1991

PHILCO: Some Recollections of the PHILCO TRANSAC S-2000

Saul Rosen

Report Number:
91-051

PHILCO: SOME RECOLLECTIONS OF THE PHILCO TRANSAC S-2000

Saul Rosen

CSD-TR-91-051
June 1991

PHILCO: SOME RECOLLECTIONS OF THE PHILCO TRANSAC S-2000

Saul Rosen

CSD-TR 91-051
June 3, 1991

Abstract

In 1954 engineers at Philco Corporation invented the Surface Barrier Transistor, the first transistor that was suitable for use in high speed computers. Philco set up a computer activity, eventually a Computer Division, and in 1957 introduced the Philco Transac S-2000, the first large scale transistorized scientific computer system offered as a product by a computer manufacturer. In the Spring of 1958 I was hired by Philco to organize a programming systems department to provide software support for the new computer system. This paper presents part of the history of the Philco computer effort from the point of view of one participant. There were problems that were solved and difficulties that were overcome, and there were a number of successful installations. However, the Philco Computer Division did not have adequate resources to remain competitive in an area dominated by IBM, and Philco withdrew from the general purpose computer field in 1965.

PHILCO: SOME RECOLLECTIONS OF THE PHILCO TRANSAC S-2000

Saul Rosen
Department of Computer Science
Purdue University
West Lafayette, IN 47907 CSD-TR-91-051

June 3, 1991

Abstract

In 1954 engineers at Philco Corporation invented the Surface Barrier Transistor, the first transistor that was suitable for use in high speed computers. Philco set up a computer activity, eventually a Computer Division, and in 1957 introduced the Philco Transac S-2000, the first large scale transistorized scientific computer system offered as a product by a computer manufacturer. In the Spring of 1958 I was hired by Philco to organize a programming systems department to provide software support for the new computer system. This paper presents part of the history of the Philco computer effort from the point of view of one participant. There were problems that were solved and difficulties that were overcome, and there were a number of successful installations. However, the Philco Computer Division did not have adequate resources to remain competitive in an area dominated by IBM, and Philco withdrew from the general purpose computer field in 1965.

1 INTRODUCTION

This article is not meant to be a complete history of the Philco 2000 computer project. I doubt that the material for such a history exists, since corporations do not usually keep detailed records, except perhaps financial records, of their day to day activities. I don't know if anyone will make the considerable effort it would require to find and study the relevant documents that still exist, and to interview the surviving participants. Perhaps, with the growth of academic interest in the History of Computing, an enterprising graduate student might undertake to do that before it is too late. I myself cannot undertake that kind of historical research. Yet I think the story of Philco's early years in the computer field should be told, and I don't think anyone else will try to tell it. I have therefore decided to write as much of the story as I know, since I was deeply involved during the most important time. This article is a memoir, a history of the Philco computer division from the point of view of my own participation in it. I kept no notes or records or memoranda from those days. The only data that I have are some manuals and documents published then. My basic source, essentially my only source, is my own memory of what happened in the late 1950s and early 1960s. My memory is generally good, but the things I am writing about happened a long time ago, and there may therefore be some inaccuracies, and perhaps some exaggerations. Some people who were important in the effort may find their contribution slighted or even omitted. That can't be avoided in this kind of presentation.

1.1 The Surface Barrier Transistor

I was not there at the beginning, and I do not know the details of the decision process that led Philco to take its plunge into the computer field. I was working at the Moore School in Philadelphia in 1954, and I remember learning with some surprise that engineers at Philco had invented a new kind of transistor, the Surface Barrier Transistor, that would make it possible to use transistors in high speed scientific computers. Up to then transistors had been too slow for such use, and the first transistorized computers had therefore been intermediate speed computers designed for data processing applications. Philco, originally known as the Philadelphia Corporation, had been producing radios since their earliest days, and had done some development work in commercial television. A breakthrough in solid state technology was not the kind of thing one would expect from this stodgy old company.

1.2 Philco Enters the Computer Field: SOLO, the TRANSAC S-1000

I left Philadelphia and spent two years at Wayne (now Wayne State) University in Detroit. When I returned to Philadelphia in the Fall of 1956 as manager of the Eastern Applied Mathematics Section of the ElectroData Division of Burroughs Corp., Philco had come into the computer field - designing and building high performance transistorized computers. The area around Philadelphia had been the birthplace of the electronic digital computer, and some of the most important of the early computers, the ENIAC, the EDVAC, BINAC, and UNIVAC had been developed in Philadelphia. Burroughs had established its research division in the Philadelphia area, and the very influential IAS computer had been developed at the Institute for Advanced Study in nearby Princeton, New Jersey. It was thus possible to recruit a group of experienced computer design people for the Philco computer projects

The National Security Agency was actively supporting new developments that might produce faster, more reliable computers for cryptographic and other security work. They were interested in the Surface Barrier transistors, and in June 1955 entered into a contract with Philco to produce a computer that used these components. The project was called SOLO, since the idea was to provide a powerful workstation for individual users. The computer was later also called the TRANSAC S-1000. The design was based on an existing vacuum tube computer, the Univac 1103, another computer whose initial development by Engineering Research Associates in St. Paul, Minnesota, had been subsidized by the NSA. It is only in the past few years that security agencies have admitted to and publicized their involvement in the design of many of the early computers. The U. S. Navy was the official contracting agency for the S-1000. Surface Barrier transistors were used in a number of other early high speed transistorized computers, including the TX-0 at Lincoln Laboratories which was one of the first. They were used in the Univac LARC supercomputer.

2 INTRODUCING THE TRANSAC S-2000

Philco also contracted to design and build another, larger transistorized computer, the CXPQ for the Navy. Sometime in late 1956 or early 1957 the decision was made at Philco to design and build a large scale computer system that would be offered for sale as a product. Philco entered the general purpose scientific computer field. The new computer was named the Transistorized Automatic Computer, TRANSAC S-2000. It would be faster and more powerful than the IBM 704 and 709 that were then the standard large scale scientific computers, and it would have all of the advantages that transistors have over vacuum tubes. In retrospect this decision to attack IBM in an area in which it was so strong and firmly established seems like an excess of bravado. Philco executives must have believed that their Surface Barrier transistor gave them a big lead over potential competitors. They probably thought that IBM's huge investment in existing vacuum tube computers would keep IBM from moving rapidly into a new generation of computers. They certainly underestimated the resources that would be needed for a profitable incursion into the large scale computer field.

Henry Argento was vice president in charge of Philco's Government and Industrial Division and he brought in Herm Affel to head up the computer activity, later the Computer Division. The proposed product, the S-2000, had to be sold and John M. (Mac) Nisbet was hired as head of computer marketing. There were two major activities, engineering and marketing. Dr. Morris Rubinoff, who was on the faculty of the Moore School of Electrical Engineering of the University of Pennsylvania was head of Engineering. He was on leave from the University while he held that position. Rubinoff himself along with some of the other engineers responsible for the design of the S-2000 had been involved in the IAS computer project, and in many ways the S-2000 can be considered to be a modern (for its time) version of the IAS computer.

2.1 Some Features of the S2000 (Segal et al. 1958)

One of the advantages of the TRANSAC S-2000 for scientific computing was the use of a 48-bit word, as compared to the 36-bit word in the IBM 700 series computers. Floating point arithmetic used a 12 bit exponent and a 36 bit mantissa. Central memory was 32,768 words of core memory with an access time of about 10 microseconds. In the early advertising of the system there was great emphasis on the asynchronous nature of the design. Thus Philco engineers were building a 10 microsecond core memory based on the memories designed at MIT. If it turned out that they could get the memory to run with an 8 microsecond cycle, it would improve the performance of the system without requiring any redesign. On the other hand if it were a bit slower than 10 microseconds, the system would still work with a slight degradation of performance. This feature may have made it easier to move to a 2 microsecond memory that was soon needed to remain competitive with IBM after IBM introduced its 7090 system.

Each 48-bit word could contain two 24-bit instructions. The last eight bits of an instruction contained the command[1] The first four bits referred to an index register, or no index

[1]Address modification during the running of a program was used extensively in those early days, and for that reason I felt that the address should have been put at end of the instruction word and the command

register if zero, and the next 12 bits were the operand address. The design permitted up to 15 index registers, but all of the machines built had 8 index registers. These index registers were more like base registers, but they were not called that since this was before the IBM 360 made memory addressing through the use of base registers popular.

3 SOFTWARE FOR THE TRANSAC S-2000

The responsibility for software support of the new computer systems was in marketing. Mac Nisbet had come to Philco from the Burroughs Philadelphia computer sales activity whose offices were close to those of my own Electrodata Applied Mathematics Section. Applied Mathematics was an old name for our group, and most of our activities were in areas that would now be called software support. Nisbet and I knew each other; he was aware of the Datacode programming system I had produced for the Burroughs 205 computer (Rosen 1957), and he was apparently impressed with my technical expertise in the computer field. He needed someone to head a programming systems group and he called me and suggested that I apply for the position.

I was indeed ready for a change. Burroughs was not doing very well. They had decided to compete in the medium scale computer market with the Burroughs 220, probably the last completely new vacuum tube computer, introduced at a time when most users were hoping that their next system wouldn't use tubes. My Eastern Applied Mathematics Section was a bit of an extravagance. I was offered a chance to improve my own position in the company by moving out to the home office of the ElectroData Division in Pasadena. I was not enthusiastic about moving to California, and even though the position offered to me was technically a promotion, I did not consider it enough of one to warrant a cross country move.

I had become involved with ACM activities - first on the languages committee that eventually led to Algol, and then as Managing Editor of a new journal, the Communications of the ACM. Early in 1958 Ed McCollister, the director of marketing for ElectroData, visited me in my Philadelphia office. He told me that he could not offer me anything more than I had already been offered in Pasadena. He was resigned to the fact that I would not move from Philadelphia, and while he was willing to maintain the status quo, he felt that in view of my growing reputation in the field I would probably not be satisfied to remain very long in the position I then held.

Professor Saul Gorn of the University of Pennsylvania and Professor Alan Perlis of Carnegie Institute (now Carnegie Mellon University) were software consultants to the Philco computer project. They recommended the team of Anatol Holt and William Turanski for the position of Manager of Programming Systems. Holt and Turanski were two brilliant men who worked for John Mauchly's Univac Applied Research Center (UARC). They were the designers of the Generalized Programming (Fleximatic) system for the Univac I and II, and were building a more ambitious system, GPX (Generalized Programming Extended) for the Univac LARC computer. At Univac Mauchly's group was in strong competition with

at the beginning. I was told that the decision to place the command in the last eight bits had to do with convenience of engineering layout, and no one had thought it would make any difference for programming. It was only much later that I realized that this and other design decisions, including the 2's complement representation of numbers, followed the design of the IAS machine.

Grace Hopper's automatic programming group. Even though Mauchly's group may have had people with stronger technical and theoretical credentials, it seemed clear even then that Hopper's group was going to win out, and that some of the good people at UARC would be receptive to offers elsewhere. (In the next year I recruited several of them to work for me at Philco.)

Holt and Turanski had been working together for a number of years, and they would only be interested in an offer that would permit them to continue their effective collaboration. This might have been an excellent opportunity for Philco to start its programming systems effort with two very strong people, but Mac Nisbet didn't feel comfortable with the prospect. He wanted a manager for a group, not two managers. Also I think he felt that they were perhaps a bit too theoretical and research oriented. I recall that he told me he talked to them; I don't know if it was a formal interview or just a discussion. I don't think he talked to anyone else about the job. He invited me down to the Philco plant on Wissahickon Ave. in the Germantown district of Philadelphia and told me that the position was mine if I wanted it. The salary offered, $18,000 a year, was attractive in 1958. I accepted, but with some misgivings. Philco was going to have a difficult time and might not succeed at all in establishing itself in the computer field. The job I was undertaking was going to be extremely difficult. It was something I thought I could handle, and handle well, if the S-2000 project was really viable, and if I would get enough support in terms of manpower and other resources. Looking back on it now I feel that I was a bit naive; Quixotic might be the right word - I think that it is the right word to describe the whole Philco computer effort.

3.1 The Beginnings of a Programming Systems Department

I started at Philco in April 1958, and my misgivings were even stronger after my first few days at work there. During my interview visit I had been shown a computer that was close to completion. I assumed, perhaps I was even told, that I would soon be able to use it to start developing S-2000 programs. It was the CXPQ computer, and it was quite different from the S-2000. The first S-2000 wasn't nearly that far along, and it was going to be five or six months before there was going to be any computer to use for software development. The CXPQ, a one of a kind computer built on special contract with the Navy, was destined for the David Taylor Model Basin and would be shipped out as soon as it could be made to work.

There was no programming staff at all. I was really starting from scratch. There were two people tentatively assigned to the systems programming activity, but after a brief period of evaluation I suggested as tactfully as I could that they be assigned to other activities.

On one of my early days there things began to look brighter when I discovered that there really was a good systems programmer at Philco, working in the Engineering department. A young Philco employee, I. Bennett Goldberg, dropped in to talk to me. He was part of the S-2000 design group, and now that the central processor design was complete, or at least frozen, he looked forward to the possibility of joining my new Programming Systems group. I would have to get the Engineering group in which he worked to approve the transfer but he didn't think there would be much difficulty. I think that he had been a bit of a gadfly in the design group, pointing out subtle deficiencies that some engineers would rather have ignored. In any event there was no problem. Bennett Goldberg was a tremendous asset to the Programming Systems activity right from the beginning. Aside from being a brilliant and productive programmer, he was an invaluable resource person since he knew the S-2000 down to the smallest detail. He had a deep understanding of the machine and how to use it that I don't think anyone else had at that time.

At the beginning the major activity was recruiting, and we were able to attract some good programmers. Several of our programmers came from the Remington Rand Univac division Philadelphia operation which then employed a large number of systems programmers. People I interviewed told me that Remington Rand had had one of its periodic major reorganizations and the new regime there had announced that henceforth age and seniority in the company were going to be two of the most important factors in determining salary increases and promotions. Since most programmers were recent college graduates who had been with the company a short time their prospects there seemed limited, and a number of them applied for our openings.

3.2 Getting Started

How does one get started on the development of software systems for a newly developed large scale computer? The thing to do is to start with an existing computer, and to develop the initial software there. The terms "cross assembler" and "cross compiler" may not yet have been invented, but we were certainly aware of the concepts. The most sensible thing to do would have been to obtain the use of an IBM 704 for TRANSAC software development, but that was never seriously considered. There are several reasons why this reasonable approach did not seem reasonable to us then. The only way to obtain adequate use of a 704 would have been to order and set up and run our own 704 installation. That would have been an expensive, time-consuming and resource consuming operation. Also there was the feeling at Philco that IBM was the enemy. We were out to replace their 704's and 709's with our new computer. We didn't want to support them by buying their equipment. The less they knew about what we were doing the better, and we didn't want their salesmen and maintenance engineers around. There was an IBM 650 in the Philco accounting office, but we felt that the 650 was too different and too slow and too busy to be of any use. We decided not to try to obtain the use of any other computer. We would somehow make it on the 2000 itself. A paper tape reader was hooked up to the first running system. A short program was keyed in at the console to load programs in octal from paper tape. Only a few short programs had to be written in octal. One of the first of these was an elementary assembler that could then be rewritten in its own language. It could be assembled and then improved and reassembled to a level at which it could be used as the language for software development until a more sophisticated assembler was developed.

We started almost immediately on the design of the assembler that would be provided to customers. We were aware of the design of the first SAP assembler that had been done for the IBM 704, and some of our programmers had been involved with UNIVAC software, so we had a good deal of information about their elaborate multi-pass assemblers. Our design was a compromise. It was basically a two-pass assembler like the SAP assembler, but there were additional optional passes between the two main passes in which macro expansions could be carried out, and in which special code generators could be invoked (Rosen et al. 1959).

3.3 Three Major Philco Customers

Before we could deliver any computers it would be necessary to provide an Algebraic Compiler. As a representative of Burroughs I had been a member of the ACM language committee which sent a subcommittee to meet with a corresponding European committee in Zurich in the spring of 1958 to design the International Algebraic Language, Algol 58. The progressive thing for us at Philco to do would have been to lead the American computer industry by implementing Algol 58 which, some thought, was bound to replace Fortran and become the international standard. Some of the computer science people I had worked with on the Language committee expressed disappointment and disapproval when I announced that we would use Fortran as our Algebraic language. I don't remember whether I told them then that I really had no choice in this matter. Several of our early customers had been involved in the development and use of Fortran from its inception, and they expected us to provide them with a way to continue using Fortran on the TRANSAC S-2000.

By the time I joined Philco three companies with a great deal of prestige among computer users, United Aircraft and the Westinghouse Bettis and General Electric Knolls Atomic Power Laboratories, had committed themselves to purchase TRANSAC computers. Westinghouse Bettis and GE KAPL are research laboratories that were set up under contract with the U.S. Naval Reactor Board to develop nuclear power plants for the Navy's submarine fleet. They have to be able to share computer codes, and have therefore always ordered the same kind of large scale computer equipment.

These three customers all were important users of the IBM 704, a computer that had almost a monopoly in the large scale scientific computer field in the mid 1950's. They were active members of the SHARE organization of users of the 704. They consulted with each other and participated jointly in the drastic decision to move away from IBM computers and to cast their lot with Philco, a newcomer in the computer field. Why did they do so? I imagine that they were disappointed by IBM's recently announced successor to the 704, the 709 system, which had some nice new input-output features but which was still essentially a 704. The 704 with its magnetic core memory and its index registers and its hardware floating point arithmetic was a great computer when it was introduced in 1954, but it was approaching obsolescence in 1958. It was a vacuum tube computer, and its 36-bit word length was inadequate for many applications. The Philco TRANSAC S-2000 would be a much more modern machine. The 48-bit word length was attractive, and the TRANSAC would outperform the 704/709 in many important ways. I think the most important factor was that the 709 still used vacuum tubes, and the Philco TRANSAC would permit them to move into the modern age of transistors.

I think that another important factor in their decision to move away from IBM was their displeasure with the philosophy and the direction of software development at SHARE and at IBM. Roy Nutt and Bill Melcher of United Aircraft had developed the Share Assembly Program (SAP) that was in use at almost all IBM 704 installations. For the 709 a SHARE committee dominated by representatives of the west coast Aerospace industry had designed the Share Operating System (SOS) and the Share Compiler and Translator (SCAT) which were going to be the standard software systems on the 709, and to which IBM had committed a major implementation effort. SCAT was a very complicated assembly system that represented a major departure from the simple elegance of SAP. Also the design of SOS was incompatible with Fortran which had only recently achieved production status. Roy Nutt had been an important member of the IBM Fortran project (on loan from United Aircraft). Herb Bright, who was the technical person in charge of computer procurement at Westinghouse Bettis, was one of the first enthusiastic users of Fortran. They were disappointed that the SOS project had apparently ignored the existence of Fortran. They also realized that the existence of Fortran would make it possible to move many programs from the IBM 704 to the TRANSAC S-2000 without a major reprogramming effort. I don't know if IBM ever considered the Fortran language to be a proprietary product. They raised no objection to our use of their language, and as far as I know we at Philco were the first group to implement a compiler for the Fortran language on a non-IBM computer. Many others have done so since. It turned out that the SOS effort was not very successful, and until it was replaced by IBSYS most users of the 709 (and 7090) used a much simpler Fortran Monitor System based on SAP and Fortran.

3.4 ALTAC, A Fortran Language Processor

The software system that I envisioned for the S-2000 would be based on the powerful and versatile assembly system. Higher level language processors would be translators into assembly language. The only such translator on our original schedule was Altac, a translator from an Algebraic language into TAC Assembly language (Rosen and Goldberg 1959). The Algebraic language was to be Fortran, or as close to Fortran as we could make it. We expected that in the future there would be other languages and other translators from those languages into TAC.

While with Burroughs, I had studied the Fortran language and had given considerable thought to the possibility of implementing Fortran on the new Burroughs 220 computer. I had begun the design of a translator and had convinced myself that I would be able to produce a translator from Fortran into a suitable intermediate language in 6 to 9 months of effort, and I had started the preliminary design of such a translator. I don't know how well I would have carried out that project. It was still in its initial stage when I joined Philco, and there I could no longer devote myself to that kind of activity. I soon became convinced that our first programmer, Bennett Goldberg, would be able to carry out a Fortran translator project, probably much better than I would have been able to, even if I could have devoted myself to just that. That became his assignment, and he carried it out far better than I had any right to expect.

The approach used, translating a higher level languages into assembly language and then using the standard assembler to produce object modules, seems very attractive. It has been used many times since, and the concept keeps recurring, since it seems so natural, but it had serious drawbacks on a magnetic tape system. It made compilation take much longer than it should have taken. Assemblers, especially those that provide extensive macro expansion facilities, are notoriously slow. The attractive modular approach that produces standard assembly language first may take as much as twice as long as would a well-designed compiler into an intermediate language that transforms easily into machine code. These were important considerations on magnetic tape systems where the time needed for multiple tape passes (including rewinds) made compilation a slow process. Even though we had a relatively fast tape system our compilation times were much longer than I had originally (perhaps naively) expected.

3.5 Compatibility Considerations

There was a great deal of discussion back then about the relative advantages of compilation over interpretation. We decided that we would go as far as possible with compilation, even in the input-output area in which the IBM Fortran interpreted format statements at run time. As a result we ended up generating large amounts of input-output code. We had overlooked an important advantage of interpretive systems, the fact that the interpreting code is used over and over again and need not appear in memory repeatedly. We had sacrificed a large amount of space for a relatively small gain in execution speed. This was an especially serious problem in connection with some large Fortran source programs we had to compile that contained literally hundreds of READ, WRITE and FORMAT statements, far more than we had ever expected to see.

Although it was not initially part of the language, the fact that the IBM Fortran interpreted Format statements at run time made it possible for IBM to add a feature that permitted users to read in Format statements at run time. That was a feature we could not offer in our Altac system. It is one of many illustrations of the fact that if you want to be compatible you must consider compatibility of implementation as well as compatibility of language (Rosen 1961).

Another compatibility problem occurred in connection with subroutine parameters. Early in the design we had made a decision which limited the number of parameters that could be passed to a subroutine to 32, and that was a limit that could not be increased without a major redesign. It was only quite a bit later that we found out that IBM Fortran permitted up to 64 parameters. It really doesn't make sense to pass large numbers of parameters to a subroutine, especially in Fortran where a full list of parameters had to be used with every reference to the subroutine. The COMMON statement of Fortran makes it possible for routines and subroutines to share storage, and the use of very long lists of parameters was almost always an indication that the programmer did not understand how to use COMMON. There were many such programmers. We were surprised and sometimes amazed by the Fortran programs we saw. It took some time for us to become fully aware of the fact that we were going to have to deal with programs that were written by people who knew little or nothing about programming, and who neither knew nor cared about the effect of their programming practices on the efficiency of either the compilation or the running of their programs.

3.6 The First S-2000

The first S-2000 gradually took shape on the production floor. It had the amateurish look of computer products in development in those days, and I was worried about the impression we would make on visitors who represented sophisticated customers, people from United Aircraft and Westinghouse Bettis and G. E. KAPL. Actually I received reassurance from those customers, who had participated in the IBM 701 and 704 developments when they were in similar preliminary development stages. Eventually it became clear that we had to have a running machine soon if we were to come close to meeting the delivery schedules, and we agreed to demonstrate the CPU in operation by a rather arbitrarily chosen date in the Fall of 1958. The system wasn't really working by the agreed upon date, but it was very close. I remember the group of customers and Philco people standing around the machine while our engineers fussed over it trying to get it to run. A cathode ray tube display hooked up to the memory access register showed the pattern of memory reference, which seemed mostly random noise. After what seemed a very long time during which we watched meaningless green lines running through the display, one of the engineers had a flash of insight - a little more capacitance on one of the signal lines might do it. He found a small capacitor of the right size and held it in place across two wires. The CRT suddenly cleared up and settled into a regular pattern. There was no question about it. The machine was running through an iterative program. We had a computer, not just a collection of components. From that time forward we never doubted that the S-2000 would run. We still didn't know how well it would run.

Things moved ahead quickly after that. We soon had a rudimentary assembler running, and the machine could actually be used to run programs. The tapes and the drum worked as did the first UBC (see below). The first S-2000 was considered ready for installation, not yet for an outside customer, but for Philco's Western Development Laboratories in Palo Alto. The machine was shipped, and our software development efforts were left high and dry - without any software development machine at all. As soon as that first machine was installed and running I decided to send Bennett Goldberg out to Palo Alto to continue with the Fortran development. The rest of us would wait until the second main frame, the United Aircraft machine, would be ready for advanced hardware debugging and also, incidentally, for software development.

3.7 Magnetic Tapes

Large scale first and second generation computers were magnetic tape machines. The first TRANSAC S-2000 had a magnetic drum system on-line, but that was the only 2000 that had one, and the magnetic drum soon disappeared from the product line. Magnetic tapes provided peripheral storage as well as direct input to and output from the computer. An offline device (the UBC or Universal Buffer Controller) provided card-to-tape, tape-to-card and tape-to-printer utilities. An efficient smooth-functioning tape system was essential to the operation of a large computer installation. The design, operation, and maintenance of such a tape system for the S-2000 turned out to be extremely difficult, and the efforts in that direction, though often heroic, were only partially successful.

There were very few companies making high performance digital tape units in those days. I don't know what alternatives had been considered, but by the time I joined Philco the decision had already been made to use Ampex FR-300 tape drives. Ampex had made its reputation in the area of analog tape recording. They were famous for the tape system that they had developed for recording programs for television. The FR-300 was their first venture into digital tape systems for high speed digital computers. The FR-300 was designed to provide transfer rates of 90,000 characters per second and had other desirable characteristics. Philco was only one of a number of manufacturers that planned to use these new tape units.

The tapes were 1 inch wide, with two six-bit characters recorded in each frame. Fixed length blocks were used, with block start and block end patterns, and clock pulses for each character prerecorded on each reel. In order to avoid skewing problems the two characters in a frame had separate clock bits; they also had separate parity bits.

The need to prerecord timing pulses and beginning and ending block markers turned out to be a nuisance. On the early systems large amounts of system time were used to edit blank tapes. I don't think the system designers realized how many tapes there would be or how long it would take to edit a tape. Eventually off-line pre-editing of tapes was provided as a feature of the Universal Buffer Controller.

The prerecorded blocks made it possible to do selective alteration of tapes, i.e. tapes already written could be modified by writing over one or more selected blocks inside an existing file. Most tape users considered this a risky practice, and this feature was rarely used.

Tapes could be written only in the forward direction, but they could be read either backwards or forwards. The ability to read tapes while they are moving backwards is essential to the implementation of efficient tape sorting programs. The tapes are divided into two groups. In an initial pass sorted strings are written on one group of tapes. On the next pass that group is read backwards, and the strings are merged into longer sorted strings on the other group of tapes. The roles of the two groups are then interchanged and the merges continue until the file is converted into a single sorted string. There is no time wasted in rewinds between merge passes. The backward read feature existed on the UNIVAC I, and made that machine a better sorting machine than some of its otherwise more powerful competitors. The IBM systems of that generation did not have this capability, and most computer users were therefore unaware of its value.

I think that enough time has elapsed to permit me to write about the early Ampex tape units in a way that I would not write about contemporary products. Companies and their products change completely over periods of 30-35 years. The reader should certainly be aware that my comments about products delivered then have no relevance to the Ampex Corporation or to its products in the 1990s.

Having said all that, I will go on to say that the performance of the FR-300 was very disappointing. Cornelius Eldert Jr. was the engineer in charge of the tape system for the S-2000. I had known Neil Eldert when we were both at the Moore School several years earlier, and I would occasionally drop by and chat with him about the tape system. Shortly after the first tape drives were delivered he seemed completely discouraged. He had samples of tape that had been stretched and twisted while in use on the tape drives, and others with edges frayed and damaged so as to make them useless. It was clear that a great deal of work would have to be done by Philco and by Ampex before those tape units could be used as part of a working computer system.

The tape controller that was under development had a number of interesting and innovative features. It was designed to switch up to 16 tape units onto up to four channels which had direct access to the central magnetic core memory. It provided a transistorized crossbar switch. When an instruction like Read or Write was addressed to a tape unit, the hardware would seek a non-busy channel, and if one was available, would switch the tape to that channel. The tape controller was then at the forefront of the state of the art of high speed transistor technology.

Very few four-channel controllers were built. One that I know of was installed on an S-2000 at United Aircraft Corporation in Hartford. Most of the controllers that were built had two channels. In a typical tape processing operation two tape channels are needed in order to be able to read and write simultaneously. It was not easy to see how to make effective use of four channels. We thought about the possibility of exotic sorts in which two independent merge passes would be executed simultaneously, but no one ever tried to write such sorts. The additional hardware redundancy in the case of equipment failures may have been the most important practical value of the extra channels.

Work continued on the controller, and on trying to convert the tape drives into reliable components. The customers that we had were of course very much interested in the progress of the tape system. They were aware of some of our problems with the drives and a bit worried about our ability to overcome them.

There were contingency clauses in most of the system orders that we had. Westinghouse-Bettis insisted on adding a tape system performance requirement: that by some specified date we should be able to demonstrate a working tape system. The system would be deemed to be working if it could successfully run a set of tape system test programs developed by Westinghouse-Bettis programmers. The test programs would be submitted to us well in advance of the actual test, so that we would know the performance that they considered acceptable, and would be able to negotiate changes if we felt the tests were unreasonable. There was a good relationship between the technical people on both sides. The Westinghouse people didn't want to push our system beyond the then current state of the art in tape systems. They only wanted to know if the tape system worked.

The tape test programs came and after he had some time to study them, Neil Eldert assured me that we would be able to run them successfully. I was relieved. We were going to have a working tape system. But Neil told me that was not what he meant. He could (and did) tune and adjust the tapes to pass that test, but he didn't think the FR-300 tape units would ever perform satisfactorily at customer installations.

Eventually Ampex replaced the FR-300 tape drives with their newer and better TM-2 drives. Then George Cogar, a very resourceful engineer, came to Philco from Univac, and helped get the tape system to work satisfactorily. Cogar's stay with Philco was a brief interlude in an interesting, perhaps even legendary career in the computer field. Some interesting comments about George Cogar can be found in (Lundstrom 1987). Even after the tapes worked adequately from the point of view of the customer, they remained a problem for Philco, since they required so much preventive maintenance and such frequent replacement of components that it was not practical to charge the customer their full maintenance cost.

4 THE MODEL 211 OF THE TRANSAC S-2000

The system delivered to Philco's Western Development Laboratories was officially known as a TRANSAC S-2000 model 210. It was the only model 210 built. New and better transistor technologies made the Surface Barrier Transistor obsolete. Surface Barrier Transistors had to go through careful evaluation and selection procedures, since there were significant differences in performance characteristics among the individual transistors used. You couldn't simply replace one transistor with another with the same part number and expect identical performance. The newer Field Effect Transistors provided more uniform performance, and other desirable characteristics. Philco's version of these transistors were called Micro-Alloy

Diffusion Transistors (MADT's). A considerable redesign had to be undertaken in order to build the model 211, the MADT version of the S-2000. The redesign, and the incorporation of the new higher performance transistors produced a significantly improved computer.

4.1 Progress

My recounting of the difficulties that we faced may give a false impression about the morale of the people involved in the Philco computer program. My recollection is that morale was high, especially in the early days. There was a feeling that we were doing great things. We were the first on the market with a large scale high speed transistorized computer. We had good engineers and we were building a good programming staff. The big names among our first group of customers proved that we were not deluding ourselves. An order for a TRANSAC S-2000 from the Government of Israel attested to our international renown. We had a long list of prospects, and some of them eventually became customers.

The completion and delivery of the first systems was an exciting series of events. The computer activity was elevated to a new status, becoming the Computer Division. An attractive new building was built for the Computer Division in Willow Grove, a suburb north of Philadelphia.

The need to reengineer the model 210 into the 211 without slipping the delivery schedule posed a real challenge for the engineering staff. The customers were told that the introduction of the new MADT transistors would only require minor changes, but the effort required was not minor. Work went on all day and all night. In some cases people worked straight through for 24 hours at a stretch. There was one instance in which one of the senior design engineers passed out at his desk from exhaustion and had to be carried out to a local hospital to recover. I think that he was back at his desk the next day. I was impressed by the dedication of so many of the staff.

Our programming people were mostly later arrivals in the Philco computer effort, but most of them felt the excitement of being in on a big prestigious new venture. I don't think there were any who looked on their work as only a job that had to be done. Programmers often worked many extra hours, without pay, in order to meet self-imposed deadlines.

I mentioned earlier the fact that we were able to recruit a number of people from the faltering Univac Applied Research Center. One of these was Dr. Louis R. Lavine. (The doctorate was in Chemistry.) After a while Lou became assistant manager of our department. Lou then helped us recruit John Guernaccini who had worked with him at Univac. John had been in charge of implementing the GPX system at Univac, and he took over the completion of our TAC (TRANSAC Assembler and Compiler) system. John was very good in a number of ways that it is hard to explain. A great deal of progress had been made on TAC but it seemed to be stuck short of completion. When John Guernaccini took it over the project seemed to take on a new life. We soon had a version that was adequate for use with ALTAC. Other, more complete versions followed. In the discussion of Fortran that follows It should be recalled that we could not have had a working version of Fortran (ALTAC) until we had a working TAC assembler.

4.2 Program Debugging

The first model 211, the one being built for United Aircraft, was out on the floor for system checkout and we arranged to get time for software testing and debugging. It is hard to imagine the conditions under which this was done.

I can remember the difficulty I had convincing some of our programmers that the operator we had hired was competent to mount their tapes and run the machine. Some of those who had been trained on the Univac I wanted to remain in complete control of their tapes and their runs. One or two incidents in which program tapes were mounted as scratch tapes didn't help, even though it turned out that it was usually a programmer and not the operator who was at fault. We gradually moved to a semi-production environment, but the fact that the hardware was only partially debugged made this difficult.

We were constantly harassed by the early difficulties with the tape system. There was a time during which the tape units were stretching and mutilating tapes to the extent that each time you mounted a tape there was a good chance that would be the last time that tape could be used. How do you back up data on such a system? Once there was a power failure in the middle of a run and all 7 tapes on the 7 drives in use were stretched and made unusable. It got so bad that I remember one time issuing a directive to our programmers not to mount any tapes until we received assurance that some of the more blatant problems had been solved. Management and engineering people assured me that the problems couldn't really be that bad, and I assured them that they were. The engineering people worked on the tapes, and they did get better, but it was a long hard process.

Eventually that machine was working well enough to be shipped, and our software development machine disappeared again. We had to send a group of programmers up to United Aircraft in Hartford in order to have a machine to use. That was a hard cold winter in New England.

The next big push was on the Bettis and KAPL machines. These were being purchased by the Naval Reactor Board, and they were tough customers. They insisted on penalty clauses, so-called liquidated damages, in the event of late delivery or non-delivery of equipment. The first of the two machines was to go to KAPL, and penalties of $1000 per day would be paid in the event of late delivery.

I was dismayed to learn that our contract people had agreed that delivery of the system would include the installation of running hardware along with a running Fortran system. Their definition of a running Fortran system was that the system had to be able to compile and run a program called TIGER-2, a very large elaborate heat transfer program. GE KAPL would pay no rent on the system and Philco would have to pay them $1000 per day in penalties until that program could be successfully compiled and run.

I complained that this would put our programmers under unreasonable pressure. We had been trying to produce software under impossible conditions - with no hardware system of our own, on test systems that weren't fully operational. We had made a great deal of progress that I could point to with considerable pride, but not enough to be able to deliver a full scale production Fortran compiler by the agreed upon date.

I asked our management to delay shipping the system so that we would be able to debug the Fortran compiler. One might think that the customer wouldn't want the machine delivered without working software. They insisted on immediate delivery. They felt that they could do program development with buggy software. The fact that we couldn't compile and run the acceptance test program wouldn't keep them from compiling many other programs. The fact that our programming systems group would not have a machine to complete the debugging of the Fortran compiler was our problem. We could come up to Schenectady and get some time on their machine.

The system was shipped and several of our programmers moved along with it and settled down in a motel in Schenectady. They had the system at night for program development and debugging. After a few weeks everyone's nerves were on edge. I can remember a call from our legal department informing me that someone was threatening to sue Philco because we were driving their daughter to the verge of a nervous breakdown. After a while Jim Painter who was in charge of the debugging crew in Schenectady and Bennett Goldberg who had remained in Willow Grove were no longer on speaking terms. I remember one situation in which I was at a computer conference in California and had to stay on the phone for several hours, calling first one and then the other in order to provide a necessary communication link between them. The debugging process went on for several weeks until finally Jim Painter called me and told me that he had been able to compile and run the acceptance test program, but only after he had made a very slight change to the Fortran program. Our Fortran compiler had a bug in its handling of indefinite repeats in FORMAT statements. It would require some thought and some time to fix it. If he could just change one FORMAT statement to call for a large fixed number of repeats instead of an indefinite number of repeats the program would run. I told him to go ahead and make the change and run the acceptance test and come back home. I felt a bit dishonest, but we later did our best to fix that bug and other remaining bugs as quickly as possible.

5 COMPETITION FROM THE IBM 7090

Early in 1958 the Ballistic Missile Early Warning System project (BMEWS) issued a request for bids to supply large scale transistorized computer systems to be installed in Greenland and at other sites. From the point of view of those involved in preparing the Philco proposal the TRANSAC S-2000 would have been the best, perhaps the only choice as the computer for BMEWS. The only large scale transistorized computer that IBM had under development was the Stretch supercomputer that would be far too big and expensive and couldn't possibly be finished in time for the BMEWS project. I was therefore disappointed when I heard through a friend who was a consultant on a project related to BMEWS that IBM had been selected to supply the BMEWS computers. Some of the details that I learned then and passed along to Philco management made it clear that the computer that IBM planned to produce for BMEWS would pose a serious threat to the whole program at Philco. Sylvania was doing the BMEWS programming, and IBM offered to provide vacuum-tube 709's as interim computers on which programs could be developed and tested. IBM promised that before the end of 1959 they would be producing transistorized computers that would be logically identical to the 709, with a clock cycle one fifth of that of the 709. Its central processor would thus be five times as fast as that of the 709. The 12-microsecond core memory of the 709 would be replaced by a 2.4-microsecond core memory on the new 709TX. The Stretch specs called for a 2-microsecond memory storing words consisting of 64 bits plus 8 parity bits each. This could be converted into a memory storing pairs of 36-bit words for the 709TX. The transistor technology developed for Stretch could be adapted for use in the new machine.

The machine that IBM delivered was even faster than the one promised, since the memory cycle on the delivered machine was 2.18 microseconds. IBM initially hoped that the 709TX could be offered as a special product that would be available only to the armed forces. However, customers outside the government were soon aware that IBM had a computer compatible with the 709 and more than five times as fast, that they were selling to the government at a price only slightly higher than that of the 709. Before IBM delivered the first 709TX in November, 1959 it was renamed the IBM 7090, and was offered to all customers as part of the IBM product line.

After the initial disappointment that Philco computers were not selected for BMEWS the official Philco reaction was skepticism about IBM's ability to deliver the machines they had promised in the time and with the performance that they had promised. There were rumors of difficulties within the IBM project, and there were some real difficulties and delays and reports of poor performance on the early delivered 7090's. But there never was any doubt that IBM would work out the problems and the difficulties, and that even though the TRANSAC S-2000 had some advantages, the IBM 7090 would be a much faster computer.

Philco's 10-microsecond memory cycle looked bad compared with the 2.4-microsecond memory promised by IBM on the 7090, and a project was started to develop a 2-microsecond core memory. Philco had some very good engineers, and they soon had some demonstration circuits in which magnetic cores were being switched in a 2 microsecond cycle. These were shown to customers to reassure them that Philco too had the capability to develop very high speed memories. I don't know if there was any time limit set then, but Philco was committed to produce a 2-microsecond memory for TRANSAC.

There was talk about the fact that the S-2000 was an asynchronous machine which could run with a faster memory with little or no redesign. There was a plan to replace the core memories on some of the model 211 computers after delivery with faster, 2-microsecond memories. I don't know if this was ever done. I do know that the Philco Computer Division did not produce any 2-microsecond memories. The 2-microsecond memories that were used on the model 212 were manufactured by Ampex.

6 MY DEPARTURE FROM PHILCO

I only stayed with Philco for a little over two years. I left feeling proud of what I had accomplished there. I had started from scratch and had built up a good programming systems department. From the start we had dealt with some sophisticated customers, and we had gained their respect. We had designed and built and delivered software products, an assembler, a Fortran compiler, subroutine libraries, and service routines, that were being used by those and other customers on a daily basis. I received praise and good increases in pay from Philco management. Why then did I leave? Looking back after 30 years it is hard for me to answer that question.

My reasons for leaving were complex. One major factor was my loss of confidence in the long term prospects of Philco in the computer field. That kind of confidence was needed in recruiting good people for our programming staff. It was needed in contacts with prospective customers who had to be convinced that we were and were going to remain the best in the computer field.

All sales organizations lie to their customers and I don't think we were any worse than others. But I found it very unpleasant to be in a situation in which I had to make promises to customers that I knew I couldn't keep, and try to deliver on impossible promises made by salesmen in pursuit of orders. One example is the case of a generalized sort that we needed in order to sell an S-2000 as a data processing system for a Navy accounting center in San Francisco. Our salesman assured the customer that we could provide the sorting system that they would need in about six months, but the customer wanted to talk to me directly. All of our programmers were over-committed to ongoing projects and there was no way they could undertake a major new project. Our only hope was a new programmer whom we had just hired. He was a young man with some system programming experience. I asked him if he thought he could do a generalized sorting routine, and he assured me that he could. I asked if he would be able to do it in six months. He thought that was quite feasible. I didn't delve any further. I assured the customer that we had a sorting expert on our staff who had told me that we could deliver the needed sort in six months, more or less. The sale was made, and we all celebrated. It turned out that the new programmer knew very little about sort routines and couldn't even get started. I didn't really know a great deal about them myself. We did eventually start up a sort project under Lou Lavine who had some knowledge of sorting from his experience at Univac, and the customer did get the needed sorting routines, but more than a year later than I had originally promised.

Lou Lavine took over my job when I left Philco. I remained in the Philadelphia area working as an independent consultant for about 2 years after I left Philco, and I remained in close touch with the Philco Computer Division. The programming staff could not grow fast enough to keep up with the need for new software, and several major projects were contracted out to companies that were pioneers in the new and rapidly growing software industry. Philco had been invited to participate in the design of the new Common Business Oriented Language (COBOL), but we had not done so because I did not then feel that I could allocate any personnel to that project. However, once COBOL 60 had been specified the Defense Department announced that a COBOL compiler would be required with all future computers that they purchased. I remember talking to Bennett Goldberg about setting up a COBOL compiler project but he was not enthusiastic about it, and we just didn't have the manpower for it. After I left the decision was made to enter into a contract with Computer Sciences Corporation (CSC) for a COBOL compiler for the 2000. Note that Roy Nutt, who had been very much involved in the United Aircraft dealings with Philco, had left United Aircraft and was one of the three founders of CSC. Later, when IBM released Fortran IV, CSC wrote a Fortran IV compiler for the Philco 2000. Other software companies were called upon to help when Philco received a contract to supply a large system for NORAD, the North American Air Defense System at Colorado Springs.

7 MODEL 212 OF THE TRANSAC S-2000

There had been plans to produce a more powerful machine to follow after the model 211 even before the advent of the 7090. The much faster machine, compatible with the 211 and able to make full use of the proposed 2-microsecond memory was called the model 212. Some of the speedup was achieved through instruction lookahead, a feature now called pipelining, that IBM introduced on their Stretch computer but did not use on the 7090. The first 212 was delivered early in 1963, and it was a very powerful machine. By then Philco was also producing the Philco 1000, a small character oriented data processing machine analogous to the IBM 1401. It was in no way related to the old TRANSAC S-1000. The new Philco 1000 could be used effectively as an adjunct to the 2000 for off-line peripheral operations such as card to tape and tape to printer.

For a time the Philco 212 was probably the most powerful scientific computer on the market. Two of the early 212's delivered replaced the model 211's at Westinghouse Bettis and GE KAPL. In a conversation I had years later with Ben Mount who had been in charge of the computing system at Bettis he praised the performance of the model 212. He said it was so good that it may have made their whole involvement with Philco worthwhile.

The supercomputer of the mid 1960s was the Control Data 6600, originally developed for the Livermore Laboratories, and later offered as the premier product of the growing Control Data Corporation. It was offered to Bettis and KAPL as a replacement for the Philco machines with a rash promise that their major programs written only in Fortran would run on the CDC 6600 at least five times as fast as they ran on the 212, and this in spite of the fact that the 212 programs had some machine language inserts that speeded up the inner loops. Years later, when I was in charge of a CDC installation at Purdue I heard that some of the excellence of the code generation of the CDC 6000 Fortran compilers was due to heroic efforts on the part of software people at CDC to meet the performance requirements of the Bettis-KAPL contract. I was also told by Jack Brennan, a former Philco computer salesman, that CDC had apparently not been aware that Philco had developed a high speed multiply option that was installed on the 212's at Bettis and KAPL. The performance of the 6600 was very impressive, but CDC could not meet the requirement that they run the specified programs five times as fast as the 212's at Bettis and KAPL, and they had to renegotiate the contracts at a reduced price.

8 THE MERGER WITH FORD, THE END OF THE TRANSAC LINE

At some time, I think it was in 1961, the name TRANSAC was dropped, and the name of the main product line became the Philco 2000. Some time after that the Philco Corporation was taken over by (merged into) the Ford Motor Company. There was temporary optimism in the computer division. Philco had never had the resources to become a serious contender in the large scale computer field. Ford certainly had the resources to compete in any field in which it wanted to invest. However, Ford management decided, probably wisely, that they did not want to try to compete with IBM as a manufacturer of large scale general purpose digital computers. The proceedings of the 1964 Fall Joint Computer Conference contains an article about a Philco 2000 model 213 and about even more advanced systems to follow (Bright 1964), but no 213's were ever built, and the computer division was phased out shortly after that. Several of the programmers I had brought into Philco from Univac: John Guernaccini, Arnie Shapiro, and Hal Siegal, then formed a company they called Keystone Computer

Associates. They prospered by offering software services to former Philco customers and to others. I left the Philadelphia area in the fall of 1962 to join the newly established Department of Computer Sciences at Purdue University. About a year later Lou Lavine left Philco and joined the Educational Testing Service in Princeton, N.J. Herb Bright, who had left Westinghouse and joined Philco in 1961, replaced Lou Lavine as head of the Programming Systems department.

Some time after the demise of the computer division at Philco Herb Bright moved to Washington where he organized Computation Planning Inc., a new company that specialized in the area of computer security. I was sad to read in an obituary published in Vol. 10 No. 3 (1988) of the Annals of the History of Computing that Herb Bright died on November 28, 1986. I was also saddened by an obituary in Vol. 12 No. 4 (1990) of the Annals that pointed out that Roy Nutt died on June 14, 1990. Neither obituary mentions the fact that Herb Bright while at Westinghouse Bettis, and Roy Nutt (along with Walt Ramshaw) at United Aircraft were essential participants in the effort to establish the Philco TRANSAC S-2000 as a factor in the large scale scientific computer field. They became customers before there was a working S-2000 system, and they showed confidence in the Philco program in spite of the difficulties that it encountered. They and their organizations offered advice and encouragement and criticism that contributed greatly to the early successes of the program. It is no reflection on them that the effort was not ultimately successful.

References

1. Bright, H. S., 1964. "A Philco Multiprocessing System." AFIPS Conference Proceedings Vol. 26 Part II, 1964 Fall Joint Computer Conference pp. 97-141. Spartan Books Inc. Washington, D.C.

2. Lundstrom, David E., 1987. "A Few Good Men From Univac." MIT Press

3. Rosen, S., 1957. "Datacode, a Compiler for the Datatron 205." Published by the Electrodata Division, Burroughs Corporation.

4. Rosen, S., 1961. "Altac, Fortran, and Compatibility." Preprints of Papers presented at the 16th National Meeting of the ACM.

5. Rosen, S., J. H. Brown, and Carl Calo., 1959. "TAC, The Transac Assembler-Compiler." Preprints of Papers presented at the 14th National Meeting of the ACM.

6. Rosen S. and I. B. Goldberg., 1959. "Altac, the Transac Algebraic Translator." Preprints of Papers presented at the 14th National Meeting of the ACM.

7. Segal, R. J., J. L. Maddox, and P. Plano., 1958. "Performance Advances in a Transistorized Computer System: The TRANSAC S-2000." Proceedings of the Eastern Joint Computer Conference. Dec. 3-5, 1958. pp. 168-174.