Early Computers in The Netherlands

Eda Kranakis
Wetenschapsdynamica, Universiteit van Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam

1. Introduction

Several computer development projects were established in The Netherlands shortly after World War II, including one at the Mathematical Centre (MC) in Amsterdam, one at the PTT (Post, Telephone, and Telegraph) in the Hague, and one at the Philips Company. These programs led to the construction of some 10 different machines up to 1960. Two of the machines were marketed commercially - that is, multiple copies were produced and sold internationally. The aim of the present paper is to review the history of the early Dutch computers, and to explain the differing contexts in which they were developed.

2. The Mathematical Centre program

2.1. The ARRA

The MC program must be viewed within the context of the goals of that institution, which was a product of institutional reorganization and renewal following World War II. The concept behind the MC was quite radical in the sense that what had traditionally been a 'pure,' academic field - mathematics - was now to be placed immediately at the service of science and industry. One division within the new centre was the Rekenafdeling, or Calculating Department. Here, for a fee, any business or industry could have scientific or technical calculations carried out.[1]

For the first several years, the calculating equipment in the Rekenafdeling consisted of desk-top accounting machines mainly operated by women. The often complex problems posed by clients were broken down into smaller units and these were farmed out to the women at the machines (the process has been described as a kind of primitive programming). As an example of the type of calculations done, one problem concerned the landing of aircraft on aircraft carriers. The trajectories of the flights had to be carefully analyzed, so multiple photographs were taken of planes as they landed, about 20 photographs per second. From these photographs, the precise trajectories of the planes had to be calculated, in the form of lists of coordinates in three-dimensional space. Each trajectory was said to involve about 200 sets of coordinates, and each set of coordinates took about 2-3 minutes to figure.[2]

DEVELOPMENT OF COMPUTERS IN THE NETHERLANDS

Year	TH Delft	PTT	MC	Philips
1944	Plan			
1945				
1946				
1947	ARCO		Diff. Anal.	
1948	ARCO		Diff. Anal.	
1949	ARCO		ARRA-I	
1950	ARCO	ZERO	ARRA-I	
1951	ARCO	ZERO	ARRA-I	Prel. Stud.
1952	ARCO	PTERA	ARRA-II	Prel. Stud.
1953	ARCO	PTERA	ARRA-II	PETER
1954	ARCO	ZEBRA	FERTA, ARMAC	PETER
1955		ZEBRA	FERTA, ARMAC	PETER
1956		ZEBRA	X-1	PETER
1957		ZEBRA	X-1	PASCAL
1958		ZEBRA	X-1	PASCAL
1959			X-1	PASCAL
1960				

From the beginning, however, the founders of the MC were aware of the importance of computers and in fact had decided that such a machine was essential for the Rekenafdeling. But computers could not be bought commercially at that time, so it was decided that one should be built. The computing department therefore came to have two missions: to do mathematical calculations and to build computers. For this latter task, the department's director, A. van Wijngaarden, hired two promising young experimental physics students, Bram J. Loopstra and Carel S. Scholten, who started work in August of 1947. Scholten recalled:

> In the summer of 1947 I was on holiday somewhere in the eastern part [of The Netherlands], and I got a postcard saying would I please join the Mathematical Centre, in order to build computers. Well, I had never heard of the Mathematical Centre, which was a small wonder of course, because it had been founded only the year before. And I didn't know what, as they said then, a calculating machine was. (Well, I knew what calculators were, desk machines). And so, knowing neither my employer nor the subject, I said yes. And that was that[3].

Scholten and Loopstra began with several serious handicaps. First of all, there was no money in the beginning for either of them to travel to England or the United States to learn from those who had already built such machines. Second, they had little access to relevant literature, in part because there had been no exchange of books and periodicals between The Netherlands and the

Allied countries during the war, and even for a year or so after because of the poor economic situation. (At first they were only able to find two articles: one about Vannevar Bush's differential analyzer, an analogue computer, and one about the ENIAC). Third, as students they had been given no training in the use of relays or vacuum tubes. And while this was in part because they were trained as experimental physicists rather than as, say, electrical engineers, Scholten and others have pointed out that a more important reason was that this kind of equipment was virtually unavailable in The Netherlands in the early years after the war.

The picture that emerges then is of two young researchers attempting to reproduce an important technological development without the benefit of any of the expertise, experience, or equipment that was available to researchers in England and the United States, and with no prior training in electronics. (Scholten characterized his and Loopstra's first experiments with vacuum tubes at the MC as good material for a slapstick comedy film.)[4]

Based on the two articles they had initially found, Scholten and Loopstra decided at first to build a differential analyzer. Scholten remarked:

> Here you can see the influence of a good article: the article about the ENIAC was horrible, you just couldn't understand it. Well, it said a lot of words, and the words were arranged into sentences, which were grammatically correct, but that's all. Whereas the article by Vannevar Bush was a different story... . So the mere fact that we could understand that article and convince ourselves that such a thing could work, we started on it, but we bet on the wrong horse.[5]

The fact that the ENIAC employed 18,000 vacuum tubes was also an important factor that led them to opt for an electro-mechanical, analogue machine rather than an electronic, digital one, because they knew that it would be impossible to obtain such a large supply of vacuum tubes. (In fact, in the beginning they were obliged to get a lot of their equipment from a second hand flea market.) Within a year or so plans were changed, however. At the instigation of Van Wijngaarden, (who in 1947 had toured the most important computer installations in England and America), the effort shifted to designing a digital rather than an analogue computer. Obtaining equipment was still a problem, because funds were scarce. Van Wijngaarden and Loopstra managed to purchase, at a very low cost, a group of Siemens high speed relays salvaged from an English war dump, and thereafter work began on a *relay* computer. It was named the ARRA, an acronym for Automatische Relais Rekenmachine Amsterdam (Amsterdam Automatic Relay Computer).

The ARRA was under construction from 1948 to 1952, but it never became a workable machine. The problem was two-fold. First, too many different technical elements were incorporated into the design. Scholten traced the cause of this in part to his group's relationship with the Philips Research laboratory:

We acted sort of as pioneers for the Nat Lab* of Philips, already then, because we had very good contacts with them. And whenever the Nat Lab thought of something new, in the way of technique, we were usually presented with a couple of items, and they would say, 'Would you like to try it and see what you can do with it?' And then we would try to incorporate it somewhere. For instance, they gave us a very small cathode ray tube. It had two anodes, and the beam could be on either one of them. So we could use it as a switching element. We wanted to try it - it was stimulating, it was new! But so the whole of the ARRA was, well, to put it very unkindly, just a sort of patchwork.[6]

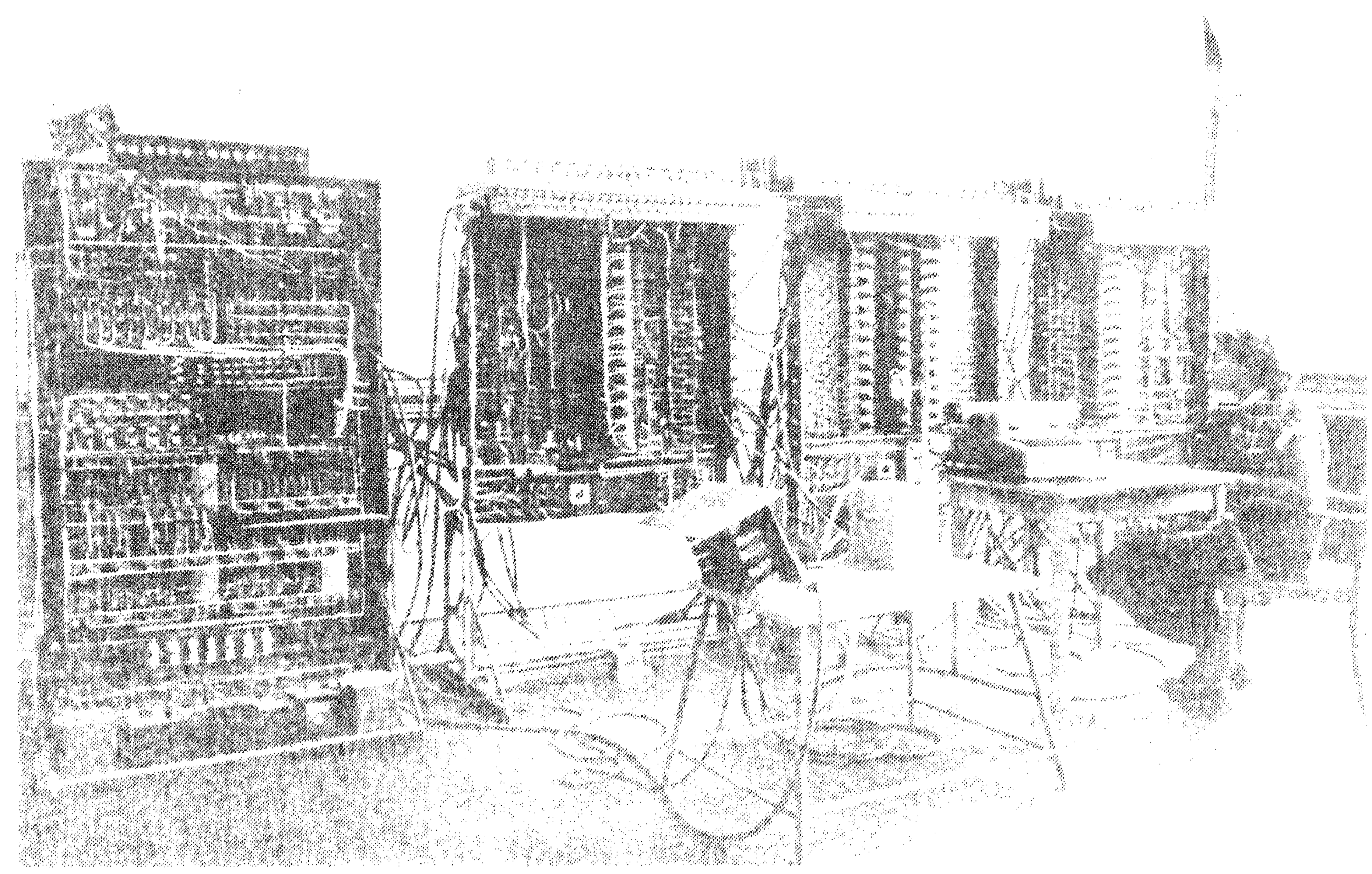

FIGURE 1. The ARRA

The other problem with the ARRA was more serious. The machine worked with an operation complete signal, which meant that at regular intervals (on the order of a millisecond), a signal was given to the machine indicating that the previous cycle had been completed and that a new cycle should be initiated. The frequency of these signals, however, was established on the basis of an assumed minimum switching time which in reality could not be depended upon. This was mainly due to the behaviour of the relays. The relays each had a fixed arm and a movable arm, and a circuit was either closed or opened

*Nat Lab = Natuurkundig Laboratorium

when the movable arm either made or broke contact with the fixed arm. But what often happened was that the movable arm began to bounce after it made contact with the fixed arm, and as long as this bouncing continued (which was often beyond the assumed minimum switching time) a new cycle could not actually be initiated, although the signal to initiate one would still be given. The practical result of all this was that the computer's output could not be trusted. As Scholten explained:

> In the worst of all possible cases you would get an output which could be right but then perhaps it wasn't. For example, you might get a neatly typed list of figures which could be the correct ones, but you have no idea whether they are or not. If the machine gives up and collapses altogether, it is better, of course, because you know there is something wrong. [7]

There were also some unavoidable, practical problems. For example, the laboratory was located fairly close to the Amstel brewery, and it seemed that whenever the wind blew from that direction, the relays would stop working properly. To solve the problem, a number of the women 'calculators' were recruited to clean the hundreds of relays, but this only worked temporarily.[8] They also tried covering the relays with condoms, but this idea was given up because, as the story goes, the relays could no longer 'multiply'.

In the summer of 1952, the ARRA was officially 'brought into service' with a formal ceremony. Yet in fact, on that occasion, it ran the only program it was ever to run, a program to generate a table of random numbers (presumably because random errors in such a table would not be noticed).[9]

2.2. Gerrit Blaauw and the ARRA II

The failure of the ARRA made it clear that to build a viable computer demanded considerable technical knowledge that could not easily be reinvented on demand. Consequently, that fall, Van Wijngaarden hired a young Dutchman, Gerrit A. Blaauw, who had just completed a Ph.D. under Howard Aiken at the Harvard Computation Laboratory.

Blaauw had first become aware of computers in 1946, when he read a Dutch newspaper article about the Mark I computer* that had been built at the Harvard Computation Laboratory. The machine seemed to combine perfectly two of his main interests: mathematics and electrical engineering, and from that point on his mind was made up to go into that field. He received an undergraduate degree in electrical engineering and ultimately was accepted for graduate work at Harvard under Howard Aiken.

As a graduate student, Blaauw at first taught courses on the use of the Mark I. Subsequently he had a research assistantship which involved him in extensive testing of the Mark III. He also designed parts of the Mark IV computer.

*Actually, the Mark I computer was completed 1943 and made public in the US in 1944, but it became known in Holland only after the war.

In particular, he designed the control system and also the system of connections for transmitting data into and out of the computer - in Blaauw's words, 'the logical part of getting data ready for transmission'.[10]

Blaauw thus came to the Mathematical Centre with a good deal of practical experience about designing and operating computers. He was to use this knowledge in a renewed effort to build a viable computer at the MC.

One of the first exchanges that took place between Blaauw and Scholten and Loopstra, was a long discussion in which, as Scholten put it, Blaauw told them 'some facts of life':

> Stick to one technology, and one technology only; don't rely on minimum propagation time. And he also taught us how to use the components you have in a reliable way.[11]

In effect, what Blaauw introduced into the program at the MC was philosophy of design that he had learned from Aiken at Harvard. According to Blaauw, Aiken's goal was not simply to design computers that would work, but to design computers that would work *reliably*. Accordingly, his design philosophy embodied a set of principles to insure reliability. For example, he never attempted to push a technology to its limit - that is, to make components work at the limit of their speed and performance capability. He preferred to leave it 'to future development to improve the speed'. Aiken also shied away from using the very newest technologies that were not yet 'tried and true'. He also made use of some more specific techniques to insure sound construction and easy maintenance. For example, wires were color coded to avoid errors and confusion, and a strict system of checks and cross-checks was established to insure correct wiring. Aiken also used pluggable components, which made for easy replacement. (In the ARRA, all components were soldered in place, which made replacement time consuming and tedious.)[12]

As opposed to the five years spent to build the ARRA, the new computer took only 13 months to build. It ran its first program in December of 1953 and was used constantly for the next three years. For bureaucratic reasons, it was called the ARRA II, but it was a totally new machine, and not merely a revision of the ARRA. It was entirely electronic rather than electromechanical. The memory was a magnetic drum with a capacity of 1024 words (each with a length of 30 bits), extremely limited by today's standards. (Magnetic drums were slower than other types of memory, but they were cheaper, which is why they were chosen by the MC group*.)[13]

The small capacity of the drum meant that only relatively simple problems could be programmed on the computer. The more complicated problems still had to be done with the desk-top calculating machines. Thus, there was no rapid transition from calculating machines to computers at the MC. In fact, the shift was so slow and gradual (over a period of 15 years) that those who

*The magnetic drum for the ARRA II was actually made at the repair shop of KLM, because there they had machine tools of the requisite precision and accuracy.

worked there cannot remember exactly how or when it occurred. What they do remember is that the desk-top calculating machines were still in use in 1962, but that over the years their function changed from doing all the calculations, to doing only the most difficult calculations, and finally to verifying the correctness of computer calculations. And as the computers gradually began to take priority over the desk-top machines, the women who had operated the desk-top machines gradually evolved into computer programmers.[14]

2.3. The FERTA and the ARMAC

Following the ARRA II, two other computers were built at the MC between 1953 and 1956. The first, known as the FERTA*, was basically a copy of the ARRA II and was built specifically for the Fokker company. Its principal job was to perform the flutter calculations involved in the design of the Fokker F-27 'Friendship'[15],

The second computer, the ARMAC**, was built for the MC itself; it replaced the ARRA II in 1956. It was designed entirely by Scholten and Loopstra, for by this time Blaauw had left the MC to work at the IBM development lab in Poughkeepsie, New York. The ARMAC was faster than the ARRA II, and it had a larger memory, 3584 words (still very small by today's standards). The main memory was a magnetic drum, but it also had a smaller, faster, magnetic core memory (which was a recent innovation at the time). The magnetic core memory was used as a buffer to store the entire contents of one track of the magnetic drum. The advantage was that instructions could be accessed from the buffer many times faster than from the drum, which meant that programs could be made to run much faster.[16] Yet the design of this buffer-memory system involved sequential addressing, which meant that the programmer always had to keep track of the order of the instructions on the drum. (In the case of ARRA II, addressing was non-sequential, so this was never a problem.)

Now it happened that the programmer of the ARMAC, Edsger W. Dijkstra (presently regarded as one of the world's leading computer scientists) was by nature a very methodical person, who favored a rigorous, logical approach to scientific research. (He has always disliked the kind of clever tricks that manage to produce viable solutions to technical or scientific problems but that have, in themselves, no deep, logical foundation.) Given his own character, then, Dijkstra came to see the limitation of the ARMAC as a blessing in disguise, because it pushed him further in a direction toward which he had already been groping. It led him to a very strict, disciplined style of programming in which emphasis was placed on structuring programs with great care, on achieving logical clarity and conciseness, and on verifying the correctness of a program theoretically, in advance. This approach has since become one of the hallmarks of Dijkstra's scientific style.[17]

*FERTA = Fokker Elektronische Rekenmachine Type ARRA
**ARMAC = Automatische Rekenmachine Mathematisch Centrum

FIGURE 2. The ARMAC

2.4. Transition of the MC computer group to the private sector

After completion of the ARMAC, the MC computer group made a transition to the private sector. The reasons were two-fold. First, commercial computers were now readily available, which meant that the original justification for the group's existence within the MC was no longer valid. Second, J. Engelfriet, the director of a Dutch insurance company (Nillmij) had expressed an interest to Van Wijngaarden in backing a commercial venture in computers. (The company was interested in using computers for its administration.) Consequently a new firm, Electrologica, was founded in 1956, and over a period of two years, nearly the entire MC computer group, some 45 persons, transferred to the new company.

During that period, between 1956 and 1958, the MC-Electrologica group designed a general-purpose, commercial computer, the X-1. The first X-1 came into operation in August 1958, and ultimately some 40 of the machines were sold. The X-1 was notable for its speed and for the fact that it made use of innovative technologies. In particular, it was fully transistorized and used a magnetic core memory. The speed of the machine was due in part to the fast-access memory but also to the fact that words (each 27 bits long) were transferred from the memory in parallel rather than serially.[18] Another innovative feature of the X-1, was that it was among the first computers to make use of an interrupt, which made the computer's operation more time-efficient with respect to input and output operations.

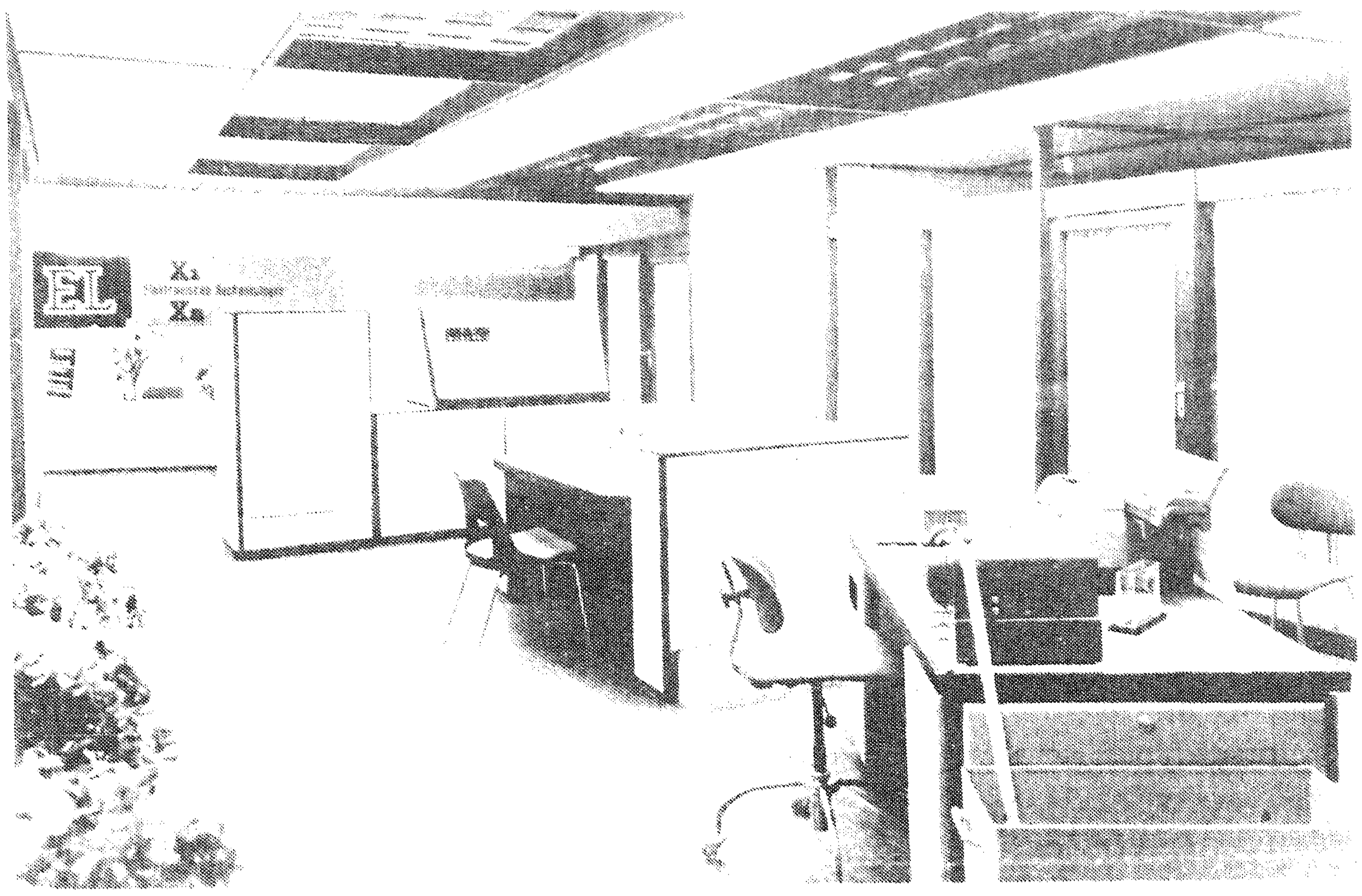

FIGURE 3. The X-1

After the X-1, Scholten and Loopstra developed a larger and faster machine, the X-8, which came onto the market in 1965. The Electrologica machines did not prove in the long run to be a commercial success, however. At first there had been a great deal of excitement in Europe about Electrologica, because it represented an alternative to IBM. (According to Scholten, there was at that time in Europe, and particularly in Germany, a 'severe anti-IBM movement'.) But the first machines did not have a very good 'mean time between failures' and in general the company could not withstand the competition from IBM. According to Scholten, IBM was more successful partly for technical but also for organizational creasons:

> I guess that perhaps discipline in their [IBM's] factory was better than ours. Just the ordinary things like soldering or choosing the right sort of components were done better there. But I think that as far as performance goes, we really didn't differ too much from IBM, because they had a very bad mean time between failure also. But they were a big company which could easily plan to compete us out of Europe. Our area was very small, of course. We sold machines in Holland, in Germany, in France (with difficulty), in Switzerland, and perhaps one in the northern part of Italy - and that was it.[19]

One particular advantage of IBM was that, being a very large, well-established company, they were in a better position to provide their customers with prompt repair and servicing. For Electrologica, competition on this score proved to be impossibly expensive, and the company went into debt, eventually reaching an accumulated loss of some 20 million guilders. Subsequently, in 1968, Electrologica was acquired by Philips, but Philips initiated its own design program, and did not continue along the path established by Electrologica.

3. The PTT program

The person responsible for initiating a computer project at the PTT was L. Kosten, who headed the mathematical section of the PTT research group. Kosten's main research interest was teletraffic, the study of the behaviour and characteristics of telephone systems. His concern was to analyze these systems mathematically, in order to be able to predict, for example, levels of congestion in particular systems.

Yet as the telephone networks continued to become larger and more complex, their mathematical analysis became increasingly lengthy and cumbersome, indeed to the point of exceeding human computational capabilities. The only way to handle this complexity was to simplify the mathematical models. For there was, in Kosten's words:

> a certain interaction between the character of these models and the means available to build them. (And by models I mean the mathematical extraction of the problem you have at hand in the technology.) The models change with the means available ... [and so] when the means were not available, you had to modify the models a bit - you put water in your wine, let me say it that way.[20]

What Kosten and other teletraffic analysts therefore did was to search for means to model telephone networks physically rather than mathematically. They developed special machines - traffic analyzers - which imitated the behaviour of these systems.

Kosten himself spent ten years developing such machines, from 1940 to 1950. Research on the principles began somewhat earlier, around 1937, in particular on how to devise a physical system that would imitate the randomness of telephone calls. The solution Kosten found was to use a selector from a telephone exchange linked to a Geiger counter:

> The old [telephone exchanges] had electro-magnetic switches - and there were selectors that turned around and hunted for free lines and so on. And I used one of those which was turned around by a motor. So I let it turn around (it was in effect a simulation of a roulette wheel), and at a certain moment the motor was stopped by a Geiger-Müller counter activated by radiation. It was rather clumsy, and not very accurate, and it took a lot of time, but it worked.[21]

Between 1940 and 1942, Kosten built a prototype traffic analyzer which incorporated the 'randomness apparatus'. From 1942 to 1945, plans were made for a larger machine, but these were dropped in 1947 because, according to Kosten, the design was made obsolete by the new developments in electronics and computers.

These special-purpose traffic analyzers were never entirely satisfactory because they were not adaptable. In particular, radical technological changes in the telephone systems (such as the introduction of link systems) 'yielded situations that could not be coped with'. In other words, the operation of the traffic analyzers could not be altered to reflect changes in the technological systems they were supposed to model. But a general-purpose computer, with its ability to be programmed, had the potential to serve as an adaptable, all purpose traffic analyzer. It could also be an aid in carrying out the very complex mathematical calculations that teletraffic engineers constantly had to deal with.[22]

Kosten only gradually came to see computers as an alternative to dedicated traffic analyzers, however. At first his interest was to apply the new technologies developing around computers to build a better traffic analyzer. He developed such a design between 1947 and 1950, with cooperation from Philips. The new traffic analyzer design was to incorporate a memory device (to keep track of engaged and disengaged lines), so Kosten began experimenting with various types of memories originally developed for computers - notably, magnetic drums and Williams tubes. And for the 'randomness apparatus', several random bit generators were developed. Each one consisted of a flip-flop activated at random intervals by a particle of cosmic radiation that caused ionization in a gas discharge tube. What is, in retrospect, ironic about these expensive and elaborate random bit generators, is that the job could have been done instead with a general-purpose computer, by means of an algorithm for generating pseudorandom bits, i.e. through programming. It was when Kosten recognized this possibility that he shifted the attention of his group to developing a general-purpose computer.[23]

To design the computer, Kosten hired a young, scientist-inventor from Delft Technische Hogeschool, W.L. van der Poel. Unlike Scholten and Loopstra, Van der Poel knew what a computer was and in fact was already in the process of constructing one for the optics department at Delft.

Van der Poel's interest in calculating machines dated back to his childhood:

> I was very bad at mental calculation, so even in my elementary school days, so from 8 to 12, say, I had an old cash register, for doing my sums, and I had the Napier's rods to do the multiplications. And from that time I have always been fascinated by the mechanics of computation.[24]

While still a high-school student, before World War II, Van der Poel began to haunt the libraries in search of literature on computing machines and he managed to learn most of what had been done up to about 1940 (after which, according to him, the flow of literature ceased because of the war).

During the war, in 1944 to be precise, Van der Poel designed his own calculating machine, a rather sophisticated one considering his youth (he was only 18 at the time) and the difficult circumstances in which he had to work. In particular, Delft Technische Hogeschool, where Van der Poel would have begun classes, was closed down, so he had no access to equipment or technical advice: he worked entirely on his own. (He did, however, enroll in a technical correspondence course for a while, but this was no substitute for a university-level engineering education.) The machine Van der Poel designed during the war was to be built from telephone stepping switches, which meant that it would work in base 10 rather than base 2. It was not programmable, but it did have programming levels built into it, so that it could not only add and subtract, but also multiply and divide and calculate sine functions.[25]

The machine was never built, but only a few years later, in 1947, Van der Poel began to build a computer in the optics department at Delft Technische Hogeschool, where he had since become a student. It was a programmable, digital, relay computer, designed specifically for optical calculations. Like Scholten and Loopstra, Van der Poel had essentially no budget to work with, so there was no question of building an elaborate, high-tech machine like the ENIAC. But luckily around this time Van der Poel had become acquainted with Kosten from the PTT, and from Kosten he managed to obtain, at no cost, a supply of 600 discarded relays. They all had to be rewound, and Van der Poel did this himself using a lathe, since there was no winding machine available. When first experimenting with the relays, Van der Poel encountered the same difficulty as Scholten and Loopstra. He initially regarded 30 milliseconds as an ample estimate for the relays' switching time, but he discovered that, in practice, with the bouncing that would occur and so forth, switching times were often on the order of 900 milliseconds.

In 1950 Van der Poel graduated from Delft and began working with Kosten at the PTT. The building of the optical computer at Delft was thereafter left to other students, and in that makeshift, low-budget environment, it is not surprising that the project took years to complete. In fact there was no one point at which the machine was completed, because improvements, additions, and adjustments continued to be made over a period of years. But sometime between 1952 and 1954 it began to be used on a more or less regular basis, and it continued to be used until about 1965. It was so slow - it could not calculate much faster than a human - that it was nicknamed TESTUDO (Latin for turtle). But the TESTUDO was very reliable: it could be set into operation at night and it would grind away at its calculations, unattended, until morning. (The calculations mainly involved the conversion of sine to cosine, which used the expression $\sqrt{1-x^2}$, so this operation was actually built into the machine.)

Meanwhile, Van der Poel at the PTT had shifted his focus to designing a fully electronic computer (i.e. composed of vacuum tubes rather than relays). It was named PTERA*. In order to gear up for this machine and test the basic

*PTERA = PTT Elektronische Reken-Automaat

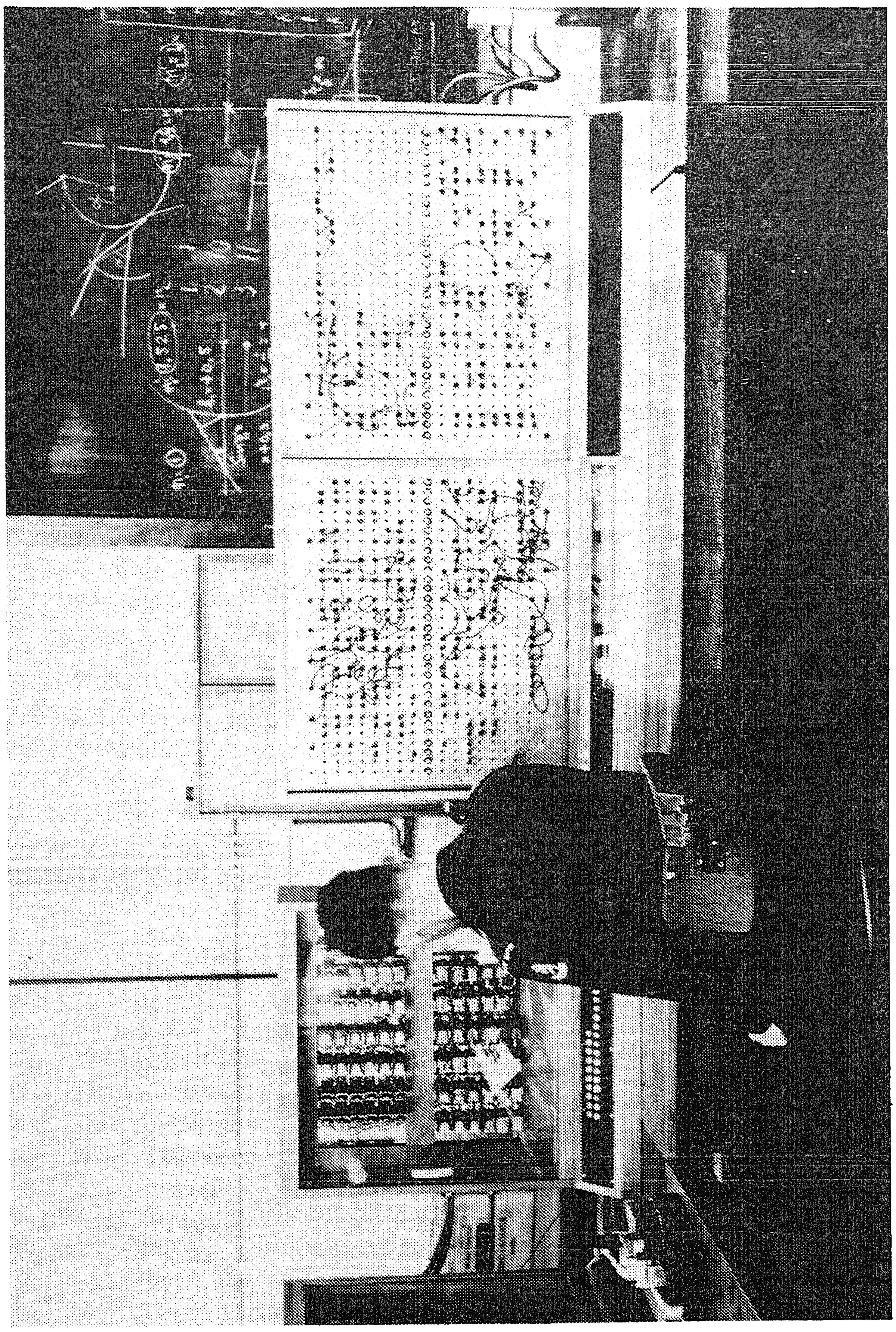

FIGURE 4. The TESTUDO

concept, an ultra-simple, relay computer was built between 1950 and 1952, the ZERO*. The ZERO's memory was limited to one track on a magnetic drum, or 32 words. The arithmetic unit was equally limited, and it had an ultra-simple control box consisting only of 6 relays, controlled by four 'functional bits'. Particular circuits would open or close depending on whether these functional bits were set to 0 or 1. 'Programming' was done by choosing and manipulating the functional bits in such a way that desired mathematical operations were carried out. The only built-in functions were addition and subtraction. Thus even multiplication and division had to be programmed as a series of repeated additions or subtractions, using the functional bits.

The ZERO was in operation for only two months and was never regarded as anything more than an experimental, 'toy' computer. The real computer which followed, the PTERA, was completed in September of 1953. According to Van der Poel, there was considerable pressure within the PTT to get it into operation as fast as possible - and again within a tight budget - so there was no time to work out the design carefully; it was, in his words, a 'weird' machine, and not particularly innovative or creative. Nevertheless, the machine was used at the PTT for several years and its computing power was said to be equivalent to the ENIAC's, although it operated much more slowly. In contrast to the ENIAC's 18,000 vacuum tubes, however, the PTERA employed only 700, with an additional 120 relays. Like the ARRA II, the PTERA had a magnetic drum memory with a capacity of 1024 words (each 31 bits long).[26]

Following completion of the PTERA, Van der Poel undertook the design of a commercial computer, which was eventually dubbed ZEBRA**. To manufacture the computer, the PTT at first approached Philips, but they declined to get involved in the project. So the PTT turned to a British company, Standard Telephones and Cables and they made an agreement to manufacture the computers. The design was completed in 1956, and the first machines were delivered in 1958. They remained in production until about 1964, and overall some 55 were sold world-wide. Van der Poel developed the logic design, but left the construction details entirely to the company. At first the machines were made with vacuum tubes, but eventually a fully transistorized version was produced (still based on the same design, however).

The design of the ZEBRA was acknowledged to have been quite unique. Interestingly, the basic underlying concept of the machine came not from the PTERA, but from the ZERO. In particular, Van der Poel returned to the idea of using functional bits, which in the ZERO had represented merely a temporary, makeshift solution to the problem of a control box. In the ZEBRA, however, this idea was extended so that the control consisted not of four, but of 15 functional bits. Each instruction word, which was 33 bits long, included these 15 functional bits, and by setting the values of the various bits at either 0 or 1, the programmer was able to control the operation of the machine

*The name ZERO was not an acronym

**ZEBRA = Zeer Eenvoudig Binair Reken-Apparaat

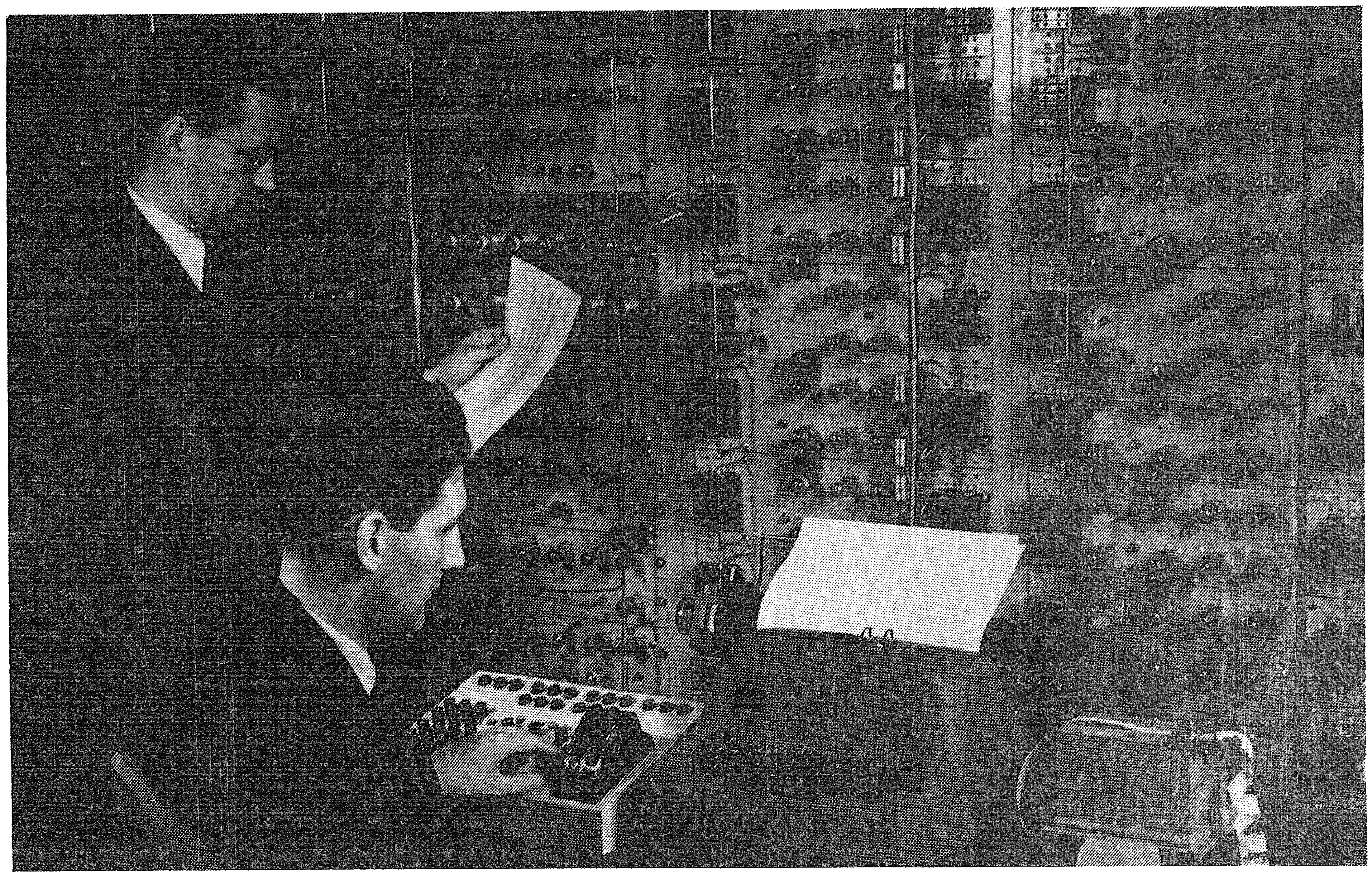

FIGURE 5. The PTERA - with Kosten and Van der Poel (seated)

directly. To give a few examples, one bit (the 'A' bit) controlled whether information stored on the drum was sent to the arithmetic unit or to the control unit; another (the K bit) controlled whether the registers (for temporary storage of information within a computer) were connected to the drum or to the control unit; two other bits (D and E) controlled whether information was either read from or written to, respectively, the drum or the registers. Thus, depending upon how these functional bits were set up in any given instruction, the machine might be made to draw a new instruction from the drum at the same time that it was performing an addition with a number drawn from a register; or it might be made to form a new instruction in the control unit by drawing an instruction from the drum and then modifying it by the contents of a word in the register.[27]

In practical terms, the use of these functional bits made the ZEBRA at once structurally simple and extremely flexible. It was structurally simple in that no operations other than addition and subtraction were built in, which meant that the machine had comparatively few parts and demanded correspondingly little maintenance. It was flexible in the sense that the system of using functional bits made each instruction potentially very powerful. As a consequence, long, complex routines could eventually performed with very few instructions. Or, by clever use of the functional bits, a program could be made to run very fast. The opportunities were seemingly endless, limited only by the ingenuity of the programmer. (Consider the fact that 15 bits, each with a possible value of 0 or 1 gives rise to 2^{15}, or 32,768 combinations.)

What eventually happened was that a veritable programming subculture arose around the ZEBRA. (Since there were some 55 of these machines built, there were hundreds of ZEBRA programmers around the world, and dozens in The Netherlands.) One particular technique that evolved was called 'underwater programming'. This referred to programs that were, on the surface, very short and simple: such a program might contain only 7 or 10 lines. But when run, the instructions (by way of the functional bits) would re-execute themselves many times over and even alter themselves in the process so that ultimately a very long and intricate computation would be carried out. The term, 'underwater programming' was intended to convey the analogy with the activity that goes on in a lake or sea, undetected from the surface.[28]

Van der Poel dubbed the myriad of clever programming techniques that evolved around the ZEBRA 'trickology'. His choice of this particular term was not haphazard, but reflected a life-long interest in clever puzzles. Van der Poel made his first puzzle as a small boy - it was something like a Chinese wood block puzzle - and he presently has a collection of about 1500 such puzzles, a number of which he has made himself. Of those he has made, one of his favorites was a kind of combination lock that only locked itself together; when the right combination was entered, all the components of the lock came apart. Van der Poel acknowledges a direct connection between his interest in puzzles and his interest in calculators and computers, moreover. For him the operating mechanism of a mechanical desk calculator is like an elaborate puzzle and likewise the design of an electronic computer or an integrated circuit. Thus the particular character of the ZEBRA's design, and the fact that it lent itself to an elaborate programming 'trickology', can be seen in some sense as outgrowths of Van der Poel's fascination with complicated, ingenious puzzles.

The style, or perhaps one should say the ideology of programming fostered by Van der Poel with the ZEBRA was thus remarkably different from that advocated by Dijkstra with the ARMAC. And perhaps not surprisingly, each disapproved of the other's work in certain respects. Dijkstra regarded Van der Poel's trickology approach as just so much wasted ingenuity, because it was machine dependant and made no essential contribution to the *science* of computing. Van der Poel regarded Dijkstra's emphasis on verifying programs theoretically as pedantic and generally unworkable in practice.

4. The Philips program

The one other computer design program that was established in The Netherlands by the early 1950's was at The Philips Nat Lab. In virtually every way, of course, Philips was better equipped than anyone else in the country to undertake such a project and ultimately to produce commercial computers. The company already manufactured electronic components; it had an excellent research staff with experience in this field; it was one of the largest and wealthiest companies in The Netherlands; it had branches in other countries (e.g. England) and close ties with other companies (e.g. IBM, AT&T) so that it was in an ideal position to keep abreast of the very latest developments in computers and electronics. (In fact, by 1949 it was already beginning to receive commercial orders for computer components.)

Yet Philips did not actually begin building a computer until late 1953, although some preliminary studies were carried out beginning in 1951, including one project to develop a very small, fast magnetic drum. By 1960, a total of three computers had been produced at Philips, all for internal use. The first, the PETER*, was built between 1953 and 1956; the other two, PASCAL and STEVIN**, were built between 1956 and 1960. The STEVIN was a copy of the PASCAL, slightly modified for business use.

The PETER was a small but relatively fast computer. It had a drum memory (1924 words) plus a smaller magnetic core memory (32 words). Compared to the PTERA, the ARRA II, and the ARMAC, which required 10-50 milliseconds (about 10^{-2} seconds) for an addition, the PETER required only about 15 microseconds. (i.e. 15×10^{-6} seconds). Such speed was achieved because the machine worked partly in parallel and because it had a fast drum and an even faster magnetic core memory. Yet the PETER was not very reliable: a sign posted above the machine read 'Don't count on it!'. The unreliability was due partly to makeshift construction. The soldering was not professionally done, so there was considerable trouble with bad connections. The bad

*PETER = Philips Experimentele Tweetallige Elektronische Rekenmachine

**PASCAL = Philips Akelig Snelle Calculator; STEVIN = Snel Tel En Vermenigvuldig Instrument connections were moreover difficult to detect, because '99.9% of the time they functioned properly'. In addition, many of the components were makeshift, which made them unreliable as well, and this problem was compounded by the fact that some of them incorporated germanium diodes which were very heat sensitive:

> Because the power supply units and the electronic valves were producing a considerable amount of energy, the temperature in the room where PETER was located increased in the course of the day. This temperature change influenced the behaviour of the Ge-diodes resulting in glitches, an unpredictable setting or resetting of the flip-flops. By adjusting the power supply voltages PETER could be called to order for some time. If one failed, however, to adjust the power supplies before lunch-time, one could forget performing calculations in the afternoon.[29]

Despite all these problems, the PETER worked for three years and performed a wide range of scientific calculations concerning, e.g., crystallography, switching functions, and cyclotron design.

The original aim in building a second computer, the PASCAL, was simply to produce a more reliable and more professionally built version of the PETER. But gradually the project evolved into the development of an entirely new machine. And when it was completed, the PASCAL was one of the fastest computers in the world. (It was many times faster than the ZEBRA or the X-1.)[30] The PASCAL had several levels of memory, including a magnetic drum, a magnetic core memory (capable of storing 2048 words)*, and a special, fast 'B-memory' which was used to modify and temporarily store instructions.

According to one of the PASCAL's designers, H.J. Heijn, a number of features helped to make this computer very fast. First, the machine operated at a high frequency - that is, with a short cycle time (660,000 cycles per second). The frequency was set by the drum and was ultimately dependant upon the specific design of the drum. In particular, the frequency depended upon such features as: (1) the density of pulses on the drum, which was a function of the drum's magnetic coating; and (2) the design of the reading and writing heads, including the distance between the heads and the drum. The unique design of the drum reflected some of the particular strengths of the Philips research laboratory (e.g. an extensive knowledge of magnetic materials and state-of-the-art knowledge about the design of reading and writing heads).[31]

*The original plan was to have a magnetic core memory of 1024 words × 42 bits, or a total of 43008 bits. At that time, however, one of the directors of the research lab argued that this was extravagant, that 40000 bits was already more than enough. (Bear in mind that personal computers today commonly have memories a dozen times larger.) But the director changed his mind after visiting the United States and hearing that memories of 4096 words were in the process of development. Thereafter he suggested that the size of the PASCAL's memory be increased, and eventually it was doubled.

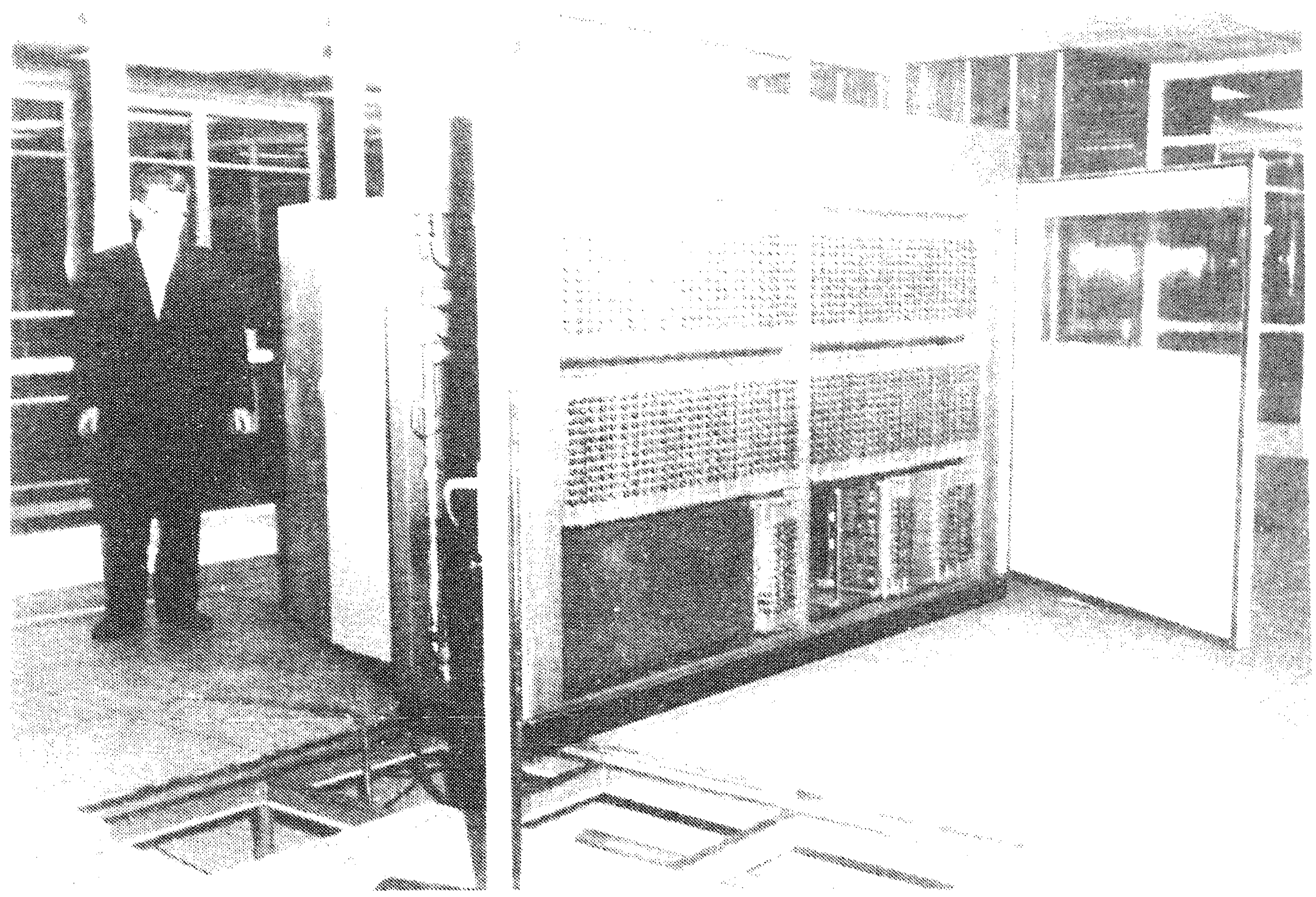

FIGURE 6. The PASCAL

The second feature that helped to make PASCAL so fast was that the problem of carry propagation (i.e. the digits that have to be carried in additions and multiplications) was solved in a new way which made the whole process much faster. (In fact a patent was obtained for the design of the new carry mechanism.) Third, the use of the B-register, which was a relatively new innovation and involved the use of specially designed transistors, helped to speed up the machine. Finally, the units for transferring data into and out of the computer (i.e. from the drum to the core memory and vice-versa) were designed in such a way that transfer was very rapid and could proceed while calculations were being performed.[32]

The PASCAL was used in the company's research labs for scientific and technical calculations related to research on cryogenics, semiconductors, television, hot air engines, and other problems. The STEVIN, which differed from the PASCAL only in the number and variety of the peripherals that were linked to it (e.g. Bull punch-card readers) was used by the Philips administration.

Despite the success of the Philips machines (and the fact that other computer designers were quite impressed by them), Philips did not immediately attempt to develop a commercial computer. Moreover, the company declined requests to build computers for other organizations. In particular, the PTT had at first asked Philips to produce the ZEBRA and the MC group had originally wanted it to produce the X-1, but in both cases Philips refused. In effect, the company's very strengths became a source of weakness when it came to computers. Because it was an electronic *component* manufacturer, there was emphasis within the company to preserve this particular market (and it was a large and flourishing market at the time). Within this framework, Philips made an agreement with IBM not to enter the computer market in return for a guarantee that IBM would buy a certain share of its electronic components from Philips. One of the principal reasons why Philips undertook to build a computer at all was to get a better idea of how computer *components* should be designed.[33]

It was not too long, of course, before Philips saw that this had been a short-sighted agreement and that computers represented an innovation that they could no longer afford to ignore.* Yet to be able to cultivate this new technology, the company's directors had to make a place for it within the company's overall organizational structure: a new department needed to be set up, or a decision made to add a computer section to an existing department; a substantial budget needed to be allocated; a fairly large research and development team needed to be set up; extensive marketing research needed to be done; and so on.[34] But the company was slow in making these changes, partly as a result of indecision, partly as a result of infighting between different departments who wanted to 'claim' the computer project, and partly, according to A. Duyvestijn, as a result of financial policies geared toward the consumer goods market (in particular a policy that investments should return a profit within two years - an impossibility in the case of the first generation computers). In fact Philips initiated a program to develop commercial computers only around 1963.

5. Significance of the invention of computers: the Dutch example

Existing computer histories, which concern mainly British and American developments, have implicitly portrayed the invention of computers as largely a byproduct of military research. The history of Dutch computers, however, reveals that the forces behind this development were much broader than that. In The Netherlands, it was not just the military that needed to do complex calculations: scientists and engineers throughout the country were swamped with them. For example, at the PTT, in order to create a viable telex network, very complex and extensive calculations had to be done to determine how atmospheric ionization was influenced by changes in the sun's radiation. Other complicated calculations had to be done to keep undersea cables in working order:

> These cables had valve amplifiers submerged at certain distances along the floor of the sea. Well, a valve does not have an eternal life, so sometimes the cable went out of service. And how do you determine the exact spot where it is? You can do that by sending a

* One immediate cause of this change in perspective that is cited by representatives of Philips was the fact that IBM built its own factory to produce computer components, thereby reneging on the agreement with Philips.

> pulse down the line and seeing where it reflects. And out of the reflections, you can calculate exactly which amplifier is out of service, and then you can send a ship there which hooks up the cable and replaces the amplifier. And so that all involved a lot of complex calculations.[35]

Then there were calculations that needed to be done in order to dimension and structure the telephone networks effectively. (It was partly these which led engineers like Kosten to develop teletraffic analyzer machines.) There were also calculations for filter designs:

> The whole coming-up of telephone networks with multiple lines compressed on a single band, with all little slices of band-widths allocated to each telephone. You know nowadays on one of these glass [fiber-optic] cables there are 20,000 telephone calls on a single wire. But you have to pack and unpack it; and in those days [the 1950's] the packing and unpacking, the modulation and demodulation was done by hard-wired valve amplifiers, and it was a rather complicated matter and we had to do a lot of filter calculations. Well filter calculations involve a lot of complex computation with elliptical functions and solving of higher-order equations - all that sort of thing.[36]

The examples mentioned here so far represent just a few of the calculations that were being done at the PTT. At Delft, at Fokker, at the MC, in fact everywhere you look, it was also the same story. One problem carried out at the MC, concerning the determination of a large set of values for flutter calculations, occupied several members of the full-time staff for a period of two years.

The situation appears to have been analogous in the realm of business computing, moreover. For example, the 'giro' system of the PTT grew so rapidly after World War II that it eventually reached a point of crisis:

> Every clerk in the central office had about 10,000 accounts under his supervision, and had to do the additions and the subtractions from his batch of accounts by hand. So the growth of the traffic and the growth in the number of accounts required a proportional growth of personnel. Now, at that time we had 6,000 people in the central office, and we had to put a stop on the number of accounts, because we just couldn't handle any more.[37]

The development of computers was thus a response to - and an agent of - increasing social and technical complexity. The fact that this technology at first emerged mainly within a military context in the United States and Britain only reflects the fact that, for social, political, and historical reasons, there was more willingness to invest money and manpower to automate military computing than other branches of technical or business computing.

6. Conclusion

In reflecting upon the early development of Dutch computers, one is at first struck by the considerable diversity of the designs: no less than 10 distinct computer designs were created in the period up to 1960. Yet when analyzed as a group, it becomes evident that the designs arising at any given time were always conceived within a relatively limited technological framework. Thus, in the period from 1948 to 1952 the Dutch designers worked with electromagnetic relays and magnetic drums: in the period from 1952 to 1956 they worked with vacuum tubes and magnetic drums; in the period after 1956 they shifted to transistors, magnetic core memories, and larger, faster magnetic drums (256 tracks, as opposed to 64 tracks for the early computers; and operating at 6000 as opposed to 3000 r.p.m.). Hence, the overall trend was toward faster computers with larger, faster memories.

The question therefore arises: which was more significant in social terms, the diversity of these early computer designs, or their overall evolution? It seems clear that the latter was most important in determining how the early computers were and could be used. For example, it was not practicable with the ARRA II or the PTERA to automate a payroll - the computers were too small and slow to handle this kind of operation; in fact the possibility of using computers for large-scale administrative automation emerged in The Netherlands only in the late 1950's and it was years later before this possibility could be realized on any significant scale.

To put the matter another way, early computer designers were very limited with respect to the social and economic realms toward which their designs could be directed. It is not simply that they were not interested in extending the computer's use in the early years, it is that they were not able to extend it, because they did not have the technological know-how to be able to produce larger or faster or cheaper or more reliable computers that could accommodate a greater diversity of applications. Nor could such know-how simply have been generated on the spot, instantaneously. On the contrary, extending the technical capabilities - and hence the social and economic potential - of computers demanded many man-years of research. The early Dutch projects, with their mistaken paths and idiosyncracies as well as their successes, were an essential stage in achieving a broader social and economic role for the computer.

Notes

1 On the history of the Mathematical Center, see G. Alberts, F. van der Blij, and J. Nuis, eds., *Zij mogen uiteraard daarbij de zuivere wiskunde niet verwaarloozen* (Amsterdam: Centrum voor Wiskunde en Informatica, 1987).

2 Interview with C.S. Scholten, March 1987; interview with E.W. Dijkstra, July 1987; interview with A. Duyvestijn, April 1987. See also Alberts et al, eds., pp. 276-285 (transcription of an interview with A. van Wijngaarden).

3 Scholten interview. See also C.S. Scholten, 'Computers Ontwerpen, Toen', *Informatie* 22, No. 4 (April 1980), pp. 337-8.

4 Scholten, 'Computers Ontwerpen, Toen,' p. 338.

5 Scholten interview. See also Scholten, 'Computers Ontwerpen, Toen,' p. 338.
6 Scholten interview.
7 Scholten interview.
8 N.C. de Troye, 'From ARRA to Apple,' Retirement Address, Philips International Institute, June 10, 1987, pp. 5-6.
9 Scholten interview; E.W. Dijkstra, 'A Programmer's Early Memories', in *A History of Computing in the Twentieth Century*, ed. N. Metropolis et al. (New York: Academic Press, 1980) pp. 564-565.
10 Interview with G.A. Blaauw, May 1987.
11 Scholten interview. See also Scholten, 'Computers Ontwerpen, Toen', p. 340.
12 Blaauw interview.
13 Dijkstra, 'A Programmer's Early Memories', pp. 565-569: Nederlands Studiecentrum voor Administratieve Automatisering, *De Ontwikkeling van Elektronische Rekenmachines in Nederland* (Amsterdam: Studiecentrum voor Administratieve Automatisering, 1967), pp. 6-8, 20; *De Automatisering Gids* (22 February 1984), p. 5; (7 March 1985), p. 5.
14 Interview with J. Berghuis, October 1987; interview with E.W. Dijkstra and Mrs. Dijkstra. (The latter was one of the calculator-programmers at the MC.)
15 'Rekenmachine in Gebruik bij Fokker', Van Telraam tot Computer, No. 17, *Automatisering Gids* (March 28, 1984), p. 5.
16 Dijkstra interview; Scholten interview; *De Ontwikkeling van Elektronische Rekenmachines in Nederland*, pp. 6, 8-10, 19.
17 This theme is interwoven throughout Dijkstra's *Selected Writings on Computing: A Personal Perspective* (New York: Springer Verlag,1982).
18 Scholten interview; Dijkstra interview; E.W. Dijkstra, *Communication with an Automatic Computer* (Ph.D. Thesis, University of Amsterdam, 1959), passim; *De Ontwikkeling van Elektronische Rekenmachines in Nederland*,
19 Scholten interview.
20 Interview with L. Kosten, March 1987.
21 Ibid.
22 The complexity of the problems teletraffic scientists were attempting to handle can be seen in the fact that one researcher at Bell labs, W.S. Hayward, spent 6 months to model 'one-half hour of real time' (personal communication from W.S. Hayward).
23 Kosten interview; W.L. van der Poel, 'The Early History of Computing at PTT', *Delft Progress Report* 5 (1980), pp. 174-183.
24 Interviews with W.L. van der Poel, February, June, August, October 1987.
25 Ibid.
26 Van der Poel interviews. Additional information about the PTERA was culled from articles appearing in Dutch newspapers when the machine was put into service in September 1953; from *De Ontwikkeling van Elektronische Rekenmachines in Nederland* pp. 12-13, 23; and from *Automatisering Gids* (19 april 1984), p. 5.

27 Van der Poel interviews; *Stantec ZEBRA Manual* (London: Standard Telephones and Cables, 1958).

28 Dijkstra interview; van der Poel interviews; W.L. van der Poel, 'Microprogramming and trickology', *Digitale Informationswandler,* ed. W. Hoffmann (Braunschweig: Vieweg, 1961), pp. 269-311.

29 N.C. de Troye, 'From ARRA to Apple'.

30 The time required for an addition on the PASCAL was 7-8 microseconds, and for a multiplication, 70 microseconds. The figures for the ZEBRA were, respectively, 300 microseconds and 11 milliseconds; for the X-1, 64 microseconds and 500 microseconds.

31 Interview with H.J. Heijn, September 1987.

32 Ibid.

33 Heijn interview; Duyvestijn interview; Berghuis interview.

34 Other companies naturally faced the same situation. For example, the director of National Cash Register recalled that 'the vital question that faced NCR in the 1950's was: Should the company enter the electronic data processing field? If the answer was yes, we knew it meant many millions of research development dollars, plus the retraining of manufacturing, marketing, and service personnel. If the answer was no, we knew full well that future growth would be seriously limited. We would be living from then on with a ceiling on opportunity. So we decided once more to move ahead'. Quoted from Katharine Davis Fishman, *The Computer Establishment* (New York: McGraw-Hill, 1981), p. 170.

35 Van der Poel interviews.

36 Ibid.

37 Ibid.

ACKNOWLEDGEMENTS

I would like to thank all of those interviewed for their time and patience, for their helpfulness in pointing out source materials, and for their willingness to share their knowledge and insights. The research for this paper was supported by the Nederlandse Organisatie voor Technologisch Aspectenonderzoek (NOTA).