benefits to be very valuable and enjoyable, and I am sure that others do and can enjoy these benefits as well. Finally, I found it to be quite an unexpected and pleasant honor to be awarded the ACM Distinguished Service Award for 2002 at their annual award banquet in San Diego, California in June 2003.

Chapter 7: Georgia Tech Days

Late in 1979 or early 1980 Rich DeMillo, who was a professor at Georgia Tech at the time, called me to talk about their search for a new Director of their School of Information and Computer Science. He asked me if I would be interested in being a candidate for the position. I assured Rich that I was quite happy at IBM Research and was really not looking for a change. Yet, after much urging by him, I agreed to send him my resume even though I told him that I didn't expect to leave IBM. As discussed earlier, I knew Rich from the computer science research community and from doing some joint research with him. Also, Nancy Lynch, another theoretician in computer science who I also knew, was at Georgia Tech, and Rich indicated that they were both interested in the possibility of me coming there. A short time later I was asked to come down to give a talk and be "interviewed". This was getting serious. After all, I had been with IBM for about 30 years by then. The research activities in the Mathematical Sciences Department were very interesting. I had reached a rather senior level in the department, and had wonderful colleagues. Nevertheless, I went down to Georgia Tech in Atlanta to see what was actually going on there. After another trip or two, I was offered the position of Director of the School. Not being willing to burn my bridges with IBM, I asked them for a one year leave of absence so that I could see if the Directorship of the School would be interesting to me, as well as a new challenge. Rather than a leave of absence, IBM granted me a leave much like the sabbaticals I had taken earlier. This way I would continue as an IBM employee and a contract between Georgia Tech and IBM would compensate IBM for my salary for the school year. I had made it clear to IBM that I might actually decide to stay at Georgia Tech after that year, but the arrangement was still approved, making this change very easy, since if I found out that I did not want to stay there I could return to my research home at IBM. Thus, we moved again. We rented a house in Atlanta near Emory University, fairly close to Georgia Tech, for August 1980 through June 1981 from a professor at Emory who was going on sabbatical in Paris for the year.

Why would I consider such a change? Someone actually asked me some-

time later why I had applied for this job when there were so many other opportunities that would have been much more attractive. I informed him that I had not applied, but it was rather they who had approached me. In any case, a main reason for my interest was that Georgia Tech was an excellent institution with a strong reputation, particularly in engineering, and I had been assured by them that they wanted me to build a strong computer science program there. This was a new challenge. Something that I could not do at IBM Research, and there was a real need for more quality graduate (and undergraduate) programs in computer science. I felt that if I could succeed in building an excellent computer science program at Georgia Tech that it would be a service to the computer science community as a whole. The Dean of the College of Science and Liberal Studies, in which the school resided, was Dean Henry Valk, a physicist, and he promised me good support for hiring new faculty. However, he seemed quite shocked when I told him that when hiring new faculty we would also have to supply them with an office and office furniture, and in the case of computer scientists what “office furniture” meant was including in each of their offices a computer terminal connected to the school computers. This was a new concept to him, as direct access to the computing facilities was not available to the faculty then. Most computing jobs for the faculty, as well as students, were processed in a batch mode by submitting the jobs to the computing center as decks of cards, with later pickup of the results at the computing center service window. My request for computer terminals in offices sounds very outdated now, but desktop workstations were not even conceived of at that time. With continued assurance of support for such hiring and facility support, however, I decided to accept the challenge.

Thus, in August, 1980, still formally an IBM employee, I became the Director and a Professor of the School of Information and Computer Science at Georgia Tech. I went there untenured, however, as the Board of Regents had a practice of requiring at least five years service in academia prior to granting tenure. I was granted three years prior service due to all of my previous teaching and sabbatical activities, but it was certainly somewhat strange to be the Director of the School, reporting directly to the Dean of the College, without having tenure. Georgia Tech, of course, had a Col-

lege of Engineering, for which it was internationally recognized. In addition there was a College of Business, a College of Architecture, and the College of Science and Liberal Studies (COSALS), which included our School of Information and Computer Science (ICS). COSALS also housed the schools of Physics, Mathematics, Chemistry and Psychology, as well as a number of “departments” such as English. The terminology “department” referred to a unit that simply taught service courses required by most disciplines, but did not offer degrees in their discipline. Whereas “school” referred to units that offered both undergraduate and graduate degrees. ICS was quite different than most of the other units within COSALS. Most of the COSALS schools had few majors and offered many service courses, but ICS had a large number of undergraduate majors, a healthy masters program and a small PhD program, and offered almost no service courses. As I learned more about the College, I found that this difference was cultural, and that ICS was culturally more like the schools in the Engineering College rather than those in COSALS. I found it interesting that Bill Sangster, the Dean of Engineering, often told me that ICS would be welcome in the Engineering College. My response became; “Thanks Bill, but that is not my decision to make.” Anyway, I’m sure that many of my ICS faculty would have staunchly resisted such a move, especially those with mathematics, psychology and information science backgrounds. More importantly in this first year, I had to understand the major problems in ICS and devise approaches to overcome them. There were indeed many problems. The School was founded as a School of Information Science and later changed its name to Information and Computer Science. Many of the original faculty members, including the founding Director, were less than thrilled over this name change and change in direction. Information Science and Computer Science cultures were quite different in styles of research as well as subject matter. Thus, this led to tensions in the faculty as well as difficulty in course offerings. Since most of the incoming undergraduate students wanted a computer science education, and the numbers were rapidly growing, this meant that we aggressively had to recruit new faculty. In the 1980’s this was no easy task. New PhD’s in computer science were few, and industrial laboratories as well as academic departments were actively recruiting. Demand far exceeded supply. Yet, we did quite well, hiring young

faculty from good programs, and I attempted to support them well and give them leadership responsibilities to build courses and research programs in their specialties. Over a few years we built or improved our program in most of the core areas of computer science, and became known for real strengths in systems, networking, artificial intelligence and even software engineering. We couldn't actually label our software engineering courses with titles that included the term "engineering", however, since the Engineering College had extreme sensitivity to anyone else using the term "engineering". If a course had engineering in its title they insisted that it be a course offered by the Engineering College.

I held monthly faculty meetings open to all faculty and with an agenda that could be added to by any faculty member. Junior faculty had the same opportunities as senior faculty to bring up ideas and participate in discussions. Also, we encouraged active graduate and undergraduate curriculum committees to review and revise the curricula. Our new faculty participated strongly, and this helped us in updating both the graduate and undergraduate programs. Of course I eventually had to make final decisions, as well as steer directions, but I felt that this open approach to participation was essential to build our programs and encourage cooperation amongst the faculty. We had different problems with our graduate and undergraduate programs. Our undergraduate enrollments were expanding exponentially, so eventually we succeeded in putting a cap on the number allowed to be undergraduate majors. Installing a cap on enrollment in a program was counter to the Georgia Tech practices, but after much arguing we convinced the administration that we could not hire fast enough to sustain such growth, nor maintain quality without a cap. This cap held for several years until the extreme pressure subsided, and we had installed some filters on getting good grades in some of the basic courses before a student was admitted into the ICS major. For the graduate program we had an opposite problem. When I arrived we had about 25 PhD students on our books, but less than 1/2 of them were actually active graduate students. Others had taken jobs, left, and had never officially been removed from the books. Even some that were there were not progressing toward their degrees. This was due partly to the lack of sufficient research grant and contract funds for graduate research positions, so they were forced

to take other jobs to support themselves, which didn't relate to their PhD research. The result was that we needed to purge the books, encourage the active students to progress with their PhD research, and get more funds for graduate research assistantships. Along with this, we needed to build up our research facilities, attract new high quality graduate students, and start new research projects; just to name a few. I encouraged faculty, especially the new faculty, to apply for research grants and provided an incentive to do so by letting them buy out of some of the teaching duties by a small amount of funds toward their salary. Also, Tom Stelson, the Vice President for Research and the head of the large Georgia Tech Research Institute associated with Georgia Tech, had a very attractive program for attracting new graduate students to come to Georgia Tech. He would promise funding from GTRI funds for two graduate research assistantships for each GRA funding promised by the School. This allowed us to offer many more potential graduate students funding when they arrived, than would have been possible using School funds only. We also set our GRA funding level somewhat above those of competing schools, and these two approaches helped us grow our PhD student body. Rarely did Tom Stelson have to use the monies he promised in his program, since the number of students that actually entered the program was substantially less than had been given offers, but he had no problem backing up his promises when he had to, as there were many GTRI projects that were happy to take on our new graduate students to work with them at the start of their graduate education.

Before I stepped down as Director in 1987 we had made significant progress, attaining an active graduate student body of around 75 students with good GRA support. Our graduate and undergraduate programs were substantially improved and it was gratifying to see that our national ranking was showing this. The National Research Council ranking rated our program as the most rapidly improving program in computer science, which made me feel that the efforts were worthwhile indeed. I viewed this only as a plateau, however, as much more needed to be done before our programs could reach into the top 10 to 15 in the country.

One of the dangers that I faced upon becoming the Director of ICS at Georgia Tech, and attempting to build its computer science program, was

that I would spend all of my time on these administrative tasks and stop doing any computer science research or any other computer science activities. I guarded against this religiously, since I had no intention of “losing my profession” and simply becoming an administrator. Thus, I taught courses in the modeling of parallel computation, graph theory, and eventually networking. My first PhD student at Georgia Tech started with me in the Fall of 1981. He was Tat Choi, actually a PhD student in Electrical Engineering. He came to me to see if I would be willing to advise him on his thesis in communication protocols for which he had some initial ideas and was not able to find any E.E. professor that was interested in this area. Even though I had not worked in protocols before, I immediately saw a connection between this area and my current parallel computation area of research. Thus, we started a study of some of the literature in communication protocol modeling and analysis, coming across the model of communicating finite state machines, which was being applied to protocol analysis. Finite state machines, of course, were something else that I had worked on before, so this combination of finite state machines and parallel computation experience made this new area for me easy to enter. This work with Tat Choi started my interests in networking and communication protocols, and this has now been my main area of research for over 20 years. Another enticing feature of this area is that it appeared to be in its infancy with most of the approaches being ad hoc in nature. Thus it appeared to be ripe for the development of some more formal approaches. Tat’s thesis was the development of a decomposition technique based on the communicating finite state machine model that had the benefit of materially simplifying protocol analysis. I used some of this work, and other literature developing protocol analysis and testing techniques to start teaching a graduate course on protocols. This resulted in two more PhD students, Bert Lundy and Murali Rajagopal working with me on their PhD theses. As I worked with these students I developed the practice of not allowing any interruptions while we worked together in my office. This protective measure enabled me to protect time for research and continue publishing technical papers. The administrative demands were intense, but I had no intention of stopping research, and this new field of protocol research led to many interesting problems.

Almost as soon as I arrived at Georgia Tech in August 1980, Dr. Pettit, the Georgia Tech President, asked me if I could help them on quite a different problem they were facing. Their current payroll system for Georgia Tech was a complex computer system handling five different payroll cycles for different classifications of employees. The system did not meet the U. S. Government A-21 regulations for payroll, so they had to get a new payroll system up and running in order to still qualify for government funding. With all the government grants and contracts held by faculty in the academic units, as well as GTRI, it was crucial to meet these A-21 regulations. They had foreseen this and had a new payroll system under development. This development had been going on for several years and Dr. Pettit knew that with the progress being made it would be nowhere ready by the required deadline. Thus, Dr. Pettit was asking me to see if I could help to ensure that the new payroll system would be up and running in time. I told him that I was no expert in payroll systems, but I would be glad to see what I could do. To be effective this required that I hold a title that went beyond simply a School Director. Thus I was given the additional title of “Acting Associate Vice President for Information Technology”, which I held from August 1980 to September 1981 when a new person took over the position without the “acting” part. I then started to look into what was going on for this project. I found out that a commercially available payroll system had been purchased by Georgia Tech and this was being followed by modifications that were being carried out by a group of programmers in the Computing Center so that the program would be suitable for this complex of five payroll cycles that they had in place with their current system. The Finance unit led by the Georgia Tech Vice President of Finance was also involved, since they were the people that produced the payroll. It had gotten to the point between these two groups that they were communicating their respective needs via memos, rather than discussion, and it looked like there was considerable friction between them. To ascertain the status of the project, monitor its progress, and promote a better working relationship, I instituted weekly Friday afternoon meetings in the ICS Conference room that adjoined my office. Attendance was mandatory, including the Head of the Computing Center and the Vice President for Finance. This quickly got the attention of all parties involved. Friday

afternoons were not the favorite times for meetings for many, but I felt that immediate prodding and questioning was needed for me to understand what was going on in detail and to assure that progress was being made. This allowed me to understand better and also for each group to hear what the others were saying. I remember early in this process of asking for a Pert Chart of the project being developed, resulting in blank stares from everyone. No one knew what a Pert Chart was. Clearly a complete view of the project had not been developed, rather a piecemeal approach was being taken with the programming group responding to requests for new features from the finance group, resulting in programming changes being made without much concern about their complexity, the time required to make the change, or the side effects that might result. This Pert Chart request resulted in me getting a “Pert Chart like” diagram produced for the project, giving an overview of the work to be done and the bottlenecks involved in finishing the project. Since there was still quite a bit more to be done, I ordered that no more modifications be requested or implemented (e.g. an automatic bank deposit feature) until we had an operational system. I said, of course it would be very nice to add such features, but first we needed a core payroll system up and running. With this approach, and continued Friday afternoon meetings, the payroll system was finally completed in around nine months. Then, to test the system I asked that this new system be run in parallel with the old system, which would still be producing the actual payroll, for three months, so that the new system could be checked against the current system to ensure consistency. Amazingly, all checked out OK, and I reported back to Dr. Pettit. Needless to say, he was very pleased. The A-21 deadline had been met. However, I told him: “Yes, the project is complete, but now it is time to go and find a truly commercial system that will serve Georgia Tech needs. This new system as developed works, but it is full of program patches and could fail at any time. Payroll systems are probably the most prevalent computing systems in use, and surely there should be options for a reliable and commercially supported system that would satisfy Georgia Tech needs.” I don’t think this advice was taken, however, but my success at getting this system completed was very much appreciated, particularly by the Vice President for Finance. Later, as special needs for ICS arose that required added

funds, my work on this payroll project paid off in various ways for the School. Fortunately, however, Georgia Tech found and hired someone else to fill the Vice President for Information Technology position, Jesse Poore, so I was relieved of my acting position so that I could return my full attention to the ICS Director duties.

As a newcomer to Georgia Tech and Director of one of their Schools, I found myself appointed to many committees for the Institute. This was quite a change from industrial research; i.e., where deliberations and recommendations of committees are an integral part of academic decision processes, in industry decisions are often made by a small “inner group” more rapidly and then announced. The culture was quite different in this respect. Another cultural shock came via a phone call from Dr. Pettit in the summer of 1981 asking me to give the commencement address for the Fall 1981 ceremonies. I would have loved to decline, but my better judgment told me that this was something I better accept. Thus, on September 4, 1981 I presented my first, and only, commencement address, “Is the U.S. Leading in Technology?”. This talk was aimed at the threats, especially Japan at that time, of the countries investing heavily in technological education and research while the U.S. was holding ground, at best. This was backed up through many statistics. In this way I could urge the graduating students, who at Georgia Tech are primarily technology oriented, to go out into the world and excel, thereby helping to keep the U.S. in the lead in industry and research and development. As we know now, Japan and some of these other countries later met economic difficulties, which have helped the U.S. maintain its technological lead. Even so, as we now experience a downturn in the information technology sector due to over exuberance of the past few years, we not only see substantial changes in industrial research but may again have concerns about the U.S. leadership position prevailing.

As a growing School, both in size and reputation, we faced some critical needs for faculty offices, laboratories for new specialties we were developing, more classroom space and student areas. In Fall 1980 our offices and one laboratory were housed in the semi-basement of the Rich Building; the building that housed the Computing Center offices on the upper floor and their computer facility on the floor we also occupied. Not only were the ICS offices

uninviting accommodations for attracting new faculty, but they were small and few in number. Thus, with the invaluable help of Lucio Chiaraviglio, my Associate Director, who had also been Acting Director until I was hired, we carried out our plans for more and better space. Privately I dubbed this our “Space Wars”. Generally, it is very difficult to get new space at an academic institution without getting a new building or extension constructed. Academic or administrative units are very reluctant to relinquish space, even though they are not fully utilizing the space they have. There is really no real incentive to do so. As budgets go, they do not pay rent for their space, and unless they are moving to new and better quarters, they hold tenaciously to the space they “own”. The argument is: After all, we use it to some extent and will be needing it soon as we initiate new programs. Yet, with much arguing, and I’m sure with making some enemies, we succeeded in our quest. First, we got a small attached-building, a temporary kind of structure, to supplement our office space. Later we took over the upper floor of the Rich Building, including the move of my office to that previously occupied by the Computing Center Director. Finally, in the mid 1980’s we succeeded in getting a new building design started by an architect hired by the Institute. Thus, by the time I was to leave Georgia Tech in 1989 the building, which was also to house the Computer Engineering program of the E.E. School as well as some chemistry laboratories and lecture halls on the lower floor, was almost ready for occupancy. Unfortunately it did not work out for me to see the fruits of this space war, but the building has served well since for the “College of Computing”.

What caused this space war success? I believe that it was primarily due to the marked improvement of the ICS rankings and popularity. We had hired very good young faculty and a few more advanced faculty, and their efforts built our program. Hiring senior faculty, however, was very difficult. The Board of Regents practice of not granting tenure for incoming senior faculty was a major deterrent, since other Universities could give attractive offers that included tenure upon entry. Through the initial years we received good support from Dean Valk, but he stepped down within a few years of my coming, and a new Dean, Les Karlovitz, a mathematician, took over COSALS. Thus, by the mid 1980’s I found considerably less enthusiasm for support

for ICS. We had reached a reasonable national stature, comparable or above other COSALS units. Our research and graduate programs were flourishing with healthy grant support, and our improved undergraduate program was very popular. We received CSAB accreditation in the first round of CSAB accreditation in 1986. Yet I became convinced that the Georgia Tech administration was happy with our position and had other priorities for improvement. With this feeling, and not being interested in simply running an organization that had much room for further gains, I resigned as Director in July 1987. This met with considerable surprise. Lucio had urged me to continue, as I had discussed this with him ahead of time. The ICS faculty were shocked and quite disappointed. Demetrious Paris, the long time Director of Electrical Engineering, probably the most powerful School at Georgia Tech, and our obvious competitor with their Computer Engineering program, knew me well as we had had many discussions over the years, and some battles. He saw me in the lunch room shortly after my resignation was announced and came up to me saying; “Ray, what happened? I thought everything was going well.” I told him that it was, but after 7 years it was time for me to step down and let someone else take over with new ideas. I had no plans of leaving ICS or Georgia Tech, however. I expected to continue as a “plain old professor”, get some research grants to do my research in networking and protocols, teach my courses, and work with students. I had good colleagues and good students that I could work with, and the resignation simply allowed me to return to my primary interests of research, along with some teaching. Also, I could now take my first real academic sabbatical for 1987-88.

I arranged to spend October - December at the Computer Science Department of the University of Texas at Austin with Simon Lam and Mohamed Gouda, two people that were well known for their networking research. I gained much during this time, and had the opportunity to work very closely with Mohamed Gouda writing some papers on protocols together. February and March 1988 I spent at Maryland as a visitor at UMIACS. Vic Basili, the Chair of Computer Science there, whom I had worked with on the Computing Research Board, facilitated this arrangement for me. I didn't get a chance to work closely with people at Maryland then, but I did get a chance to continue my research on protocols and have a good two months there. In

April I returned to Georgia Tech, settled into a new office, and in the Fall taught my graduate communication protocols course. From this offering Murali Rajagopal, an E.E. graduate student, became my last Georgia Tech PhD thesis student.

Not long after returning to Georgia Tech in April 1988 two nearly simultaneous events occurred. I was approached by the search committee of the University of Maryland to apply for their Computer Science Department Chair position, and John Hopcroft from Cornell called me about another possible position as Director of a new NASA venture for starting the “Center of Excellence in Space Data and Information Sciences” (CESDIS) at NASA Goddard Space Flight Center. This new center was under the management of the University Space Research Association (USRA). John was serving as Interim CESDIS Director until a Director could be hired who would actually get CESDIS established. The position was also associated with the University of Maryland, wherein the CESDIS Director would also be a full professor at Maryland, and NASA funding through USRA would be provided to Maryland for the part-time CESDIS Director position. Dilemma upon dilemma. Here I was ready to settle back into a more relaxed professorial life at Georgia Tech, and these two new challenges faced me. These were two different challenges, and after several visits back to Maryland I had some decisions to make. Should I accept a department chairman offer from this highly respected department at Maryland, or the CESDIS Director offer which was a new kind of challenge? After much deliberation I decided to accept the CESDIS Directorship, as this was unlike the challenge I had faced at Georgia Tech being essentially a department chairman, and it had the intriguing potential of obtaining academic computer science research support from NASA for projects at various universities. Thus, having just recently stepped down as Director of ICS at Georgia Tech, this CESDIS position was a completely new challenge. I became CESDIS Director in July 1988, while still at Georgia Tech and continued there until April 1989 when I moved to Maryland. CESDIS was still nonexistent. I traveled often from Atlanta to Maryland to select proposals sent to CESDIS for research funding, considered applicants for the head staff position with CESDIS, and talked to Milt Halem at NASA Goddard about space for CESDIS. This interim time gave me the chance to

settle matters at Georgia Tech as well as finding a house in Maryland. Thus, at the end of March 1989 I retired from Georgia Tech, becoming a Professor Emeritus there, and started at Maryland in April 1989, both as a professor and as CESDIS Director, looking to build this completely new organization.

Chapter 8: University of Maryland and CESDIS

Before arriving in Maryland on April 1, 1989 I made numerous visits to Maryland to start establishing CESDIS. With the help of USRA I interviewed candidates for the head administrative position for CESDIS, hiring Nancy Campbell who had considerable administrative and budget experience from previous positions. Nancy thus became the first CESDIS employee, and remained with CESDIS for its complete life. She was primarily responsible for setting up the initial CESDIS facilities at NASA Goddard, hiring the administrative staff and keeping in touch with both USRA and NASA. I say that she was the first employee of CESDIS since she was full-time with CESDIS, whereas I was officially employed by the University of Maryland as a Professor, and my contract with USRA provided the funds for my half-time position as CESDIS Director. A call for proposals for CESDIS had been initiated by John Hopcroft before my hiring, and these proposals were being received from numerous universities. We established an evaluation procedure for these proposals, filtering out the 10 to 12 that seemed most promising, and then along with Milt Halem and some others from NASA and USRA we selected four for our initial funding. Milt Halem originated the concept of CESDIS, got the initial funding from NASA, and CESDIS was housed in his NASA Directorate at Goddard. Thus, with our initial funding from NASA we started our research program with projects at George Washington University, Duke University, North Carolina University and Stanford University. Also, we had funding for two Assistant Professors at Maryland to work with CESDIS part time, both for research and support of our program. The four projects consisted of the development of a scientific data visualization system at George Washington, an image pattern recognition system for interactive analysis and visualization at UNC, a parallel data compression of space and earth data project at Duke, and a computationally assisted analysis of auroral images system at Stanford. Even though there were several other very worthy proposals, the funds that were given were not sufficient for added proposal funding. CESDIS funding did provide me with some GRA support and we also had some monies to start a seminar series, conduct some workshops and initiate a technical report series. These provided additional connections and

awareness for CESDIS between university computer science and NASA. To provide oversight and advice we also established an advisory board of about six well-known computer scientists to meet at least once a year.

We attempted to get the members of our four funded projects to spend periods of time with us at NASA Goddard during our first summer of operation in 1989, to carry out their research and interact with NASA scientists. We had hoped that this would establish fruitful interactions, but we ran into a major problem. We did not have a research facility set up for them. I had been assured that there were NASA computers that would be available for our CESDIS researchers, but this was not the case. There were very few UNIX workstations at Goddard then, and they were already being used by the NASA researchers full time. Thus, our CESDIS summer visitors could not use them for their research without inconveniencing exactly those NASA researchers who would be our best contacts. For this reason we had only very short visits that first summer, but learned our lesson in that we would need to establish our own laboratory to be effective. We lobbied for more space to set up this laboratory, purchased attractive workspace separators, installed a group of UNIX workstations and got them connected to the Internet. This then allowed us at Maryland to spend more time at our CESDIS facility, as well as provide access to research machines for the four projects.

Since I arrived at Maryland on April 1, 1989 I did not have any teaching duties for the Spring semester at the university, so this allowed me to devote most of my attention to getting CESDIS established at NASA Goddard. I did, however, have my PhD student Murali Rajagopal from Georgia Tech still working with me on his thesis, as well as John Guthrie, a Georgia Tech computer science graduate student who had started to work with me on some research and had followed me to Maryland to continue his graduate studies. Thus, I spent time with Murali when he visited either at the university or at our CESDIS offices. Also, John Guthrie was trying to get his research to a point at which he could put together a PhD research proposal that he could defend. Murali progressed well, receiving his PhD in September 1990 with his thesis on protocol conversion, but John never completed his proposal, dropped out for awhile, restarted later with another advisor, but again failed to find his way to completing a proposal and getting his PhD. He certainly

was quite capable and a good scholar in his understanding of the various research areas, but for some reason, unknown to me, could not seem to be able to create enough new material of his own.

Thus, after this initial startup period of the Spring and Summer of 1989 at Maryland I started my teaching duties in the Fall 1989 semester by teaching my graduate course on Communication Protocol Specification, Analysis and Design. Through this offering I attracted two University of Maryland graduate students to work with me on their PhD's; Sanjoy Paul and later Zafar Choudhry, both EE students. For the Spring 1990 semester I taught the undergraduate course CMSC452, "Elementary Theory of Computation", which is an introduction to automata theory and formal languages. This course was much like some courses I had taught previously as an adjunct professor when I was at IBM, so it was a natural advanced undergraduate course for me to teach. However, it required considerable new preparation for me since this course had additional material and also used a text that I had not used previously. As time went on I taught this course almost every year. It was a popular course, and I found it fun to teach as the material was well developed theory, it had very nice ties between the automata models and the formal language structures for programming languages, and had many ties to other systems courses.

As a newly arrived senior faculty member at Maryland I was also asked to take on quite a few service duties. This was similar to my initial time at Georgia Tech, and it may be common in general that a new senior faculty member be asked to serve in many capacities. Thus, I was involved with faculty hiring on various hiring committees. I also served on a beginning course development committee and on the Dean's College APT Committee for 1990-92. From 1991-93 I served as the department representative on the development of a Masters Program on Telecommunications. This effort was led by Dr. Bill Destler, who was Chairman of the Electrical Engineering Department at that time. Bill later became the Dean of the Engineering College and later assended to Provost. This Masters in Telecommunications became a program administered by the Electrical Engineering Department with cooperation from the Computer Science Department and the Management School. It also was supported with considerable funding from industry,

and many of it's students came from local industry, taking evening courses designed for this audience.. I taught for several years for this program, but eventually dropped out when it became apparent that to continue teaching in the program would mean additional teaching loads over the normal computer science department needs. Other activities I served on, or chaired, over time included various search committees, the Department Council, the department grievance committee, the University Senate, department coordinator for computer engineering ABET accreditation, the Dean's Facilities Task Force and the department's awards committee. Even after retirement I have continued serving on several committees to help support the department.

The CESDIS activities continued to require much of my time in addition to my professorial activities. As time progressed CESDIS supported NASA in many ways that only indirectly related to computer science research. Continuous NASA funding for CESDIS proved very difficult to achieve, thereby causing CESDIS funding to be very uneven and unpredictable. This made it very difficult to plan ahead or even maintain the funding for our initial projects. Some of these projects were not renewed after their initial grants ended, and no new calls for proposals were sent out. Although we still had some computer science research going on, as well as a very strong support activity for NASA's participation in the interagency high performance computing and networking initiative, I could not develop any research collaboration with the NASA Goddard networking activities and my networking and protocol research. Thus, somewhat after five years as the founding Director of CESDIS, I decided to resign and become a "plain old professor" again, this time at Maryland. I was not interested in simply managing a service organization that had little computer science research, connections to university computer science research, or to the networking areas of my research. CESDIS continued on for a second five years under a new Director, but it never reached the potential I had hoped for. I believe there were several factors that contributed to this. First, and foremost from my point of view, is that NASA really didn't understand academic computer science research nor have a mission to support computer science as a discipline. NASA provides good support for space and earth science, with numerous grants and contracts for research projects in these areas. Its support for "science" is limited, how-

ever, as it requires substantial money to support its engineering activities for its rocket, manned space flight and space station programs. Thus, computer science could not effectively compete for funds with programs that are considered more central to NASA's primary mission. I didn't realize this problem until after some time as CESDIS Director, so I was undoubtedly somewhat naive in that respect. Yet, I find it hard to understand how this can prevail when all of NASA data gathering and analysis, as well as large portions of its flight operations depend on computing. Certainly computer science research should be capable of improving their use of computers, their computer programming and their networking capabilities.

Another thing I learned concerning the CESDIS operation was that even though there was initial strong support from Milt Halem's operation at NASA Goddard, there was not any dedicated support at the higher levels of NASA. Since funds for their science programs were already quite constrained, and the top administrators were from the earth and space science disciplines, or engineers that supported them, the adding of yet another science was not something they were comfortable in supporting. Rather, they wanted computing research projects that directly supported their other missions, and these were more short-term developmental in nature, rather than that of research. Even with some of the initial CESDIS projects being very closely connected to overall NASA needs, they were not closely enough tied to individual NASA projects to get enthusiastic support. In talking to other respected computer scientists that had previous NASA contacts, I learned that these views that I gained over time were similar to the views that they had come to as well. Other issues probably compounded these CESDIS problems, such as; its location within a Directorate at NASA Goddard rather than at some higher level in the NASA administration, my view of support for academic computer science research may have been somewhat idealistic, and the overall shrinking of NASA's budget during those years.

Thus, even though I was quite disappointed that CESDIS didn't thrive, I found it quite a relief in stepping down as Director in October 1993. As mentioned earlier, during these years as CESDIS Director I was also a professor in the computer science department at Maryland and taught one course per semester there. This one course load was the standard load in the de-

partment for faculty that were involved in research, so even though one-half of my salary was supported by the CESDIS contract, my teaching load at Maryland was not reduced. Conversely, as CESDIS Director it seemed that USRA and NASA expected me to be at CESDIS full-time. I believe this is a problem with joint appointments quite generally, but it did not occur to me until several years of direct experience. So, removing myself from the CESDIS Directorship materially reduced my workload overall, as well as eliminating my concerns about the lack of funding and progress that CESDIS was facing. It did, however, eliminate my CESDIS summer funding, as well as the graduate research assistant funding for my PhD students, so now there was more reason for me to seek other research funding. I continued to teach my usual load, an automata and formal languages undergraduate course each fall semester and my graduate communication protocol course each spring semester. For research grant support I was fortunate enough to be approached by a group that wanted to start quite a different activity with NASA concerning making the NASA flight operations more efficient. This was to be done through NASA Goddard again, but this time with direct NASA headquarters involvement, and there was a desire to have the grant from NASA go directly to a university. Thus, I became the grant principal investigator for the project entitled "Reducing the Complexity of NASA's Space Communications Infrastructure" which was funded from July 1994 to July 1999. This project was a delight to work on. Rhoda Hornstein from NASA Headquarters led the project activity. It included subcontracts with various small companies and individuals, support for me and several graduate students, and had NASA participants from NASA Headquarters, Goddard, and the Jet Propulsion Laboratory. My students and I did some modeling for some aspects of the problem, but the project was wide ranging and included annual workshops and the publishing of seven joint papers in various conferences. I found it quite interesting to see how some areas of computer science research could be molded into techniques that applied to satellite mission operations, an area that seemed quite remote from computer science. Yet, we saw how to apply Petri nets for modeling of a satellite mission operation, and how reusability could be used in reducing mission operation costs. Although my graduate students initially viewed working on this grant as a diversion

from their PhD research, I know that they benefited greatly through the experience of working with such a diverse team. Also, I received an NSF grant for communication protocol research that provided very desirable funding for my main research activity on protocols from 1995 through 2000.

One aspect of our undergraduate program at Maryland that had concerned me for a while was that there was no undergraduate networking course that was being offered. A course was listed in the catalog, but had not been offered in recent years. Thus, no longer having the CESDIS duties, I convinced the Department Chair to let me restart the undergraduate networking course in the fall of 1994. About 25 students took this course, both from computer science and electrical engineering. Subsequently, the course was then taken over by some new young faculty that joined the department. They added a course project and markedly improved the course in other ways. This course has since become quite popular with many students electing to take it since then. A new graduate networking course was later added, so now a more reasonable set of courses in networking exists for students in the department.

My PhD students from Maryland have been Sanjoy Paul and Zafar Choudhry from Electrical Engineering, and Hong Liu, Junehwa Song, Jun-Cheol Park and Khaled Arisha from Computer Science. Their theses topics were in protocol conformance testing, protocol analysis, synthesis, performance estimation, and passive testing. However, Junehwa Song worked on multimedia authoring systems with much of his work culminating from spending time at IBM Research and working with people there. In addition to these PhD students, over 20 students wrote scholarly masters degree papers under my direction. Most of my PhD students initially went to industrial research positions after graduation. Tat Choi joined GTE Research in Massachusetts, but later joined ATT. I saw him several years ago when he was at the Holmdel Labs in New Jersey, but the ATT activities have moved out of that location now. Bert Lundy joined the Naval Postgraduate School in Monterey, California where he is now an Associate Professor. I visited him in October 2002 when I was there giving a paper at a conference. I have not seen Murali Rajagopal for several years, but the last time we exchanged email he was with a startup in California. Sanjoy Paul joined ATT Bell Labs in Murray Hill, New Jersey working in a networking department led by Nick Maxemchuk. Upon the

splitup of ATT that formed Lucent Technologies Sanjoy went with Krishan Subnani to their Bell Labs in Holmdel. He is there again as the director of a department, but did spend some time as the technical vice president of a startup before returning to Bell Labs. Hong Liu initially joined Bell Core where he had spent a summer or two as an intern, but he left them for a startup as well, but is now with Neustar. Zafar Choudhry joined a company in Virginia doing contract work for the Army and he was located at the Pentagon working on their network security systems. I have not heard from Zafar since 2001, and I believe he has moved on to other positions. Junewha Song joined IBM Research in Hawthorne N.Y. where he had spent time on a fellowship, but after several years there he decided to return to his homeland, Korea, where he is now a professor at KAIST. Also, Jun-Cheol Park initially wanted to go to industrial research, but at the time he graduated such positions in the U.S. were quite hard to find, so he returned to Korea as well, first in an industrial lab, but has since moved into academia in Korea. My final PhD student, Khaled Arisha is with a Honeywell research facility in Columbia, MD where he was working part time while finishing his PhD. Since he is very close I get to see him occasionally.

I have tried to keep in touch with all of my PhD students as I consider them to be my professional children. However, this is difficult as they move far away and change positions. I have emailed most of them at least once a year during the Christmas and New Years holidays, but unfortunately have lost touch with some who have not provided me with their latest email addresses.

In 1996 I was given the opportunity by Krishan Sabnani to consult for Bell Laboratories at Holmdel, N.J. to work with them on networking research. This consulting has continued through 2004, and has given me the opportunity to work with some very good colleagues on passive testing of networks and multicasting. The passive testing work stems from an initial paper in this area by David Lee, Krishan Sabnani and Arun Netravali, where I generalized their approach to a distributed model which is a variant of the communicating finite state machine model that has been used for protocol analysis. I have continued to work with David Lee on passive testing since then, and this led to the thesis work of Khaled Arisha on passive testing. More recently I have built up a research collaboration with Sneha Kasera of

Bell Labs on multicast pricing, by proposing some interesting formulations on trees that give some insight into the pricing benefits of multicast.

I have found it stimulating to be associated with industrial research again, both to investigate new areas arising from industrial research problems and also to see how these industrial research laboratories have changed their mode of operation since the days that I was at IBM Research. In all of my time at IBM Research, from 1957 to 1980, I was never told what to work on, nor was I assigned to a specific project. Not all of IBM Research was that way, but a fairly large number of people were allowed and encouraged to work on those ideas that they found interesting rather than being assigned to company related projects. Bell Laboratories also had that characteristic during those days. This has changed markedly since then. There is much more pressure to work on problems that will lead to company products, to build prototype systems, and work with business units of the company to see that the ideas actually make it into the product line. I'll discuss this more in the last chapter, but it is certainly one facet of the substantial changes that have occurred over the years. In fact, these changes in the industrial research environment have, in my opinion, been a major factor in causing many top notch industrial researchers to move to academic positions. Sneha Kasera has moved to computer science at the University of Utah as an assistant professor and David Lee is moving to Ohio State University as a Regents Professor in their Computer and Information Sciences Department. Thus, I have lost some of my best research colleagues at Bell Labs, but still plan to work with them in the future.

Chapter 9: Retirement Number Three

After starting my career with IBM on August 7, 1950 shortly after receiving my Bachelor's Degree, then retiring from IBM in 1981 to join Georgia Tech, then retiring a second time, this time from Georgia Tech in 1989, to join the University of Maryland, I decided to retire for the third, and I believe the last, time effective June 30, 2002. A nearly fifty two year career in computing is not such a bad record! As I discussed retiring with Larry Davis, our Department of Computer Science Chair in Spring 2002, I mentioned that even though I would officially be retiring as of July 1, 2002, I didn't know exactly when I could clear out my office. He immediately responded; "Ray, why would you want to do that?" What a marvelous surprise! This way I would be able to keep my office, continue to do my research here at Maryland, and also help the department in various ways. Also, this meant that I did not yet have to face the question of what I would do with all the papers, books and journals that I have accumulated over these years. I was granted Professor Emeritus status and was treated royally with a retirement dinner on May 13, 2002, a retirement workshop on May 14, 2002 where Dick Karp, Shmuel Winograd and Mostafa Ammar presented talks, and this was followed by a luncheon and reception. Many people were invited. Mohamed Gouda came from Texas and Rao Kosaraju from John Hopkins. All my family plus my brother and sister-in-law came, as well as Rhoda Hornstein from NASA. Rhoda was the project leader on the NASA project "Reducing the Complexity of NASA's Space Communications Infrastructure" on which I had a grant from 1994-99.

In June 2002 Khaled Arisha and I went to Las Vegas to present one of our papers at a conference there. Since I owed him a dinner for finishing his PhD with me it seemed like a perfect time to do this. My wife had accompanied me on the trip, so we told Khaled that we would take him to New York for dinner. Of course we meant the New York New York Hotel in Las Vegas, just a block from the conference hotel. I had made it a recent practice to take my PhD students out for a dinner after they received their PhD's, and I had not had a chance to do this earlier for Khaled. He enjoyed the dinner celebration as well as the fake invitation to New York. At another conference

I presented another paper of ours in Monterey, California in October 2002. On this trip I got to visit with Bert Lundy, my past student from Georgia Tech who is now an Associate Professor at the Naval Postgraduate School in Monterey. Then in November Marilyn and I went to Paris for the 10th ICNP that I mentioned earlier. I presented my “From Whence ICNP?” address in the opening session there as well as a paper that I coauthored with two graduate students, Arun Vasan and Moustafa Youssef, who had taken my communication protocol course in the spring 2002 semester. This paper was a combination and extension of their two term papers from the course. I’m pleased that they are continuing to work with me on some other protocol and networking research. I also was a coauthor of another paper with David Lee and some of his colleagues at the Bell Labs China Research Laboratory in Beijing, where David Lee became the initial Director of this Laboratory several years ago and has built quite an impressive research operation there. Thus, I am continuing my research in protocols and networking even though I am officially retired. It’s nice to be able to continue to do what I enjoy so much, and to find people that still want to work with me. I haven’t taught classes since I retired, however, but may do so again later on occasion, so that has changed. Many things are the same, however. I go to my office almost every day. We travel to our house in Atlanta quite often, and I am still visiting Bell Labs for consulting. While in Atlanta I also go to visit my colleagues at Georgia Tech most times, thereby using my Professor Emeritus status there as well.

I cannot predict what might happen in the future, but I look forward to staying active in research and service as long as I am able. It is quite amazing to be in a field that has progressed so far in fifty years, but still has so many interesting research problems to investigate. The opportunity to continue with an office in the department and consult at Bell Labs has been a godsend. Even though difficulties abound with our economy, the funding of computer science research, and research job opportunities in academia and industry, there is no lack of research issues to look into, as well as no lack of efforts needed to help our computer science discipline continue to mature and become more robust.

Chapter 10: Changes Over the Years

As is very well known and should be obvious from what computing was like in the 1950's that I discussed in my early chapters, there have been very significant advances made in computing over the last fifty plus years. In 1950 IBM did not even have any computers on the market, but soon came in strong with mainframe computers, peripheral devices, operating systems, compilers and application software. For years it dominated the market, grew rapidly, and set the de facto standards for computing equipment. It was slow to adapt to new ways of computing, however, being late in the PC market, but still being able to gain a large market share once it decided to offer PC's. It was when desktop workstations and networking came in that this dominance slowly eroded. Its initial concept of networking was centered around a central mainframe computer with networked "slaves", rather than peers of networked machines. In the software area it also lagged. It's contract with Microsoft to build the PC operating system was, in my opinion and with hindsight, a tremendous flaw. It enabled Microsoft to own the operating system even though it was done under contract to IBM. Possibly this came about from IBM having had little success in making money on its own software products, but ever since, Microsoft has been able to dominate the market with its operating systems and become the standard platform for the development of PC applications software. Technology advances from vacuum tubes to transistors to integrated circuits have played a major role in computers becoming more powerful, much smaller and much faster, with many new companies being born in the process. Computer science as a discipline cannot claim that it was a major player in the computer hardware advances, and initially at least this was the domain of physics and electrical engineering, with some contributions in computer architecture from computer engineering.

Where then does computer science enter? Even though this is an over simplification, most people viewed computer science to be more on the software than hardware side. Initially as computer science started to appear as a separate discipline with academic departments being established, the faculty members came primarily from mathematics and electrical engineering. Numerical analysis, logic, combinatorial mathematics, and even programming

for mathematicians that used computers, came from the mathematics side to become important components for computer science. Electrical engineers, on the other side, often viewed computers as systems, so their interests were more along the system organization side with computer organization as well as with logical design and new computing devices. Thankfully, these demarcations no longer exist in any strong sense. Computer science and computer engineering have evolved into two disciplines that have much in common. I have often argued that as far as their undergraduate degree education goes they are about ninety percent the same. I have had a few computer engineer colleagues argue with me, claiming that it is less than that, the lowest estimate one gave was seventy percent. Even so, there is much in common and I expect that this will continue with them even having more similarity in the future. One sees signs of this with numerous departments offering a single undergraduate degree called “Computer Science and Engineering”. Also, computer science and computer engineering programs often reside in a single department, with many being in an Engineering College. Although this may be desirable from a commonness and efficiency point of view, one can as strongly argue that it is long overdue to form Colleges of Computing where many related activities could reside centered around the pervasiveness of computing that has arisen in our society, along with computing having become an important line of investigation in science, giving us computational science to supplement the traditional scientific domains of theoretical science and experimental science. Some institutions have established Colleges of Computing, but there is still considerable resistance to this approach. Many scientists do not consider this “computing area” to be a science at all, and indeed there is a long way to go for our scientific principles to be developed. Other considerations also play a role, such as the desirability of having undergraduate majors associated with a college. Here computer science excels, whereas some of the other sciences have few undergraduate majors supplying sources of funding from their tuition. In my view much still has to be sorted out, and this will require considerable time, effort and probably political clout. Idealistically we could imagine computing to be one overarching discipline having many different specialties. Could you imagine, for example, one professional society to represent all computing interests? Will this happen?

I even don't see how ACM and the IEEE/CS could come together due to the differences in structural organization as well as the natural competitiveness and resistance for empires to fall, and this is only one component of the problem. Also in academia, the differences in views between the engineering side of our discipline and the science side of our discipline are huge. (Excuse me for saying here that both are "our discipline" but I really feel that they are. Computer science is different than the other hard sciences in that it has both science and engineering intermingled, and most undergraduates work in the discipline after finishing their bachelors' degrees.) Another factor that has had an effect as computer science and computer engineering have matured is its explosive growth due to student interest related to industrial job opportunities. This played a major role in universities starting undergraduate computer science programs. I described this briefly as I discussed the start of accreditation and CSAB. Some universities realized that they would lose many potential undergraduate students to other institutions if they did not offer a computer science or computer engineering degree. For those universities without strong engineering programs this still gave them the option of attracting students wanting a computer education by starting a computer science program, and many did. Also, many university computer science departments reside in colleges other than engineering by individual choice or for historical reasons. Since colleges of engineering will want to keep their computer engineering programs, and colleges of science will want to keep their computer science programs, reorganization into colleges of computing will be difficult to achieve.

Student interest in computing has remained high from its beginning. There have been some plateaus in enrollment that have occurred, but no major decreases. Will this continue? Certainly continued rapid growth cannot continue. Eventually we could experience the cyclic student enrollments that are common in engineering, or more desirably this may come to some steady state or slow growth state. We may be seeing some of this effect now with the bursting of the dot-com bubble and the current decline of attractive job offers coming to newly graduating students having bachelors' degrees in computer science, or for that matter also in computer engineering. We will have to wait to see how the problems with the technology industry will

apply pressures on our academic programs. As would be expected, these will probably be uneven and have delayed effects. It may create a temporary decrease in undergraduate enrollments, which now appears to be occurring, along with an increase in students wanting graduate education. However, without good job opportunities for students getting Masters and PhD degrees this increase could also come to a halt. Certainly, the opportunities to get attractive faculty offers directly after completing one's PhD have decreased due to the economic downturn, and industrial research has been shrinking quite dramatically. For this reason, as well as the maturing of the field, I would expect that more fresh PhD's will take Post Doctoral positions for several years before obtaining good academic faculty or industrial research positions.

With the growth of computer science and computer engineering academic programs there has been a commensurate growth in academic research. This has stimulated the growth in scholarly journals in the field both by professional societies and commercial publishers. Also many new conferences have been established. Both the journals and the conferences are often quite specialized to particular research areas. This means that few researchers can maintain a broad spectrum of the computing field, as was the common case when computing was in its infancy. Although this is a natural occurrence as a field grows, it can also mean that potential side effects of new developments will not be foreseen. With this growth of journals and conferences there has also been a shift in how some faculty view publication of their research. For many years the publication of results in archival journals was of primary importance, and this is still the case for most well established scientific disciplines. Yet, with the rapid pace of developments, the lag between paper submission to a journal before publication, the high quality of some conferences and the distribution of research software and its use by others, the recognition of one's research efforts has found many new outlets. This has led some computer science researchers both in industry and academia to downgrade the importance of archiving their results by journal publication. Although some accept this, there are still dangers in this approach. For academic promotion and tenure we either must comply with the standard methods of evaluation, or get new methods understood and accepted. The

widespread use of a researcher's software system by other researchers certainly deserves to be considered as a positive peer review of that researcher's accomplishment, but care must be taken to assure that the results are archived and, for experimental results, that the experimental results can be repeated and verified by others. It seems to me that much still remains to be done to make the case that computer science research accomplishments should be viewed by different measures than in any other scientific discipline.

Computer science and computer engineering have contributed significantly to the advancements we have experienced in computing and networking. For much of the first fifty years the advances have been what I would call "inward looking". That is, they were in fields like computer architecture, programming languages and system software. In the 1950's higher-level languages were introduced to free the programmer from programming directly in machine level code. Along with this compilers were developed, largely by ad hoc techniques, to automatically transform these programs in Cobol and Fortran (among others) into machine code. Some felt that this compiled code was very inefficient compared to hand coded machine programs. This led to code optimization techniques for compilers. Then the disciplines of automata theory and formal languages were applied to compiler development, turning the process into a more routine scientific approach, rather than ad hoc process. Boolean algebra was shown to apply to the logical design of computer circuitry, and computer architectures were enhanced by separating the functions in the control and arithmetic units so that various phases of the control and arithmetic instructions could be processed simultaneously. Pipelined and look-ahead control units were introduced, and these computer architecture enhancements gave rise to more compiler optimization techniques to be developed to take advantage of these added units, along with detailed data flow analysis of programs. Although Cobol and Fortran forged the way in showing that higher level languages could be used to ease the burden of programming, many other programming languages were developed that were more elegant or had features that were particularly useful for some types of applications. For a while Pascal was widely taught in computer science undergraduate programs. Later C and its variants became popular, with Java now being quite popular especially for web-based applications. I expect that new program-

ming languages will continue to be developed as new ways to do computing come into being. Software engineering developed as a separate computer science specialty for the development of techniques to improve programming techniques aimed at producing programs that had fewer bugs. Operating systems were developed for the scheduling of programs and the assignment of computing resources to program operation. As programs grew in size and complexity, and the amount of data grew both for scientific and business applications, the issue of the efficient management of data became more important, and database management became an important area for study. All of these areas required years of research and development resulting in many new mathematical and empirical techniques being developed. Theoretical computer science played a central role in some of these developments as well as in algorithm development and analysis. Yet, the bulk of these developments were aimed at making the computing process faster, more efficient, or easier for programs to be developed. Extensive computer networking came along more recently, and its evolution from ad hoc to scientific is much more in its infancy. With it, new issues of privacy and security of information arise, as well as many other issues of the efficient transfer of information from one user to another. Although there is still much to be done in each of these areas of computer science and computer engineering, a more “outward looking” aspect of computing is becoming much more important. That is, how can computing be made more natural for the benefit of the average person? Can it be more “transparent” so that a person is not even aware that computing is involved? Certainly this is happening as computing devices get embedded into our automobiles, appliances, telephones, etc. What is the role of computer science and engineering in this outward look? The areas of computer human interaction, speech recognition, pattern recognition and picture processing are examples of such areas. There are many more, some using artificial intelligence approaches and some using very mathematical techniques. I foresee this extension of our computer science and engineering discipline into these outward looking areas to be very important for the future well being of the discipline. Success in reaching out in this way should lead to many new areas and problems to investigate, including the development of new special purpose computing devices and the discovery of new mathematical problems

to bring into the computing realm. If we look at electrical engineering, which I consider to be an excellent example of a discipline that has embraced new areas, we should be doing the same thing in computing. Computer science has been slow to fully embrace the computer networking area, but it now is. We should look into new areas more aggressively. What about quantum computing, micro technology, biological computing, and others that I have not even heard of yet? I feel that there is much room to expand for our discipline and if we in computer science and engineering do not embrace these new fields they will become the domain of other disciplines. Much of this research may span several disciplines that we do not consider to be purely computer science, and boundaries between disciplines and departments can make it difficult to do research across these barriers. One commonly used approach to accomplish this, beyond direct departmental expansion into a new area, is to establish separate research centers or laboratories so that faculty from different departments or colleges can collaborate through these research units, rather than directly under a given department.

What about computer science industrial research? When I joined IBM Research in 1957, IBM was a very healthy company. It was no longer growing by 25% per year like it was doing in 1950 when I first joined, but it was still doing very well in the computer and data processing business. It clearly saw the rapid technological changes that were occurring in the area and knew that it would need a strong research operation to keep up. It also saw the great example of AT&T Bell Laboratories with the numerous examples of discoveries and patents that they had produced. I'm sure this played a major role in their decision to put priority on establishing a strong research laboratory. Having a strong patent portfolio is an important asset for a company, especially when it needs to negotiate with another company for sharing some of their patent rights. Those were times when both AT&T and IBM were strong and dominated their markets, AT&T from the viewpoint of being a somewhat regulated monopoly and IBM from having market dominance. These conditions have changed dramatically. Competition has come into the telecommunications market by federal government actions, and IBM, which also had some government suits brought against it due to charges of monopolistic practices, found competition increase, but this was probably

due more from its slowness to enter emerging markets. Both the computer and telecommunication industries now experience extreme competitive pressures, both nationally and internationally. For these reasons neither AT&T nor IBM have the same financial base to support research as they did before, and the other companies that have entered these areas have not built dominant research organizations since they have had problems of their own. Of course Bell Labs became part of Lucent Technologies after the multiple splits of AT&T, and thereby had parts that went to other companies as well. With these corporate changes Bell Labs is now much smaller and has a much narrower focus than before. Also IBM went through major changes, moving somewhat away from the production of hardware into being a full information service organization. The fact remains, however, that there is now less ability for these companies to support the broadly based research that they did during their less competitive days. Other strong research laboratories that had computer research have also diminished or disappeared, e.g., Xerox, Control Data Corporation and Digital Equipment Corporation. Microsoft has come in more recently with a noticeable research effort with many very well known researchers, but it certainly does not dominate industrial research like AT&T and IBM did earlier. Another reason often given for the changes that have occurred in industrial research is the more rapid rate of product development from initial concept to the product reaching the market. I have no doubt that this has played a significant role in these changes, but I believe that the increase in competition played a much larger role. In fact, competition itself may have been the root cause of the shortened product development cycles. In some sense competition has been good for the United States. As competition grew many new products and services developed. They became smaller, cheaper and had many new functions. Yet, the monopolistic situation also had some benefits. For example, I never considered my phone bill to be terribly expensive when it was a phone company under the AT&T umbrella that supplied my phone service, yet AT&T was split up under federal government pressure, even though we had the best phone system in the whole world. This caused the base funding for Bell Labs that existed from AT&T to disappear. It was no longer the case that every researcher that AT&T hired for Bell Labs would increase its profit. I like to present this argument to current Bell

Labs researchers because it is basically true, due to the bottom line of AT&T having been regulated through the various state public service commissions which would allow only a certain level of profit over their expenses. Thus, as expenses increased, so would the profit dollars. Few if any of my Bell Labs colleagues seem to have thought of this in this way. I'm not advocating that this previous situation was best for the country and its economy, but it certainly helped in having strong industrial research.

I would like to see a resurgence of industrial research in our discipline. As the economy recovers we may see some strengthening, but it is difficult to imagine that we will see any industrial research organization again that is as strong and broadly based as AT&T Bell Laboratories was in the 1950's. It may be possible for the government to take some actions to stimulate industrial research, but there are dangers that some kinds of measures could stifle innovation. Possibly stronger tax incentives for research expenses would help to act somewhat like the regulations on profit margins that existed in previous years for AT&T.

With all these changes there has been a shift of balance between academic and industrial research. Many more faculty and academic research departments now exist. Much less basic research is being done in industrial research laboratories, and considerable development and short term research is being done in academia. I believe that this loss of balance creates a great danger for our country and the technological leadership we have experienced due to the healthy interplay between academic and industrial research that existed. I see no easy solution to this problem. NSF has had programs to stimulate joint academic and industrial projects, and academia has aggressively pursued industrial funding and cooperation, but these efforts have not, in my opinion, created a good balance between industrial research and academic research. Beyond the lack of balance, the decrease in basic research is an additional concern. As noted, this has almost disappeared in industrial research laboratories due to the competitive pressures, but the shift of grant funding for university research to more short-term and developmental projects has also shifted interest away from basic research in the academic computing community. I expect that this will create long-term problems in the U.S. dominance in computing. We see our technological lead being diminished by

other factors as well. More technological jobs are moving offshore. At first it was the manufacturing of electronic equipment that migrated to countries that had cheaper labor. More recently we see basic computer coding being done more cheaply in India, Russia, China and elsewhere. These are not elementary manufacturing jobs anymore, but have moved up into the “white collar” area. How far will this go in the future? As it moves, the U.S. may not be able to keep the lead it has had for so many years, and I see no real focus yet on this issue. The United States certainly has been a leader in innovation, and its policies have provided a good environment to encourage this. With continued strong governmental support for research, both academic and industrial, this can continue to be true. Service industries can also thrive, but I hope we will still be able to be leaders in developing new areas and specialties in scientific areas, especially in the computing sciences.